Lecture Notes in Artificial Intelligence 3435

Edited by J. G. Carbonell and J. Siekmann

Subseries of Lecture Notes in Computer Science

Peyman Faratin
Juan A. Rodríguez-Aguilar (Eds.)

Agent-Mediated Electronic Commerce VI

Theories for and Engineering of Distributed Mechanisms and Systems

AAMAS 2004 Workshop, AMEC 2004
New York, NY, USA, July 19, 2004
Revised Selected Papers

Springer

Series Editors

Jaime G. Carbonell, Carnegie Mellon University, Pittsburgh, PA, USA
Jörg Siekmann, University of Saarland, Saarbrücken, Germany

Volume Editors

Peyman Faratin
Massachusetts Institute of Technology
Computer Science and Artificial Intelligence Laboratory
Massachusetts Avenue, Cambridge, 02139, USA
E-mail: peyman@mit.edu

Juan A. Rodríguez-Aguilar
Artificial Intelligence Research Insitute (IIIA)
Spanish Council for Scientific Research (CSIC)
Campus de la Universitat Autònoma de Barcelona, 08193 Bellaterra, Spain
E-mail: jar@iiia.csic.es

Library of Congress Control Number: 2005938890

CR Subject Classification (1998): I.2.11, K.4.4, C.2, H.3.4-5, H.5.3, I.2, J.1

LNCS Sublibrary: SL 7 – Artificial Intelligence

ISSN 0302-9743
ISBN-10 3-540-29737-5 Springer Berlin Heidelberg New York
ISBN-13 978-3-540-29737-6 Springer Berlin Heidelberg New York

Springer is a part of Springer Science+Business Media

springer.com

Printed in Germany

Typesetting: Camera-ready by author, data conversion by Scientific Publishing Services, Chennai, India
Printed on acid-free paper SPIN: 11575726 06/3142 5 4 3 2 1 0

Preface

The design of intelligent trading agents, mechanisms, and systems has received growing atttention in the agents and multiagent systems communities in an effort to address the increasing costs of search, transaction, and coordination which follows from the increasing number of Internet-enabled distibuted electronic markets. Furthermore, new technologies and supporting business models are resulting in a growing volume of open and horizontally integrated markets for trading of an increasingly diverse set of goods and services. However, growth of technologies for such markets requires innovative solutions to a diverse set of existing and novel technical problems which we are only beginning to understand. Specifically, distributed markets present not only traditional economic problems but also introduce novel and challenging computational issues that are not represented in the classic economic solution concepts. Novel to agent-mediated electronic commerce are considerations involving the computation substrates of the agents and the electronic institutions that supports trading, and also the human–agent interface (involving issues of preference elicitation, representation, reasoning, and trust). In sum, agent-mediated electronic trade requires principled design (from economics and game theory) and incorporates novel combinations of theories from different disciplines such as computer science, operations research, artificial intelligence, and distributed systems.

The collection of above-mentioned issues and challenges has crystallized into a new, consolidated agent research field that has become a focus of attention in recent years: *agent-mediated electronic commerce.*

The papers in this volume originate from the 6th Workshop on Agent-Mediated Electronic Commerce (AMEC VI), held in conjunction with the 3rd International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS) in July 2004. The AMEC VI workshop continued with the tradition and built upon the success of the previous AMEC workshops.

Thus, the primary goal of this workshop was to continue to bring together novel work from diverse fields such as computer science, operations research, artificial intelligence and distributed systems that focus on modeling, implementation, and evaluation of computational trading institutions and/or agent strategies over a diverse set of goods. Along this direction, areas of particular interest included:

– Distributed (scalable) algorithmic mechanism design
– Mechanisms for unreliable, dynamic, and asynchronous environments
– Mechanisms for incomplete and/or imperfect information environments
– Mechanisms for information goods and services
– Mechanisms for security, privacy, accounting, verification, and auditing
– Distributed (agent and mechanism) learning models
– Agents strategies in multi-institutional environments

- Economic and game theoretic specification, design, and analysis
- Bargaining, voting, and auction mechanisms
- Distributed reputation and trusted mechanisms
- User–agent interface design
- Agents that support bidding and negotiation
- Empirical evaluation of human–agent trading
- Eliciting human preferences and requirements
- Simulation and evaluation of properties of novel and complex mechanisms
- Goods, services, and contract description languages
- Mechanism description, verification, and testing languages
- Machine learning for mechanism identification problem
- Agent–mediated electronic system architectures and design principles
- Implemented agent-mediated electronic-commerce systems
- Mechanisms for business (supply chains, coalitions, and virtual enterprises)
- Mechanisms for Internet (congestion, routing, overlay, peer-to-peer, ad hoc networks)
- Mechanisms for novel applications

The workshop received a total of 39 submissions, from which 14 were selected for full presentation during the workshop. After the workshop, the authors were asked to submit their revised versions for publication in this volume. The result is that the volume contains 15 high-quality papers that can be regarded as representative of the field.

We have arranged the papers in the book around three major topics:

- Mechanism design
- Trading agents
- Tools

The first section contains eight papers dealing with a variety of issues on mechanism design. Dash et al. design an auction mechanism for allocating multiple goods when the buyers have interdependent valuations that turns out to be a generalization of the Vickrey–Clarkes–Grove (VCG) mechanism. Conitzer et al. study two related problems concerning the VCG payment scheme: the problem of revenue guarantees and that of collusion. Motivated also by the problems of the VCG payment scheme, Faltings introduces a new mechanism that sacrifices Pareto-efficiency to achieve budget balance while being both incentive compatible and individually rational. Also motivated by problems with side-payment schemes, Jurca et al. present a mechanism that discovers (in equilibrium) the true outcome of a transaction by analyzing the two reports coming from the agents involved in the exchange. Larson et al. lay out mechanism design principles for deliberative agents: agents whose actions are modelled as part of their strategies. Juda et al. devise an options-based market infrastructure that enables bidders to use a dominant, truthful strategy across multiple, sequential auctions. A different, more empirical approach is taken by Phelps et al., who report on an evolutionary game-theoretic comparison of two double-auction market designs.

Finally, Kelly focuses on the computational realization of mechanisms by analyzing the use of generalized knapsack solvers for multi-unit combinatorial auctions.

The second section brings together a collection of papers on trading agents in a wide range of trading scenarios. Debenham et al. propose an agent bidding strategy that is not based on traditional game theory models but rather information theoretic; maximum entropy inference to determine the agent's actions taking into account the uncertain data he handles in actual-world scenarios. Gerding et al. design and empirically compare different bargaining strategies for selling agents when negotiating with many buyers. They show that bilaterally exchanging multiple offers combined with a random offer generation mechanism suffices to closely approximating Pareto-efficiency. Furthermore, they also analyse the versatility of combined strategies. Sherstov et al. report on the development and analysis of three autonomous stock-trading agents within the framework of the Penn Exchange Simulator, a novel stock-trading simulator. Approaches based on reinforcement learning, trend following, and market making are presented, evaluated individually against a fixed opponent strategy, and analysed comparatively. Pardoe et al. research on strategies for a different type of scenario: the trading agent competition supply chain management. They study the selling strategy of a supply chain agent to generate the set of bids to customers in simultaneous reverse auctions that maximizes the agent's expected profit. Sarne et al. focus on the analysis of agents' strategies for the dual parallel search in partnership formation applications. As a framework application they choose the classic voice communication partnerships application in an electronic marketplace. The authors manage to provide efficient means for the agents to calculate their distributed equilibrium strategies so that they can improve their expected utilities.

Finally, the third section contains two papers dealing with tools aimed at supporting the enactment of digital markets. On the one hand, the work by Michael et al. focuses on a scripting language and a run-time system that allow for the specification and monitoring of market mechanisms using rights and obligations. On the other hand, Reyes-Moro et al. introduce a bundling procedure intended to assist buyers when deciding whether to auction a bundle of goods as a whole or as separate, smaller bundles.

We would like to conclude by thanking the members of the Program Committee. They were able to produce a large number of high-quality reviews in a very short time span. Furthermore, we would also like to thank the authors for submitting their papers to our workshop, as well as the attendees and panelists for their valuable insights and discussions. Needless to say that these helped authors to improve the revised papers published in this book.

June 2005

Peyman Faratin
Juan A. Rodríguez-Aguilar

Workshop Organization

Organizing Committee

Peyman Faratin	Massachusetts Institute of Technology, USA
Juan A. Rodríguez-Aguilar	IIIA-CSIC, Spain

Program Committee

Martin Bichler	Technical University of Munich, Germany
Chris Brooks	University of San Francisco, USA
Monique Calisti	Whitestein Technologies, Switzerland
Jesús Cerquides	University of Barcelona, Spain
Vincent Conitzer	Carnegie Mellon University, USA
Frank Dignum	Universiteit Utrecht, The Netherlands
Marc Esteva	IIIA-CSIC, Spain
Boi Faltings	EPFL, Switzerland
Shaheen Fatima	Liverpool University, UK
Andrew Gilpin	Carnegie Mellon University, USA
Amy Greenwald	Brown University, USA
Nick R. Jennings	Southampton University, UK
Jeff Kephart	IBM Research, USA
Kate Larson	Carnegie Mellon University, USA
Kevin Leyton-Brown	University of Stanford, USA
Asuman Ozdaglar	Massachusetts Institute of Technology, USA
Julian Padget	University of Bath, UK
David Pennock	Overture Services, USA
Joerg Muller	Siemens, Germany
Simon Parsons	Brooklyn College, USA
Antonio Reyes-Moro	iSOCO, Spain
Jeff Rosenschein	Hebrew University, Israel
Rahul Sami	Massachusetts Institute of Technology, USA
Onn Shehory	Carnegie Mellon University, USA
Yoav Shoham	Stanford University, USA
Gerry Tesauro	IBM Research, USA
William Walsh	IBM Research, USA
Mike Wellman	University of Michigan, USA
Steven Willmott	UPC, Spain
Makoto Yokoo	Kyushu University, Japan

Table of Contents

Mechanism Design

Trading Agents

Tools

Revenue Failures and Collusion in Combinatorial Auctions and Exchanges with VCG Payments*

Vincent Conitzer and Tuomas Sandholm

Computer Science Department, Carnegie Mellon University,
5000 Forbes Avenue, Pittsburgh, PA 15213
{conitzer, sandholm}@cs.cmu.edu

Abstract. In a *combinatorial auction*, there are multiple items for sale, and bidders are allowed to place a bid on a *bundle* of these items rather than just on the individual items. A key problem in this and similar settings is that of *strategic bidding*, where bidders misreport their true preferences in order to effect a better outcome for themselves. The *VCG payment scheme* is the canonical method for motivating the bidders to bid truthfully. We study two related problems concerning the VCG payment scheme: the problem of revenue guarantees, and that of collusion. The existence of such problems is known by many; in this paper, we lay out their full extent.

We study four settings: combinatorial forward auctions with free disposal, combinatorial reverse auctions with free disposal, combinatorial forward (or reverse) auctions without free disposal, and combinatorial exchanges. In each setting, we give an example of how additional bidders (colluders) can make the outcome much worse (less revenue or higher cost) under the VCG payment scheme (but not under a first price scheme); derive necessary and sufficient conditions for such an effective collusion to be possible under the VCG payment scheme; and (when nontrivial) study the computational complexity of deciding whether these conditions hold.

1 Introduction

In a *combinatorial auction*, there are multiple items for sale, and bidders are allowed to place a bid on a *bundle* of these items rather than just on the individual items. A rapidly growing body of computer science literature is devoted to the study of combinatorial auctions, and, to a lesser extent, variations of it, such as combinatorial reverse auctions (where the auctioneer seeks to procure certain items) and combinatorial exchanges (where bidders can offer goods for sale as well as express demand for goods—even within the same bid). One of the main reasons for the computer science community's interest in combinatorial auctions and exchanges is the hardness of the *clearing problem*. The clearing problem is to label bids as accepted or rejected to maximize the total value of the bids accepted (or, in the case of a reverse auction, to minimize their total value), under the natural constraint that the corresponding allocation of items does

* This work is supported in part by NSF under CAREER Award IRI-9703122, Grant IIS-9800994, ITR IIS-0081246, and ITR IIS-0121678.

P. Faratin and J.A. Rodríguez-Aguilar (Eds.): AMEC 2004, LNAI 3435, pp. 1–14, 2005.

not require more items than are available (or, in the case of a reverse auction, under the constraint that all the desired items are procured). For example, the combinatorial auction clearing problem is NP-complete [15] (even to approximate [16]). Much research has focused on developing worst-case exponential time algorithms as well as approximation algorithms for the clearing problem [12,16,17,6].

Another key problem in auctions and exchanges (combinatorial or not) is that in general, the bidders may not bid their true valuations for the goods. For example, under a first-price payment rule, where bidders pay the value of their accepted bids, a bidder that bids her true valuation is entirely indifferent whether her bid is accepted or not. Thus, in order to benefit from the auction or exchange at all, bidders necessarily need to "shave" their bids, that is, report a lower value than their true value. The problem with untruthful bidding is that the clearing algorithm can only base the final allocation of the goods on the reported valuations, and thus the final allocation may not be optimal relative to the bidders' true valuations. Thus, economic efficiency may be lost. Additionally, by a result known as the *revelation principle*, for any nontruthful mechanism, there is a truthful mechanism that performs just as well (under some assumptions on the strategic behavior of the bidders) [11]. It turns out, however, that by changing the payment rule, it is possible to motivate bidders to report their true valuations. The best-known such payment rule is the Vickrey-Clarke-Groves (VCG) scheme [18,3,8]. Here, a bidder must pay the total value of the bids that would have been accepted if she had not participated, minus the total value of bids that did get accepted (excluding her own bids). Because the bidder pays the externality she imposes on the other bidders (based on their reported valuations), she will bid to maximize the value of the final allocation—measured by her true valuation and the others' reported valuations. If the clearing algorithm always finds the optimal allocation, bidding her true value will always effect this.[1] Not only is the VCG payment scheme the best-known payment scheme for motivating truthfulness, if the setting is general enough, given certain requirements, it (or its generalization to *Groves mechanisms*) is also the only one [7,9,20].

Unfortunately, there are also many problems with VCG mechanisms. They introduce the problem of lying auctioneers; they are bad from a privacy perspective; they are vulnerable to collusion; and they may lead to low revenue for the auctioneer. (In this paper, we will focus on the last two problems, which are closely related.) While most researchers in combinatorial auctions and exchanges acknowledge these problems, we believe that their severity may not be fully appreciated. For one, the Vickrey auction for a single item (the second-price sealed-bid auction) has some nice properties that unfortunately do not generalize to multi-item settings. For example, in a single-item Vickrey auction, it is not possible for colluders to obtain the item at a price less than the bid of any other bidder. Additionally, for the single-item Vickrey auction, various types of

[1] Of course, in general, the clearing problem may be too hard to always be solved optimally; and in general, the VCG scheme will not motivate bidders to bid truthfully when the final allocation may be suboptimal. A growing body of research is dedicated to finding approximation schemes for the (forward auction) clearing problem that still motivate bidders to bid truthfully, or attempting to prove that this is impossible in general [13,10,2,9]. Throughout this paper we will assume that the auctions and exchanges are cleared optimally.

revenue equivalence with (for example) first-price sealed-bid auctions hold. As we will show, in the multi-item setting these properties do not hold at all (and can be violated to an arbitrary extent). We hope that a greater awareness of these issues will help bridge the gap between theory and practice in mechanism design for combinatorial auctions and exchanges.

To that end, in this paper we give some detailed worst-case results about collusion and revenue. For the various variants of combinatorial auctions and exchanges, we study the following single problem that relates both issues under consideration:

Given some of the bids, how "bad" can the remaining bidders make the outcome?

"Bad" here means that the bidders are paid an inordinately large amount, or pay an inordinately small amount, relative to the goods they receive and/or provide. This is closely related to the problem of making any revenue guarantees to the auctioneer. But it is also the collusion problem, if we conceive of the remaining bidders as colluders.

As it will turn out, our fundamental problem is often computationally hard. Computational hardness here is a double-edged sword. On the one hand, if the problem is hard, collusion may not occur (or to a lesser extent) because the colluders cannot find the (most) beneficial collusion. On the other hand, if the problem is hard, it is difficult to make strong revenue guarantees to the auctioneer.

2 The VCG Mechanism

All the results in this paper hold even when all bidders are *single-minded*: that is, they are interested in only one bundle of items (or, in the case of a reverse auction, can provide only one bundle of items). In this case, every bid corresponds to a unique *utility function*, namely the one for which this bid would have been a truthful revelation of the bidder's valuation for the bundle.

The VCG payment scheme proceeds as follows: accept bids so that the resulting allocation maximizes the sum of the bidders' utilities as implied by the bids, not taking payments into account. (This is simply maximizing the sum of the values of the accepted bids in a forward auction, or minimizing their sum in a reverse auction.) Call this sum of utilities a. Then, to determine winning bidder i's payment, remove that bidder's bid, and see what the maximum sum of the utilities (disregarding payments) would have been with only the remaining bids. Call this sum of utilities b_i. Winning bidder i then must pay the second sum of utilities, minus the original sum of utilities of the other bidders—that is, the externality she imposed on the other bidders. (Thus, if the value of winning bidder i's bid is v_i, the payment is $b_i - a + v_i$.) We observe that this payment can be negative, if the bidder's presence actually makes the other bidders better off (disregarding the payments).[2] For instance, in a reverse auction where goods are disposable, each winning bidder's payment will be nonpositive (the other bidders need to supply fewer items when this bidder is present, so (disregarding payments) this bidder makes the others better off).

[2] This can only happen if the bidder's presence actually makes more allocations to the other bidders possible. For instance, in a forward auction with free disposal, when we remove a bidder we can always throw away the items allocated to her and keep the allocation to all other bidders the same. Thus, payments cannot be negative in this setting.

3 Combinatorial (Forward) Auctions

3.1 Review

In a combinatorial auction, there is a set of items $I = \{A_1, A_2, \ldots, A_m\}$ for sale. A bid takes the form $b = (B, v)$, where $B \subseteq I$ and $v \in \Re$. The clearing problem is to label bids as accepted or rejected, to maximize the sum of the values of the accepted bids, under the constraint that no item occurs in more than one accepted bid. (This is assuming *free disposal*, that is, items do not have to be allocated to anyone.) We say a bid is *truthful* if the value attached to the bundle is the bidder's utility for that bundle.

3.2 Motivating Example

Consider an auction with two items, A and B. Suppose we have collected two bids (from different bidders), both $(\{A, B\}, N)$. If these are the only two bids, one of the bidders will be awarded both the items and, under the VCG payment scheme, have to pay N. However, suppose two more bids (by different bidders) come in: $(\{A\}, N+1)$ and $(\{B\}, N+1)$. Then these bids will win. Moreover, neither bidder will have to pay anything!

This example demonstrates a number of issues. First, the addition of additional bidders may actually decrease the auctioneer's revenue from an arbitrary amount to 0. Second, the VCG mechanism is not revenue equivalent to the sealed-bid first-price mechanism in combinatorial auctions, even when all bidders' true valuations are common knowledge—unlike in the single-item case. (The first-price mechanism will generate positive expected revenue for these valuations; we omit the proof because of space constraint.) Third, even when the other bidders by themselves would generate nonnegative revenue for the auctioneer under the VCG payment scheme, it is possible that two colluders can bid so as to receive all the items without paying anything.

The following proposition sums up the properties of this example.

Proposition 1. *In a forward auction (even with only 2 items), there exists a family of instances (sets of bids) such that: 1. The winning bidders pay nothing under the VCG payment scheme; 2. If the winning bids are removed, the remaining bids actually generate revenue N under the VCG payment scheme; 3. If these bids were truthful (as we would expect under VCG), then if we had run a first-price sealed-bid auction instead (and the bidders knew each other's true valuations), any equilibrium would have generated revenue $\Theta(N)$.*

3.3 Characterization

We now proceed to characterize the settings where the colluders can receive all the items for free.

Lemma 1. *If the colluders receive all the items at cost 0, then for any positive bid on a bundle of items by a noncolluder, at least two of the colluders receive an item from this bundle.*

Proof. Suppose that for some positive bid b on a bundle B by a noncolluder i, one of the colluders c receives all the items in B (and possibly others). Then, in the auction where we remove that colluder's bids, one possible allocation gives every remaining bidder all the goods that bidder received in the original auction; additionally, it gives i all the items in bundle B; and it disposes of all the other items c received in the original auction. With this allocation, the total value of the accepted bids by bidders other than c is at least $v(b)$ more than in the original auction. Because the total value obtained in the new auction is at least the value of this particular allocation, it follows that c imposes a negative externality of at least $v(b)$ on the other bidders, and will pay at least $v(b)$. ■

Lemma 2. *Suppose all the items in the auction can be divided among the colluders in such a way that for any positive bid on a bundle of items by a noncolluder, at least two of the colluders receive an item from this bundle. Then the colluders can receive all the items at cost 0.*

Proof. For the given partition of items among the noncolluders, let each colluder place a bid with an extremely large value on the bundle consisting of the items assigned to him in the partition. (For instance, twice the sum of the values of all noncolluders' bids.) Then, the auction will clear awarding each colluder the items assigned to him by the partition. Moreover, if we remove the bids of one of the colluders, all the remaining colluders' bids will still win—and thus none of the noncolluders' bids will win, because each such bid requires items assigned to at least two colluders by the partition (and at least one of them is still in the auction and wins these items). Thus, each colluder (individually) imposes no externality on the other bidders. ■

Combining these two lemmas, we get:

Theorem 1. *The colluders can receive all the items for free if and only if it is possible to divide the items among the colluders in such a way that for any (nonzero) bid by a noncolluder, the items in that bid are spread across at least two colluders.*

3.4 Complexity

Definition 1 (DIVIDE-SUBSETS). *Suppose we are given a set S, as well as a collection of subsets of it, $R = \{S_1, \ldots, S_q\}$. We are asked whether S can be partitioned into n parts $T_1, T_2, \ldots, T_n$ so that no subset $S_i \in R$ is contained in one of these parts.*

Theorem 2. *DIVIDE-SUBSETS is NP-complete, even when $n = 2$.*

Complexity proofs are omitted because of space constraint.

4 Combinatorial Reverse Auctions

4.1 Review

In a combinatorial reverse auction, there is a set of items $I = \{A_1, A_2, \ldots, A_m\}$ to be procured. A bid takes the form $b = (B, v)$, where $B \subseteq I$ and $v \in \Re$. The clearing problem is to label bids as accepted or rejected, to minimize the sum of the values of the accepted bids, under the constraint that each item occurs in at least one accepted bid. (This is assuming *free disposal*, that is, items do not have to be allocated to anyone.) We say a bid is *truthful* if the value attached to the bundle is the bidder's cost for providing that bundle.

4.2 Motivating Example

Consider a reverse auction with m items, $A_1, A_2, \ldots, A_m$. Suppose we have collected two bids (from different bidders), both $(\{A_1, A_2, \ldots, A_m\}, N)$. If these are the only two bids, one of the bidders will be chosen to provide all the goods, and, under the VCG payment scheme, be paid N. However, suppose m more bids (by different bidders) come in: $(\{A_1\}, 0), (\{A_2\}, 0), \ldots, (\{A_m\}, 0)$. Then these bids will win. Moreover, each bidder will be paid N under the VCG payment scheme. (Without this bidder, we would have had to accept one of the original bids.) Thus, the total payment that needs to be made is mN.

Again, this example demonstrates a number of issues. First, the addition of additional bidders may actually increase the total amount that the auctioneer needs to pay. Second, the VCG mechanism requires much larger payments than a first-price auction in the case where all bidders know each others' valuations (and the equilibrium is in pure strategies[3]). (The first-price mechanism will not require a total payment of more than N for these valuations in any pure-strategy equilibrium; we omit the proof because of space constraint.)

Third, even when the other bidders by themselves would allow the auctioneer to procure the items at a low cost under the VCG payment scheme, it is possible for m colluders to get paid m times as much for all the items.

The following proposition sums up the properties of this example.

Proposition 2. *In a reverse auction, there exists a family of instances (sets of bids) such that: 1. The winning bidders are paid mN under the VCG payment scheme; 2. If the winning bids are removed, the remaining bids allow the auctioneer to procure everything at a cost of only N; 3. If these bids were truthful (as we would expect under VCG), then if we had run a first-price sealed-bid reverse auction instead (and the bidders knew each other's true valuations), any equilibrium in pure strategies would have required total payment of at most N. (However, there are also mixed-strategy equilibria with arbitrarily large expected total payment.)*

4.3 Characterization

Letting N be the sum of the values of the accepted bids when all the colluders' bids are taken out, it is clear that no colluder can be paid more than N. (With the colluder's bid, the sum of the values of others' accepted bids is still at least 0; without it, it can be at most N, because in the worst case the auctioneer can accept the bids that would be accepted if none of the colluders are present.) In this subsection, we will identify a necessary and sufficient condition for the colluders to be able to each receive N.

Lemma 3. *If a colluder receives N, then the items that she has to provide cannot be covered by a set of noncolluders' bids with cost less than N.*

[3] Perhaps surprisingly, the first-price combinatorial reverse auction for this example (with commonly known true valuations corresponding to the given bids) actually has mixed-strategy equilibria with arbitrarily high expected payments. We omit the proof because of space constraint.

Proof. If they could be covered by such a set, we could simply accept this set of bids (including those that were accepted already) rather than the colluder's bid, and increase the total cost by less than N. Thus, the colluder's VCG payment is less than N. ■

Thus, in order for each of the n colluders to be able to receive N, it is necessary that there exist n disjoint subsets of the items, each of which cannot be covered with a set of noncolluders' bids with total value less than N. The next lemma shows that this condition is also sufficient.

Lemma 4. *If there are n disjoint sets of items $R_1, \ldots, R_n$, each of which cannot be covered by a set of noncolluders' bids with cost less than N, then n colluders can be paid N each.*

Proof. Let colluder i (for $i < n$) bid $(R_i, 0)$, and let colluder n bid $(R_n \cup (S - \bigcup_i R_i), 0)$. Then the total cost of all accepted bids with all the colluders is 0; but when one colluder is omitted, the items she won cannot be covered at a cost less than N (because her bid contained one of the R_i). Thus, each colluder's VCG payment is N. ■

The next lemma shows that the necessary and sufficient condition is equivalent to being able to *partition* all the items into n sets, so that no element of the partition can be covered by noncolluders' bids with total value less than N. That is, we can restrict our attention to the case where the sets exhaust all the items.

Lemma 5. *The condition of Lemma 4 is satisfied if and only if it is possible to partition the items into $T_1, \ldots, T_n$ such that no T_i can be covered by a set of noncolluders' bids with cost less than N.*

Proof. The "if" part is trivial: given T_i that satisfy the condition of this lemma, simply let $R_i = T_i$. For the "only if" part, given R_i that satisfy the condition of Lemma 4, let $T_i = R_i$ for $i < n$, and $T_n = R_n \cup (S - \bigcup_i R_i)$. We observe that this last set can also not be covered at a cost of less than N because it contains R_n. ■

Combining all the lemmas, we get:

Theorem 3. *The n colluders can receive a payment of N each (simultaneously), where N is the sum of the values of the accepted bids when all the colluders' bids are removed, if and only if it is possible to partition the items into $T_1, \ldots, T_n$ such that no T_i can be covered by a set of noncolluders' bids with cost less than N.*

4.4 Complexity

Definition 2 (CRITICAL-PARTITION). *We are given a set of items S, a collection of bids (S_i, v_i) where $S_i \subseteq S$ and $v_i \in \Re$, and a number n. Say that the* cost *of a subset of these bids is the sum of their v_i; and that the cost $c(T)$ of a subset $T \subseteq S$ is the lowest cost of any subset of the bids whose S_i cover T.*

We are asked whether there exists a partition of S into n disjoint subsets $T_1, T_2, \ldots, T_n$, such that for any $1 \leq i \leq n$, $c(T_i) = c(S)$.

Theorem 4. *Even when the bids are so that a partition $T_1, \ldots, T_n$ is a solution if and only if no set $S - T_i$ covers all items in a bid, CRITICAL-PARTITION is NP-complete (even with $n = 2$).*

5 Combinatorial Forward or Reverse Auctions Without Free Disposal

5.1 Review and Equivalence

A combinatorial forward auction without free disposal is exactly the same as one with free disposal, with the exception that every item must be allocated to some bidder. Here, bids with a *negative* value may still be useful, as they allow us to remove some of the items—which may allow us to accept better bids for the remaining items.

Similarly, a combinatorial reverse auction without free disposal is exactly the same as one with free disposal, with the exception that no additional items can be procured. Here, bids with a *negative* value may occur—the (nondisposable) item may be a liability to the bidder and the bidder could be happy to be rid of it.

In both cases, we seek to identify a subset of the bids that constitutes an exact cover of the items—in the former case we try to maximize the sum of the values in the cover, in the latter, to minimize it. Because bids may carry both negative and positive values, we may simply switch the plus and minus signs in the reverse auction, and the optimization problem will be the same. Moreover, after the switch, the value of a bid in a reverse auction is exactly the utility the bidder would derive from having that bid accepted (disregarding payments). It follows that we can simply run a reverse auction as a forward auction by switching the signs. This makes intuitive sense: when items are nondisposable, they can be either assets or liabilities.

In the rest of this section, we will discuss forward auctions without free disposal only, because results immediately carry over to reverse auctions without free disposal.

5.2 Motivating Example

Consider a forward auction with two nondisposable items, A_1 and A_2 . Suppose we have collected two bids (from different bidders): both $(\{A_1, A_2\}, N)$. If these are the only two bids, one of the bidders will be awarded both the items and, under the VCG payment scheme, have to pay N. However, suppose two more bids (by different bidders) come in: $(\{A_1\}, N + M)$ and $(\{A_2\}, N + M)$, with $M > 0$. Then these bids will win. Moreover, because without free disposal, we cannot accept either of these bids without the other, each of these bidders will *be paid* M under the VCG payment scheme!

Again, the example demonstrates a number of issues. First, additional bidders may change the auctioneer's revenue from an arbitrarily large positive amount to an arbitrarily large negative amount (an arbitrarily large cost). Second, the VCG mechanism may require arbitrarily large payments from the auctioneer, where a first-price auction would actually generate revenue for the auctioneer, even in the case where all bidders know each other's valuations, for any equilibrium in pure strategies.[4] (The first-price mechanism will generate a revenue of at least N for these valuations in any pure-strategy equilibrium; we omit the proof because of space constraint.)

[4] Similarly to the case of the combinatorial reverse auction with free disposal, there are mixed-strategy equilibria in the first-price auction where the auctioneer is forced to make arbitrarily large payments—we omit the proof because of space constraint.

Third, even when the other bidders by themselves would generate positive revenue for the auctioneer under the VCG payment scheme, it is possible that two colluders can make the auctioneer pay each of them an arbitrarily large amount.

The following proposition sums up the properties of this example.

Proposition 3. *In a forward auction without free disposal (even with only two items), there exists a family of instances (sets of bids) such that: 1. Each winning bidder is paid M under the VCG payment scheme (where M depends only on the winners' bids); 2. If the winning bids are removed, the remaining bids actually generate revenue N to the auctioneer under the VCG payment scheme; 3. If these bids were truthful (as we would expect under VCG), then if we had run a first-price sealed-bid auction instead (and the bidders' knew each other's true valuations), any equilibrium in pure strategies would have generated a revenue N. (However, there are mixed-strategy equilibria with arbitrarily large cost to the auctioneer.)*

5.3 Characterization

In this subsection, we will identify a necessary and sufficient condition for the colluders to be able to each receive an arbitrary amount.

Lemma 6. *If each colluder receives a payment of more than $2\sum_d |v(b_d)|$ (where d ranges over the noncolluders), then for each colluder c, the set of all items awarded to either her or a noncolluder ($A_c \cup \bigcup_d A_d$, where A_b is the set of items awarded to bidder b and d ranges over the noncolluders) cannot be covered exactly with bids from the noncolluders.*

Proof. Say that the sum of the values of accepted noncolluder bids is D (which may be negative). Suppose that for one colluder c, the set of all items awarded to either her or a noncolluder ($A_c \cup \bigcup_d A_d$) can be covered by a set of noncolluder bids of combined value C (which may be negative). Then removing colluder c can make the allocation at most $|C|+|D|$ worse to the other bidders (relative to their reported valuations), because we could simply accept the bids of combined value C and no longer accept the bids of combined value D, and keep the rest of the allocation the same. Thus, under VCG, that colluder should be rewarded at most $|C| + |D| \leq 2\sum_d |v(b_d)|$. ■

Thus, in order for each colluder to be able to receive an arbitrarily large payoff, it is necessary that there are n disjoint subsets of the items such that, when taken together with the remaining items, no such subset can be covered exactly by the noncolluders' bids (while the set of remaining items can, by itself, be covered exactly by the noncolluders' bids). The next lemma shows that this condition is also sufficient.

Lemma 7. *If it is possible to partition the items into $T_1, \ldots, T_n, T_{n+1}$ such that for no $1 \leq i \leq n$, $T_i \cup T_{n+1}$ can be covered exactly with bids from the noncolluders; and T_{n+1} can be covered exactly with bids from the noncolluders; then for any $M > 0$, n colluders can place additional bids such that each of them receives at least M.*

Proof. Let colluder i place a bid $(T_i, M + 3\sum_d |v(b_d)|)$ (where d ranges over the noncolluders). All these bids will be accepted, because it is possible to do so by also accepting the noncolluder bids that cover T_{n+1} exactly; and these noncolluder bids will have a combined value of at least $-\sum_d |v(b_d)|$, so that the sum of the values of all accepted bids is at least $(3n-1)\sum_d |v(b_d)|) + nM$. (We observe that if we do not accept all of the colluder bids, the sum of the values of all accepted bids is at most $(3(n-1)+1)\sum_d |v(b_d)|) + (n-1)M = (3n-2)\sum_d |v(b_d)|) + (n-1)M$.) Now, if the bid of colluder i is removed, it is no longer possible to accept all the remaining $n-1$ colluder bids, because $T_i \cup T_{n+1}$ cannot be covered exactly with noncolluder bids. It follows that the total value of all accepted bids when i's bid is removed can be at most $(3(n-2)+1)\sum_d |v(b_d)|) + (n-2)M$. When i's bid is not omitted, the sum of the values of all accepted bids other than i's is at least $(3(n-1)-1)\sum_d |v(b_d)|) + (n-1)M$. Subtracting the former quantity from this, we get that the VCG payment to i is at least $\sum_d |v(b_d)|) + M$. ■

The next lemma shows that the necessary and sufficient condition is equivalent to being able to partition all the items into n sets, so that no element of the partition can be covered exactly by the noncolluders' bids. That is, we can restrict our attention to the case where $T_{n+1} = \{\}$.

Lemma 8. *Lemma 7's condition is satisfied if and only if the items can be partitioned into $R_1, \ldots, R_n$ such that no R_i can be covered exactly with bids from the noncolluders.*

Proof. For the "if" part: given R_i that satisfy the condition of this lemma, let $T_i = R_i$ for $i \leq n$, and $T_{n+1} = \{\}$. Then no $T_i \cup T_{n+1} = R_i$ can be covered exactly with bids from the noncolluders, and $T_{n+1} = \{\}$ can trivially be covered exactly with noncolluder bids. For the "only if" part: given T_i that satisfy the condition of Lemma 7, $R_i = T_i$ for $i < n$, and let $R_n = T_n \cup T_{n+1}$. That R_n cannot be covered exactly by noncolluder bids now follows directly from the conditions of Lemma 7. But also, no R_i with $i < n$ can be covered exactly: because if it could, then we could cover $T_i \cup T_{n+1} = R_i \cup T_{n+1}$ using the bids that cover R_i exactly together with the bids that cover T_{n+1} exactly (which exist by the conditions of Lemma 7). ■

Combining all the lemmas, we get:

Theorem 5. *The n colluders can receive a payment of at least M each (simultaneously), where M is an arbitrarily large number, if and only if it is possible to partition the items into $R_1, \ldots, R_n$ such that no R_i can be covered exactly with bids from the noncolluders.*

5.4 Complexity

Definition 3 (COVERLESS-PARTITION). *We are given a set S and a collection of subsets $S_1, S_2, \ldots, S_q \subseteq S$. We are asked whether there is a partition of S into subsets $T_1, T_2, \ldots T_n \subseteq S$ such that no T_i can be covered exactly by the S_i. (That is, each cover of one T_i will either use two intersecting S_i, or include some elements outside T_i.)*

Theorem 6. *Even if there is a singleton S_i for all but two elements a and b, and $S_0 = \{a, b\}$, and $n = 2$, COVERLESS-PARTITION is NP-complete.*

5.5 An Easier Collusion Problem

So far, we phrased the collusion problem so that *each* colluder should receive M, where M is an arbitrary amount. An easier problem for the colluders is to make sure that *together*, they receive M, where M is an arbitrary amount. Such a collusion is less stable (because some of the colluders may be receiving very little). Nevertheless, as we will show, such collusions are possible whenever a weak (and easily verified, given the others' bids) condition holds: at least one item has no singleton bid on it. (A singleton bid is a bid on only one item.) We first show that this condition is necessary.

Lemma 9. *If at least one colluder receives a payment of more than* $\sum_d |v(b_d)|$ *(where* d *ranges over the noncolluders), there is at least one item* s *on which no noncolluder places a singleton bid.*

Proof. If each item has a singleton noncolluder bid placed on it, then when we remove a colluder's bid, we can simply cover all the items in it with singleton bids (with a combined value of at least $-\sum_d |v(b_d)|$), and leave the rest of the allocation unchanged. It follows that the VCG payment to the colluder can be at most $\sum_d |v(b_d)|$). ■

We now show that the condition is sufficient.

Lemma 10. *If there is at least one item* s *on which no noncolluder places a singleton bid, then if one colluder bids* $(\{s\}, 0)$*, and the other colluder bids* $(S - \{s\}, M + 2\sum_d |v(b_d)|)$ *(for* $M > 0$*), the total payment to the colluders is at least* M*.*

Proof. The colluders' bids will be the only accepted ones (because colluder 2's bid has a greater value than all other bids combined). If we removed colluder 2's bid, the total value of the accepted bids would be at most $\sum_d |v(b_d)|$), so colluder 2 will pay at most this much under the VCG scheme. If we removed colluder 1's bid, colluder 2's bid could no longer be accepted (because $\{s\}$ cannot be covered by itself), and thus the total value of the accepted bids could be at most $\sum_d |v(b_d)|$). It follows that colluder 1 is paid at least $M + \sum_d |v(b_d)|$). So the total payment to the colluders is at least M. ■

Combining the two lemmas, we get the desired result:

Theorem 7. *Two (or more) colluders can receive a total payment of* M*, where* M *is an arbitrarily large number, if and only if there is at least one item that has no singleton bid placed on it by a noncolluder.*

6 Combinatorial Exchanges

6.1 Review

In a combinatorial exchange, there is a set of items $I = \{A_1, A_2, \ldots, A_m\}$ that can be traded. A bid takes the form $b = (\lambda_1, \ldots, \lambda_m, v)$, where $\lambda_1, \ldots, \lambda_m, v \in \Re$ (possibly negative). The clearing problem is to label bids as accepted or rejected, so that the sum of the accepted vectors has its first m entries ≤ 0, to maximize the last entry of the sum of the accepted vectors. (This is assuming *free disposal*, that is, items do not have to be allocated to anyone.) We say a bid is *truthful* if the value attached to the λs is the bidder's utility for receiving λ_i units of good i, for every i. We will also use the notation $(\{(A_{i_1}, \lambda_{i_1}), (A_{i_2}, \lambda_{i_2}), \ldots, (A_{i_k}, \lambda_{i_k})\}, v)$ for representing a bid in which λ_{i_j} units of item A_{i_j} are demanded (and 0 units of each item that is not mentioned).

6.2 Impossibility of a Revenue Guarantee (or a Bound on Collusion)

In a combinatorial exchange with at least two items A_1 and A_2, let q_1 (respectively, q_2) be the total number of units of A_1 (respectively, A_2) offered for sale in bids so far (or, in bids by noncolluders). Now consider the following two bids (possibly by colluders): $(\{(A_1, q_1+1), (A_2, -q_2-1)\}, M+\sum_d |v(b_d)|)$ and $(\{(A_1, -q_1-1), (A_2, q_2+1)\}, M+\sum_d |v(b_d)|)$, where $M > 0$ and d ranges over the original (noncolluding) bids. Both these bids will be accepted (for otherwise, the total value of the accepted bids could be at most $M + 2\sum_d |v(b_d)| < 2(M + \sum_d |v(b_d)|)$). Moreover, if we remove one of these two bids, the other cannot be accepted (because its demand cannot be met), so the total value of the accepted bids can be at most $\sum_d |v(b_d)|)$. It follows that the VCG payment to each of these two bidders is at least M. This proves the following theorem:

Theorem 8. *In a combinatorial exchange with at least two items (even with free disposal), for any set of bids by noncolluders, two colluders can place bids so that each of them will receive at least M, where M is an arbitrary amount. Moreover, each one receives exactly the items the other provides, so that their net contribution in terms of items is nothing.*

7 Conclusion

The VCG mechanism is the canonical payment scheme for motivating the bidders to bid truthfully in combinatorial auctions and exchanges; if the setting is general enough, under some requirements, it is the only one. Unfortunately, it also introduces many problems. In this paper, we focused on the related problems of revenue guarantees and bidder collusion. While many researchers in the area are aware of the existence of these problems under the VCG mechanism, their full extent was (to our knowledge) unknown. Besides studying how severe these problems can be, we also studied the computational problem of deciding how "bad" additional bids could make the final outcome. This problem is of interest both to the auctioneer, for the purpose of coming up with a revenue guarantee based on a subset of the bids that she knows; as well as to colluding bidders, for the purpose of deciding how to bid most effectively given the noncolluders' bids. Hardness of this problem is undesirable because of the first purpose, but desirable because of the second purpose.

For combinatorial forward auctions with free disposal under VCG, we showed that as few as two colluders may receive all the items at zero cost, even in cases where the noncolluders' bids by themselves would have generated positive revenue for the auctioneer; whereas running a first-price auction instead would have generated at least a constant fraction of that revenue. We gave a necessary and sufficient condition for the colluders to be able to receive all the items at zero cost. We showed that deciding whether this condition is satisfied is NP-complete even with only two colluders.

For combinatorial reverse auctions with free disposal under VCG, we showed that n colluders may be able to receive a total payment of nN (N each), where N is the total payment the auctioneer would have to make if the colluders were not present; whereas running a first-price auction instead would have required a cost of only N in any pure-strategies equilibrium. (Though arbitrarily bad mixed-strategy equilibria exist.) We gave a necessary and sufficient condition for n colluders to be able to receive a payment N each. We showed that deciding whether this condition is satisfied is NP-complete.

For combinatorial forward auctions without free disposal (which we showed are equivalent to combinatorial reverse auctions without free disposal), we showed that the colluders may be able to each receive (simultaneously) an arbitrarily high payment; whereas running a first-price auction instead would have given the same revenue to the auctioneer as the VCG auction without the colluders. We showed a necessary and sufficient condition for n colluders to be able to each (simultaneously) receive an arbitrarily large payment. We showed that deciding whether this condition is satisfied is NP-complete even with only two colluders. We also showed that colluders can make the sum of payments to them (as opposed to all individual payments simultaneously) arbitrarily large if and only if at least one item does not have a singleton bid on it (only two colluders are necessary).

Finally, for combinatorial exchanges (with or without free disposal), we showed that two colluders can *always* each get an arbitrarily large payment (simultaneously), with each colluder receiving exactly what the other provides (so their net contribution is zero).

8 Future Research

We believe future research should address the revenue and collusion problems that VCG introduces in combinatorial auctions and exchanges. A few avenues are available here. First, it may be possible to design other truthful mechanisms (possibly Groves mechanisms) which do not have these issues. They could be designed by hand for general settings; alternatively, using automated mechanism design [4,5], they could be designed by a computer for the specific setting at hand. (For example, no-collusion constraints could be given to the automated mechanism design software in the same manner that incentive-compatibility constraints are given to it now.) Alternatively, we may switch our attention to mechanisms that are not truthful direct revelation mechanisms. As we demonstrated, simple first-price mechanisms (which are decidedly nontruthful) already avoid some of the pitfalls of VCG pointed out in this paper. All of these issues should also be studied in the context of *iterative* combinatorial auctions [14,19,1].

References

1. Lawrence Ausubel and Paul Milgrom. Ascending auctions with package bidding. *Frontiers of Theoretical Economics*, 1, 2002. No. 1, Article 1.
2. Yair Bartal, Rica Gonen, and Noam Nisan. Incentive compatible multi-unit combinatorial auctions. In *Theoretical Aspects of Rationality and Knowledge (TARK)*, 2003.
3. Ed H. Clarke. Multipart pricing of public goods. *Public Choice*, 11:17–33, 1971.
4. Vincent Conitzer and Tuomas Sandholm. Complexity of mechanism design. In *UAI*, 2002.
5. Vincent Conitzer and Tuomas Sandholm. Self-interested automated mechanism design and implications for optimal combinatorial auctions. In *ACM-EC*, 2004.
6. Rica Gonen and Daniel Lehmann. Optimal solutions for multi-unit combinatorial auctions: Branch and bound heuristics. In *ACM-EC*, 2000.
7. J Green and J-J Laffont. Characterization of satisfactory mechanisms for the revelation of preferences for public goods. *Econometrica*, 45:427–438, 1977.
8. Theodore Groves. Incentives in teams. *Econometrica*, 41:617–631, 1973.
9. Ron Lavi, Ahuva Mu'Alem, and Noam Nisan. Towards a characterization of truthful combinatorial auctions. In *FOCS*, 2003.
10. Daniel Lehmann, Lidian Ita O'Callaghan, and Yoav Shoham. Truth revelation in rapid, approximately efficient combinatorial auctions. *JACM*, 49(5):577–602, 2002.
11. Andreu Mas-Colell, Michael Whinston, and Jerry R. Green. *Microeconomic Theory*. Oxford University Press, 1995.
12. Noam Nisan. Bidding and allocation in combinatorial auctions. In *ACM-EC*, 2000.
13. Noam Nisan and Amir Ronen. Computationally feasible VCG mechanisms. *ACM-EC*, 2000.
14. David Parkes. iBundle: An efficient ascending price bundle auction. In *ACM-EC*, 1999.
15. Michael Rothkopf, Aleksandar Pekeč, and Ronald Harstad. Computationally manageable combinatorial auctions. *Management Science*, 44(8):1131–1147, 1998.
16. Tuomas Sandholm. Algorithm for optimal winner determination in combinatorial auctions. *Artificial Intelligence*, 135:1–54, 2002.
17. Tuomas Sandholm, Subhash Suri, Andrew Gilpin, and David Levine. CABOB: A fast optimal algorithm for combinatorial auctions. In *IJCAI*, 2001.
18. W Vickrey. Counterspeculation, auctions, and competitive sealed tenders. *J. Finance*, 1961.
19. Peter Wurman and Michael Wellman. AkBA: A progressive, anonymous-price combinatorial auction. In *ACM-EC*, 2000.
20. Makoto Yokoo. The characterization of strategy/false-name proof combinatorial auction protocols: Price-oriented, rationing-free protocol. In *IJCAI*, 2003.

A Mechanism for Multiple Goods and Interdependent Valuations

Rajdeep K. Dash, Alex Rogers, and Nicholas R. Jennings

School of Electronics and Computer Science, University of Southampton,
Southampton, SO17 1BJ, UK
Fax: +44 (0) 23 8059 2865
{rkd02r, ar2, nrj}@ecs.soton.ac.uk

Abstract. This paper reports on the design of an auction mechanism for allocating multiple goods when the buyers have interdependent valuations. We cast the problem as a multi-agent system consisting of selfish, rational agents and develop an auction mechanism which is efficient, incentive compatible and individually rational. We first discuss the necessary assumptions that *any* mechanism developed for this scenario should satisfy so as to achieve the aforementioned properties. We then present our mechanism and show how it is a generalisation of the Vickrey-Clarke-Groves mechanism.

1 Introduction

Auction mechanisms have been proposed as a solution for a variety of task and resource allocation problems that occur in multi-agent systems (MAS) [2,11,13]. A common requirement of these systems is that agents of limited complexity can participate fairly without the risk of being exploited by more complex agents indulging in strategic behaviour. As such, *direct* mechanisms which are *incentive compatible* are most often considered, as under these mechanisms, the equilibrium strategy of all agents is simply to truthfully report their type to the auctioneer[1]. Of these mechanisms, the Vickrey-Clarke-Groves (VCG) mechanism is the most widely studied because in addition to the above properties, it ensures that the resulting allocation is efficient (i.e. it maximises the global welfare) and that the mechanism is *individually-rational* (i.e. it guarantees any agent joining the mechanism derives a non-negative utility) [5].

However, a key shortcoming of the VCG mechanism is that it relies on private independent valuations to achieve these desirable properties. Such private valuations arise when an agent forms its valuation of the goods or services based solely on its own observation or signal (e.g. the value of a particular car to an agent depends solely on the agent's own perception of the car's use and is not dependent on the valuations of other bidders). However, the more general case is that valuations are actually *interdependent* (e.g. if the agents' valuations were to consider not only the car's use, but also the potential re-sale value of the car in the future, the valuation would clearly be dependent

[1] The *revelation principle*, which states that any mechanism can be transformed into an *incentive compatible* and *direct* revelation mechanism (ICDRM), thereby guarantees that if a more complex mechanism achieves some desirable properties then there is a corresponding ICDRM that can also achieve them.

P. Faratin and J.A. Rodríguez-Aguilar (Eds.): AMEC 2004, LNAI 3435, pp. 15–29, 2005.

on the valuations of other bidders). Now, in auctions with interdependent valuations, the desirable properties of the VCG mechanism no longer hold and the auction is not guaranteed to be efficient.

Interdependent valuations occur most commonly within multi-agent systems when agents have noisy or uncertain estimates of the true value of a good. For example, consider the case of agents bidding for a service in some form of computational economy (as is found, with web services or grid computing). In such cases, the value of a service to an agent is often dependent on the time of response between submitting a request and receiving the desired service. However, in many such cases, the dynamic and open nature of most of these systems means that each agent is only likely to have limited previous experience of a given service and thus it will only have an imprecise estimate of its expected response time. Now if the agent knew the response time of other agents who have used this service (e.g. by asking them about their previous experience or by deducing it from their bidding behaviour), it would be able to form a more accurate estimate of the future response time (by cross-correlating from a broader set of experiences). Hence each agent's valuation is dependent on the signals (in this case, the response time) observed by the other agents bidding for the service and thus we again have interdependent valuations.

Another instance where interdependent valuations have been documented is in the FCC spectrum auctions [3] where it was found that bidders formed their valuations based around the beliefs and actions of other bidders. In these auctions, each bidder wanted to infer from the bidding actions of the other bidders how much they valued the spectrum licenses that were being offered. Thus, whilst each bidder had carried out independent research to gauge the market profitability of these spectrum licenses (i.e. how much money can an agent potentially make by using the license if it wins it), they wanted to use the information gained by the other bidders as well.

To overcome the independent valuation limitation, a number of researchers have developed efficient auctions for interdependent valuation scenarios where a single item is allocated (see section 2 for more details). However, in this work we are interested in the case of multiple items being allocated (i.e where agents may be interested in combinations of items such as a bundle of services). This extension also allows us to consider the important case of combinatorial allocations. These allocations deal with items exhibiting complementarities and substitutabilities and are shown to be more efficient than multiple concurrent auctions of single goods [11,13]. Such allocations occur in many real world scenarios such as the grid services and FCC spectrum auction.

To this end, we develop a novel direct mechanism that can allocate multiple items in an interdependent valuation scenario where each agent receives a single-dimensional signal (for example, a time of response in the computational economy or market profitability in the case of the FCC spectrum). We restrict our attention to single-dimensional signals because in an interdependent valuation scenario it is not possible to develop an efficient auction for multi-dimensional signals [6][2]. Moreover, the single-dimensionality of the signal is not overly restrictive because in many cases the

[2] However, Mezzetti [8] shows that if we adopt a two-stage approach to the auction design, we can then achieve efficiency and incentive-compatibility.

necessary information can be encompassed into a representative single-dimensional signal. In developing this mechanism, we advance the state of the art in the following ways:

1. We extend the standard VCG mechanism to deal with interdependent valuations in the case of multiple goods in which agents receive a single-dimensional signal.
2. We show that our mechanism is general and demonstrate that it reduces to the VCG mechanism for multiple goods in the case of private values.
3. We prove the economic properties of our mechanism. In particular, we show that it is incentive-compatible, individually rational and efficient. We also analyse its computational properties and show that the mechanism does not impose any additional computational load on the agents, but does so in the case of the centre (as compared to an independent valuation scenario).

The remainder of the paper is organized as follows: section 2 presents related work. Section 3 then develops our auction mechanism for the interdependent valuation scenario. We then provide an explanatory example that highlights how the mechanism works in section 4. In section 5 we prove the economic and computational properties of the mechanism. Finally, we conclude and suggest areas of future work in section 6.

2 Related Work

The VCG mechanism and its various extensions have been used in a variety of resource and task allocation scenarios that occur in MAS [11,13,7,5]. However, in these scenarios, work has invariably concentrated on private valuation situations. Specifically, in the case where an agent observes a single-dimensional signal about the objects it wishes to bid on and this signal determines its valuation. This single-dimensional signal is often referred to as the *type* of the agent.

Recently, however, a number of researchers have started to consider interdependent auctions [7,4,6]. In particular, there are currently two main approaches to finding an efficient mechanism for the allocation of items with interdependent valuations. Krishna considers a direct mechanism for efficient allocations for multi-unit single items with single-dimensional signals [7]. In this case, agents submit their interdependent valuation functions, as well as their signals, to a central auctioneer who then decides on the efficient allocation. The payment scheme was then devised so that the agents are incentivised to reveal their signals truthfully.

On the other hand, Dasgupta and Maskin developed an indirect efficient mechanism for the case of two non-identical items, again with single-dimensional signals [4]. In their case, agents make contingent bids rather than submitting their valuation functions and observed signals (i.e. agent 1 submits a range of bids which describes its bid when agent 2 bids a particular value and vice versa). Thus the bidding is more complex than in Krishna's mechanism because the agents have to submit bids based on what other agents might bid, rather than just revealing their valuation function and signals. This bidding becomes even more complex in the indirect mechanism they have developed for the case where multiple items need to be allocated.

Given this, in this paper, we adopt the approach by Krishna, since the bidding is more straightforward for the agents. Specifically we develop a direct mechanism in order to deal with the allocation of multiple items where each agent receives a single-dimensional signal. A naive extension of the VCG mechanism is known not to work in this case [7] and given this we show how to change the payment scheme in order to achieve the desirable economic properties of the VCG. We should note here that we do not concern ourselves with the problem of multi-dimensionality of these signals since it is known that allowing for multi-dimensionality of signals leads to inefficient allocations [6] in direct mechanisms. If the agents can observe the outcome of their reports, then an efficient allocation with multi-dimensional types is possible [8]. However, we believe that this is impractical in many cases because an agent might not be able to observe the outcome from a report (see [8] for an example). Thus, in this paper we consider direct mechanism where the agents can report on their types only once.

3 The Multiple Good Interdependent Mechanism

In this section, we extend Krishna's approach to develop a mechanism that is incentive-compatible, efficient and individually-rational for the case of multiple goods with single-dimensional signals. In this scenario, there is a set of agents $\mathcal{I}$. Each agent i, $i \in \mathcal{I}$, observes a signal $x_i \in \Re_+$ and forms its valuation $v_i(.)$ based on the vector of signals $\mathbf{x} = [x_1, \ldots, x_\mathcal{I}]$ (where each element in the vector is observed by one agent and is correspondingly indexed) and the particular allocation $f \in \mathcal{F}$ being implemented ($\mathcal{F}$ denotes the set of all possible allocations). Thus, $v_i : \Re_+^{|\mathcal{I}|} \times \mathcal{F} \rightarrow \Re_+$. For ease of presentation, we shall denote the set $\mathcal{I} \setminus i$ as $-i$. Furthermore, we shall at times denote $v_i(f, \mathbf{x})$ as $v_i(.)$. Our mechanism, $(\mathcal{M}, \mathbf{r})$, then consists of an allocation rule $\mathcal{M} : \Re_+^{|\mathcal{I}|} \rightarrow \mathcal{F}$ which chooses the allocations and a payment rule $\mathbf{r} : \Re_+^{|\mathcal{I}|} \rightarrow \Re_+^{|\mathcal{I}|}$ which determines the payments r_i to each agent, both being based on the reports of the signal values $\mathbf{x}$. Finally, we shall denote allocations induced by the true report of x_i (x_{-i} being truthful) as f_0^*. As x_i is decreased, it is quite natural to expect that the allocation which is deemed efficient will change because the valuations of each allocation by the agents would also change. These allocations will be denoted by f_l^* with l being the index of each successive induced allocation as x_i is decreased. Mirroring this, as x_i is increased, the successive efficient allocations are denoted by f_{-l}^*. Now, before presenting our mechanism, we shall discuss the assumptions that are critical for the auctions to be efficient.

Assumption 1. $\frac{\partial v_i}{\partial x_j} > 0 \;\; \forall i, j \in \mathcal{I}$

This implies that higher values of the signal lead to higher valuations for the agent. This restricts the signal of the agent to vary in one direction only, thereby making it impossible for an agent to have the same valuation of an allocation for two different signal values. For example, in the case of a computational economy, this would imply that the valuation always increases with rapidity of service (which is x_i).

Assumption 2. $\frac{\partial v_i}{\partial x_i} > \frac{\partial v_j}{\partial x_i} \;\; \forall i, j \in \mathcal{I}, i \neq j$

This implies that an agent's signal affects its own valuation more than it affects the valuation of any other agent. This assumption is the single-crossing condition analogue in the interdependent scenario [7,9]. Without this condition, no efficient mechanism can exist. In the case of a computational economy, this implies that the agent puts more credence on the rapidity of service it measured as opposed to the one observed by other agents.

Assumption 3. $\frac{\partial v_i}{\partial x_i}(., f_p^*) \geq \frac{\partial v_i}{\partial x_i}(., f_q^*)$ *if* $p < q$

This implies that if a higher value of x_i induces an allocation f_p^*, then agent i's value changes more rapidly in this new allocation than in the previous allocation f_q^*. This implies that on receiving a higher x_i, the centre allocates a set of goods to i in the new allocation f_p^* where i's valuation changes more rapidly, than in the previous set f_{p+1}^*. To better explain this assumption, consider a situation where there are two services to be allocated and an agent has a complementary valuation of those services. Suppose that the agent is allocated a particular service when $x_i = \alpha$. Now, if x_i is increased, there will come a point $x_i = \beta > \alpha$ when it will be efficient to allocate both services to the agent (since from assumption 2, its valuations will increase more rapidly than that of other agents). This assumption then implies that the rate of change of the valuation with respect to x_i is greater in this new allocation than in the previous one. Consider, for example, two agents bidding for two services being in a grid service economy. Then suppose that as x_i is increased, it first becomes more efficient to allocate one good (denote this allocation as f_{-1}^*) and then both goods to agent i (denote this allocation as f_{-2}^*). Then this assumption implies that $\frac{\partial v_i}{\partial x_i}(x_i, x_{-i}, f_{-2}^*) \geq \frac{\partial v_i}{\partial x_i}(x_i, x_{-i}, f_{-1}^*)$ i.e. agent i's valuation increases more rapidly with x_i when it is allocated both goods rather than only one.

Given these assumptions, our mechanism then proceeds as follows:

1. Each agent i transmits to the centre its valuation function $v_i(f, \mathbf{x})$ for all the possible allocations $f \in \mathcal{F}$. This function is also over all possible values of $\mathbf{x}$.
2. Each agent i also transmits its observed signal $\widehat{x}_i$.[3]
3. The centre then computes the optimal allocation f_0^* which is calculated as:

$$f_0^* = \arg\max_{f \in \mathcal{F}} \Big(\sum_{i \in \mathcal{I}} v_i(f, \widehat{\mathbf{x}}) \Big) \tag{1}$$

4. The centre also calculates the payment r_i made by each agent i. To do this, the centre first finds the m next best allocations as the reported signal $\widehat{x}_i$ is decreased successively, until the presence of i makes no difference to the allocations. That is, find allocations $f_1^* \dots f_m^*$ and the signal values z_i^l such that:

$$z_i^l = \inf\Big\{ y_i : \sum_{j \in \mathcal{I}} v_j(f_l^*, y_i, \mathbf{x_{-i}}) = \sum_{j \in \mathcal{I}} v_j(f_{l+1}^*, y_i, \mathbf{x_{-i}}) \Big\} \tag{2}$$

(where each allocation f_l^* is different) until:

$$z_i^m = \inf\Big\{ y_i : \sum_{j \in \mathcal{I}} v_j(f_{m-1}^*, y_i, \mathbf{x_{-i}}) = \sum_{j \in \mathcal{I}} v_j(f_m^*, y_i, \mathbf{x_{-i}}) \Big\} \tag{3}$$

[3] Of course, $\widehat{x}_i$ may not be equal to x_i. However, we prove in section 5 that it is a best strategy for the agent to set $\widehat{x}_i = x_i$.

where the allocation f_m^* is the optimal allocation when i does not exist i.e.

$$f_m^* = \arg\max_{f \in \mathcal{F}} \sum_{j \in -i} v_j(f, \mathbf{x})$$

Then the transfer[4] to buyer i is:

$$r_i = \sum_{l=0}^{m-1} \Big[\sum_{j \in -i} v_j(f_l^*, z_i^l, \mathbf{x_{-i}}) - \sum_{j \in -i} v_j(f_{l+1}^*, z_i^l, \mathbf{x_{-i}}) \Big] \tag{4}$$

The above scheme rests upon making an agent derive a utility equal to the marginal contribution that its presence makes to the whole system of agents (which is the same intuition as used in the VCG). Thus the additional part of this mechanism is to take into account the effect that an agent's signal x_i has on the overall utility of the system.

This mechanism is general and is shown (below) to reduce to the well-known multiple-good private value model if we take the case of independent valuations i.e when $v_i(\mathbf{x}, .) = v_i(x_i)$. Then the optimal allocation (from equation 1) is:

$$f_o^* = \arg\max_{f \in \mathcal{F}} \big(\sum_{i \in \mathcal{I}} v_i(f, \widehat{x}_i) \big)$$

To calculate the payment scheme, we first note that with independent valuations x_i only affects $v_i(.)$. Thus repeatedly decreasing x_i, until the stopping condition on equation 3, does not change the valuation of the other agents $-i$ on the different allocations. This then implies that in the payment (as computed by equation 4) all the terms cancel each other, except for the first and last, leading to a payment of:

$$r_i = \sum_{j \in \mathcal{I} \setminus i} v_j(f_0^*, \widehat{x}_j) - \sum_{j \in \mathcal{I} \setminus i} v_j(f_m^*, \widehat{x}_j) \tag{5}$$

This is exactly the payment scheme for the multiple-good private values model. Thus, this shows that the classical VCG mechanism is an instance of the generalised mechanism developed here. Furthermore, notice that assumption 2 is automatically satisfied in this independent valuation scenario, since $\frac{\partial v_j}{\partial x_i} = 0$ in such a scenario. Also, since an increase in x_i would only increase $v_i(., x_i)$, any increase in x_i that induces a new allocation would imply that the rate of change of $v_i(., x_i)$ with respect to x_i is higher in the new allocation than in the previous allocation. Thus, assumption 3 is also automatically satisfied in the independent valuation scenario.

4 Example of an Interdependent Valuation Scenario

In order to better explain how the mechanism operates to achieve efficiency and incentive-compatibility, in this section, we present an example that demonstrates how it computes the efficient allocation and the payments. We will also consider the assumptions which we made in section 3 and show how the mechanism fails when these do not hold.

[4] If the transfer is negative it implies that buyer i pays to the centre.

We consider a very simple case, namely that with two agents 1 and 2 bidding for two different spectrum licenses A and B. The set of possible allocations consists of four members, which are $\mathcal{F} = \{(AB, \varnothing), (A, B), (B, A), (\varnothing, AB)\}$. In this case, each agent perceives a particular signal x_i that determines the market profitability of the spectrum licenses. Table 1 shows the valuations of player 1 and 2 for each allocation as well as the sum of their valuations.

Table 1. Valuations of the players with each allocation

Allocation	$v_1(f, \mathbf{x})$	$v_2(f, \mathbf{x})$	$v_{\mathcal{I}}(f, \mathbf{x})$
$(AB, \varnothing)$	$4x_1 + 2x_2$	0	$4x_1 + 2x_2$
(A, B)	$2x_1 + x_2$	$x_1 + 2x_2$	$3x_1 + 3x_2$
(B, A)	$x_1 + x_2$	$0.5x_1 + 2x_2$	$1.5x_1 + 3x_2$
$(\varnothing, AB)$	0	$x_1 + 4x_2$	$x_1 + 4x_2$

We shall now consider how agent 1 views the mechanism as it reports its signal x_1. The explanation for agent 2 is the same and is therefore omitted. Figure 1 shows how the value of each allocation varies for agents 1, 2 and the set of agents $\mathcal{I}$, as agent 1's reported signal x_1 is increased. We denote agent 1 by i and agent 2 by $-i$ to demonstrate how this works in cases of more than two agents. Suppose that agent 1 has observed $x_1 = 1.5$ and agent 2 has observed a value of $x_2 = 2$. Then from the figure, we

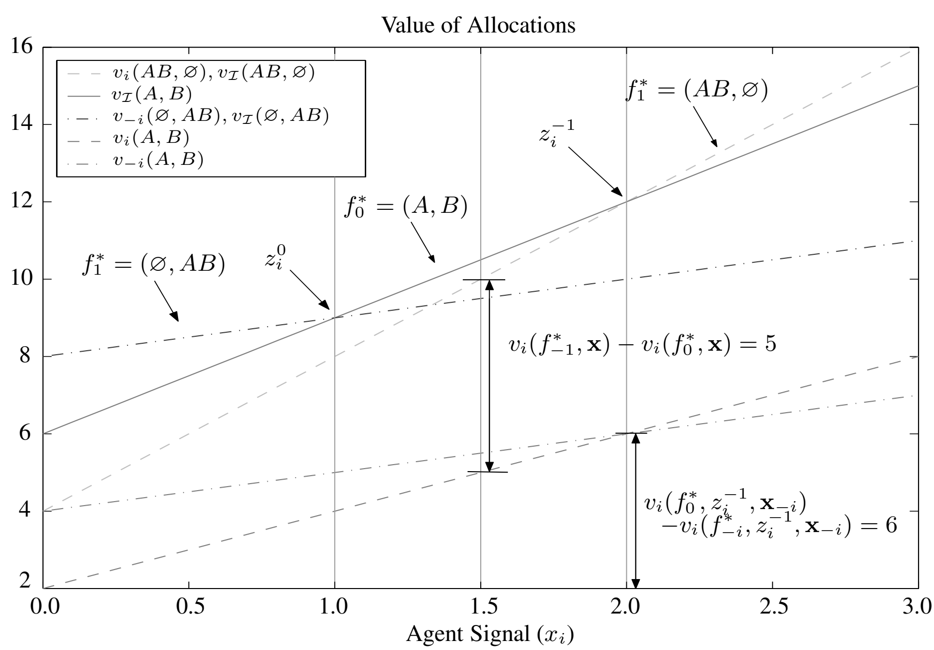


Fig. 1. Valuations of $1, 2$ and $\mathcal{I}$ for each bundle as x_1 is increased

see that the efficient allocation in this case is $f_0^* = (A, B)$ (the efficient allocation is the one that maximises the value of $\mathcal{I}$). Furthermore, the values of x_i at which it becomes more efficient to implement allocations $f_1^* = (\varnothing, AB)$ and $f_{-1}^* = (AB, \varnothing)$ are $z_i^0 = 1$ and $z_i^{-1} = 2$ respectively (shown in figure 1). Hence we can calculate the overall utility that agent 1 derives from reporting truthfully, which from equation 4, is $v_i(f_0^*, \mathbf{x}) + v_{-i}(f_0^*, z_i^0\mathbf{x}_{-\mathbf{i}}) - v_{-i}(f_1^*, z_i^0\mathbf{x}_{-\mathbf{i}}) = 5 + 5 - 9 = 1$. Now, any report in the range $1 \leq x_i \leq 2$ will induce the same allocation and transfer and thus agent 1 has no incentive to report x_i in this range different from the truthful value. If agent 1 reports $x_i > 2$, it will then derive a utility of $v_i(f_{-1}^*, \mathbf{x}) + v_{-i}(f_{-1}^*, z_i^{-1}\mathbf{x}_{-\mathbf{i}}) - v_{-i}(f_0^*, z_i^{-1}\mathbf{x}_{-\mathbf{i}}) + v_{-i}(f_0^*, z_i^0\mathbf{x}_{-\mathbf{i}}) - v_{-i}(f_1^*, z_i^0\mathbf{x}_{-\mathbf{i}}) = 10 + 0 - 6 + 5 - 9 = 0$, which is less than what it would derive from truthful reporting. Thus agent 1 would not over-report its observed value. The reason why this occurs is because, as shown in figure 1, $v_i(f_{-1}^*, \mathbf{x}) - v_i(f_0^*, \mathbf{x})$ is always less than $v_{-i}(f_0^*, z_i^{-1}\mathbf{x}_{-\mathbf{i}}) - v_{-i}(f_1^*, z_i^{-1}\mathbf{x}_{-\mathbf{i}})$ when the true value of x_i is in the range $1 \leq x_i \leq 2$. If, on the other hand, the agent reports $x_i < 1$, it would then derive a utility of $v_i(f_1^*, \mathbf{x}) = 0$ which is again less than what it would derive from truthful reporting. We have thus demonstrated how an agent finds it in its best interest to report truthfully (see section 5 for a more general proof).

The mechanism is guaranteed to work in the above example because the valuations satisfy the assumptions presented in section 3. We will now show how this mechanism would fail if ever, one of these assumptions does not hold.

In order to show what happens when assumption 1 fails, consider only the single good A. Suppose that agent 1 has a valuation of $(x_1 - 2)^2 + x_2$ for good A and agent 2 still has the same valuation of $0.5x_1 + 2x_2$. Then the auctioneer in this case has to decide only between two allocations, namely $\mathcal{F} = \{(A, \varnothing), (\varnothing, A)\}$. With these valuations, it is efficient to allocate good A to agent 2 when $2.25 - \sqrt{\left[(2.25)^2 - (4 - x_2)\right]} \leq x_1 \leq 2.25 + \sqrt{\left[(2.25)^2 - (4 - x_2)\right]}$. If $x_1 \leq 2.25 - \sqrt{\left[(2.25)^2 - (4 - x_2)\right]}$ agent 1 obtains the good and pays $2x_2$ according to equation 4. If $x_1 \geq 2.25 - \sqrt{\left[(2.25)^2 - (4 - x_2)\right]}$, then agent 1 again obtains good A, but this time, it pays 6 (again using equation 4). Thus, it is always in the interest of agent 1 to state that its signal is in the lower range if its signal happens to occur in either of these ranges. Although assumption 1 may seem to be required only for our mechanism to work, this is not so, as it is required for *any* efficient, incentive-compatible mechanism [9].

Now consider that the valuations of the good A are such that $v_1((A, \varnothing), \mathbf{x}) = 2x_1 + x_2$ and $v_2((\varnothing, A), \mathbf{x}) = 3x_1 + x_2 - 6$ (thus assumption 2 is not satisfied). In this case, it is efficient to allocate A to agent 1 when $x_1 < 6$ and to agent 2 otherwise. However, it is not possible to achieve an efficient mechanism in this case, since agent 1 will always state $x_1 < 6$ no matter what the real value of x_1 is. In the case of our mechanism, agent 1 pays $x_2 - 6$ if it allocated the good. Since $v_1(A, \varnothing)$ is always higher than this, agent 1 will thus lie and always state a value of $x_1 < 6$. This problem can again be shown to extend to be symptomatic of any mechanism rather than our mechanism [4]. Notice that with the original valuations in table 1, such a situation would not arise.

We next consider valuations that break assumption 3. Here the valuations of agents 1 and 2 for the allocation $f = (AB, \varnothing)$ are $v_1'((AB, \varnothing), \mathbf{x}) = 0.5x_1 + 2x_2$ and

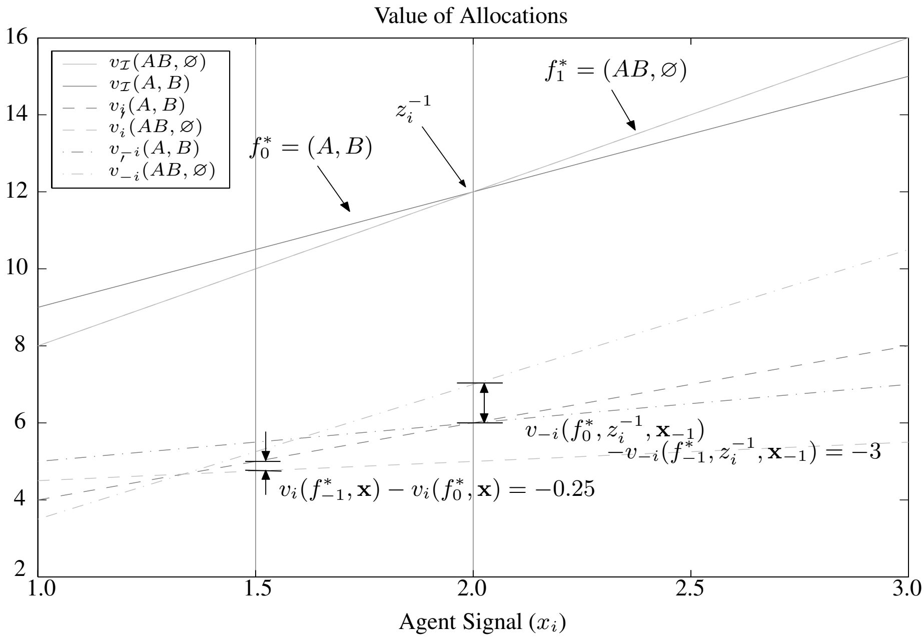


Fig. 2. Modified valuations of $1, 2$ and $\mathcal{I}$ for allocations $(AB, \varnothing)$ and (A,B) as x_1 is increased

$v_2'((AB,\varnothing),\mathbf{x}) = 3.5x_1$ as shown in figure 2[5]. Since $v_{\mathcal{I}}$ remains the same for all the allocations, then z_i^{-1} is still the same as shown in figure 2. Using these modified valuations, agent 1 derives a higher utility of 1.75 (using equation 4 and the valuation function) if it reports $x_i > 2$ thereby leading to the mechanism no longer being incentive-compatible. The reason this occurs is because if assumption 3 is broken we then have that $v_i(f_{-1}^*, \mathbf{x}) - v_i(f_0^*, \mathbf{x}) > v_{-i}(f_{-1}^*, z_i^{-1}\mathbf{x_{-i}}) - v_{-i}(f_0^*, z_i^{-1}\mathbf{x_{-i}})$ as shown in figure 2. As a result, the agent has an incentive to lie and quote a higher value than z_i^{-1}. Notice that this did not occur with the original valuations. Again this assumption is required in order to find an efficient, incentive-compatible mechanism and is thus not idiosyncratic to our mechanism [4].

Having thus illustrated the working of our mechanism and the necessity of the assumptions via the use of an example, we now turn to formally proving the properties of our mechanism.

5 Properties of the Mechanism

We next prove the properties of our mechanism. We first consider the economic properties; namely that it is incentive-compatible, efficient and strategy proof, whilst

[5] Of course, in practice, agent 2 having a valuation for nothing is highly unlikely to occur. However, we need to use this particular valuation in this case due to the simplicity of our example in order to demonstrate what happens when one of the assumptions fails.

intuitively explaining why the mechanism has the aforementioned properties. We then consider the computational properties of the mechanism, showing that the mechanism does not impose any added complexity on the agents' bidding process compared to what it would already face in an independent value scenario. However, it does increase the complexity of calculating the payment, a computational load borne by the centre.

5.1 Economic Properties

Proposition 1. *The mechanism is incentive-compatible in ex-post Nash Equilibrium.*

A mechanism is *incentive-compatible* in ex-post Nash Equilibrium if it is a best response strategy for the players to reveal their types truthfully even after they have complete information about the signal values $\mathbf{x}$.

Proof. Let $v_{-i}(.) = \sum_{j \in -i}(v_j(.))$ and $v_{\mathcal{I}}(.) = \sum_{i \in \mathcal{I}}(v_i(.))$. Suppose now that all players except i report their signals truthfully (i.e. $\widehat{\mathbf{x}}_{-i} = \mathbf{x}_{-i}$). Let the optimal allocation when i reports truthfully be f_0^*. We can then analyse the utility $u_i(.)$ that agent i derives by reporting a certain $\widehat{x}_i$. There are two cases that should be analysed namely when $\widehat{x}_i < x_i$ and $\widehat{x}_i > x_i$. The utility of an agent on reporting $\widehat{x}_i = x_i$ is:

$$u_i(f_0^*, \mathbf{x}) = v_i(f_0^*, \mathbf{x}) + \sum_{l=0}^{m-1} \Big(v_{-i}(f_l^*, z_i^l, \mathbf{x_{-i}}) - v_{-i}(f_{l+1}^*, z_i^l, \mathbf{x_{-i}}) \Big) \tag{6}$$

Now suppose an agent reports $\widehat{x}_i \neq x_i$ but this does not change the optimal allocation f_0^* implemented. Then, $u_i(f_0^*, \mathbf{x}) = u_i(f_0^*, \widehat{x}_i, \mathbf{x_{-i}})$. This is because if the allocation does not change then the agent derives the same value $v_i(f_0^*, \mathbf{x})$ and payment as the signals $z_i^0 \ldots z_i^m$ are computed by the centre. Now consider the case that an agent reports $\widehat{x}_i < x_i$ such that this changes the allocation. Then some other optimal allocation, which is necessarily one of the allocations $f_1^*, \ldots, f_m$, is implemented. Denote the resulting allocation when $\widehat{x}_i < x_i$ as f_n^* (i.e. $z_i^n < \widehat{x}_i \leq z_i^{n-1}$).

The utility that the agent gets from this new allocation is then:

$$u_i(f_n^*, \mathbf{x}) = v_i(f_n^*, \mathbf{x}) + \sum_{l=n}^{m-1} \Big(v_{-i}(f_l^*, z_i^l, \mathbf{x_{-i}}) - v_{-i}(f_{l+1}^*, z_i^l, \mathbf{x_{-i}}) \Big) \tag{7}$$

The difference, $D_n = u_i(f_0^*, \mathbf{x}) - u_i(f_n^*, \mathbf{x})$ between truthful reporting and under reporting (as given by equations 6 and 7 respectively) is:

$$\begin{aligned} D_n &= v_i(f_0^*, \mathbf{x}) - v_i(f_n^*, \mathbf{x}) + \sum_{l=0}^{n-1} \Big(v_{-i}(f_l^*, z_i^l, \mathbf{x_{-i}}) - v_{-i}(f_{l+1}^*, z_i^l, \mathbf{x_{-i}}) \Big) \\ &= v_i(f_0^*, \mathbf{x}) + v_{-i}(f_0^*, z_i^0, \mathbf{x_{-i}}) - v_{-i}(f_n^*, z_i^n, \mathbf{x_{-i}}) - v_i(f_n^*, \mathbf{x}) \\ &\quad + \sum_{l=1}^{n} \Big(v_{-i}(f_l^*, z_i^l, \mathbf{x_{-i}}) - v_{-i}(f_l^*, z_i^{l+1}, \mathbf{x_{-i}}) \Big) \end{aligned}$$

Since $\frac{\partial v_{-i}(f_l^*, \boldsymbol{x})}{\partial x_i} \geq 0$, we thus have:

$$\begin{aligned} D_n >& v_i(f_0^*, \mathbf{x}) + v_{-i}(f_0^*, z_i^0, \mathbf{x_{-i}}) - v_{-i}(f_n^*, z_i^n, \mathbf{x_{-i}}) - v_i(f_n^*, \mathbf{x}) \\ >& v_i(f_0^*, \mathbf{x}) - v_i(f_0^*, z_i^0, \mathbf{x_{-i}}) - v_i(f_n^*, \mathbf{x}) + v_i(f_n^*, z_i^n, \mathbf{x_{-i}}) + v_{\mathcal{I}}(f_0^*, z_i^0, \mathbf{x_{-i}}) \\ &- v_{\mathcal{I}}(f_n^*, z_i^n, \mathbf{x_{-i}}) \end{aligned}$$

However, by construction we know that $v_{\mathcal{I}}(f_0^*, z_i^0, \mathbf{x_{-i}}) > v_{\mathcal{I}}(f_n^*, z_i^n, \mathbf{x_{-i}})$ and from assumption 3 we also know that $v_i(f_0^*, \mathbf{x}) - v_i(f_0^*, z_i^0, \mathbf{x_{-i}}) > v_i(f_n^*, \mathbf{x}) - v_i(f_n^*, z_i^n, \mathbf{x_{-i}})$. We thus have $D_n \geq 0$. On the other hand, if an agent reports $\widehat{x}_i > x_i$ and this induces an allocation f_{-n}^*, then the utility it derives is:

$$u_i(f_{-n}^*, \mathbf{x},) = v_i(f_{-n}^*, \mathbf{x},) + \sum_{l=-n}^{m-1} \Big(v_{-i}(f_l^*, z_i^l, \mathbf{x_{-i}},) - v_{-i}(f_{l+1}^*, z_i^l, \mathbf{x_{-i}},) \Big) \quad (8)$$

The difference, $D_{-n} = u_i(f_0^*, \mathbf{x}) - u_i(f_{-n}^*, \mathbf{x})$ between truthful reporting and under reporting (as given by equations 6 and 7 respectively) is:

$$\begin{aligned} D_{-n} &= v_i(f_0^*, \mathbf{x}) - v_i(f_{-n}^*, \mathbf{x}) - \sum_{l=-n}^{-1} \Big(v_{-i}(f_l^*, z_i^l, \mathbf{x_{-i}}) - v_{-i}(f_{l+1}^*, z_i^l, \mathbf{x_{-i}}) \Big) \\ &= v_i(f_0^*, \mathbf{x}) - v_i(f_{-n}^*, \mathbf{x}) - \sum_{l=-n}^{-1} \Big(v_{\mathcal{I}}(f_l^*, z_i^l, \mathbf{x_{-i}}) - v_{\mathcal{I}}(f_{l+1}^*, z_i^l, \mathbf{x_{-i}}) \Big) \\ &\quad + \sum_{l=-n}^{-1} \Big(v_i(f_l^*, z_i^l, \mathbf{x_{-i}}) - v_i(f_{l+1}^*, z_i^l, \mathbf{x_{-i}}) \Big) \\ &= v_i(f_{-n}^*, z_i^{-n}, \mathbf{x_{-i}}) - v_i(f_{-n}^*, \mathbf{x}) - v_i(f_0^*, z_i^{-1} \mathbf{x_{-i}}) + v_i(f_0^*, \mathbf{x}) \\ &\quad - \sum_{l=-n+1}^{-1} \Big(v_i(f_l^*, z_i^{l-1}, \mathbf{x_{-i}}) - v_i(f_l^*, z_i^l, \mathbf{x_{-i}}) \Big) \end{aligned}$$

Using assumption 3 implies that $D_{-n} \geq 0$. We thus see that i derives highest utility when reporting $\widehat{x}_i = x_i$.

Proposition 2. *The mechanism is efficient.*

This implies that the centre finds the outcome such that $f^* = \arg\max_f \sum_{i \in \mathcal{I}} v_i(f, \mathbf{x})$.

Proof. The above is a result of the incentive-compatibility of the mechanism. Since the goal of the centre is to achieve efficiency, then given truthful reports, the centre will achieve efficiency.

Proposition 3. *The mechanism is individually rational.*

A mechanism is *individually rational* if there is an incentive for agents to join it rather than opting out of it. We begin by assuming that the utility an agent derives from not joining the mechanism is 0. Then, we need to prove that the utility an agent derives in the mechanism is always ≥ 0.

Proof. Given that the agents are incentivized to report truthfully, agent i derives utility:

$$u_i(f_0^*, \mathbf{x}) = v_i(f_0^*, \mathbf{x}) + \sum_{l=0}^{m-1} \Big(v_{-i}(f_l^*, z_i^l, \mathbf{x_{-i}}) - v_{-i}(f_{l+1}^*, z_i^l, \mathbf{x_{-i}} \Big)$$

$$= v_i(f_0^*, \mathbf{x}) + \sum_{l=0}^{m-1} \Big(v_{\mathcal{I}}(f_l^*, z_i^l, \mathbf{x_{-i}}) - v_{\mathcal{I}}(f_{l+1}^*, z_i^l, \mathbf{x_{-i}}) \Big)$$

$$- \sum_{l=0}^{m-1} \Big(v_i(f_l^*, z_i^l, \mathbf{x_{-i}}) - v_i(f_{l+1}^*, z_i^l, \mathbf{x_{-i}}) \Big)$$

Since $v_{\mathcal{I}}(f_l^*, z_i^l, \mathbf{x_{-i}}) = v_{\mathcal{I}}(f_{l+1}^*, z_i^l, \mathbf{x_{-i}})$ (from equation 2):

$$u_i(f_0^*, \mathbf{x}) = v_i(f_0^*, \mathbf{x}) - \sum_{l=0}^{m-1} \Big(v_i(f_l^*, z_i^l, \mathbf{x_{-i}}) - v_i(f_{l+1}^*, z_i^l, \mathbf{x_{-i}}) \Big)$$

$$= v_i(f_0^*, \mathbf{x}) - v_i(f_0^*, z_i^0, \mathbf{x_{-i}}) + v_i(f_m^*, z_i^m, \mathbf{x_{-i}})$$

$$+ \sum_{l=1}^{m-1} \Big(v_i(f_l^*, z_i^l, \mathbf{x_{-i}}) - v_i(f_l^*, z_i^{l+1}, \mathbf{x_{-i}}) \Big)$$

From equation 3, $v_i(f_m^*, z_i^m, \mathbf{x_{-i}}) = 0$. Now, since $\frac{\partial v_i(K, \boldsymbol{x})}{\partial x_i} \geq 0$, thus $u_i(f_0^*, \mathbf{x}) > 0$.

5.2 Computational Properties

In order for a mechanism to be of use in real world scenarios, we must not only consider its economic properties but also its computational complexity. An important distinction is to differentiate between the computational load which is imposed on the agents within the auction and that imposed on the auctioneer or centre. Specifically, we will analyse the computational properties of the mechanism as opposed to that faced by agents in a standard VCG mechanism. In so doing, we aim to quantify the computational cost that the added richness of this mechanism (namely the ability to express interdependent valuation) imposes.

Outcome Determination. In our mechanism, the centre will need to solve equation 1, which is similar to the winner determination equation in the VCG mechanism, in order to determine the efficient allocation. In both cases the computation involves solving a combinatorial allocation problem which is, in the general case, NP-hard [12]. In fact, the size of the set over which the optimisation is carried out is the same in both cases since this is determined by the number of items $|M|$. Thus our mechanism imposes no additional computational load in terms of the centre calculating the allocation. However, in terms of calculating the payments to the agents, our mechanism does impose a larger computational load. In the case of the VCG mechanism, calculating the payment involves performing the winner determination problem $|\mathcal{I}|$ times over the reduced set of agents $\mathcal{I} \setminus i$ (see [5] for more details). However in our case, the centre needs to successively reduce the value of the report from each agent (and calculate the optimal allocation at each stage) until it reaches an allocation which is the optimal one for the reduced set of agents $\mathcal{I} \setminus i$ (see equations 2 and 3). In the worst case scenario, we have to traverse through all possible allocations (except the efficient one) when calculating the different z_i^l for each agent $i \in \mathcal{I}$. For m goods in a combinatorial auction, this requires $2^m - 1$ calculations and is thus exponential in complexity. However, typically, the number of allocations that need to be traversed (i.e the K_l^i) will be much less than 2^m and there is some redundancy between the calculation of the K_l^i in between the agents in $\mathcal{I}$. We will exploit this redundancy in future work so as to reduce the computational load on the centre.

Preference Formulation. In the case of a direct mechanism such as the VCG mechanism or our mechanism, the agents do not have additional computational load in formulating their preferences over all possible outcomes. This is because the agents transmit their observed signal θ_i to the centre and thus do not actually compute $v_i(K, \theta)$ over all $K \in \mathcal{K}$. Rather it is the centre which performs this calculation for each agent when solving the winner determination problem. Thus, our mechanism in this case does not add any computational load on the agents.

Strategy Selection. In the VCG mechanism the agent knows *a priori* that it has a dominant strategy, and thus this computational problem does not arise. In our case, an agent has an *ex-post* Nash strategy. Thus if all the agents are behaving rationally, there is no computational load on the agent in this particular case. However, if it becomes common knowledge that some agent is not playing its best-response strategy (i.e. some agent is not rational) then the agents will have to search through their space of strategies again to find their best-response.

Thus, we can observe that there is no additional computational load on the agents when compared with a standard VCG mechanism. Thus we can use the computationally efficient bidding languages developed for VCG mechanisms [11,10]. This is important since in many proposed applications, whilst the centre may have significant computational power, the agents will be represented by distributed devices of limited computational power.

6 Conclusions and Future Work

In this paper we considered an important class of auctions in which the bidders have interdependent valuations (based on a single dimensional signal measured by each bidder) and bid for multiple goods. In this context, we have significantly extended the standard VCG mechanism and proved that the ensuing mechanism has the ideal economic properties of being efficient, incentive compatible and individually rational. Our mechanism is general and reduces to the VCG mechanism whenever there are independent valuations (as seen in section 3). Thus, we can visualise our mechanism being used even in MAS where the designer is unsure whether the valuations are interdependent or not.

Whilst we have presented our mechanism in terms of resource allocation, it can be easily converted into a task allocation scenario. In such a scenario, agents will first submit cost functions instead of valuation functions. Then, we need to perform a minimisation instead of a maximisation in equations 1, 2 and 3 and take supremums instead of infimums in equations 2 and 3. With these changes, the mechanism still conserves both its computational and economic properties in the task allocation scenario.

Our future work in this area concerns two issues. The first issue will concentrate on how to design mechanisms which take into consideration multi-dimensional signals. Such signals are known to better characterise the preferences of agents in certain MAS such as in a procurement auction where both the price and date of delivery are important [1]. These mechanisms are known not be efficient in an interdependent scenario as a result of the impossibility result due to Jehiel and Moldovanu [6]. However, we aim to calculate the loss in efficiency when taking into consideration multi-dimensionality. The second issue is concerned with the question of the computational complexity of the resulting mechanism. We have seen that allowing for interdependent valuations comes at the cost of additional computational complexity on the centre. We intend to investigate methods to reduce this load, by reducing the space of allocations that need to be considered when computing the payments to the agents (as discussed in section 5). Our aim is to achieve a mechanism whose complexity is no greater than that of performing the task of winner determination in the underlying auction.

Acknowledgements

The authors would like to thank R.Mason and anonymous reviewers for helpful comments. Rajdeep K. Dash is funded by a BAE Systems studentship and an ORS scholarship. This research was undertaken as part of the ARGUS II DARP (Defence and Aerospace Research Partnership). This is a collaborative project involving BAE SYSTEMS, QinetiQ, Rolls-Royce, Oxford University and Southampton University, and is funded by the industrial partners together with the EPSRC, Ministry of Defence (MoD) and Department of Trade and Industry (DTI).

References

1. M. Bichler and J. Kalagnanam. Bidding languages and winner determination in multiattribute auctions. *European Journal of Operational Research*. to appear.
2. S.H. Clearwater, editor. *Market-Based Control- A paradigm for distributed resource allocation*. World Scientific, 1996.
3. P. Cramton. The fcc spectrum auctions: An early assessment. *Journal of Economics and Management Strategy*, 6(3):431495, 1997.
4. P. Dasgupta and E. Maskin. Efficient auctions. *Quarterly Journal of Economics*, 115:341 388, 2000.
5. R. K. Dash, D. C. Parkes, and N. R. Jennings. Computational mechanism design: A call to arms. *IEEE Intelligent Systems*, 18(6):4047, 2003.
6. P. Jehiel and B. Moldovanu. Efficient design with interdependent valuations. *Econometrica*, 69(5):123759, 2001.
7. V. Krishna. *Auction Theory*. Academic Press, 2002.
8. C. Mezzetti. Mechanism design with interdependent valuations: Efficiency and full surplus extraction. Technical report, University of North Carolina, February 2003.
9. R. Mirrlees. An exploration in the theory of optimum income taxation. *Review of Economic Studies*, 38:175208, 1971.

10. N. Nisan. Bidding and allocation in combinatorial auctions. In *ACM Conference on Electronic Commerce*, pages 112, 2000.
11. D. Parkes. *Iterative Combinatorial Auctions:Achieving Economic and Computational Efficiency*. PhD thesis, University of Pennsylvania, May 2001.
12. M. H. Rothkopf, A. Pekec, and R. M. Harstad. Computationally manageable combinatorial auctions. *Management Science*, 44:11311147, 1998.
13. T. Sandholm. Making markets and democracy work: A story of incentives and computing. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*, pages 16491671, 2003.

A Budget-Balanced, Incentive-Compatible Scheme for Social Choice

Boi Faltings

Artificial Intelligence Laboratory (LIA),
Swiss Federal Institute of Technology (EPFL),
IN-Ecublens, CH-1015 Ecublens, Switzerland
boi.faltings@epfl.ch

Abstract. Many practical scenarios involve solving a *social choice problem*: a group of self-interested agents have to agree on an outcome that best fits their combined preferences. We assume that each outcome presents a certain utility to an agent and that the best outcome is the one that maximizes the sum of these utilities. We call a mechanism for solving social choice problems *incentive-compatible* if for each agent, the behavior that maximizes its own utility is also the one that maximizes the group's utility.

One way to achieve incentive-compatibility is the Vickrey-Clarke-Groves (VCG) tax ([5]) mechanism. However, it produces a surplus of taxes that cannot be redistributed to the agents and can severely reduce agents' utilities. Game theory has shown that it is not possible to have a general scheme that is incentive-compatible, budget-balanced and guarantees a Pareto-efficient solution.

We present a scheme that sacrifices Pareto-efficiency to achieve budget balance while being both incentive-compatible and individually rational. On randomly generated social choice problems, the scheme results in significantly better overall agent utility than the VCG tax mechanism.

1 Social Choice Problems

Many practical situations involve social choice: a group of agents has to choose from a fixed set of choices an outcome that best fits their combined preferences. For example, a group going out have dinner together has to choose a restaurant that fits everyone's preferences. Tenants of a building have to decide on features of a planned renovation. Spectrum has to be divided up among different mobile telephone providers.

A mechanism for solving a social choice problem takes as inputs declarations of the agents' utilities for each outcome, and outputs as a solution the optimal choice and possibly other information.

Social choice problems become difficult to solve when agents have conflicting preferences, as each agent will exaggerate its preferences to obtain a better outcome for itself. It is possible to counteract this tendency using *tax* schemes where agents have to pay for the preferences they claim. An example of such tax schemes are auctions: the social choice is to decide who receives the good, and

P. Faratin and J.A. Rodríguez-Aguilar (Eds.): AMEC 2004, LNAI 3435, pp. 30–43, 2005.

the winner has to make a payment that depends on how strongly he claims to value the good. Note that while auctions are a special case of social choice, reverse and double auctions as the outcome space is not independent of the agents (an agent may not agree to give up an item).

In an *incentive-compatible* (IC) mechanism, the incentives of each agent are aligned with those of the group: the behavior that optimizes the utility of an individual agent also optimizes the utility of the group. When utility optimization is left to the social choice mechanism, this often corresponds to each agent being best off declaring its preferences truthfully; this is called *truthful* or *strategyproof*. Such a mechanism makes life easy for the agents since they do not have to speculate to obtain the best outcome. It also avoids choosing a suboptimal outcome because of such speculation.

A well-known mechanism for achieving IC is the Vickrey-Clarke-Groves (VCG) tax ([5]) mechanism. It assumes that the mechanism chooses an outcome that maximizes the sum of agents' utilities (called the *Pareto-efficient* outcome, PE), and makes each agent pay a tax that is calculated so that the agent cannot gain from misreporting its utility. Furthermore, the VCG tax is *individually rational* (IR) in that the tax paid by an agent never exceeds the utility gain it gets from participating in the optimization as opposed to letting the other agents pick the outcome.

Any tax mechanism produces a surplus of taxes that cannot be redistributed to the agents without loosing the incentive-compatible property, i.e. they are not *budget-balanced* (BB). In game theory, it has been shown that all incentive-compatible mechanisms that apply to general social choice problems and always generate a Pareto-efficient outcome must use a tax of a form similar to the VCG tax ([7,9]). It has further been shown that such a mechanism cannot be budget-balanced ([7,10]).

In the special case of auctions, the surplus can be used to pay the sellers of the goods; the resulting VCG scheme is called the *Vickrey* auction protocol. However, in many cases, there is no use for this surplus. It reduces agents' utilities, and creates incentives for the receiver of the surplus to manipulate the setting to maximize taxes. For example, in spectrum allocation, governments can obtain huge windfall profits by creating scarcity, but in so doing hurt the public in general.

Since the lack of budget-balance may cause huge losses to the involved agents, we consider IC and IR mechanisms that do not always choose the Pareto-efficient outcome, but achieve budget balance. In particular, we propose a novel randomized scheme that can be applied to any tax mechanism to achieve budget balance. It preserves all IC and IR properties of the underlying tax scheme, but generates suboptimal solutions, i.e. it is not Pareto-efficient. On randomly generated constraint optimization problems, it can be seen that the loss of agent utility due to the lack of PE is in general much smaller than the loss they would incur through the taxes in a VCG mechanism. The mechanism has much better performance than previous proposals, and shows an interesting new direction for solving social choice problems.

We model social choice problems as general constraint optimization problems. In Section 2, we formally define the framework and give an example of a social choice problem formulated in this way. In Section 3, we review the properties of tax schemes. Section 4 introduces our new mechanism, and Section 5 presents examples and experimental results. Section 6 presents extensions of the mechanism. Section 7 presents related work, and Section 8 the conclusions.

2 Modeling Social Choice Problems

A social choice problem can be simply formulated a a choice among a set of possible outcomes. However, it is often useful to further structure this outcome space. In particular, we assume that the outcome space is the set of solutions to a *constraint satisfaction problem* (CSP). It is defined by a set of variables that can be assigned values in associated domains. A solution is a combination of value assignments to all variables such that a set of *constraints* is simultaneously satisfied. Note that the space of possible outcomes is independent of the agents' actions; thus, there can only be negative externalities and it is for example not possible to model a double auction as a social choice problem.

Based on the CSP formulation, we model social choice problems as multi-agent constraint optimization problems. These are CSP where a set of *agents* has declared relations that specify the utilities they attach to different value combinations. Formally, they are defined as follows:

Definition 1. *A discrete* multi-agent constraint optimization problem *(MCOP) is a tuple* $< A, X, D, C, R >$ *where:*

- $A = \{A_1, .., A_k\}$ *is a set of agents.*
- $X = \{x_1, .., x_n\}$ *is a set of variables.*
- $D = \{d_1, .., d_n\}$ *is a set of domains of the variables, each given as a finite set of possible values.*
- $C = \{c_1, .., c_p\}$ *is a set of constraints, where a constraint* c_i *is a function* $d_{i1} \times .. \times d_{il} \rightarrow \{0, 1\}$ *that returns* 1 *if the value combination is allowed and* 0 *if it is not.*
- $R = \{r_1, .., r_o\}$ *is a set of relations, where a relation* r_i *is a function* $d_{i1} \times .. \times d_{il} \rightarrow \Re$ *giving the utility of choosing each combination of values.*
- R_i *is the subset of* R *that gives the relations associated with agent* A_i*.*

A solution to an MCOP is a consistent assignment to the underlying CSP that maximzes the sum of the agents' utilities. Formally, we define:

Definition 2. *An* assigment V *is a combination of values* $x_1 = v_1 \in d_1, .., x_n = v_n \in d_n$*.*

We write $r_i(V)$ *and* $c_i(V)$ *for the result of applying* r_i *or* c_i*, respectively, to the relevant variables with the assignments in* V*.*

An assignment V *is* consistent *if all constraints are satisfied, i.e.* $(\forall c_i \in C) c_i(V) = 1$.

An MCOP is solvable *if there is at least one consistent assignment.*

We write V_R^* for the consistent assignment such that the sum of the utilities obtained by the relations corresponding to R is maximal; if there are several such assignments it is the one that is lexicographically smallest. We use $v_R^*(x_i)$ for the value of x_i in that assignment.

The solution *to a MCOP is the assignment V_R^* where R is the set of all relations of the MCOP.*

In this paper, we only consider solvable MCOP. When an MCOP is unsolvable, the social choice problem itself has no solution.

As an example, consider the following social choice problem. A building has 4 tenants, represented by agents A_1 through A_4. he tenants have to agree on three architectural features $X = \{x_1, x_2, x_3\}$ of a planned renovation. For the three features, we have options $x_1 \in d_1 = \{A, B, C\}, x_2 \in d_2 = \{A, C\}$ and $x_3 \in d_3 = \{B, C\}$. For structural reasons, the combination of $x_2 = A$ and $x_3 = C$ is not allowed, but all other combinations are feasible. This restriction is modelled by a constraint.

Each agent attaches different utilities to a feature or feature combination. Specifically:

- for agent A_1 and any of the three features, choosing A has a utility of -1, B a utility of 0 and C a utility of +1.
- for agent A_2, the utility is determined by the combination of features x_1 and x_2 according to the following table:

(x_1, x_2)	(B,C)	(C,C)	(A,C)	(B,A)	(C,A)	(A,A)
utility	2	2	1	-1	-1	-3

- for agent A_3, the utility is determined by the combination of features x_1 and x_3 according to the following table:

(x_1, x_3)	(A,C)	(C,C)	(A,B)	(C,B)	(B,C)	(B,B)
utility	2	2	1	-1	-1	-3

Finally, agent A_4 attaches the following utilities to combinations of assignments:

(x_1, x_2, x_3)	utility	(x_1, x_2, x_3)	utility	(x_1, x_2, x_3)	utility
(A,A,B)	3	(B,A,C)	1	(B,C,C)	-1
(B,A,B)	3	(B,C,B)	1	(C,A,C)	-1
(A,A,C)	1	(C,A,B)	1	(C,C,B)	-1
(A,C,B)	1	(A,C,C)	-1	(C,C,C)	-3

The optimal solution to this problem is the combination $x_1 = C, x_2 = C, x_3 = C$, which provides a total utility of 4. However, this solution can only be found if agents report their true utilities. In this example, agent A_4 could exaggerate its utility for combination $x_1 = B, x_2 = A, x_3 = B$ to 10. This would give it an overall supposed utility of 5, and make it the one chosen by the mechanism. Agent A_4 has improved its own utility from -3 to +3, but the true overall utility has decreased from +4 to -2. Other agents may follow similar reasoning and perturb the result further.

3 Achieving Incentive-Compatibility

The incentives of each agent can be aligned with those of the group by making each agent pay a tax reflecting the cost that their preferences are causing to others. A well-known mechanism is the Vickrey-Clarke-Groves (VCG) tax mechanism ([14,5,8]). Its application for multi-agent decision making has already been proposed in [6].

In the VCG mechanism, each agent pays the difference in other agents' utilities of the optimal solution when it is not present and the optimal solution when it is. Recall that R_i is the set of relations R_i imposed by agent A_i. The tax, called the Clarke tax, is then:

$$VCGtax(A_i) = \sum_{r_l \in R \setminus R_i} r_l(V^*_{R \setminus R_i}) - r_l(V^*_R)$$

The VCG tax has the effect of making the objectives of each individual agent that of optimizing the sum of all agent's utilities.

If the optimization is left to the social choice mechanism, this makes it a dominant strategy equilibrum for each agent to declare its utilities truthfully.

In our example, we consider the following solutions:

Solution	$u(A_1)$	$u(A_2)$	$u(A_3)$	$u(A_4)$	total
$v^*_R = (C, C, C)$	3	2	2	-3	4
$v^*_{R \setminus R_1} = (A, C, B)$	0	1	1	1	3
$v^*_{R \setminus R_2} = (C, C, C)$	3	2	2	-3	4
$v^*_{R \setminus R_3} = (B, C, B)$	1	2	-3	1	1
$v^*_{R \setminus R_4} = (C, C, C)$	3	2	2	-3	4

Thus, the VCG tax payments of the agents would be:

Agent	VCG tax
A_1	3-1 = 2
A_2	2-2 = 0
A_3	4-2 = 2
A_4	7-7 = 0

The example shows that the truth-inducing property of the tax can come at a high cost: in this example the total tax paid, 4 units, completely erases the utility of 4 that the agents jointly get out of their renovation.

4 Budget Balanced Mechanisms for Social Choice

The main difficulty with applying tax schemes to social choice is that they generate a surplus of taxes that reduces overall agent utility and creates unwanted incentives for whatever party gets this surplus. We now show a simple scheme that is always strictly budget balanced, but produces sub-optimal solutions. We assume that the agents are solving an MCOP whose variables, domains and constraints are fixed and known. Furthermore, we are going to assume that the MCOP is solvable, i.e. it has at least one consistent assignment.

The basic idea is to randomly select an agent or a group of agents whose relations will receive a lower priority in the optimization. In return, this agent or group of agents will be paid the tax collected from the remaining agents. The scheme is by definition budget balanced since all taxes are paid between the agents themselves. Since the agents receiving the tax have no influence on the declarations and thus the taxes of the remaining agents, the scheme preserves all incentive-compatibility properties of the tax scheme itself. However, it chooses solutions that are not optimal for all agents and is thus not Pareto-efficient.

We now present several budget balanced social choice mechanisms based on this idea. We assume that the tax scheme is that of a VCG tax. However, the mechanisms can be applied with any tax scheme, including the first-price tax where agents pay the declared utilities in the chosen solution.

In this paper, we consider the following mechanism:

Mechanism 1. *1. Each agent $A_i \in \mathcal{A}, i = 1..k$ is asked to state its relations.*

2. *Choose an* excluded coalition E *of one or more agents using a method that does not depend on the relations stated by the agents.*
3. *Compute the assignment:*
$$S_E = V^*_{R \setminus R_E}$$
where $R_E = \bigcup_{A_l \in E} R_l$. Optionally, if there are several equally optimal assignments, choose the one with the best utility according to the relations in R_E.
4. *Make each agent A_i pay to agents in E the VCGtax for the solution S_E:*
$$\begin{aligned} pay(A_i \to E) &= VCGtax_{-E}(A_i) \\ &= \sum_{r_m \in R \setminus (R_i \cup R_E)} r_m(V^*_{R \setminus (R_i \cup R_E)}) - r_m(V^*_{R \setminus R_E}) \end{aligned}$$
and distribute the tax among the agents in E according to some predetermined scheme.

The excluded coalition can be chosen by any mechanism that does not depend on the utility declarations of the agents. In the interest of fairness, it will often be useful to make this choice randomly. The excluded coalition can consist of one or more agents. In most cases, it will be best to choose only a single agent and let the optimization take into account the relations of a maximum number of agents. However, we will see later that larger coalitions may be useful in certain circumstances.

We now show several properties of Mechanism 1.

Proposition 1. *Mechanism 1 is incentive-compatible* ex-post.

Proof. Consider an agent A_i.

When $A_i \in E$, the agent's declarations have no influence on the outcome nor its tax (which is equal to 0), so it cannot gain by misreporting.

When $A_i \notin E$, then the chosen solution is optimal for a social choice problem where A_i is included. Each agent pays the VCG tax corresponding to that problem. This tax is known to be incentive-compatible *ex-post*.

Proposition 2. *Mechanism 1 is individually rational* ex-post*.*

Proof. Consider an agent A_i.

When $A_i \in E$, the mechanism chooses a solution that does not consider the utilities of A_i. Its utility is not worse than if it had not participated in the mechanism at all, and it pays no tax. Thus, it is individually rational to participate.

When $A_i \notin E$, A_i is included in the optimization, and it pays the VCG tax. This scheme is known to be individually rational ex-post.

Proposition 3. *Mechanism 1 is budget balanced ex-post.*

Proof. All taxes are paid to agents in the excluded coalition E, so no tax surplus or deficit remains to be distributed.

For the example problem, assume that the mechanism chooses to randomly leave out each of the 4 agents individually with probability 1/4. We then have the following payments:

Solution	tax(A_1)	tax(A_2)	tax(A_3)	tax(A_4)
$v^*_{R\setminus R_1} = (A, C, B)$	-5	2	1	2
$v^*_{R\setminus R_2} = (C, C, C)$	5	-7	2	0
$v^*_{R\setminus R_3} = (B, C, B)$	0	0	-2	2
$v^*_{R\setminus R_4} = (C, C, C)$	0	0	0	0
Average tax	0	-5/4	1/4	1

The mechanism is obviously not Pareto-efficient, as it does not always chose the optimal solution as the final result. However, while the VCG tax mechanism chooses the Pareto-efficient solution, the tax payments cause considerable utility loss. As we have seen before, in the optimal solution of this example, the joint utility of the agents is 4 and is completely eaten up by the sum of the VCGtaxes which is also 4. Thus, the expected utility to the community of agents is 0. In contrast, in Mechanism 1 the expected utility is

$$1/4 \cdot (3 + 4 + 1 + 4) = 12/4 = 3$$

which is significantly better. In fact, for this example it is better than the VCG mechanism no matter what agent is excluded from the optimization.

The mechanism may have a choice of several solutions S_E that have equal utility for the agents except E, but different utilities for agents in the excluded coalition E. This can lead to variations in the efficiency of the mechanism. For example, the solution (A, A, B) is also optimal for $v^*_{R\setminus R_2}$, but it has a total utility of -1. If this had been chosen as the solution in the scenario given above, the expected utility in Mechanism 1 would be 7/4 instead of 3 (but still better than in the VCG mechanism). This shows the importance of choosing the best solution for agents in the excluded coalition as well. This optimization could also be done by letting agents in E choose which of several equivalent solutions S_E is to be chosen to minimize their costs.

5 Examples and Experimental Results

As another example, consider an auction of a single item among three agents A_1, A_2 and A_3. It can be represented by a variable x that represents the final allocation of the good by an integer $1, 2$ or 3 indicating which agent gets the good. Let the agents' valuations be expressed by the relations r_1, r_2 and r_3 on x as follows:

$x =$	1	2	3
r_1	a	0	0
r_2	0	b	0
r_3	0	0	c

and assume that $a < b < c$, i.e. A_3 values the good the most. We assume that the mechanism chooses as excluded coalition a single agent, where each of the three agents is chosen randomly with probability 1/3. We have the following solutions $S_i = v^*_{R \setminus R_i}(x)$:

Solution	S_1	S_2	S_3
x=	3	3	2

which gives us the following expected taxes and utilities:

A_i	$E[tax]$	$pr(x = i)$	$E[u(A_i)]$
A_1	1/3 (-b)	0	b/3
A_2	0	1/3	b/3
A_3	1/3 b	2/3	2c/3 - b/3

We can verify that no agent has an incentive to misreport its valuation:

- if agent A_1 overreports a valuation a' so that $a' > b > a$, its true expected utility drops from $b/3$ to $a/3$. Underreporting has no effect.
- if agent A_2 underreports a valuation b' so that $b > a > b'$, then its true expected utility drops from $b/3$ to $a/3$. If it overreports b' so that $b' > c > b$, then its true expected utility drops to $2b/3 - c/3$.
- if agent A_3 underreports a valuation c' so that $c > b > c'$, then its true expected utility drops to $c/3$. Overreporting has no effect.

In comparison, in a VCG tax scheme, the Vickrey auction, agent A_3 always gets the good and pays tax b, and both other agents get nothing. Only agent A_3 has an expected utility of $c-b$. Thus, agents A_1 and A_2 are always better off, whereas A_3 is better off only as long as $c \leq 2b$. This condition is likely to be satisfied in competitive markets where valuations tend to be close to one another.

The major difference with classical auction schemes is that this way of allocating the good does not produce any revenue for a third party. Such a *revenue-free auction* is often desirable for public goods such as airport slots, water or pollution rights, and the use of distribution networks.

It is possible to construct cases where in spite of the wasted tax, the VCG mechanism would still achieve better overall efficiency. This would arise when leaving one agent out of the optimization would give only a marginally better result for the remaining agents, but a significantly worse result for the agent that was excluded. Consider the following example: n people have to go from Geneva to London. They can get a group ticket for up to $n-1$ people on Swiss at a cost of \$100 per person, or a group ticket on British Air for up to n people at a cost of \$110 per person. They can also buy individual tickets for a business jet at a cost of \$10′000 per person.

The optimal solution for any subset of $n-1$ agents it to buy a group ticket on Swiss, forcing the remaining agent to buy an individual ticket. Mechanism 1 will chose one of these solutions, so the total cost to all agents is $\$100(n-1)+\$10'000$, and no agent pays any tax. For $n = 10$, this amounts to \$10′900.

A VCG mechanism chooses the overall best solution: buy a group ticket on British Air. The total expense is $\$110n$, but on top of this each agent has to pay a tax of $\$(110 - 100)(n - 1)$. Thus, the total expense for all agents is $\$110n + \$10n(n - 1)$. For $n = 10$, this amounts to \$2′000, significantly less than with Mechanism 1. In fact, the VCG scheme will be better as long as $n < 33$.

On randomly generated problems, we have observed that on average the VCG tax is much larger than the degradation in solution quality incurred by using a suboptimal solution. Figure 1 shows a comparison on randomly generated problems with 5 to 11 variables. The problems involve as many agents as variables.

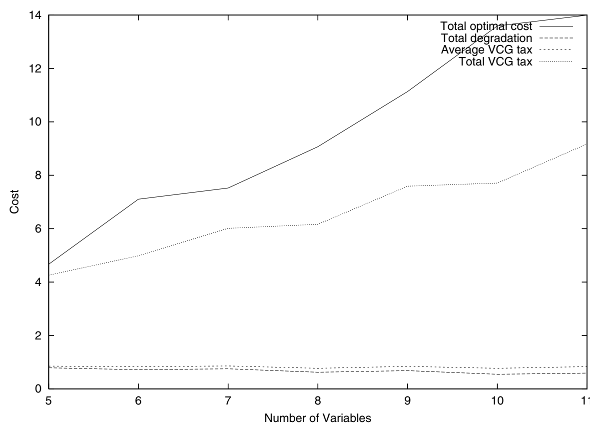


Fig. 1. Costs for randomly generated constraint optimization problems where each variable belongs to a different agent. It shows the total cost to all agents of the optimal solution, the total VCG tax paid by all agents, the degradation of the total cost when the suboptimization solution is used, and the average VCG tax paid by one agent.

For each agent, we randomly generate several binary relations (between 2 variables) where each value combination is assigned a valuation between 0 and 1 with a uniform random distribution, and the number of constraints is limited so that the problem is solvable. We apply Mechanism 1 where a each agent is excluded individually with equal probability.

Figure 1 shows the following quantities:

- the solid line shows the total cost of coexistence in the optimal solution. This is the difference of the total agent utilities if each of them could choose the solution it preferred and the ones obtained in the optimal joint solution.
- the dashed line (lowest in the figure) shows the average degradation in the total agent utility when one agent's constraints are not taken into account in the optimization.
- the lightly dotted line (second lowest in the figure) shows the average VCG tax that an agent would have to pay in the VCG (Clarke) tax mechanism.
- the densely dotted line (second highest in the figure) shows the total of all VCG taxes that would have to be paid by all agents.

The experiment shows that the VCG tax mechanism makes agents pay an amount of tax that is almost comparable to the utility loss due to their coexistence. While the tax per agent tends to decrease slightly with the number of agents, the total amount of taxes and thus the loss of social welfare continues to increase with problem size. On the other hand, the cost of the degradation incurred by having a single agent excluded of the optimization is much smaller. In fact, it is comparable or even below the average VCG tax for a *single* agent. This means that on average, even the agent that is excluded from the optimization would tend to get a comparable or better utility than it can expect in the VCG mechanism!

Another experiment has been conducted on resource allocation in the transportation (or communication) network shown by the graph in Figure 2, where each agent has different (randomly varied) costs for the arcs. We randomly generate tasks which require using a path between two points in the network. For each task, we calculate up to three shortest disjoint paths and define a decision variable whose domain is the cross product of the path to use and the agent that executes it. A further value is provided that corresponds to not executing the task at all. Constraints specify that no two tasks be assigned to paths that share an arc. Each agent is asked to evaluate the costs for the three different paths and thus state its utility (task payoff - cost) if it were assigned the task and the corresponding path.

Figure 3 shows the performance of Mechanism 1 compared to an optimization with a VCG mechanism. The bars show the average total utility to all agents in the optimal (shaded) and optimal with one agent excluded (black) solutions. The curves show the average total amount of tax in a VCG mechanism, and the resulting net total utility. It can be seen that even as the number of tasks increases, the utility of the suboptimal solution remains close to that of the optimal solution. While the total amount of VCG tax levels off as the number

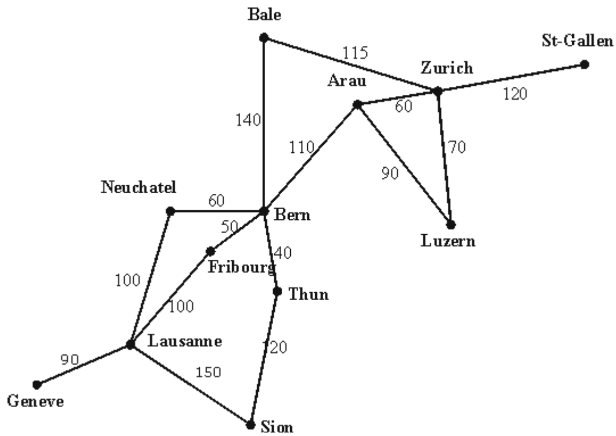


Fig. 2. Network used for resource allocation experiment

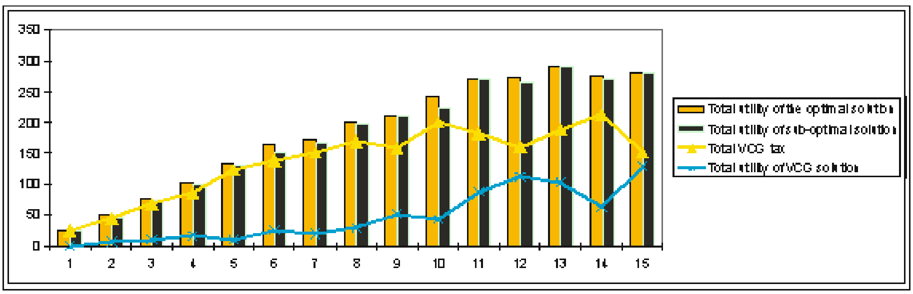


Fig. 3. Utilities achieved by different mechanisms in the experiment

of tasks increases, there is still a large amount of tax that needs to be wasted and causes utility loss to the agents.

6 Collusion

It is a well-known fact that VCG tax schemes are vulnerable to collusion: coalitions of agents can act together to achieve unfair advantages over the others. In our example, suppose that agent A_4, who fares particularly badly in the optimal solution, bribes agent A_3 to help it impose solution $x_1 = A, x_2 = A, x_3 = B$. The result lowers A_3's utility by 1 but increases A_4's utility by 6, so A_4 can pay A_3 3 units for its trouble and both will benefit from the manipulation. A_3 and A_4 can impose this solution by each adding a ternary relation between all three variables that would give utility 100 to this value and 0 to all other combinations. Since the solution would remain the same if either A_3 or A_4 was removed, no agent would pay any taxes, so the manipulation comes for free.

In auctions, collusion can be avoided by using mechanisms that make bidders pay their bid rather than the second highest bid. Similarly, we can define a tax mechanism for social choice, which we call the *first-price tax*, where every agent pays as tax the utility gain it gets in the chosen solution, i.e.:

$$Ftax(A_i) = \sum_{r_l \in R_i} r_l(V_R^*)$$

Obviously, the first-price tax is not incentive-compatible, as agents have an interest to claim a lower utility for the optimal solution and a higher utility for the case where they are excluded. However, it does protect against overclaiming true utilities since an agent that does so will be forced to pay a higher tax than its true utility should the particular solution be chosen. Furthermore, there are ascending-price elicitation schemes, such as the Ausubel auction, that allow first-price schemes to become incentive-compatible in certain cases.

Mechanism 1 can be readily adapted to use a first-price tax rather than a VCG tax as the underlying tax scheme. It is instructive to see what the effect of such a modification is in the example of the revenue-free auction. We would obtain the following expected taxes and utilities:

A_i	$E[tax]$	$pr(x = i)$	$E[u(A_i)]$
A_1	1/3 (-c)	0	c/3
A_2	b/3-c/3	1/3	c/3
A_3	c/3	2/3	c/3

Thus, in this example, if all agents report the truth, the mechanism will make them all have equal expected utility. Note however that it is obviously in the interest of all agents to speculate by underclaiming utilities. This process can be supported using an ascending-price mechanism similar to English auctions that can be readily adapted to this scenario.

7 Related Work

The social choice problem has been the subject of considerable interest in economics, game theory and more recently computer science research, and there is therefore a large amount of related work that is impossible to survey accurately and completely here. Examples of surveys are Moulin ([9]) and recently Chung and Ely ([4]), but there are numerous others. Game theory has largely focussed on the feasibility rather than design of actual mechanisms. Fundamental results ([7,10]) show that it is impossible to have a budget balanced, Pareto-efficient and incentive-compatible mechanism for the general case.

A first approach to design feasible mechanisms is based on relaxing one of the conditions. Vickrey-Clarke-Groves mechanisms ([14,5,8]), in particular the Clarke tax ([5]), are Pareto-efficient, incentive-compatible and individually rational, but in general not budget balanced.

One can easily imagine mechanisms which are incentive-compatible, budget balanced but not efficient, such as simply choosing a solution at random or choosing a solution that is optimized for agents according to a fixed priority sequence. The approach described here is similar, but comes much closer to an optimal result and thus provides much higher utility to the agents.

Parkes et al. ([12]) have investigated VCG mechanisms which are only approximately incentive-compatible in order to achieve budget balance. Since it is not known a priori what benefits manipulation can bring in a particular case, such a mechanism places a burden on agents who need to evaluate potential manipulations for possible gains.

The dAGVA mechanism ([3,1]) is an example of a mechanism that is Pareto-efficient, incentive-compatible, and budget balanced and individually rational on average. However, it requires a-priori knowledge of the true probability distributions of the agent's preferences, which is rarely available in practice.

A second approach is to relax the requirement of a general mechanism that works for all valuation structures, and design a mechanism specifically for a particular scenario. Recent work on automated mechanism design ([13]) has shown that given the exact valuations for each of the agents' types, it is computationally feasible to search for mechanisms that have all desired properties. However, this process requires that the uncertainty about agent's preferences is limited to a finite set of types rather than continuous valuations.

8 Conclusions

The internet has enabled the creation of networked enterprises consisting of multiple agents. So far, most protocols for optimizing their collective behavior have assumed cooperative behavior but neglected the presence of self-interest.

When self-interest has to be taken into account, the best existing solutions for optimal behavior are based on auctions. However, auctions are not budget balanced: they generate a surplus that reduces agents' utilities and creates unwanted incentives for the party that receives it. This is the case in particular for the Clarke tax ([5]), the most well-known mechanism for dealing with self-interest in social choice.

Since economists have shown many impossibility results (for example, [7,10]) that prove that it is impossible to combine incentive-compatibility, optimality and budget balance, it is unlikely that this can be overcome in general.

This paper has presented a mechanism that chooses a solution that is optimal for all but a group of excluded agents. In this way, any tax or auction scheme can be made budget-balanced by returning the surplus to the excluded agents. The important observation is that the quality of the solution optimized for all but a small group of agents is very close to that of the solution optimized for all agents. In fact, the loss of utility incurred by this suboptimal solution is much smaller than the loss that would be incurred through the wasted taxes. Thus, it provides an attractive scheme for achieving budget-balance that can be applied to any auction scheme.

In contrast to other proposals for budget balanced mechanisms, our proposal is general and applies to any quasilinear utility function. It also does not require any a-priori knowledge of agent's preferences.

Acknowledgements

I would like to thank Tuomas Sandholm for numerous insights regarding incentives and computing, and Adrian Petcu, Santiago Macho-Gonzalez and Quang-Huy Nguyen for running experiments.

References

1. K. Arrow: "The property rights doctrine and demand revelation under incomplete information," in M. Boskin (ed.): *Economics and Public Welfare* Academic Press, 1979
2. K.J. Arrow: "Social Choices and Individual Values," Wiley and Sons, New York, 1963
3. C. d'Aspremont and L.-A. Gerard-Varet: "Incentives and Incomplete Information," *Journal of Public Economics* **11**, pp. 25-45, 1979
4. K.-S. Chung and J.C. Ely: "Ex-Post Incentive-Compatible Mechanism Design," *Working Paper*, Department of Economics, Northwestern University
5. E.H. Clarke: "Multipart pricing of public goods," *Public Choice* **11**, pp. 17-33, 1971
6. E. Ephrati and J. S. Rosenschein: "The Clarke tax as a consensus mechanism among automated agents," *Proceedings of the 9th National Conference on Artificial Intelligence*, pp. 173-178, San Jose, California, July 1991.
7. J. Green, J.J. Laffont: "Incentives in public decision making," *Studies in Public Economics* **1**, North-Holland, 1979
8. T. Groves: "Incentives in Teams," *Econometrica* **41**, pp.617-663, 1973
9. H. Moulin: "Axioms of Cooperative Decision-making," *Econometric Society Monographs***15**, Cambridge University Press, 1988
10. R.B. Myerson, M.A. Satterthwaite: "Efficient Mechanisms for Bilateral Trading," *Journal of Economic Theory* **29**, pp. 265-281, 1983
11. N. Nisan and A. Ronen: "Computationally feasible VCG Mechanisms," *Proceedings of the 2nd ACM Conference on Electronic Commerce*, 2000
12. D.C. Parkes, J. Kalagnanam, M. Eso: "Achieving Bduget-Balance with Vickrey-based Payment Schemes in Exchanges," *17th International Joint Conference on Artificial Intelligence (IJCAI-01)*, pp. 1161-1168, 2001
13. T. Sandholm: "Automated mechanism design: A New Application Area for Search Algorithms," in F. Rossi (ed.): *Proceedings of the 9th International Conference on Principles and Practice of Constraint Programming (CP-2003)*, pp. 19-36, 2003
14. W. Vickrey: "Counterspeculation, Auctions, and Competitive Sealed Tenders," *Journal of Finance* **16**, pp. 8-37, 1961

An Options-Based Method to Solve the Composability Problem in Sequential Auctions

Adam I. Juda and David C. Parkes

Division of Engineering and Applied Sciences, Harvard University,
33 Oxford Street, Cambridge MA 02138
{juda, parkes}@eecs.harvard.edu

Abstract. Current auctions often expose bidding agents to two difficult, yet common, problems. First, bidding agents often have the opportunity to participate in successive auctions selling the same good, with no dominant bidding strategy in any single auction. Second, bidding agents often need to acquire a bundle of goods by bidding in multiple auctions, again with no dominant bidding strategy in any single auction. This paper introduces an options-based infrastructure that respects the autonomy of individual sellers but still enables bidders to utilize a dominant, truthful strategy across multiple auctions.

1 Introduction

Many authors (e.g., [8]) have written about a future in which commerce is mediated by automated trading agents. Yet, we believe that one leading place of resistance is in the lack of optimal bidding strategies for any but the simplest of market designs. Although it is popular to appeal to computational mechanism design [5], and try to design *truthful* auctions to address this problem, it is nevertheless clear that a single truthful mechanism cannot exist for all transactions in which an agent has an interest. Somewhere, at some point, there must be boundaries between mechanisms [20].

This work proposes a new *options-based* market infrastructure, that can enable simple yet optimal bidding strategies, while retaining the seller autonomy that is the defining feature of the most successful of today's electronic markets. Although eBay acts as an intermediary of sorts, eBay does *not* gather up the goods for sale by multiple sellers and run them within a single coordinated event. Rather, eBay is an open environment in which each seller chooses: a) when to bring a good to market; and b) the kind of auction to use. Buyers then pick-and-choose across auctions, before submitting bids and making purchases.

One problem encountered in environments like eBay is the *exposure problem*. People would like to acquire multiple items, but may end up only holding a subset of those items at the end. For example, imagine Alice would like to buy both Peanut Butter (PB) and Jelly (J), but has to participate in two different auctions in order to acquire both items. Alice may bid enough to win the PB, but may subsequently not win the J. In doing so, she is left exposed, having spent money in acquiring the PB, but unable to derive any benefit without also acquiring the other item she desired. Another problem occurs when *multiple copies* of an item are offered for sale sequentially. For example, Alice may want PB, and could potentially bid in either a 1 o'clock or 3 o'clock auction. While Alice would prefer to participate in whichever auction will have the lower

P. Faratin and J.A. Rodríguez-Aguilar (Eds.): AMEC 2004, LNAI 3435, pp. 44–58, 2005.

winning price, she can not determine beforehand which auction that may be, and could end up winning in the wrong auction. In general, we refer to problems including the *exposure problem* and the *multiple copies* problem as the "composability problem," because they relate to issues with composing strategies across the boundaries of multiple mechanisms [14].

Combinatorial auctions (CAs) [18] can be used to solve the exposure problem by auctioning many different goods simultaneously, allowing bidders to submit desired bundles as their bids (e.g., Alice may submit that she wants both PB "AND" J). However, there are a number of reasons to doubt that CAs can extend to address fully the problem of composability. First, the computational requirements for deriving solutions to combinatorial auctions is worst-case exponential in the number of bids (unless $\mathbf{P} = \mathbf{NP}$). In an eBay environment, the presence of millions of different goods (with an exponentially larger number of possible *bundles* of goods) makes the design of a single mechanism impractical. Second, CAs do not resolve *temporal issues*, assuming instead that all agents are present at the same time. Third, CAs assume an unrealistic market scope, with one market-maker able to control and bring together all participants into a single market.

Retail stores have customers that face similar strategic problems as the composability problem, and they have devised different policies to alleviate the problems that their customers face. Return policies alleviate the *exposure problem* by allowing customers to return goods at the purchase price. Price matching alleviates the *multiple copies* problem by allowing agents to receive from sellers after product purchase the difference between the price paid for a good and a lower price found elsewhere else for the same good. These two retail policies provide the basis for the scheme proposed in this paper. In particular, we propose an *options-based infrastructure* to address the composability problem.

To participate in the options scheme, a seller must agree to sell an *option* for her good, which will ultimately lead to either a sale of the good, or (if the option is not exercised) going back to the market and offering another option on the same good. The process by which a seller sells options and leaves/returns to market can be made easier via a proxy, whereby a seller may tell a proxy her patience and good for sale, after which the proxy could then sell an option for that good at auction, observe the status of the sold option, and resell another option if the initial option is not exercised and the seller's patience has yet to expire.

Buyers can collect portfolios of options before deciding which to exercise. We provide buyers with *mandatory* proxy agents, that carefully control the number of outstanding options that each buyer can hold, and yet still follow one of the dominant bidding strategies that an agent could follow if there was no proxy in the system. A buyer reports her value and patience to a proxy agent upon arrival, and then agrees to let her proxy: a) bid in auctions to acquire options; b) exercise options to maximize reported utility once the buyer's patience is expired. The proxy agents are essential to prevent a buyer from obtaining options on which they have no intention of exercising. The options-based protocol makes truthful and immediate revelation to a proxy a dominant strategy for buyers, whatever the future auction dynamics.

A benefit for sellers as demonstrated through simulation is that the options-based protocol maintains a market even with buyers with complementary values for goods. In comparison, we show that a traditional market can fail, because it quickly becomes difficult for buyers to participate in the market without becoming exposed to partial bundles and losing money. Thus, the options-based scheme has appeal to both buyers and sellers.

1.1 Related Work

The composability problem was previously observed in Wellman & Wurman [20], in the context of a discussion about the boundaries that must inevitably exist between mechanisms. This theme was continued by Parkes [14] and Ng et al. [13] in the context of *strategyproof computing*, in which the goal is to promote the deployment of strategyproof mechanisms within an open and shared infrastructure. The problem has often been identified in the context of simultaneous ascending price auctions, where it is termed the *exposure problem* [4].

Previous work to address the problem has considered two different directions. First, one can change the mechanism to define an expressive bidding language and incentive-compatibility. This is the approach taken in the work on combinatorial auctions (see Rothkopf et al. [18]). Second, one can attempt to provide agents with smarter bidding strategies. This is the approach taken in the work of Boutilier et al. [2], Byde et al. [3], Anthony & Jennings [1], and Reeves et al. [7]. Unfortunately, it seems hard to design artificial agents with equilibrium bidding strategies, even for a simple simultaneous ascending price auction.

Iwasaki et al. [10] have considered *options* in the context of a single, monolithic, auction design to help bidding agents with marginal-increasing values avoid exposure in a multi-unit homogeneous item auction problem. Sandholm & Lesser [19] have considered options in the form of *leveled commitment contracts* for facilitating multi-way recontracting in a completely decentralized market place. Rothkopf & Engelbrecht-Wiggans [17] discuss the advantages associated with the use of options.

Recent work of Porter et al. [16] has considered auctions with uncertainty in an agent's ability to successfully complete a task. As in our work, there is uncertainty for a seller, in their setting due to whether or not a task will be performed. The chief difference is that bidders in their model *know* their fault probabilities, and the authors can design a mechanism around the revelation of this information.

Finally, a recent direction taken in computational mechanism design is that of *online* mechanisms [15] and *online* auctions [11,9], in which agents can dynamically arrive and depart across time. We leverage a price-based characterization in Hajiaghayi et al. [9] to provide a dominant-strategy equilibrium for buyers within our options-based protocol, creating a decentralized, truthful, option-based implementation of an online combinatorial auction.

2 The Composability Problem

To illustrate the composability problem, consider the following simple example in which a bidder does not have a dominant strategy equilibrium even though the individual auctions in the world are strategyproof.

Example 1. Alice values acquiring PB before Wednesday for \$10. Bob will hold a Vickrey auction for PB on Monday, and then again on Tuesday. In which auction(s) should Alice participate? How much should Alice bid in each auction?

Clearly, Alice no longer has the simple strategy that an individual Vickrey auction offers. By participating in the Monday auction, Alice might win and so would forgo bidding in the Tuesday auction. However, the Tuesday auction may yield a better price. But if Alice decides to skip the Monday auction and only participate in the Tuesday auction, she may not win the Tuesday auction when she could have won the Monday auction, or may have been better off participating in the Monday auction and securing a lower price than the Tuesday auction. Alice has no dominant bidding strategy.

Example 2. Alice values PB and J at \$10. Bob holds a Vickrey auction for PB on Monday. Charlie holds a Vickrey auction for J on Tuesday. How much should Alice bid in each auction?

Again, Alice has no dominant bidding strategy. If Alice wins the PB for $\$y$, it is in Alice's best interest to bid \$10 in the second auction, as winning the J derives a value of \$10 for Alice while losing the auction for J garners Alice no value. However, if Alice loses the second auction, she will have a net surplus of $-\$y$, having spent money on PB but receiving no benefit from it. Even if Alice wins the J, it is possible that Alice will have to spend as much as $\$(10 + y)$ to obtain the set, and so again may lose as much as $\$y$ in surplus.

2.1 The Model

Consider a world with goods, $\mathcal{G}$, buyers $\mathcal{B}$ and sellers $\mathcal{S}$. Buyer $i \in \mathcal{B}$ has a value $v_i(L) \geq 0$ for each subset of goods $L \subseteq \mathcal{G}$. Let $\mathbb{V}$ denote the domain of agent valuations. Let $T = \{1, 2, \ldots\}$ denote time periods. Each buyer has an arrival time, $a_i \in T$, and departure time $d_i \in T$. (Sometimes we refer to the *patience* of a buyer, which is simply $d_i - a_i$.) Let $D_i \subseteq \mathcal{G}$ denote the goods of interest to agent i, defined as

$$D_i = \{k \ : \ \exists L \subset \mathcal{G} \ \ \text{s.t.} \ \ v_i(L \cup k) > v_i(L)\} \tag{1}$$

These are the goods that, when taken with some combination of other goods, have non-zero value to the agent. Buyers are indifferent between receiving a bundle of goods at any time before and up to her departure time. Each seller, $j \in \mathcal{S}$, sells a *single* item $k_j \in \mathcal{G}$.[1] All individual auctions in our model will therefore be for a single item. Seller j has an arrival time, $a_j \in T$, and a departure time $d_j \in T$, when they will leave the system if no agent has yet obtained the right to purchase their good.

Agents arrive or leave at the end of each period, but multiple auctions can be sequenced within a period. Let A^t denote this sequence of auctions in period t, each one associated with a single seller. From buyer i's perspective, let A_i denote the sequence of auctions that occur during the time in which buyer i is in the market. For example, if buyer i arrives in period 1 and departs in period 3, then $A_i = (A^1, A^2, A^3)$.

[1] This is the main limitation in our current model, but a good first-order approximation to eBay style markets.

We require that each individual auction is strategyproof (SP). Following Parkes [14] we term this *local strategyproofness*, to emphasize that it does not imply that a buyer has a dominant strategy when bidding across multiple such auctions. The utility to a buyer, given goods L and price p, is defined as $u_i(L, p) = v_i(L) - p$. An individual auction can be defined in terms of an *allocation rule*, $x(b)$ and a *payment rule* $p(b)$, given bids $b = (b_1, \ldots, b_N)$. Bid b_i reports a value $b_i(L)$ for each bundle of goods L in the auction, and need not be truthful. In particular, agent i receives $x_i(b) \subseteq \mathcal{G}$ goods and makes payment $p_i(b) \in \mathbb{R}$.

Definition 1 (locally strategyproof). *An auction A, defined as (x, p), is (locally) strategyproof when its bidding language is expressive given valuations $\mathbb{V}$, and given the goods available in the auction, and when truthful bidding is a dominant-strategy for an agent that can only bid in this one auction.*

Formally, truthful bidding, $b_i = v_i$, is a *dominant* bidding strategy when $v_i(x_i(v_i, b_{-i})) - p_i(v_i, b_{-i}) \geq v_i(x_i(\hat{v}_i, b_{-i})) - p_i(\hat{v}_i, b_{-i})$, for all bids b_{-i} from other agents, and for all $\hat{v}_i \neq v_i$. For instance, a single-item Vickrey auction is locally-SP for all agents that will bid only in that auction.

When facing a sequence of auctions, a bidding strategy, b_i for buyer i defines the bid that the agent will make in each auction, and can be contingent on: i) her own value; ii) her beliefs about other agents; iii) the outcomes and feedback from earlier auctions.

Definition 2 (The composability problem). *The composability problem exists for an agent facing a sequence of auctions A_i, when each auction in A_i is locally-SP, but the agent does not have a dominant bidding strategy across the sequence of auctions.*

In fact, the composability problem exists more often than not! In what follows, we assume that all goods in an agent's valuation function are available in A_i, and that each auction is locally-SP. All proofs are omitted in the interest of space. First, consider a single-minded buyer, and let (B_i, w_i) denote that the buyer demands bundle B_i at price w_i. Refer to auction $j \in A_i$ as an *interesting auction* when the good $k_j \in B_i$.

Proposition 1. *The composability problem exists for a single-minded buyer whenever there are two or more interesting auctions.*

Proof omitted for space. The effect of multiple auctions is that the agent must anticipate the level of competition, and prices, in future auctions when deciding how to bid. For instance, an agent that wants a single item and faces a sequence of Vickrey auctions does not have a dominant bidding strategy, but would prefer to bid in the auction with the lowest second price.

Next, we can consider an agent that demands multiple disjoint bundles $B_{im} = \{B_{i1}, B_{i2}, \ldots, B_{iM}\}$, with an additive value w_{im} for each bundle.

Proposition 2. *An agent with additive values across bundles faces the composability problem whenever there is at least one bundle B_{im} for which the composability problem exists.*

Proof omitted for space. We can also consider an agent with a general valuation that cannot be expressed as an additive value across disjoint bundles, which precludes a single-minded agent.

Proposition 3. *An agent i with a general valuation faces the composability problem whenever it faces two or more interesting auctions.*

Proof omitted for space. Here, there must exist bundles that are either substitutes or complements. If the bundles are substitutes, an agent faces the problem of determining which bundle to pursue (analogous to the problem an agent faces when the same single item is sold at multiple auctions). If the bundles are complements, an agent can face the exposure problem (analogous to when a single bundle contains multiple items).

3 The Opportunity to Use Options

An option is a right to acquire a good at a certain price, called the *exercise price*. For instance, Alice may obtain from Bob the right to buy PB from him at an exercise price of $3. What makes options unique is that the *right* to purchase a good at an exercise price does not imply the *obligation* to purchase a good at an exercise price. Therefore, when Alice obtains an option from Bob, Bob is not guaranteed that Alice will actually exercise the option at the exercise price and obtain the good. This flexibility makes options useful in addressing the composability problem. Buyers can put together a collection of options on goods, and then decide whether to exercise each option.

Options are typically sold, obtained at a price called the *option price*. However, options obtained at a non-zero option price can not generally support a simple dominant bidding strategy, as an agent must compute the expected value of an option [6] to justify the cost. This computation requires a model of the future, which in our setting requires a model of the bidding strategies and the values of other bidders. This is the very reasoning that we are trying to avoid by introducing options! Instead, we consider *costless options*, where the option price is zero. This will require some care.

The basic problem arises because agents are always (weakly) better off with an option than without an option, whatever its exercise price, because an agent can always choose for free not to exercise an option won. Therefore, an agent would be interested in obtaining a costless option at *any* exercise price (including infinity), subsequently choosing to exercise the option only if doing so would result in a gain of surplus. However, multiple bidders pursuing options with no intention of exercising them could cause market efficiency to unravel. We address this issue through mandatory *proxy agents*, which intermediate between buyers and the market.

4 Auctions for Options

In our scheme, sellers run an auction for costless options on goods, and buyers bid through mandatory proxy agents. These proxy agents are critical to addressing the potential for an inefficient allocation of options through hoarding. Proxy agents, coupled with auctions for options, make it a buyer's dominant strategy to truthfully reveal her valuation and patience. Proxies follow a dominant bidding strategy for a buyer (by bidding at a value high enough that no higher bid could make the agent *strictly* better off), but restrict a buyer from pursuing options on which it is *indifferent*, such as a second option for a good when only one instance of the good is desired, or an option with an exercise price that could never be exercised for positive surplus.

We now define the two main elements of our market:

Seller Auctions. Each seller j sells a costless option in a Vickrey auction. The option is issued to the highest bidder, with an exercise price equal to the second-highest bid, and is set to expire at the end of the buyer's patience. Sellers also agree, by joining the market, to allow the proxy representing a winning buyer to *adjust downwards* the exercise price if the proxy discovers that it could have achieved a better price by waiting to bid in a later auction for the same good (i.e., sellers agree to price match their competitors). Sellers can run additional auctions if the options are returned.

Proxy Agents. Each buyer i must submit to a proxy an expressive language bid, $\hat{v}_i$, reporting values for desired bundles. Each buyer also must submit a departure period, $\hat{d}_i$, to the proxy. The proxy computes the *maximal value* for each item k desired by i as $v_i^{\max}(k) = \max_{L \supseteq k} \hat{v}_i(L)$.[2] The proxy bids this price in any auction for item k, until it holds an option on this item. At that point, the proxy tracks future auctions on that item, determines what the world would look like if it had delayed its entry into the market until that later auction, and will reduce the exercise price of the option it has if it discovers that it could have secured a lower price by waiting to bid in that later auction.[3] The proxy determines this information by asking each future auction to report the identities (can be pseudonymous) of the winner and second-highest winner, together with their bid prices. The identities are necessary because they are used by the agent when creating its view of the world had it decided to delay its entry.[4] Finally, at the end of the buyer's reported patience $\hat{d}_i$, the proxy exercises options to maximize reported value $\hat{v}_i$, solving $\max_L \hat{v}_i(L) - \sum_{k \in L} p_{\text{opt}}(k)$, where $p_{\text{opt}}(k)$ is the option's exercise price, and $p_{\text{opt}}(k) = \infty$ if the proxy does not hold an option. All other options are returned. No options are exercised when no combination of options are priced below a buyer's reported value.

The proxy agent forces a link between the valuation function $\hat{v}_i$ used to acquire options and the valuation $\hat{v}_i$ used to exercise options. Without this, agents could

[2] While sellers of the same item type k' may not have different reserve prices for their goods (due to potential conflicts in being able to price match), sellers may agree (or be required) to have a universal reserve price for each item type, $rp(k')$. In such a scenario, bidding agents can incorporate this information into their bids for multi-item bundles because it provides a tighter lower-bound on the price; specifically, $v_i^{\max}(k) = \max_{L \supseteq k} \left(\hat{v}_i(L) - \sum_{k' \in L, k' \neq k} rp(k') \right)$.

[3] However, the proxy does *not* at any point acquire a second option for the good. Rather, it retains the single option it has been holding, but reduces its exercise price to the later price.

[4] In particular, the proxy maintains a *candidate* agent, $cand_k^t$, for each item k on which it holds an option. Agent $cand_k^t$ is the agent still present in the market that is currently not allocated an option for k, but would have been by now had the proxy delayed its entry. There may be no such candidate agent if the displaced winner leaves the market without winning in a subsequent auction, at which point the state of the market looking forward is unaffected by i winning its option. Initially, $cand_k^t$ is set to the highest outside bidder in the auction in which proxy i wins an option for k. In subsequent auctions for k: if $cand_k^t$ wins, the exercise price for the option held by i is adjusted to the minimal of its current value and the second-highest price in this new auction, and the second-highest bidder in this new auction becoming the new candidate; else, the proxy's price is adjusted to the minimal of its current price and the highest bid price in this new auction.

indifferently acquire options with exercise prices too high to ever be exercised. The proxy agent also ensures that no buyer can hold more than one option on each good, and can hold options on no goods outside its demand set. Without this, agents could indifferently obtain options that they have no intention of exercising. These two properties help to provide a well-functioning market.

Example 3. Alice desires PB and J for \$30. Bob desires PB for \$5. Charlie desires J for \$10. All agents have a patience of 2 days. On day one, a J auction is held, where Alice's proxy bids \$30 and Charlie's bids \$10. Alice wins an option to purchase the J for \$10. On day two, a PB auction is held, where Alice's proxy bids \$30 and Bob's bids \$5. Alice wins an option to purchase the PB for \$5. At the end of the second day, Alice's proxy holds an option to buy PB for \$5 and an option to buy J for \$10, and so exercises both options, spending a total of \$15 to acquire her entire desired bundle.

Example 4. Alice desires PB for \$20. Bob desires PB for \$10. Charlie desires PB for \$5. All agents have a patience of 2 days. On day one, an auction is held for Peanut Butter where each agents' proxy bids their value, and Alice's proxy wins an option to buy PB for \$10, and Alice's proxy notes that Alice prevented Bob from winning the option it now holds. On day two, another auction for PB is held where only Bob and Charlie's proxies bid. Bob's proxy wins an option to buy PB for \$5. Alice's proxy looks at its notes and observes that had Alice delayed her entry until now, Bob would not be bidding. Therefore, Alice would have won today's auction at Charlie's bid price of \$5, and Alice's proxy adjusts Alice's option price down to \$5. At the end of the second day, both Alice and Bob hold options to buy PB for \$5, and so both proxies exercise their options, each spending \$5.

Example 5. Alice desires a red hat for \$20 "XOR" a blue hat for \$10 (i.e., Alice's value is \$20 for a red hat, \$10 for a blue hat, and \$20 for both). Bob desires a red had for \$15. Charlie desires a blue hat for \$30. On day one, a red hat auction is held where Alice's proxy bids \$20 and Bob's proxy bids \$15, resulting in Alice winning an option for the red hat with an exercise price of \$15. On day two, a blue hat auction is held where Alice's proxy bids \$10 and Charlie's proxy bids \$30, resulting in Charlie winning an option for the blue hat with an exercise price of \$10. At the end of day two, Alice's proxy exercises her red hat option and Charlie's proxy exercises his blue hat option.

4.1 Truthful Bidding to Proxy Agent

What remains to be shown is that it is a dominant strategy for the buyer to truthfully reveal her value and arrival and departure time to her proxy agent. The proof builds on the price-based characterization of time-strategyproof auctions in Hajiaghayi et al. [9].

Lemma 1. *An online unit-demand auction is time and value strategyproof when:*

- *it constructs a price function for the agent over time that is independent of the type reported by the agent,*
- *it allocates the good to the agent at the minimal price during period $[a_i, d_i]$ in the market, in a period no earlier than this minimal price, and only when the agent's value is higher than this minimal price.*

Theorem 1. *It is a dominant strategy for a buyer to truthfully reveal her valuation function and patience to her proxy agent in the options-based market.*

Proof. (sketch) The options scheme constructs an agent-independent price schedule, $p_i^t(k)$, for each item k (in period t), defined as the highest bid received among those agents not holding an option at time t, not including i herself and not including any agents that would have already won options had i never entered the system. The proxy agent holds an option for item k at time t at the *minimal* price from $\hat{a}_i$ to t, whenever this price is less than the maximal value the agent could have for the item (given possible future prices). Overstating the value on a bundle can lead to the proxy holding an option on some item, k, at some price greater than it would ever want to pay. Understating the value can lead to the proxy missing a useful option on some item, k, and will not reduce the price otherwise. Strategies that misstate arrival and departure are not useful because reporting a later arrival or earlier departure can forfeit opportunities, while reporting a later departure, $\hat{d}_i > d_i$, is not useful because the agent will not receive its goods until after d_i. Thus, there is no useful manipulation of the final options on individual items, and finally the proxy makes a purchasing decision by looking at prices on options and exercising those that maximize reported utility. ■

5 Experiments

Up to this point we have focused solely on buyers. However, sellers can also benefit because the options-based scheme fixes the market failure that exists when buyers have complex values but face a sequence of auctions. The experimental results in this section demonstrate that there are many scenarios in which the average buyer surplus in a market without options is *negative*. In such a world, buyers would not enter the market to begin with (such a decision is not individually-rational) and there would be no market formation.

We simulate a simple market to better understand the economic effect of the options scheme, for both buyers and sellers. We construct values for buyers according to a *quadratic method* [12], which is parameterized with (M, γ). Each buyer receives value on M bundles, and each bundle contains γ distinct goods. The value of one bundle, B_{im} for buyer i, is determined by first choosing γ (distinct) items uniformly at random and assigning each item k a value $v_k \sim U[0,1]$. The value on the bundle is defined as $w_{im} = \sum_{k \in B_{im}} v_k + \sum_{k' \in B_{im}, k' \neq k} v_k v_{k'}$. Each seller sells a single item, chosen uniformly at random from the set of all goods. We choose to model an *identical* reserve price for all goods and for all sellers, which is set to $rp(k) = 0.5$ for each good k unless stated otherwise.[5]

Buyer patience is set to 50, while seller patience is set to 100. We vary the buyer entry-rate and seller entry-rate to model different levels of supply and demand. We compare the options-based market with a market in which there is a sequence of Vickrey auctions for traditional goods. To model sell-side auctions in each round we choose to run f^t auctions in each period t. Upon arrival, sellers wait in a queue for their auction

[5] Even though all sellers use the same reserve price, we assume that the buyers do not know this, and do *not* pursue the alternate bidding strategy introduced in Footnote 2.

to be scheduled, on a first-come first-served basis. The rate, f^t, is adjusted to keep the wait time, defined as the time that a seller needs to wait to have her auction scheduled, below 5 periods.[6] In the options world, a seller returns to the end of the queue if there are no bidders in her auction, or if the winner returns her option. In the non-options world, a seller only returns to the queue if she fails to sell her item.

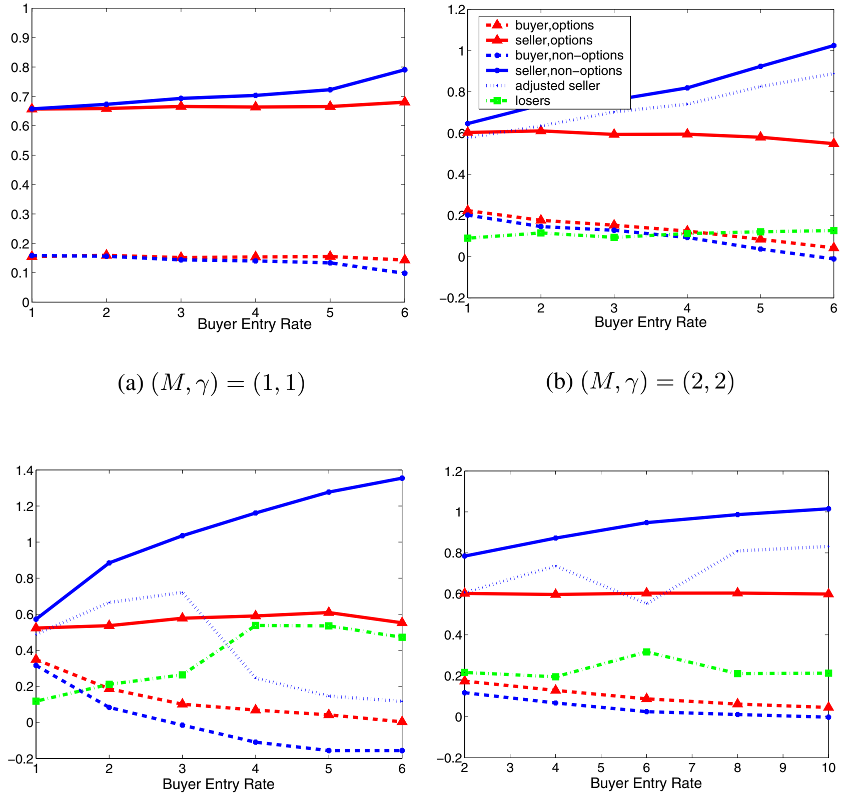
Finally, we need a model of buyer strategies. In the options world, we assume that each buyer reports her true value to her proxy immediately upon arrival. In the non-options world, we need to adopt a bidding strategy for buyers, to provide a meaningful comparison with the performance of the options world. Ideally we would adopt an equilibrium bidding strategy, but this analysis is not available for such a complex game. Instead, we adopt a "sunk-aware" bidding strategy, following the ideas in Reeves et al. [7]. At any point in time an agent has purchased goods L^t, for price $\sum_{k \in L^t} p(k)$, where $p(k)$ was the price paid in the Vickrey auction for good k. Consider some item $k' \notin L^t$. The agent estimates her value for k' as

$$\hat{v}_i^t(k') = \max_{L | k' \in L} \left[v_i(L) - \alpha \sum_{k \in L^t} p(k) - \sum_{k \in L, k \notin L^t, k \neq k'} \hat{p}_\beta^t(k) \right] \quad (2)$$

where $\hat{p}_\beta^t(k)$ is defined as the average price for item k in the last $\beta > 1$ auctions for k (we assume the agent has access to this information).[7] In the event of an auction for item k', the agent then bids this estimated value, $\hat{v}_i^t(k')$.

Parameter $\alpha \in [0, 1]$ determines how sunk-cost aware the bidder will act. With $\alpha = 0$ the agent ignores the sunk cost and continues to bid aggressively. With $\alpha = 1$ the agent considers the cost of items already purchased, as though it is deciding over again whether to buy those items and the new items. Thus, for higher α the agent is more conservative, and bids less aggressively. We model buyers in the non-options world as leaving the market as soon as they have purchased a bundle for which they have positive value, or at time d_i (whichever occurs first). The sunk-cost parameter α was selected so as to maximize buyer performance in the market for each (M, γ) and each level of buyer entry-rate. *Buyer surplus* is measured as the average *value-normalized* surplus upon exit from the market, with value-normalized surplus for a buyer that purchases bundle B_{im} at price p defined as $(v_{im} - p)/v_{im}$, for true value v_{im}. Note that this can go negative, when a buyer pays more than her value for a bundle. We normalize in this way to remove dependencies on absolute values of goods in our empiric analysis. In the non-options world, when a buyer can fail to put together a complete bundle, we substitute $-p/V$, where $V = \max_L v_i(L)$, i.e. the value of the *most-preferred* bundle. *Seller surplus* is measured as the ratio of total revenue generated by all sellers divided by the total value of goods allocated to buyers. *Losers* are the percentage of buyers that leave the market with negative surplus. In some markets without options buyers can lose money, even when bidding conservatively. In this case, we also calculate the *adjusted-seller* surplus, by factoring out any revenue that sellers were achiev-

[6] In practice, we set $f^t = \lceil \frac{N^t}{5} \rceil$, where N^t is the number of sellers in the queue in period t.
[7] β was set to 25 in experimentation.



(a) $(M, \gamma) = (1, 1)$ (b) $(M, \gamma) = (2, 2)$

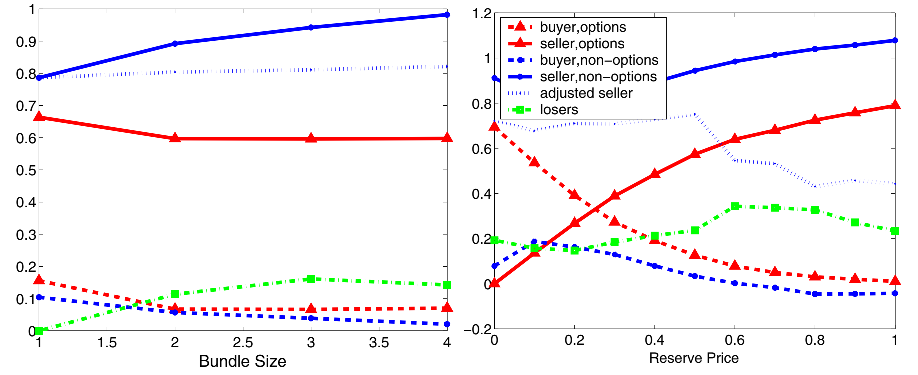
(c) $(M, \gamma) = (4, 4)$ (d) $(M, \gamma) = (3, 2)$

Fig. 1. Buyer and seller surplus vs. buyer entry rate, for different valuations parameterized with (M, γ). We also plot the percentage of buyers that are *losers*. Subplots (a)—(c) hold the *seller entry rate* fixed, whereas the *seller entry rate* in (d) is scaled with *buyer entry rate* at a 2:1 ratio.

ing from buyers that were losing money (which we would not expect in a sustainable equilibrium).[8]

Figure 1 illustrates buyer and seller surplus against an increasing buyer entry-rate, with each subplot dedicated to a different structure (M, γ) for buyer valuations. We consider values of $(M, \gamma) \in \{(1, 1), (2, 2), (4, 4)\}$ in subplots (a), (b) and (c) respectively, with the seller entry-rate set to 3, 6 and 12 in each scenario (this increases supply in line with increased buyer demand as the number of items M demanded in each bundle increases). Figure 1 (a) demonstrates that the options and non-options world produce

[8] This adjustment is deemed conservative, with a more realistic revenue for sellers in these scenarios expected to be lower. If the negative surplus bidders had never entered the market, not only would their income be lost to sellers, but also the prices paid by those people who remained would be lower due to the decreased competition.



(a) $(M, \gamma) = (2, N)$ (b) $(M, \gamma) = (3, 3)$

Fig. 2. Buyer and seller surplus, for different valuations parameterized with (M, γ). Subplot (a) shows the effects of increasing the number of items in the bundles in a valuation. Subplot (b) shows the effects of varying the reserve price.

very similar results when the population has the most simple of valuations, with seller surplus increasing and buyer surplus decreasing as demand increases. Similar results were experimentally confirmed for all $(M, \gamma) = \{(2, 1), (3, 1), (4, 1), (5, 1)\}$.

In Figure 1 (b) and (c) the non-options world "breaks" when demand gets too high because the composability problem becomes more challenging. The average buyer incurs negative surplus and one would reasonably expect that buyers would not enter this market in the first place. On the other hand, buyer surplus in the options world remains positive, indicating that buyers would continue to enter the market. This suggests the existence of scenarios in which introducing options can create new markets. Further evidence for market breakdown in the non-options world can be found by considering the *Losers* rates. Figure 1 (c) shows that the *Losers* rates is near 0.5 when there is high demand, indicating that nearly half of the agents who are entering the market are losing surplus upon exit.

In Figure 1 (d), we consider $(M, \gamma) = (3, 2)$, and scale the seller entry-rate continuously as we scale the buyer entry-rate, keeping the seller entry at twice the buyer entry. Whereas scaling the world degrades the buyer surplus in the non-options world to the point of being negative, seller surplus in the options world is steady and accompanied by positive buyer surplus.

Figure 2 (a) fixes $M = 2$ and increases the number of items in each bundle. We scale the buyer entry-rate from 16 to 8 to 5 to 4, while the seller entry-rate is fixed at 16, to keep supply and demand in the same proportion. The exposure problem is present in this scenario, and indeed we see a significant drop in buyer surplus and a rise in the percentage of buyers with negative surplus in the non-options world. Figure 2 (b) illustrates the effect of changing the reserve price on all items. Buyer entry-rate is 3, seller entry-rate is 9, and buyers demand 3 bundles, each with 3 items. A higher reserve price in the non-options world drives buyer surplus negative, and results in market failure. On the other hand, sellers in the options world are able to raise reserve prices to increase their surplus, and buyers can still manage to obtain positive surplus from the market even when reserve prices are set very high. Noteworthy, although the seller surplus is increasing in Figure 1, one should appreciate that this is a *relative* metric. In fact, we observe that the *total* seller surplus tends to decrease in the options world as buyer entry-rate grows (even though the normalized surplus remains quite flat). Less buyers complete their desired bundles and less buyers eventually exercise their options. Of course, we believe this is preferable to the complete market failure in the non-options world. However, while markets continue to form with options, trades are more infrequent as buy-side competition increases— which suggests that it would be interesting to explore additional methods, such as a prequalification stage, the use of stronger reserve prices, or the "throttling" of buyer entry-rate and pooling into separate markets. We reserve these topics for future study.

6 Conclusion

We introduced an options-based auction protocol to address the composability problem that exists when buyers with complex values must bid in sequences of simple auctions. Our approach combines costless options with proxy agents, which acquire, maintain, and exercise options on the agent's behalf and best interest. Simple trading agents have dominant bidding strategies in our options-based market, even though the markets remain fundamentally disintermediated. We believe that options-based markets may provide an interesting new class of market designs for eBay-like electronic markets.

Future work should aim to better situate this work within the context of the theory of strategyproof *online* auctions. Future work may also address and resolve the strategic problems facing sellers in this work. While it is not a dominant strategy for sellers to try and keep prices on their goods artificially high (as doing so may prevent options from being exercised if the prices maintained are at a prohibitively high level), it is true that straightforward truthful behavior may not always be in the best interest of sellers in the current model. Furthermore, an investigation of the role of false-name behavior [21] should also yield interesting results. While buyers do not want to engage in false-name behavior as multiple buyers, and sellers do not wish to engage in false-name behavior as multiple sellers, we believe there are manipulations for buyers pretending to be sellers and for sellers pretending to be buyers.

Additionally, future work on the empirical aspects of this project should aim to utilize better benchmarks when analyzing the model, including the use of real data. In particular, there are three areas where real data could be particularly helpful to this model. First, we believe there is ample opportunity for further exploration as to modeling the arrival of sellers and timing of auctions in this setting, perhaps using data from eBay as a foundation. Second, the high rates of buyers that are losing surplus in our simulation of buyers in the non-options model when demand is high is cause to believe that agents may follow a different bidding strategy than the one assumed here. Real world data can be of great assistance in helping to empirically determine what those strategies might be. Third, real world data can also help in developing accurate valuation models for some set of niche goods in an existing market.

Acknowledgments

Useful comments and suggestions were received during seminar presentations of earlier versions of this work in DEAS, HBS, and AMEC VI as part of AAMAS '04, as well as from Michael Rothkopf and Loizos Michael. This work is supported in part by NSF grant IIS-0238147.

References

1. P. Anthony and N. R. Jennings. Developing a bidding agent for multiple heterogeneous auctions. *ACM Trans. On Internet Technology*, 2003.
2. C. Boutilier, M. Goldszmidt, and B. Sabata. Sequential auctions for the allocation of resources with complementarities. In *Proc. 16th International Joint Conference on Artificial Intelligence (IJCAI-99)*, pages 527–534, 1999.
3. A. Byde, C. Preist, and N. R. Jennings. Decision procedures for multiple auctions. In *Proc. 1st Int. Joint Conf. on Autonomous Agents and Multiagent Systems (AAMAS-02)*, 2002.
4. M. M. Bykowsky, R. J. Cull, and J. O. Ledyard. Mutually destructive bidding: The FCC auction design problem. *J. of Regulatory Economics*, 2000. .
5. R. K. Dash, N. R. Jennings, and D. C. Parkes. Computational-mechanism design: A call to arms. *IEEE Intelligent Systems*, pages 40–47, November 2003. Special Issue on Agents and Markets.
6. A. K. Dixit and R. S. Pindyck. *Investment under Uncertainty*. Princeton University Press, 1994.
7. D.M.Reeves, M.P.Wellman, J. K.MacKie-Mason, and A.Osepayshvili. Exploring bidding strategies for market-based scheduling. *Decision Support Systems*, 2004. Special issue dedicated to the Fourth ACM Conference on Electronic Commerce (EC'03).
8. A. Greenwald and J. O. Kephart. Shopbots and pricebots. In *Proc. 16th International Joint Conference on Artificial Intelligence (IJCAI-99)*, pages 506–511, 1999.
9. M. T. Hajiaghayi, R. Kleinberg, and D. C. Parkes. Adaptive limited-supply online auctions. In *Proc. ACM Conf. on Electronic Commerce*, pages 71–80, 2004.
10. A. Iwasaki, M. Yokoo, and K. Terada. A robust open ascending-price multi-unit auction protocol against false-name bids. *Decision Support Systems*, 2004. Special issue dedicated to the Fourth ACM Conference on Electronic Commerce (EC'03).
11. R. Lavi and N. Nisan. Competitive analysis of incentive compatible on-line auctions. In *Proc. 2nd ACM Conf. on Electronic Commerce (EC-00)*, 2000.
12. K. Leyton-Brown, M. Pearson, and Y. Shoham. Towards a universal test suite for combinatorial auctions. In *Proc. 2nd ACM Conf. on Electronic Commerce (EC-00)*, 2000.
13. C. Ng, D. C. Parkes, and M. Seltzer. Strategyproof Computing: Systems infrastructures for self-interested parties. In *1st Workshop on Economics of Peer-to-Peer Systems*, Berkeley, CA, June 2003.
14. D. C. Parkes. Five AI challenges in Strategyproof Computing. In *Proc. IJCAI'03 Workshop on Autonomic Computing*, Aculpulco, Mexico, 2003.
15. D. C. Parkes and S. Singh. An MDP-based approach to Online Mechanism Design. In *Proc. 17th Annual Conf. on Neural Information Processing Systems (NIPS'03)*, 2003.
16. R. Porter, A. Ronen, Y. Shoham, and M. Tennenholtz. Mechanism design with execution uncertainty. In *Proc. 18th Conf. on Uncertainty in Artificial Intelligence (UAI'02)*, 2002.
17. M. H. Rothkopf and R. Engelbrecht-Wiggans. Innovative approaches to competitive mineral leasing. *Resources and Energy*, 14:233–248, 1992.

18. M. H. Rothkopf, A. Pekeč, and R. M. Harstad. Computationally manageable combinatorial auctions. *Management Science*, 44(8):1131–1147, 1998.
19. T. Sandholm and V. Lesser. Leveled commitment contracts and strategic breach. *Games and Economic Behavior*, 35:212–270, 2001.
20. M. P. Wellman and P. R. Wurman. Market-aware agents for a multiagent world. *Robotics and Autonomous Systems*, 24(3–4):115–125, 1998. Earlier version in Proc. 8th Eur. Workshop on Modeling Autonomous Agents in a Multi-Agent World (MAAMAW-97).
21. M. Yokoo, Y. Sakurai, and S. Matsubara. The effect of false-name bids in combinatorial auctions: New Fraud in Internet Auctions. *Games and Economic Behavior*, 46(1):174–188, 2004.

"CONFESS". Eliciting Honest Feedback Without Independent Verification Authorities

Radu Jurca and Boi Faltings

Artificial Intelligence Laboratory (LIA),
Swiss Federal Institute of Technology (EPFL),
CH-1015 Lausanne, Switzerland
{radu.jurca, boi.faltings}@epfl.ch
http://liawww.epfl.ch/

Abstract. Reputation mechanisms offer an efficient way of building the necessary level of trust in electronic markets. In the absence of independent verification authorities that can reveal the true outcome of a transaction, market designers have to ensure that it is in the best interest of the trading agents to report the behavior in transactions truthfully. As opposed to side-payment schemes that correlate a present report with future reports submitted about the same agent, we present a mechanism we have called "CONFESS", that discovers (in equilibrium) the true outcome of a transaction by analyzing the two reports coming from the agents involved in the exchange. For two long-run rational agents, we show that it is possible to design such a mechanism that makes cooperation a stable equilibrium.

1 Introduction

The availability of ubiquitous communication through the Internet is driving the migration of business transactions from direct contact between people to electronically mediated interactions. People interact electronically either through human-computer interfaces or even through programs representing humans, so-called agents. In either case, no physical interactions among entities occur and the systems are much more susceptible to fraud and deception.

Traditional methods to avoid cheating, involving strong cryptography and Trusted Third Parties (TTP's) that overlook every transaction, are very costly and sometimes even impossible to implement due to the complexity and heterogeneity of the environment. Moreover, network communities often have a strong desire of being independent of any authorities, as illustrated by the successful P2P systems.

Reputation mechanisms offer a novel and efficient way of ensuring the necessary level of trust in electronic markets. They are based on the observation that agent strategies change when we consider that interactions are repeated: the other party will remember past cheating, and changes its terms of business accordingly in the future. In this case, the expected future gains due to future

P. Faratin and J.A. Rodríguez-Aguilar (Eds.): AMEC 2004, LNAI 3435, pp. 59–72, 2005.

transactions in which the agent has a higher reputation can offset the loss incurred by not cheating in the present transaction. This effect can be amplified considerably if such reputation information is shared among a large population and thus multiplies the expected future gains made accessible by honest behavior.

Theoretic research on reputation mechanisms started with the seminal papers of Kreps, Milgrom, Wilson and Roberts [10,11,12] who explained how a small amount of incomplete information is enough to generate the *reputation effect*, (i.e. the preference of agents to develop a reputation for a certain *type*) in the finitely repeated Prisoners' Dilemma game and Selten's Chain-Store game [15].

Fudenberg and Levine [6] and Schmidt [14] continue on the same idea by deriving lower bounds on the equilibrium payoff received by the reputable agent in two classes of games in which the reputation effect can occur.

A number of computational trust mechanisms have been developed based on both direct (i.e. interaction-derived) and indirect (i.e. reported by peers) reputation information [1,2,16]. This class of mechanisms, however intuitive, are ad-hoc, do not provide rational participation incentives, and impose restrictions on the acceptable behavior of the agents.

In [4] Dellarocas presents an efficient binary reputation mechanism that encourages a cooperative equilibrium in an environment of purely opportunistic buyers and sellers. The mechanism is centralized, it works for single-value transactions, and is robust (within certain limits) against mistakes made by reporters.

One major challenge associated with designing reputation mechanisms is to ensure that truthful reports are gathered about the actual outcome of the transaction. In a typical trading interaction, e.g. an exchange between a seller (he) and a buyer (she), the buyer is required to first pay and then wait for the purchased good to be shipped to the intended destination. While the payment of the buyer can be easily verified with the authority intermediating the transaction (e.g. the credit card company), it is very difficult to verify that the seller has indeed shipped the promised good. We start from a typical assumption about online environments: the outcome of one transaction (i.e. the seller has shipped or not the good) is only known to the parties involved. Any reputation mechanism will therefore have information that is distorted by the strategic interests of the reporters.

Most real situations do not make it rational for an agent to report the truth. The private information of a buyer for example, about the trustworthiness of a seller is often regarded as an asset which should not be freely shared. Paying for the buyer's reputation report could overcome this inconvenient, however, no guarantee can be offered that the information provided is true. For example, a true positive report might create inconveniences for the reporting buyer because of decreased future availability of that particular seller. Moreover, in a competitive environment, a false negative report about a seller slightly increases the buyer's own reputation with regards to the other agents.

The problem of incentive compatibility can be addressed by paying for a reputation report, such that the payment is conditioned on the correlation with future reports (assumed to be true) about the same seller. [13] and [8] describe such schemes that make truth revelation a Nash equilibrium. A problem with these schemes however, is that they require certain constraints on the behavior of the sellers and on the beliefs of the reporting buyers: i.e. the signals observed by the buyers about the seller's behavior are independently identically distributed, and the set of seller types to which buyers assign positive probability is countable and contains at least 2 elements.

In this paper we address the problem of honest feedback elicitation in a setting in which sellers and buyers are assumed to be rational (i.e. maximize their monetary payoff) and both buyers and sellers have a persistent presence in the market. In Section 2 we prove that persistent presence is a critical assumption for the existence of an incentive-compatible reputation mechanism. Afterwards, we introduce an incentive compatible reputation mechanism and make an analysis of its equilibria. Finally, Section 3 presents some open issues and Section 4 concludes our work.

2 Truthful Feedback

We consider an environment in which the following assumptions hold:

- A rational seller interacts repeatedly with several rational buyers by trading one good of value v_i in each round i. The values $v_i \in (\underline{v}, \overline{v})$ are randomly distributed according to the probability distribution function ϕ [1].
- All transactions generate a fixed profit equal to $(\rho_B + \rho_S)v_i$, where $\rho_S v_i$ is the profit of the seller and $\rho_B v_i$ is the profit of the corresponding buyer. $\rho_B, \rho_S < 1$.
- All buyers are completely trustworthy: i.e. Each buyer first pays the seller and then waits for the seller to ship the good. The seller may defect by not shipping the promised good, and the buyer perfectly perceives the action of the seller.
- There is no independent verification authority in the market, i.e. the behavior of the seller in round i is known only to the seller himself and to the buyer with which he traded in that round.
- The seller cannot refuse the interaction with a specific buyer, and can trade with several buyers in parallel. A buyer can however end the interaction with the seller and choose to buy the goods from a completely trusted seller (e.g. a brick and mortar shop) for an extra cost representing a percentage (θ) of the value of the item bought. Once a buyer decides to terminate a business relationship with the seller, she will never trade again in this market. The seller, however, can always find other buyers to trade with.
- The buyer and the seller discount future revenues by δ_B and δ_S respectively. The discount factors also reflect the probability with which the agents are going to participate to the next transaction. $0 < \delta_S, \delta_B < 1$, and $\delta_S >> \delta_B$

[1] Following the same argumentation proposed in [3], this model is valid for settings where the act of accumulating inventory is independent from that of (re)selling it: e.g. a highly dynamic used car dealership.

modeling the fact that the seller is likely to have a longer presence in the market than the buyer.
- The buyer and seller interact in a market (possibly a different one for each transaction) capable of charging listing fees and participation taxes.
- At the end of every transaction, both the seller and the buyer are asked to submit a binary report about the seller's behavior: a positive report, $R+$, signals cooperation while a negative report, $R-$, signals defection.

We also assume that in our environment there is a *semantically well defined*, efficient Reputation Mechanism (RM). Reputation is semantically well defined when buyers have exact rules for aggregating feedback into reputation information and for making trust decisions based on that reputation information. These rules determine sellers to assign a value to a reputation report ($R+$ or $R-$), reflecting the influence of that report on future revenues. RM is efficient if the values associated by sellers to reputation reports are such that in any transaction the seller prefers to cooperate rather than defect. If $V(R+, v)$ and $V(R-, v)$ are the values associated by the seller to the positive respectively the negative reputation report generated after a transaction of value v, we have: $V(R+, v)$ + *Payoff(cooperate,v)* $>$ $V(R-, v)$ + *Payoff(defect,v)*[2]. A simple escrow service or Dellarocas' Goodwill Hunting Mechanism [3] satisfy these properties.

When perfect feedback (i.e. true and accurate) is available, a *well-defined, efficient* RM is enough to make rational sellers cooperate. Unfortunately, perfect feedback cannot be assumed. In the absence of independent verification means, we can only rely on the subjective reports submitted by the agents involved in the transaction; reports which are obviously biased by the strategic interests of the agents.

In the rest of this section we will achieve three things. First, we will draw some limits of feasibility for incentive compatible RMs. We will show that no RM can be incentive compatible when the interaction between the seller and any particular buyer can be modeled by a perfect information finitely repeated game. Second, we describe an incentive-compatible RM that exists within the feasibility bounds. Third, we analyze the equilibria of the described RM and provide an example.

2.1 Limits of Feasibility

From a game theoretic point of view, a complete information game models a situation in which the players are rational, their rationality is common knowledge and their payoffs are also common knowledge.

Reputation mechanisms cannot exist when the agents have complete information and the seller is present for a finite number of transactions in the market [10]. It is therefore common practice for RM designers to model sellers by infinite horizon players. However, no restrictions have been imposed so far on the model of the buyers' behavior. This is the problem we address in this section by showing

[2] As an abuse of notation, we will sometimes use $V(R+, v) = V(R+)$ and ignore the fact that the value of a reputation report also depends on the value of the good.

that RMs cannot be incentive compatible when agents have complete information and any particular buyer is present for a finite number of transactions in the market.

For the environment earlier described, we can prove that:

Theorem 1. *No incentive compatible RM exists in an environment in which the interaction between the seller and a particular buyer can be modeled by a one-shot complete information game.*

Proof. Consider a single-shot buyer B_i, who trades in round i with the seller having one of the two types: S_1 and S_2. The seller type S_1 cooperates with all buyers, the seller type S_2 cooperates with all buyers but B_i, whom he cheats. Let us assume that there exists an incentive compatible RM. RM will therefore be able to differentiate between the two seller types.

For the rest of the buyers, the behavior types S_1 and S_2 are indistinguishable. Because the behavior of the seller in round i is observed only by B_i (assumption presented in Section 2), the rest of the buyers do not have any information about the truthfulness of buyer B_i's report. Hence, any attempts from the rest of the buyers to bias B_i into telling the truth, even by monetary compensation on the side, would be futile since the "biaser" would have no way of confirming the information of the "biasee". On the other hand, a seller of type S_2 can make any positive (no matter how small) side-payment to buyer B_i in order to convince her to submit a false positive report instead of the true negative one, and B_i, being single-shot, will accept that payment. Therefore RM cannot be incentive-compatible. □

As a direct consequence of Theorem 1, a RM can be incentive-compatible only if it is incentive compatible for every isolated interaction between the seller and a particular buyer. The truth of this statement is evident if we consider that an incentive compatible RM should always be able to distinguish between two seller types that are undistinguishable for all the buyers except one.

As an immediate extension of Theorem 1 we have:

Theorem 2. *No incentive compatible RM exists in an environment in which the interaction between the seller and a particular buyer can be modeled by a complete information finitely repeated game.*

Proof. Let us denote by N the number of times a buyer trades with the seller, and let us denote by round i_t, $t = 1 \ldots N$, the round in which buyer B_i trades for the t^{th} time with the seller. In round i_N the buyer is a one shot buyer, and therefore the result of Theorem 1 applies. Because the outcome of round i_N (in terms of truth reporting) is common knowledge to both the seller and the buyer, it will not influence the outcome of round i_{N-1}, which thus strategically becomes the last interaction. By backward induction, it is not possible to obtain truthful reports in any of the N interactions. □

Fortunately, it still is possible to have an incentive compatible reputation mechanism if (1) either the interaction between the seller and a particular buyer can be modeled by an infinitely repeated game, or (2) agents do not have complete information. In the remains of this section we will describe a reputation mechanism that supports an incentive compatible equilibrium when agents have complete information. Moreover, in the same spirit as [11], [6] and [14] we show how uncertainty regarding the buyer's type can give birth to the reputation effect and reduce the set of possible equilibria to a more appealing subset.

2.2 The "CONFESS" Mechanism

Every round i, a seller offers for sale a good of value v_i. The market charges the seller a listing fee ε_S, and advertises the good to the buyer. The buyer pays a participation tax ε_B, to the market, and the price v_i to the seller. If the seller cooperates, he ships the good directly to the buyer; otherwise the seller keeps the payment for himself and does not ship the good. After a certain deadline, the transaction is considered as over, and the market starts collecting information about the behavior of the seller. The seller is first required to submit a report. If the seller admits having defected, a negative report ($R-$) is submitted to the RM, the listing fees ε_S and ε_B are returned to the rightful owners, and the protocol is terminated. If, however, the seller claims to have cooperated, the buyer is also asked to provide a report. At this moment, the buyer can report cooperation, report defection, or she can report defection and terminate the interaction with the seller.

If the buyer reports cooperation, a positive reputation report ($R+$) is submitted to the RM, and the listing fees ε_S and ε_B are returned. If the buyer reports defection, both players will be punished as one of them is surely lying: a negative report ($R-$) is submitted to RM, and the listing fees ε_S and ε_B are confiscated. Finally, if the buyer decides to terminate the interaction, a negative report ($R-$) is submitted to RM, and the fees ε_S and ε_B are confiscated.

From a game theoretic point of view, the above described protocol can be modeled by the extensive-form game $G = (N, (A_i), (\succsim_i), T)$, shown in Figure 1. $N = \{S, B\}$ is the set of players, the seller and the buyer respectively, $A_S = \{Cc_S, Cd_S, Dc_S, Dd_S\}$ is the action set of the seller, $A_B = \{c_B, d_B\}$ is the action set of the buyer, $\succsim_S$ is the preference relation of the seller over the set of possible outcomes (g_S is the corresponding payoff function of the seller), $\succsim_B$ is the preference relation of the buyer over the set of possible outcomes (g_B is the corresponding payoff function of the buyer), and T is the player function, or the "turn" function which prescribes which player should make the next move after every possible game history.

The outcome for the buyer is indicated as a single real value representing the buyer's payoff in the current round. The outcome for the seller is indicated as a tuple $(X; P)$, where $X \in \{R+, R-\}$ represents the filed reputation report (positive or negative), and $P \in \mathbb{R}$ is the monetary gain obtained by the seller in the current transaction. The payoff of the seller is defined by simply adding the monetary gain P with the value of the reputation report: i.e. $g_S(X; P) = V(X) + P$.

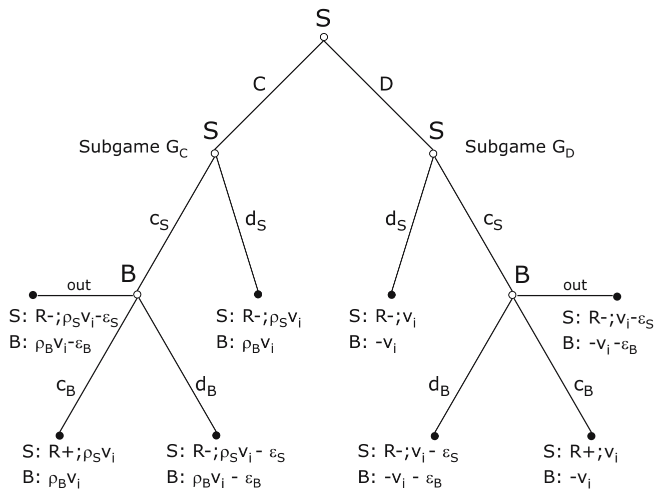


Fig. 1. Game G modeling the one-round interaction protocol

The repeated transaction between the seller and one buyer can be modeled by an infinite repetition of the stage game G, denoted G^∞, in which the overall payoff for player i is given by the average discounted sum:

$$V_i = (1 - \delta_i) \sum_{\tau=0}^{\infty} \delta_i^\tau g_i^\tau;$$

where δ_i denotes the discount factor of player i, and g_i^τ is the payoff obtained by player i in round τ.

2.3 Equilibrium Analysis

For discounted infinitely repeated games with complete information, the Folk Theorem [7] guarantees that every enforceable outcome (i.e. feasible and individually rational) can be obtained by a subgame perfect equilibrium (SPE) strategy profile when the discount factors are big enough. The results of this theorem do not apply directly to the game G^∞ because in every round t we allow the buyer to quit the game.

When the buyer terminates an interaction with a seller (chooses *out* in round t), she obtains a continuation payoff equal to:

$$\hat{V}_B^{t+1} = (1 - \delta_B) \sum_{\tau=t+1}^{\infty} \delta_B^{\tau-t-1} v_\tau(\rho_B - \theta);$$

If we denote by $\tilde{v}$ the average value of a transaction, the expected value of $\hat{V}_B^{t+1}$ is:

$$E[\hat{V}_B^{t+1}] = \tilde{v}(\rho_B - \theta);$$

Any SPE strategy profile must give the buyer at least $\hat{V}_B^{t+1}$ after every round t (otherwise the buyer can profitably deviate to *out* in round t). The minimum continuation payoff of the buyer is therefore:

$$\underline{V}_B^t = (1-\delta_B)(-v_t - \varepsilon_B) + \delta_B \hat{V}_B^{t+1}; \tag{1}$$

A payoff profile $\hat{v} = (\hat{v}_S, \hat{v}_B)$ dominates another payoff profile $v = (v_S, v_B)$ if it is better for at least one of the players and not worse for any of the players: i.e. there is $i \in \{S, B\}$ such that $\hat{v}_i > v_i$ and for all $j \in \{S, B\} \setminus i$, $\hat{v}_j \geq v_j$.

We restrict our attention to SPE strategies of G^∞ which are not dominated. A SPE strategy s is not dominated if there is no other SPE strategy $\hat{s}$ such that the the payoff profile generated by $\hat{s}$ dominates the payoff profile generated by s in G^∞. The intuition behind this assumption is that no player will choose to play a SPE strategy as long as there is another SPE strategy which can bring him a higher payoff while not decreasing the payoff of the opponent.

The above restriction limits the set of SPE strategies to the ones generating an equilibrium path containing a mixture of the action profiles (Cc_S, c_B) and (Dc_S, c_B).

Lemma 1. *All not dominated SPE strategies prescribe only the action profiles (Cc_S, c_B) and (Dc_S, c_B) on the equilibrium path in G^∞.*

Proof. See Jurca and Faltings, [9], Lemma 1. □

Let s be a mixed strategy profile such that with probability p the players play (Dc_S, c_B) and with probability $(1-p)$ the players play (Cc_S, c_B). The expected continuation payoff of the buyer is:

$$\begin{aligned} E[V_B^{t+1}] &= E\left[(1-\delta_B)\sum_{\tau=t}^{\infty} \delta_B^{\tau-t}[p(-v_\tau) + (1-p)\rho_B v_\tau]\right]; \\ &= \tilde{v}(\rho_B - p - \rho_B p); \end{aligned} \tag{2}$$

When playing in round t, the buyer knows which of the action profiles (Cc_S, c_B) or (Dc_S, c_B) are prescribed by the strategy s, and therefore the continuation payoff of the buyer is:

$$\begin{aligned} V_B^t|_{(Cc_S,c_B)} &= (1-\delta_B)\rho_B v_t + \delta_B V_B^{t+1}; \\ V_B^t|_{(Dc_S,c_B)} &= (1-\delta_B)(-v_t) + \delta_B V_B^{t+1}; \end{aligned} \tag{3}$$

depending on what s prescribes for round t. Both $V_B^t|_{(Cc_S,c_B)}$ and $V_B^t|_{(Dc_S,c_B)}$ have to be greater or equal to $\underline{V}_B^t$. The maximum value of p is therefore:

$$p \leq \overline{p} = \frac{(1-\delta_B)\varepsilon_B + \delta_B \tilde{v}\theta}{\delta_B \tilde{v}(1+\rho_B)}; \tag{4}$$

The upper bound on p limits the maximum attainable payoff, $\overline{V}_S$, of the seller in G^∞:

$$\overline{V}_S^t = (1-\delta_S)\sum_{\tau=t}^{\infty} \delta^{\tau-t}[\overline{p}v_\tau + (1-\overline{p})\rho_S v_\tau + V(R+)];$$

which has an expected value: $E[\overline{V}_S^t] = V(R+) + \overline{p}\tilde{v}(1-\rho_S) + \tilde{v}\rho_S$. By replacing (4) we obtain:

$$\overline{V}_S^t = V(R+) + \tilde{v}\rho_S + \tilde{v}(1-\rho_S)\frac{(1-\delta_B)\varepsilon_B + \delta_B\tilde{v}\theta}{\delta_B\tilde{v}(1+\rho_B)};$$

For any $p \in [0, \overline{p}]$ the strategy s can be made a SPE of G^∞ by adding minimax threats (see Fudenberg and Maskin [7] Therem 1 for how the strategy can be built). Let us observe that when $p = 0$ no false feedback is recorded by the RM, and every transaction is cooperative. "CONFESS" has therefore one incentive-compatible, efficient SPE point. Unfortunately, this equilibrium is not unique, and we can only guarantee that the maximum percentage of false reports accepted by our mechanism is $\overline{p}$.

Incomplete Information. Following the ideas from [10], [6] and [14] we can use incomplete information in order to limit the set of SPE strategies to a more desirable subset (i.e. consisting of those strategies which generate mainly true reputation reports and outcomes as close as possible to the socially efficient one).

Let us consider a perturbation of the complete information repeated game G^∞ such that in period 0 (before the first round of the game is played) the *type* of the buyer is drawn by nature out of the set $\Omega = \{\omega_0, \omega^*\}$ according to the probability measure μ. The buyer's payoff now additionally depends on her type, such that the ω_0 type buyer (or the *normal type* buyer) has the payoffs presented in Figure 1, while the ω^* type buyer (or the *commitment type* buyer) always prefers to report the truth. We say that in the perturbed game $G^\infty(\mu)$ the seller has incomplete information because he is not sure about the true type of the buyer.

We prove that in $G^\infty(\mu)$ there is a finite upper bound, k_S, on the number of times a rational seller is willing to play Dc_S, given that he always observes the commitment strategy being played by the buyer.

The intuition behind this result is the following. The seller's best response to the commitment type buyer is to always cooperate and report cooperation, i.e. (Cc_S), which gives the commitment type buyer her maximum attainable payoff in $G^\infty(\mu)$, corresponding to the socially efficient outcome. The seller however would be better off by playing against the normal buyer. As we have seen above, against the normal type buyer, the seller can get more than the cooperative outcome by randomizing between the (Cc_S, c_B) and (Dc_S, c_B) action profiles.

A normal type buyer can be distinguished from a commitment type buyer only if the seller plays Dc_S. In this situation, the normal buyer prefers to play c_B, while the commitment buyer prefers to play d_B. The normal buyer could however simulate the strategy of a commitment buyer in order to obtain the payoff of the latter (i.e. the cooperative outcome).

Because the cooperative strategy involves a loss for the seller (i.e. the potential loss of not being able to get the higher payoff that could be obtained against the normal buyer) the seller should not become "easily" convinced that he is playing against a commitment type buyer. The question is therefore, how long should the seller try to determine the true type of the buyer. Because every outcome (Dc_S, c_B) (i.e. the seller tests the type of the buyer and the buyer plays the commitment strategy) generates a loss for the seller, and because the seller cannot infinitely wait for future payoffs (the seller's discount factor is less than 1) it follows that at some point, if the seller always observes the commitment strategy being played by the buyer, he must give up trying to test the true type of the buyer, and accept playing a best response to the commitment type buyer.

Theorem 3. *If:*

1. *the seller has incomplete information in* G^∞*,*
2. *the seller assigns positive probability to the prior beliefs that the buyer is a "commitment" type and a "normal" type. i.e.* $\mu(\omega_0) > 0$*,* $\mu_0^* = \mu(\omega^*) > 0$ *and* $\mu(\omega_0) + \mu(\omega^*) = 1$*;*

Then there is a finite upper bound k_S on the number of times the seller plays Dc_S in G^∞:

$$k_S = \left\lceil \frac{\ln(\mu_0^*)}{\ln\left(\frac{(1-\delta_S)\overline{v}(1-\rho_S)+\delta_S\Phi}{(1-\delta_S)[\overline{v}(1-\rho_S)+\epsilon+\varepsilon_S]+\delta_S\Phi}\right)} \right\rceil$$

where δ_B, δ_S are the discount factors of the buyer and seller, ρ_S, ρ_B are their profit margins, $\varepsilon_B, \varepsilon_S$ are the lying fines imposed by the mechanism, $\overline{v}$ is the maximum value of a transaction, $\tilde{v}$ is the expected value of a transaction, θ is the additional fraction of the price a buyer has to pay when buying from completely trustworthy sellers, $\epsilon = g_S(R+, \rho_S v_i) - g_S(R-, v_i)$ is the loss of the seller caused by receiving a negative reputation report instead of a positive one, and:

$$\Phi = \tilde{v}(1-\rho_S)\frac{(1-\delta_B)\varepsilon_B + \delta_B\tilde{v}\theta}{\delta_B\tilde{v}(1+\rho_B)};$$

Proof. See Jurca and Faltings, [9], Theorem 1. □

The lower bound k_S restricts the set of possible equilibrium payoffs of the normal type buyer in $G^\infty(\mu)$. If a rational buyer mimics the commitment type buyer, she obtains in the worst case $\underline{V'}_B^t$; the outcome (Dc_S, c_B) in the first k_S rounds, followed by an infinite number of cooperative outcomes.

$$\underline{V'}_B^t = (1-\delta_B)\Bigg[(-v_t - \varepsilon_B) + \delta_B \sum_{\tau=t+1}^{t+k_S-1} \delta_B^{\tau-t-1}(-v_\tau - \varepsilon_B) + \delta_B^{k_S} \sum_{\tau=t+k_S}^{\infty} \delta_B^{\tau-t-k_S}\rho_B v_\tau\Bigg];$$

Any equilibrium strategy in $G^\infty(\mu)$ must guarantee the normal type buyer at least $\underline{V'}_B^t$. Let us reconsider the strategy s from the perfect information game G^∞ according to which the players play (Dc_S, c_B) with probability p and (Cc_S, c_B)

with probability $1-p$. By imposing that both $V_B^t|_{(Cc_S,c_B)}$ and $V_B^t|_{(Dc_S,c_B)}$ (Equation (3)) be greater or equal to $\underline{V'}_B^t$, the maximum value of p is:

$$p \leq \overline{p'} = \frac{(1-\delta_B)\varepsilon_B + (\delta_B - \delta_B^{k_S})(\tilde{v} + \varepsilon_B + \tilde{v}\rho_B)}{\delta_B \tilde{v}(1+\rho_B)}; \tag{5}$$

However, the constraints on p presented in Equation (4) remain valid, and therefore $p \leq \min(\overline{p}, \overline{p'})$.

Particular importance has the case in which $k_S = 1$. $\overline{p'}$ becomes:

$$\overline{p'} = \frac{(1-\delta_B)\varepsilon_B}{\delta_B \tilde{v}(1+\rho_B)}; \tag{6}$$

and as ε_B can be any positive value, $\overline{p'}$ will in the limit approach 0. In this situation, the reputation mechanism will receive false reputation reports with vanishing probability.

The result of Theorem 3 has to be interpreted as a worst case scenario. In real markets, sellers that already have a small predisposition to cooperate will defect fewer times. Moreover, the mechanism is self enforcing, in the sense that the more buyers act as commitment types, the higher will be the prior beliefs of the sellers that buyers will report truthfully, and therefore the easier it will be for the buyers to act as truthful reporters.

The following properties are also straightforward to derive as a direct consequence of Theorem 3:

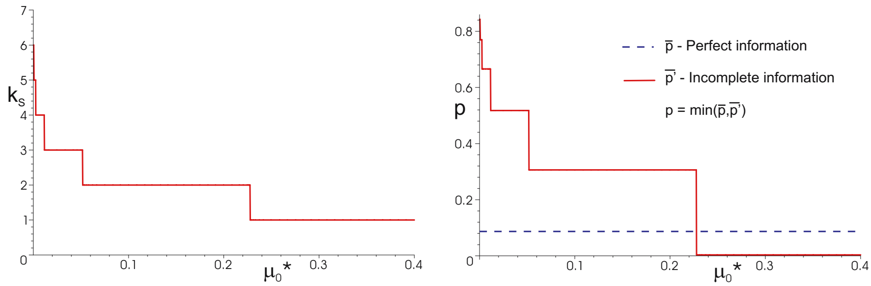
Property 1. The mechanism is bounded socially efficient.

Proof. Because of the lost exchange, outcome (Dc_S, c_B) generates a cumulated social loss of $(\rho_S+\rho_B)v_i$ every time it occurs. The perfect information equilibrium involves a possibly infinite number of rounds in which (Dc_S, c_B) is played. By limiting the number of times the seller is playing action D, we also limit to a finite number (i.e. k_S) the rounds in which the exchange does not occur. The social loss is therefore bounded above by $k_S(\rho_S + \rho_B)\overline{v}$. □

Property 2. The mechanism is weakly budget balanced.

Proof. The net payment to the mechanism is non-negative as every time there is a disagreement concerning the two reputation reports, the center gets $\varepsilon_B+\varepsilon_S$. By introducing supplementary service fees, the mechanism can be easily transformed into one that yields profit to the market. □

Numerical Example. Let us consider the example of a hotel who charges for a room a fixed price of $v = 140$ dollars a night. The profit margin of the hotel is $\rho_S = 0.95$ while the profit margin of the client is $\rho_B = 0.2$. The customer returns to the same hotel once a year with probability $\delta_B = 0.7$, and after each night spent in the hotel she is required to submit a binary reputation report about whether or not the hotel has kept its promise (in terms of a Service Level Agreement). The customer also has the option to go to another hotel which costs



(a) The upper limit k_S depending on μ_0^* (b) The probability of receiving a false report depending on μ_0^*

Fig. 2. Numerical Example

an additional 14 dollars a night ($\theta = 0.1$). The hotel discounts future revenues with $\delta_S = 0.95$.

We assume that the reputation of the hotel directly affects its occupancy (and future revenues) such that any time a hotel cheats and correctly receives a negative reputation report, it looses (in terms of future revenues) $\epsilon = 25$ dollars. When the fines ε_S and ε_B equal to 1 respectively 20 dollars, Figure 2(a) plots the value of the upper bound k_S for different values of the prior probability μ_0^*. For the same values of μ_0^*, Figure 2(b) plots the maximum value of the probability with which "CONFESS" will accept false reputation reports. When $\mu_0^* > 0.25$ the hotel will cheat at most once on a customer, and the probability of receiving a false reputation report is smaller than 0.3%.

3 Open Issues

Further benefits can be obtained if the buyers' reputation as honest reporters is shared in the market. A buyer that has once built a reputation for truthfully reporting the seller's behavior will benefit from cooperative trade during her entire lifetime, without having to convince each seller separately. The upper bound on the loss a buyer has to withstand in order to convince a seller that she is a commitment type, becomes an upper bound on the total loss a buyer has to withstand during her entire lifetime in the market. How to efficiently share the reputation of buyers within the market remains an open issue.

Correlated with this idea is the observation that buyers that use our mechanism are motivated to keep their identity. In generalized markets in which agents are encouraged to play both roles (e.g. a peer-2-peer file sharing market in which the fact that an agent acts only as "seller" can be interpreted as a strong indication of "double identity" with the intention of cheating) our mechanism also solves the problem signaled in [5] related to the ease with which agents can change their online identity. The price to pay for the new identity is the loss due to building a reputation as a honest reporter when acting as a buyer.

The mechanism can be criticized for being centralized. The market acts as a central authority by collecting listing fees from the seller and the buyer, by asking the reputation reports at the end of each transaction, and by reasoning about the outcome of the transaction. However, as the mechanism does not require any information to be transmitted from one round to another (the seller stores the reputation of the buyer) we could have the same seller and buyer interact in multiple markets (decentralized system) without having to rely on one single centralized institution.

One direction of future research is to study the behavior of the above mechanism when there is two-sided incomplete information: i.e. the buyer is also uncertain about the type of the seller. A seller type of particular importance would be the "greedy" seller type who always likes to keep the partner buyer to her minimum continuation payoff. In this situation we expect to be able to find an upper bound k_B on the number of rounds in which a rational buyer would be willing to test the true type of the seller. The condition $k_S < k_B$ would impose the constraints on the parameters of the system for which the reputation effect will work in the favor of the buyer: i.e. the seller will give up first the "psychological" war and revert to a cooperative equilibrium.

A somehow related problem is the robustness to mistakes, or imperfect monitoring of the opponent's actions. A seller's defection by mistake in a situation in which it was not rational for a seller to defect will be interpreted by the buyer as evidence of the seller's irrational behavior.

Last, but not least, we plan to adapt this truthful reporting mechanism for reputation mechanism that affect the value of future transactions. For such mechanisms the repeated interaction between a buyer and seller is much more complicated to model. A negative report submitted by a buyer at time t might lead to more beneficial trade for that buyer in the future (since the negative reputation report will attract a decrease in the price of future sold goods). Making it rational for the buyer to submit the true report involves a detailed understanding of the underlying reputation mechanism, the solution being most likely application dependent.

4 Conclusions

In this paper we formally prove that no binary reputation mechanism can be incentive compatible when the agents are rational, have game-theoretic complete information and the trusting agent (i.e. the buyer) interacts a finite number of times with the trusted agent (i.e. the seller). Moreover, we describe a truthful feedback elicitation mechanism ("CONFESS") for two long-run rational buyer and seller and give an intuitive presentation of how incentive compatibility can exist as an equilibrium. When the seller has incomplete information, the performance of our mechanism is greatly improved and we have been able to derive an upper bound on the percentage of false reports that are accepted by the mechanism. The mechanism we have presented does not require the presence of an independent verification authority, can be easily decentralized and accepts transactions of different values.

References

1. A. Birk. Learning to Trust. In R. Falcone, M. Singh, and Y.-H. Tan, editors, *Trust in Cyber-societies*, volume LNAI 2246, pages 133–144. Springer-Verlag, Berlin Heidelberg, 2001.
2. A. Biswas, S. Sen, and S. Debnath. Limiting Deception in a Group of Social Agents. *Applied Artificial Intelligence*, 14:785–797, 2000.
3. C. Dellarocas. Goodwill Hunting: An Economically Efficient Online Feedback. In J. Padget and et al., editors, *Agent-Mediated Electronic Commerce IV. Designing Mechanisms and Systems*, volume LNCS 2531, pages 238–252. Springer Verlag, 2002.
4. C. Dellarocas. Efficiency and Robustness of Binary Feedback Mechanisms in Trading Environments with Moral Hazard. MIT Sloan Working Paper #4297-03, 2003.
5. E. Friedman and P. Resnick. The Social Cost of Cheap Pseudonyms. *Journal of Economics and Management Strategy*, 10(2):173–199, 2001.
6. D. Fudenberg and D. Levine. Reputation and Equilibrium Selection in Games with a Patient Player. *Econometrica*, 57:759–778, 1989.
7. D. Fudenberg and E. Maskin. The Folk Theorem in Repeated Games with Discounting or Incomplete Information. *Econometrica*, 54(3):533–554, 1989.
8. R. Jurca and B. Faltings. An Incentive-Compatible Reputation Mechanism. In *Proceedings of the IEEE Conference on E-Commerce*, Newport Beach, CA, USA, 2003.
9. R. Jurca and B. Faltings. Truthful reputation information in electronic markets without independent verification. Technical Report ID: IC/2004/08, EPFL, http://ic2.epfl.ch/publications, 2004.
10. D. M. Kreps, P. Milgrom, J. Roberts, and R. Wilson. Rational Cooperation in the Finitely Repeated Pisoner's Dilemma. *Journal of Economic Theory*, 27:245–252, 1982.
11. D. M. Kreps and R. Wilson. Reputation and Imperfect Information. *Journal of Economic Theory*, 27:253–279, 1982.
12. P. Milgrom and J. Roberts. Predation, Reputation and Entry Deterrence. *J. Econ. Theory*, 27:280–312, 1982.
13. N. Miller, P. Resnick, and R. Zeckhauser. Eliciting Honest Feedback in Electronic Markets. Working Paper, 2003.
14. K. M. Schmidt. Reputation and Equilibrium Characterization in Repeated Games with Conflicting Interests. *Econometrica*, 61:325–351, 1993.
15. R. Selten. The Chain-Store Paradox. *Theory and Decision*, 9:127–159, 1978.
16. B. Yu and M. Singh. An Evidential Model of Distributed Reputation Management. In *Proceedings of the AAMAS*, Bologna, Italy, 2002.

Generalized Knapsack Solvers for Multi-unit Combinatorial Auctions: Analysis and Application to Computational Resource Allocation

Terence Kelly

Hewlett-Packard Laboratories, Palo Alto CA 94304 USA
kterence@hpl.hp.com

Abstract. The problem of allocating discrete computational resources motivates interest in general multi-unit combinatorial exchanges. This paper considers the problem of computing optimal (surplus-maximizing) allocations, assuming unrestricted quasi-linear preferences. We present a solver whose pseudo-polynomial time and memory requirements are linear in three of four natural measures of problem size: number of agents, length of bids, and units of each resource. In applications where the number of resource *types* is inherently a small constant, e.g., computational resource allocation, such a solver offers advantages over more elaborate approaches developed for high-dimensional problems.

We also describe the deep connection between auction winner determination problems and generalized knapsack problems, which has received remarkably little attention in the literature. This connection leads directly to pseudo-polynomial solvers, informs solver benchmarking by exploiting extensive research on hard knapsack problems, and allows E-Commerce research to leverage a large and mature body of literature.

1 Introduction

Recent years have witnessed an explosion of interest in combinatorial auctions (CAs), which permit agents to define utility over *bundles* of different types of goods. Although CAs are applicable to a wide range of allocation problems, the U.S. Federal Communications Commission's spectrum allocation problem largely motivated the 1990s surge of CA research [1, 2]. Special properties of spectrum auctions—particularly the restriction that only a single unit of each type of good is available—received much attention in E-commerce research literature. An important measure of problem size in a single-unit CA is the number of good types, and for this measure the winner determination problem (WDP) is NP-hard by reduction from the weighted set packing problem [3].

An unfortunate consequence of excessive attention to single-unit CAs has been excessive pessimism regarding efficient and exact winner determination in more general problems. The few papers that have considered multi-unit CAs (MUCAs) report that the WDP is NP hard when problem size is measured by number of good types [4, 5, 1]. Other natural measures, e.g., number of available *units* of each good, number of agents, and the length of bids, receive far less attention.

This paper follows a very different trajectory from practical motivation to conclusions regarding the computational complexity of CA WDPs. We begin with the problem

P. Faratin and J.A. Rodríguez-Aguilar (Eds.): AMEC 2004, LNAI 3435, pp. 73–86, 2005.

of allocating resources in large computing centers. The number of resource *types* in this problem is a *small constant*, whereas the number of *units* of each resource is large and variable. The optimal allocation problem is a generalized multi-dimensional knapsack problem (MDKP): allocating a bundle of goods to an agent reduces the pool of available goods, just as placing an item in a container with multiple capacity constraints (e.g., weight, volume) reduces its remaining capacity along each dimension.

The deep connection between WDPs and KPs leads to pseudo-polynomial exact algorithms for problems of fixed dimensionality. Very simple exact solvers exist whose time and memory requirements are *linear* in the number of agents, length of bids, and number of units of each resource. Such solvers are entirely practical for low-dimensional problem instances (i.e., few resource types) and are an attractive default solution method whenever their computational costs are not prohibitive. In all cases they provide a well-understood baseline for comparison with more elaborate methods.

Straightforward MUCA WDP solvers inspired by the auction-knapsack connection invite more detailed, more balanced, and more nuanced analyses than are typically performed on complex heuristic solvers designed for high-dimensional problems. Knapsack-based WDP solvers furthermore support very general combinatorial exchanges with essentially no restrictions on the expression of agent utility functions. The connection between CA WDPs and generalized KPs allows us to retain much of the flexibility and generality of integer programming [6] while exploiting the special structure of KPs to obtain simple and efficient exact solvers. In special cases such as *single-good* multi-unit auctions, textbook uni-dimensional KP solvers compare rather well with specialized WDP algorithms. Finally, WDP benchmarks can draw upon extensive Operations Research literature on hard KP instances.

The boundaries of the present investigation are as follows: We consider only one-shot sealed-bid auctions, an important subset of auction types in a comprehensive taxonomy [7]. We consider only discrete allocation (integral quantities of goods). Our results apply to the allocator of proper economic mechanisms such as the Generalized Vickrey Auction (GVA) [8] or Vickrey-Clarke-Groves (VCG) mechanisms [9], but we do not consider incentive issues surrounding auctions. Finally, although a wide range of approximation schemes for KPs have been proposed, we restrict attention to exact methods. This is appropriate in light of recent results on the necessity of exact solvers for incentive-compatible mechanisms [10,11,12,13]. A longer version of this paper [14] includes material omitted due to space limitations.

The remainder of this paper is structured as follows: Section 2 motivates interest in low-dimensional MUCAs with a discussion of resource allocation in large computing centers. Section 3 formulates our general allocation problem and explains its relation to auction winner determination. Section 4 presents a general solver for multi-unit CAs with unrestricted preference expression and analyzes its computational complexity. Section 4.2 discusses hard KP instances, Section 5 reviews related work, and Section 6 concludes with a discussion.

2 Motivation: Data Center Allocation

Large tightly-coupled computers remain popular for enterprise computing, and today entire data centers comprising large numbers of loosely-coupled hosts are offered as commercial products [15]. Resource allocation in both contexts has several properties that recommend auction-mediated negotiation, and knapsack-based optimal allocators are ideal WDP solvers for these contexts.

The number of abstract resource types in computational allocation problems is inherently a *small constant*, because only a few fundamental operations can be performed on data: data can be manipulated, stored, and transported. Corresponding resource types—processing, storage, and bandwidth—often suffice in models of computational resource allocation [16]. For reasons of fault isolation, security, and performance isolation, most computing resources are allocated in *integral* quantities; examples include CPUs, switch ports, and logical devices (LDEVs) in consolidated storage arrays. By contrast, the number of *units* of each resource is large and expands with user needs.

Data centers are partitioned so that an application's performance depends only upon the resources it receives; in auction contexts this property is sometimes called "no externalities" [17]. Multi-tiered applications for large computing environments are *horizontally scalable* by design, i.e., they exploit variable quantities of resources at each tier. Application performance exhibits both complementarities and substitutabilities across resource types. For example, one application may require minimal quantities of both memory and bandwidth in order to perform acceptably; another may compensate for lack of bandwidth by exploiting an additional CPU for data compression. The utility that accrues to an application is a complex function of the *bundle* of resources it receives; this property recommends combinatorial auctions.

While the number of applications simultaneously sharing an enterprise computing center may be large, the number of self-interested agents among whom resources are allocated may be small. Agents might correspond to departments or projects within a firm, or to firms within a consortium that jointly owns a data center. If the number of agents is so large that each agent's potential influence on allocative outcomes is negligible, competitive (i.e., non-strategic) behavior may be a reasonable normative assumption. However strategic behavior is to be expected if few agents are involved. Incentive-compatible mechanisms (in which truth-telling is a dominant strategy for agents) are therefore desirable, even for allocation within a hierarchical organization [18]. Given that the incentive properties of GVA/VCG mechanisms sometimes require *exact* WDP solvers [10, 11, 12, 13], we prefer exact solvers to approximate ones where possible.

Computational resource allocation can be formalized as a generalized knapsack problem [19]; Section 3 describes a suitable formulation. Our straightforward solver, presented in Section 4, is appropriate to the special properties of data-center allocation. Its computational complexity is exponential in the number of resource *types* but is linear in the number of available *units* of each resource and in all other natural measures of problem size. A simple implementation of the solver produces, as a side effect, a table describing the aggregate utility of any subset of the data center's resource pool, thereby providing a wealth of information about the marginal value of various resource types. This information might be useful for purposes other than allocation, e.g., capacity planning.

3 Problem Formulation

We are given R resource types and T agents. At most N_r indivisible units of resource type r are available, $r = 1,\ldots,R$. Each agent's utility function is represented by defining utility over a list of resource bundles; the list may be arbitrarily long, and may therefore may represent any utility function. If agent utility naturally takes a more compact form than a list of (bundle, utility) pairs, the former may easily be translated into the latter. The length of utility functions when defined explicitly over bundles is not prohibitive in low-dimensional cases (i.e., where the number of resource types R is a small constant).

Our goal is to maximize aggregate utility by choosing exactly one bundle from each list, subject to resource scarcity. Let B_t denote the number of bundles in agent t's utility function, and let $q_{tb} = (q_{1tb},\ldots,q_{Rtb})$ and u_{tb} respectively denote the quantities of resources in bundles and the utility of bundles, $b = 1,\ldots,B_t$. Binary decision variable $x_{tb} = 1$ if agent t receives the bth resource bundle on its list, zero otherwise. Formally, our "multi-dimensional multiple-choice knapsack problem" (MDMCK) is the following integer program:

$$\text{maximize} \quad \sum_{t=1}^{T} \sum_{b=1}^{B_t} x_{tb} u_{tb} \tag{1}$$

$$\text{subject to} \quad \sum_{b=1}^{B_t} x_{tb} = 1 \qquad t = 1,\ldots,T \tag{2}$$

$$\sum_{t=1}^{T} \sum_{b=1}^{B_t} x_{tb} q_{rtb} \le N_r \quad r = 1,\ldots,R \tag{3}$$

The inequality in Equation 3 permits unallocated goods; to forbid them we simply replace it with equality. In the latter case we can express arbitrary disposal costs of unallocated goods via an additional agent utility function. The solver of Section 4 takes a different approach: it accepts an explicit disposal cost function as an input.

MDMCK includes classic knapsack problems as special cases [19]. Extensive literature exists on these special cases, but relatively little on MDMCK itself. Kellerer *et al.* devote roughly three pages to MDMCK and identify approximate heuristic algorithms dating back to 1997 [20]. They report that to the best of their knowledge no exact algorithm for MDMCK has ever been published. In fact, Tennenholtz briefly sketched an exact solver suitable for low-dimensional MDMCK instances, without analyzing its complexity or connecting the WDP to generalized KPs [21].

Two-resource MDMCK admits simple graphical illustration (Figure 1). A bundle/utility pair in a utility function is represented as a rectangle labeled with agent ID (left). Utility functions are collections of such rectangles (center). The solution is illustrated at right: a bundle is chosen from each utility function such that utility is maximized while total resource usage does not exceed any capacity dimension.

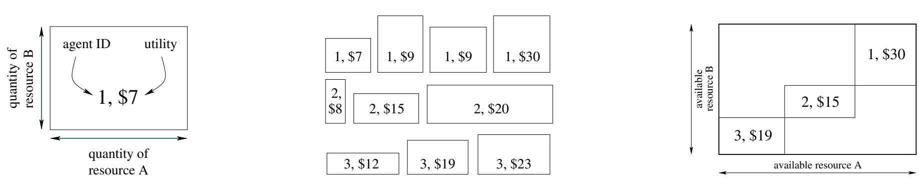


Fig. 1. 2-D MDMCK. *Left:* resource bundle. *Center:* utility functions. *Right:* optimal allocation.

3.1 Application to Auctions

In an auction setting, we refer to an agent's list of (q_{tb}, u_{tb}) pairs as its *bid*. We shall ignore the relationship between an agent's *reported* and *true* utility functions except to note that they may differ and that our allocator receives the former. The constraint of Equation 3 ensures that each agent receives exactly one bundle defined by its bid. In other words, we permit "XOR bids," which in turn permit the expression of arbitrary preferences [17].

The MDMCK formulation requires that each agent's utility depends only on the bundle of resources the agent itself receives ("no externalities" [17]). No other restrictions on agent preferences are inherent. For example, MDMCK allows goods to be "bads," i.e., free disposal is not required. Furthermore agent utility need not be "normalized" in the sense that no change in goods owned implies no change in utility.

Some prior work on single-good-type/multi-unit auctions has restricted the form of bids, e.g., demand must be monotonic in per-unit price [22] or atomic bids are forbidden [23]; monotonicity restrictions have also appeared in multi-good CA analyses [24]. In the single-good-type case, divisibility is required for existence of a uniform price that maximizes surplus *according to restricted-form bids* (which might not represent actual agent preferences). Uniform prices are sometimes desirable, e.g., for reasons of perceived fairness. The real motivation for bid restrictions, however, has sometimes been to facilitate efficient WDP algorithms [25].

Computational issues aside, the greater generality and flexibility of a MDMCK formulation makes it attractive if uniform prices are not required. The components of resource vectors q and utilities u may assume both negative and positive values, allowing agents to express willingness to engage in complex *atomic* (all-or-nothing) transactions. Thus the MDMCK formulation supports very general combinatorial exchanges, e.g., the dozen CA variants considered in Ref. [26].

3.2 Auction and KP Taxonomies

CA WDPs are often linked to set packing, even in the multi-unit case [5]. Connections with generalized knapsack problems, however, seem more natural and more useful for several reasons. First, KPs are more widely known among nonspecialists, e.g., implementors in industry; they are intuitive, memorable, and invite graphical interpretation (Figure 1). KPs are also far more widely studied. Most importantly, KPs admit pseudo-polynomial solution under restrictions that are sometimes acceptable in practice. Whereas connections with set packing have led to the pessimistic view that "CA WDPs are NP hard," the knapsack connection encourages cautious optimism.

Consider three aspects of sealed-bid auctions and their knapsack counterparts:

1. number of types of goods in an auction / dimensionality of a KP;
2. number of units of each good / capacity of KP container in each dimension; and
3. number of bundles in bids / the "multiple-choice" aspect of KP.

In each case the characteristic may be single or multiple, e.g., an auction may involve multiple units of a single good type, or single units of multiple good types. Table 1 summarizes the seven meaningful combinations of these possibilities. When KP items

Table 1. Auction types and winner-determination problems (S=single, M=multiple)

good types	units	bundles	common name / examples	winner-determination problem
S	S	S	first price	find max
S	M	S	double auctions, single-quantity bids	0-1 KP; subset-sum if #units $\propto$ utility
S	M	M	double auctions, XOR bids	multiple-choice KP (MCKP)
M	S	S	"combinatorial auctions"	weighted set packing (WSP) [3]
M	S	M	single-unit CA, XOR bids	convert to WSP via "dummy goods"
M	M	S	multi-unit CA, single-bundle bids	multi-dimensional KP (MDKP)
M	M	M	multi-unit CA, XOR bids [4]	MDMCK [19]

are partitioned into disjoint sets and we must choose exactly one item from each set, we say that a "multiple-choice" constraint applies; this corresponds to an XOR constraint across elements of a compound bid. The most general KP shown is MDMCK, which corresponds to multi-unit CAs with arbitrary XOR bids (MMM in Table 1).

It is straightforward to convert an instance of the MSM problem to an MSS instance by adding "dummy goods" to enforce multiple-choice/XOR constraints: introduce an extra good type for each agent, one unit of which is included in each of the agent's bundles and of which exactly one unit is available [4]. MMM instances can be converted to MMS instances in the same way. This transformation increases the dimensionality of problem instances, which may increase computational burdens for some solvers.

Several of the correspondences in Table 1 have been noted previously. Kothari *et al.* mention in a footnote that their single-good multi-unit WDP is similar "in spirit" to MCKP, citing a 1970s reference [22]. However they quickly dismiss the connection on grounds that MCKP leads to an infeasible formulation. In fact, simple MCKP solvers in modern texts scale rather well with problem size (see Section 5.2), and efficient specialized solvers are the subject of sophisticated recent research [27]. Holte observes that Operations Researchers have long investigated MDKPs that are substantively identical to multi-unit CA WDPs [28], contrary to claims in recent E-commerce literature that MUCA WDPs were never before studied [4]. Years later, however, MUCA WDP research that cites Holte does not mention the connection he made [29]. A very recent text on KPs discusses Holte's insight in considerable detail but does not make the connection between MDMCK and multi-unit CAs with XOR bids; instead it suggests the use of dummy goods to enforce XOR constraints for a MDKP solver [20]. Overall, we find remarkably few references to knapsack problems in recent literature on auction WDPs, and nothing approaching a comprehensive treatment of the relationship between the two in the E-commerce literature. Section 5 considers in greater detail the state of the E-commerce literature in this regard.

4 Dynamic Programming Solver

This section presents a simple dynamic programming (DP) algorithm for MDMCK; it generalizes multi-dimensional and multiple-choice KP solvers [30, 20].

Let $N = (N_1, \ldots, N_R)$ denote the multi-dimensional "size" of our resource pool, and let 0 denote the R-vector consisting entirely of zeros. We say that $a \geq b$ if every component of vector a is not less than the corresponding component of b.

Given an integer $\hat{t}$ and a resource pool size n, we define $F_{\hat{t}}(n)$ to be the optimal value of our objective function (Equation 1) for the sub-instance of MDMCK involving only agents $1, \ldots, \hat{t}$ and a resource pool of size n. $F_0(n)$ defines the utility of unallocated resources for feasible "leftovers" $n \geq 0$ and defines utility as $-\infty$ for infeasible allocations. Similarly we define $A_{\hat{t}}(n)$ as the bundle assigned to agent $\hat{t}$ by the optimal assignment for the sub-instance defined by $\hat{t}$ and n. F and A may be defined recursively:

$$F_{\hat{t}}(n) = \begin{cases} -\infty & \hat{t} = 0, \ \neg(n \geq 0) \\ D(n) & \hat{t} = 0, \quad n \geq 0 \\ \max\limits_{b \in \mathcal{B}_{\hat{t}}} \{F_{\hat{t}-1}(n - q_{\hat{t}b}) + u_{\hat{t}b}\} & 1 \leq \hat{t} \leq T \end{cases} \tag{4}$$

$$A_{\hat{t}}(n) = \arg\max_{b \in \mathcal{B}_{\hat{t}}} \{F_{\hat{t}-1}(n - q_{\hat{t}b}) + u_{\hat{t}b}\} \quad 1 \leq \hat{t} \leq T \tag{5}$$

where $\mathcal{B}_{\hat{t}} = \{1, \ldots, B_{\hat{t}}\}$ and D expresses the (dis)utility of unallocated resources. To permit unallocated goods at no cost we simply set $D = 0$; to forbid unallocated goods we set $D = -\infty$. $F_T(N)$ is the value of an optimal solution, and the corresponding choices of bundles may be recovered as $A_T(N), A_{T-1}(N - q_{TA_T(N)})$, etc.; conversion to decision variables x_{tb} of Equations 1 through 3 is trivial.

We may evaluate the dynamic program in at least two ways: by constructing tables corresponding to $F(\cdot)$ and $A(\cdot)$ in bottom-up fashion, or by recursively evaluating $F_T(N)$ and $A_T(N)$. The former strategy yields a full $F_T(n)$ table containing information about the marginal utilities of every resource type for every resource pool size $n : 0 \leq n \leq N$; this may be useful for purposes other than allocation, e.g., capacity planning. A disadvantage of the bottom-up approach is that it achieves worst-case performance on *all* inputs. Top-down evaluation may save time on some inputs by evaluating $F(\cdot)$ and $A(\cdot)$ for fewer $(\hat{t}, n)$ pairs, and may permit more space-efficient representation of the tables than naïve arrays. Top-down evaluation admits a variety of optimizations and elaborations, including lower-bound heuristics and pruning via upper bounds; with such embellishments it resembles branch-and-bound (B&B) search. A no-frills top-down C implementation of our solver runs to several dozen lines of code, comparable to succinct uni-dimensional KP solvers [31].

4.1 Computational Complexity

The worst-case time and memory complexity of a straightforward implementation of the the dynamic program are easy to analyze. We assume a bottom-up implementation that stores $F(\cdot)$ and $A(\cdot)$ values in ordinary arrays. We assume that the coefficients describing a problem instance are integers from a bounded range, and without loss of generality we assume that all coefficients are non-negative. (A natural expression of a fully general two-sided exchange WDP might be an instance of MDMCK with negative coefficients, but such an instance can be efficiently transformed to one with non-negative coefficients in a simple pre-processing step without altering the optimal values of decision variables.)

The dynamic program requires storage proportional to $T\prod_{r=1}^{R} N_r$. Evaluating Equations 4 and 5 requires time proportional to $R\sum_{t=1}^{T}(B_t\prod_{r=1}^{R} N_r)$ where the R term is due to the R-dimensional vector subtraction in the recursive calls to F. If $N_r = N$ for each resource, and if each agent defines utility over B resource bundles, then the storage requirement is $O(TN^R)$ and the time requirement is $O(RTBN^R)$. If each agent defines utility over all N^R possible resource bundles (the case of rational preferences) then the time requirement becomes $O(RTN^{2R})$.

The classic 0-1 and integer knapsack problems are NP-hard [32, 33]. MDMCK includes these as special cases, and therefore it too is NP-hard. However knapsack problems are *not* NP-hard in the strong sense, i.e., they admit pseudo-polynomial solution if dimensionality is fixed. See Papadimitriou & Steiglitz for a good discussion of pseudo-polynomial complexity analysis applied to classic KPs [33]. If we need not support problem instances with enormous coefficients, pseudo-polynomial bounds are the most natural and insightful description of algorithmic complexity. Restricting u_{tb}, q_{rtb}, and N_r to the length of modern machine words, e.g., 64 bits, is unlikely to be problematic in practical allocation problems.

For classic uni-dimensional problems, branch-and-bound algorithms are often favored over DP *except for hard problem instances*, where DP usually performs better [30, page 36]. For high-dimensional problems the computational costs of DP are prohibitive and the best method may be general integer programming (IP). Modern IP solvers support convenient and rapid solution of a wide range of WDPs [6] and compute approximate solutions to large MDMCK instances very rapidly [19].

4.2 Hard Knapsack Problems

Real-world CA WDP instances are not available for solver benchmarking, so we must rely on synthetic benchmarks. A thorough evaluation of any WDP solver should include instances intended to mimic typical inputs, such as those generated by CATS [34], as well as hard instances to expose worst-case behavior. The connection between WDPs and KPs allows us to exploit many years of research on hard KP instances for WDP solver evaluation.

There are two ways to construct hard instances of classic uni-dimensional knapsack problems. The first is to make the coefficients enormous; Chvátal describes how large they must be in order to foil a range of common solution methods [35]. We shall continue to assume that coefficients are bounded and therefore focus on the second method, which involves the relationship between bundle size and utility.

The size/utility relationship is easy to visualize in the uni-dimensional case. Figure 2, after Pisinger [36], illustrates four possibilities; Martello *et al.* and Kellerer *et al.* describe others [37, 20]. Strongly-correlated instances are among the hardest

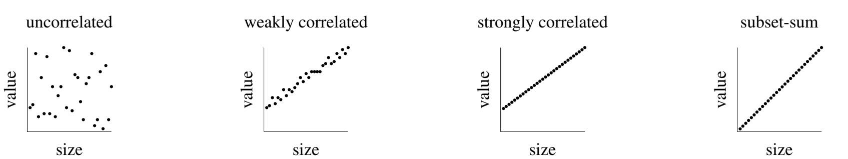


Fig. 2. Uni-dimensional KPs

for today's best KP solvers and are the subject of ongoing research [38, 37]. An extended version of this paper describes how to construct generalized multi-dimensional strongly-correlated instances [14].

It is interesting to note that early empirical evaluations of KP solvers focused excessively on "easy" problem instances, specifically the uncorrelated and weakly-correlated cases of Figure 2; only later did attention within the OR literature shift to characterizing hard instances and using them in solver benchmarks [37]. A similar pattern is evident in evaluations of WDP solvers many years later, as Andersson *et al.* have noted [6]; see also Section 5.2. It is reasonable to speculate that mis-steps in WDP benchmarking might have been avoided if connections between WDPs and KPs had been more prominent in E-commerce research.

5 Related Work

(This section has been reduced due to space limitations; see [14] for the full version.) The literature on knapsack problems is vast and growing. An excellent text by Martello & Toth [30] is now out of print, but a very recent book by Kellerer *et al.* provides updated and expanded coverage, including multi-dimensional problems and MDMCK itself [20]. Martello *et al.* review recent research on exact solutions for large hard instances of 0-1 KP [37]. Pisinger summarizes the state of the art in uni-dimensional KP research c. 1995 [36], much of which is directly applicable to subsequent research on single-good-type/multi-unit WDPs [23, 22].

5.1 WDP-KP Connections

An extensive literature search revealed little mention of the connection between auction WDPs and KPs and nothing approaching a comprehensive treatment. Recent surveys on combinatorial auctions and auction theory [3, 2, 39, 1, 40, 26] do not discuss knapsack problems. The string "knap" appears in exactly five papers among all past proceedings of the ACM Conference on E-Commerce (EC). Two are unrelated to our interests, two mention in passing a relationship between special cases of WDP and KPs [22, 5], and one uses reduction from KP to prove NP-hardness [24]. Occasionally papers in other fora note that single-good-type auction WDPs are KPs, usually to establish intractability and sometimes to note the existence of pseudo-polynomial algorithms [41].

In several cases E-commerce research has missed opportunities to build upon relevant prior work—and failed to acknowledge it—perhaps because the WDP-KP connection has been overlooked. For instance, the fact that multi-dimensional KPs do not admit fully-polynomial approximation, even in the two-dimensional case, has been known since 1979 [20, p. 252]. Twenty years later, the question of whether CA WDPs admit approximation was described as "open" [42, p. 10].

Queries to six literature search engines for "auction," "knapsack," and "auction AND knapsack" yielded results summarized in Table 2. In all cases the conjunctive query yielded far fewer hits than the two basic queries; none of the "auction AND knapsack" papers contained a detailed or systematic treatment of the WDP-KP connection. A few papers mention in passing a deep relationship between WDPs and KPs and a handful casually state that the connection is well known, without saying by whom;

Table 2. Summary of keyword searches, December 2003 and January 2004

Source	Total docs	"auction"	"knapsack"	both
Springer Link	?	152	103	zero
IEEE Xplore	990,765	313	150	zero
ACM Digital Library	125,779	802	427	10
CiteSeer	?	1,686	922	12
Science Citation Index	33,117,604	2,379	989	zero
Elsevier Science Direct	"over 4M"	5,143	2,084	11

see [14] for citations. Somewhat ironically, the only detailed discussion of the connection between combinatorial auction WDPs and generalized KPs of which we are aware occurs in a very recent text written primarily by Operations Researchers with little interest in E-commerce [20, pp. 478–482].

In summary, the WDP-KP relationship is neither noted nor exploited widely in E-commerce research at the intersection of computer science and auction theory. The remainder of this section reviews selected literature on multi-unit auction WDPs, showing how the KP literature can enhance several of these contributions.

5.2 Multi-unit Auction WDPs

Kothari *et al.* consider *single-good-type* multi-unit auctions and introduce a fully-polynomial algorithm to compute approximately surplus-maximizing allocations [22]. Bids are restricted in several ways: they are divisible, the utility they express is monotonic in per-unit price, and their length is bounded. This paper mentions in passing that its allocation problem can be solved by a multiple-choice KP solver and that fully-polynomial approximation algorithms exist for MCKP. However it offers no detailed comparison with earlier approximate MCKP solvers or with simple exact algorithms.

A textbook DP algorithm for MCKP [30, page 78] applied to the single-good multi-unit WDP supports a completely general two-sided exchange with unrestricted bids. In the special case of a forward auction with N units for sale and T agents whose bids define utility over all possible quantities $0,\ldots,N$, the (pseudo-polynomial) time and memory requirements of this very simple exact method are respectively $O(TN^2)$ and $O(TN)$. Similar computational properties apply to the special case of a reverse auction; more sophisticated algorithms with improved asymptotic bounds exist [20]. The algorithm of Kothari *et al.* computes a $(1+\varepsilon)$ approximation for the restricted-bid problem and requires $O(T^3/\varepsilon)$ time. A detailed comparison with the textbook DP solver would place the new contribution in better perspective and would illuminate the tradeoffs between computational complexity and generality that are available to us. Discussion of the need for fully-polynomial (vs. pseudo-polynomial) algorithms would help to motivate the new method.

Bassamboo *et al.* consider *online* bid processing in single-good-type multi-unit auctions with indivisible (all-or-nothing) single-quantity bids [43]. They describe a remarkably storage-efficient algorithm for maintaining a small set of potentially winning bids prior to clearing; bids that cannot potentially win at the time they arrive are rejected, permitting the bidder to adjust her bid if desired. These authors note that literature on online knapsack problems exists, but does not precisely match the auction rules they consider.

Tennenholtz notes that the multi-good-type/multi-unit WDP is "tractable" when the number of types of goods is fixed, and describes a longest-paths dynamic programming algorithm in the context of a two-good-type example [21]. It is not clear whether the intended meaning is that polynomial or pseudo-polynomial solutions exist (the former cannot be true, because this WDP includes NP-hard problems MCKP and 0-1 KP as special cases). Neither knapsack problems nor their close relationship with longest-path problems [44, p. 100] are mentioned, nor are time and memory complexity analyses presented. A later version of the paper omits the DP algorithm entirely [45].

WDP solver research for multi-good-type/multi-unit CAs has emphasized heuristic branch-and-bound algorithms [4, 5]. Such approaches are entirely reasonable, particularly for high-dimensional problems in which DP solvers are likely to be infeasible. Comparisons with DP-based KP solvers could enhance B&B investigations by encouraging more detailed analyses of worst-case time and memory requirements in terms of all measures of problem size. B&B research to date has emphasized the number of good *types*, sometimes without detailed quantitative analysis of computational requirements [4]. Furthermore, benchmarks for multi-unit CAs could draw upon extensive research on hard KP instances. Empirical evaluations of MUCA WDP solvers to date have employed similar input synthesis procedures [4,5,26], which produce multi-dimensional variants of the uncorrelated and weakly correlated cases of Figure 2; for uni-dimensional KPs, these are not hard instances.

Finally, awareness of the WDP-KP connection would support more succinct and more precise descriptions of novel WDP algorithms. Leyton-Brown *et al.*, for instance, introduce a "polynomial" subroutine for pre-processing bids for a single good type ("singletons") [4,29]. In fact, this subroutine implements the classic *pseudo*-polynomial DP algorithm for the NP-hard 0-1 knapsack problem.

6 Discussion

This paper has compared two very different trajectories of CA research, summarized in Table 3. Motivated largely by FCC spectrum auctions, most CA research over the past decade has taken the number of *types* of goods as a measure of problem size while fixing the number of *units* of each good at 1. This paper begins with the problem of

Table 3. Trajectories of CA research

	single-unit/high-dimensional	multi-unit/low-dimensional
practical motivation	spectrum auctions	computational resource allocation
# good types	variable, high	low, fixed
# units/type	fixed at 1	variable, high
WDP	weighted set packing	generalized knapsack problem
conventional wisdom	"WDP is NP-hard," rational preferences infeasible	*linear* solvers available, rational preferences okay
solver research	heuristic B&B, restricted prefs	exact DP, any preferences
OR leverage	limited, late	extensive, early

computational resource allocation in modern data centers, which involves few types of goods but many units of each. Whereas comparisons with set packing have led to the conclusion that the WDP is intractable in the single-unit/high-dimensional case, different natural measures of problem size lead us to conclude that the WDP admits pseudo-polynomial solution in the multi-unit/low-dimensional case. Realization that WDPs are special cases of MDMCK leads to a very general solver whose simplicity invites thorough analysis.

By recognizing connections between knapsack problems and winner determination, we bring a wealth of Operations Research knowledge to bear on problems central to multi-agent resource allocation. This eliminates duplication of effort by allowing E-commerce research to focus on *typical* WDP instances while leaving to the OR community the task of characterizing *hard* cases. It also allows WDP solver research to focus on novel methods only when real-world instances offer optimization opportunities that are not exploited by general-purpose KP solvers.

Straightforward dynamic-programming KP solvers offer several attractive properties, including analytic tractability and simplicity of implementation. These in turn reduce errors, which have been discovered in elaborate B&B solvers after publication [46]. If nothing else, DP provides a well-understood baseline for comparisons of more sophisticated methods and highlights tradeoffs between algorithmic intricacy and computational efficiency. Furthermore for hard instances of low-dimensional problems, DP may simply outperform alternatives. In the special case of single-good/multi-unit auctions, textbook KP solvers provide exact solutions for unrestricted inputs and scale remarkably well with problem size; at the very least, they merit detailed comparison with approximation algorithms for restricted problems.

We have shown that a practical multi-agent allocation problem involving computational resources lends itself readily to formulation as a generalized knapsack problem, and that for this low-dimensional problem an extremely simple DP solver scales to instances of non-trivial size. In future work we intend to compare the performance of DP, B&B, and integer program solvers on a range of synthetic MDMCK instances and, if possible, to characterize analytically the instances best suited to each solution method.

Acknowledgments

I thank Hans Kellerer, Anshul Kothari, Daniel Lehmann, Kevin Leyton-Brown, Jeff Mason, David Pisinger, Daniel Reeves, Yoav Shoham, Bill Walsh, and Michael Wellman for discussions and pointers to literature. Moises Goldszmidt, Eugene Nudelman, Ryan Porter and the anonymous reviewers provided detailed feedback on earlier versions. Wei Deng helped with the literature review.

References

1. de Vries, S., Vohra, R.V.: Combinatorial auctions: A survey. INFORMS Journal on Computing **15** (2003) 284–309
2. Pekec, A., Rothkopf, M.H.: Combinatorial auction design. Management Science **49** (2003) 1485–1503

3. Rothkopf, M.H., Pekec, A., Harstad, R.M.: Computationally manageable combinatorial auctions. Management Science **44** (1998) 1131–1147
4. Leyton-Brown, K., Shoham, Y., Tennenholtz, M.: An algorithm for multi-unit combinatorial auctions. In: Proc. of the 17th National Conference on Artificial Intelligence. (2000)
5. Gonen, R., Lehmann, D.: Optimal solutions for multi-unit combinatorial auctions: Branch and bound heuristics. In: ACM E-Commerce. (2000)
6. Andersson, A., Tenhunen, M., Ygge, F.: Integer programming for combinatorial auction winner determination. In: ICMAS. (2000)
7. Wurman, P.R., Wellman, M.P., Walsh, W.E.: A parametrization of the auction design space. Games and Economic Behavior **35** (2001) 304–338
8. Varian, H., MacKie-Mason, J.K.: Generalized Vickrey auctions. Technical report, Dept. of Economics, University of Michigan (1994)
9. Mas-Colell, A., Whinston, M.D., Green, J.R.: Microeconomic Theory. Oxford University Press (1995) ISBN 0-19-507340-1.
10. Nisan, N., Ronen, A.: Algorithmic mechanism design. Games and Economic Behavior **35** (2001) 166–196
11. Lehmann, D., O'Callaghan, L., Shoham, Y.: Truth revelation in approximately efficient combinatorial auctions. Journal of the ACM **49** (2002) 577–602
12. Nisan, N., Ronen, A.: Computationally feasible VCG mechanisms. In: Proceedings of the ACM Conference on Electronic Commerce. (2000) 242–252
13. Lavi, R., Mu'alem, A., Nisan, N.: Towards a characterization of truthful combinatorial auctions (extended abstract). In: IEEE FOCS. (2003)
14. Kelly, T.: Generalized knapsack solvers for multi-unit combinatorial auctions. Technical Report HPL-2004-21, HP Labs (2004)
15. Hewlett-Packard Corporation: HP Utility Data Center (UDC) overview (2004) `http://h30046.www3.hp.com/solutions/overview.html`.
16. Hewlett-Packard Corporation: An economy of IT: Allocating resources in the computing utility (2003) `http://www.hpl.hp.com/news/2003/oct_dec/computons.html`.
17. Nisan, N.: Bidding and allocation in combinatorial auctions. In: Proceedings of the Second ACM Conference on Electronic Commerce. (2000) 1–12
18. Ledyard, J.O.: Incentive compatible space station pricing. American Economic Review **76** (1987) 274–279
19. Kelly, T.: Utility-directed allocation. In: First Workshop on Algorithms and Architectures for Self-Managing Systems. (2003) `http://tesla.hpl.hp.com/self-manage03/`. Also available as HP Labs tech report HPL-2003-115.
20. Kellerer, H., Pferschy, U., Pisinger, D.: Knapsack Problems. Springer (2004) ISBN 3-540-40286-1.
21. Tennenholtz, M.: Some tractable combinatorial auctions. In: Proc. of the 17th National Conference on Artificial Intelligence. (2000) A later and longer version is Ref. [45].
22. Kothari, A., Parkes, D.C., Suri, S.: Approximately-strategyproof and tractable multi-unit auctions. In: ACM E-Commerce. (2003)
23. Wurman, P.R., Walsh, W.E., Wellman, M.P.: Flexible double auctions for electronic commerce: Theory and implementation. Decision Support Systems **24** (1998) 17–27
24. Lehmann, B., Lehmann, D., Nisan, N.: Combinatorial auctions with decreasing marginal utilities. In: ACM E-Commerce. (2001)
25. Walsh, W.: Personal communication (2003)
26. Sandholm, T., Suri, S., Gilpin, A., Levine, D.: Winner determination in combinatorial auction generalizations. In: AAMAS. (2002)
27. Pisinger, D.: A minimal algorithm for the multiple-choice knapsack problem. European Journal of Operations Research **83** (1995) 394–410

28. Holte, R.C.: Combinatorial auctions, knapsack problems, and hill-climbing search. In Stroulia, E., Matwin, S., eds.: Lec. Notes in CS. Volume 2056. Springer (2001)
29. Leyton-Brown, K.: Resource Allocation in Competitive Multiagent Systems. PhD thesis, Stanford University (2003)
30. Martello, S., Toth, P.: Knapsack Problems: Algorithms and Computer Implementation. John Wiley & Sons Ltd. (1990) ISBN 0-471-92420-2.
31. Sedgewick, R.: Algorithms in C. Addison-Wesley (1998) See page 215 of the 8th printing (August 2001) for a remarkably clear and compact integer knapsack solver in C.
32. Garey, M.R., Johnson, D.S.: Computers and Intractability: A Guide to the Theory of NP-Completeness. W.H. Freeman (1979) ISBN 0-7167-1045-5.
33. Papadimitriou, C.H., Steiglitz, K.: Combinatorial Optimization: Algorithms and Complexity. Second edn. Dover (1998) ISBN 0-486-40258-4.
34. Leyton-Brown, K., Pearson, M., Shoham, Y.: Towards a universal test suite for combinatorial auction algorithms. In: ACM E-Commerce. (2000)
35. Chvátal, V.: Hard knapsack problems. Operations Research **28** (1980) 1402–1411
36. Pisinger, D.: Algorithms for Knapsack Problems. PhD thesis, University of Copenhagen (1995) `http://www.diku.dk/users/pisinger/95-1.pdf`.
37. Martello, S., Pisinger, D., Toth, P.: New trends in exact algorithms for the 0-1 knapsack problem. European Journal of Operational Research **123** (2000) 325–332
38. Pisinger, D.: A fast algorithm for strongly correlated knapsack problems. Discrete Applied Mathematics **89** (1998) 197–212
39. Klemperer, P.: Auction theory: A guide to the literature. Journal of Economic Surveys **13** (1999) 227–260
40. Krishna, V.: Auction Theory. Academic Press (2002) ISBN 0-12-426297-X.
41. Sandholm, T., Suri, S.: Market clearability. In: IJCAI. (2001)
42. Sandholm, T.: Algorithm for optimal winner determination in combinatorial auctions. Artificial Intelligence **135** (2002) 1–54
43. Bassamboo, A., Gupta, M., Juneja, S.: Efficient winner-determination techniques for Internet multi-unit auctions. In: Proceedings of the First IFIP Conference on E-Commerce, E-Business, and E-Government. Volume 202. (2001)
44. Ahuja, R.K., Magnanti, T.L., Orlin, J.B.: Network Flows. Prentice Hall (1993) ISBN 0-13-617549-X.
45. Tennenholtz, M.: Tractable combinatorial auctions and b-matching. Artificial Intelligence **140** (2002) 231–243
46. Ghebreamiak, K.A., Andersson, A.: Caching in multi-unit combinatorial auctions. In: AAMAS. (2002)

Designing Auctions for Deliberative Agents

Kate Larson and Tuomas Sandholm

Computer Science Department, Carnegie Mellon University,
Pittsburgh, PA 15213
{klarson, sandholm}@cs.cmu.edu

Abstract. In many settings, bidding agents for auctions do not know their preferences *a priori*. Instead, they must actively determine them through deliberation (e.g., information processing or information gathering). Agents are faced not only with the problem of deciding how to reveal their preferences to the mechanism but also how to deliberate in order to determine their preferences. For such settings, we have introduced the deliberation equilibrium as the game-theoretic solution concept where the agents' deliberation actions are modeled as part of their strategies. In this paper, we lay out mechanism design principles for such deliberative agents.

We also derive the first impossibility results for such settings - specifically for private-value auctions where the agents' utility functions are quasilinear, but the agents can only determine their valuations through deliberation. We propose a set of intuitive properties which are desirable in mechanisms used among deliberative agents. First, mechanisms should be non-deliberative: the mechanism should not be solving the deliberation problems for the agents. Secondly, mechanisms should be deliberation-proof: agents should not deliberate on others' valuations in equilibrium. Third, the mechanism should be non-deceiving: agents do not strategically misrepresent. Finally, the mechanism should be sensitive: the agents' actions should affect the outcome. We show that no direct-revelation mechanism satisfies these four properties. Moving beyond direct-revelation mechanisms, we show that no value-based mechanism (that is, mechanism where the agents are only asked to report valuations - either partially or fully determined ones) satisfies these four properties.

1 Introduction

Game theory, and mechanism design in particular, have long been successfully used in economics and have recently drawn a lot of research interest from computer scientists (e.g., [7] [8]). In most of this work it is assumed that the participants, or agents, know their preferences and the goal of the mechanism is to extract this information to a sufficient extent, and select an outcome such that desirable properties are achieved. However, there are many settings where agents do not know their preferences *a priori*. Instead they may, for example, have to solve computationally complex optimization problems, query databases, or perform complicated searches in order to determine the worth of an outcome. We call the actions taken to determine preferences *deliberation*.

If there are no restrictions placed on the deliberation capabilities of agents, they could optimally determine their preferences and act as fully rational agents. However,

P. Faratin and J.A. Rodríguez-Aguilar (Eds.): AMEC 2004, LNAI 3435, pp. 87–100, 2005.

in many settings there are costs associated with deliberation. Agents are not able to optimally determine their preferences, but instead must trade off quality of valuations against deliberation cost. Decision making under costly deliberation resources is challenging even in single-agent settings. Having to interact with other agents complicates the problem further. Agents must take into account the other agents' actions in determining both how to act in the mechanism and also how to use deliberation resources.

We have proposed explicitly including the deliberation actions of agents into their strategies, and then analyzing games for *deliberation equilibria* which are fixed points in the space of strategy profiles from this enlarged strategy space [4] [5] . Using this approach, we have studied common auction mechanisms such as the first-price auction, Vickrey auction, ascending auction, descending auction, and generalized Vickrey auction. We discovered the existence of interesting strategic behavior. In each auction mechanism studied, there existed instances where, in equilibrium, agents would use their deliberation resources to determine *other agents' valuations* of the item(s) being auctioned. We coined this phenomenon *strategic deliberation*.

In this paper, we build on this body of work. Instead of looking at the properties of auctions that were designed for fully rational agents, we ask the question: "Is it possible to *design* auctions that have desirable properties for such deliberative agents?" We propose a set of weak, intuitive properties that are desirable for auctions designed for such agents. In particular, we propose that auctions should not solve the valuation problems for the agents, that strategic deliberation should not occur in equilibrium, that agents should not have incentive to misreport, and that the agents' actions affect the outcome. We show that no direct-revelation mechanism satisfies these four properties. Moving beyond direct-revelation mechanisms, we show that no value-based mechanism (that is, mechanism where the agents are only asked to report valuations - either partially or fully determined ones) satisfies these four properties.

The rest of the paper is as follows. We first provide an example application where our approach is needed (Section 2). We then given an overview of pertinent mechanism design concepts, and describe the model for computationally-limited agents (Sections 3 and 4). We show that there is a parallel to the revelation principle for our setting, but argue that the direct mechanism produced has highly impractical properties. We then propose a set of auction properties which we believe are important when the auction is to be used among deliberative agents. Our main results show that it is impossible to design a direct-revelation mechanism that satisfies those desirable properties, and furthermore, it is impossible to design any value-based mechanism that satisfies them (Section 5).

2 An Example Application

To make the presentation more concrete, we now discuss an example domain where our methods are needed.

Consider the example presented in Figure 1. An agent is trying to determine how much it values a specific product. In order to help determine its value, the agent queries a product review database to gather information about the product. Each query costs some fixed an amount, so an agent is faced with the decision of how much information to gather given the cost to acquire it. It may also be in the agent's best interest to use some of its deliberation resources (i.e. money to pay for queries) to (partially) determine

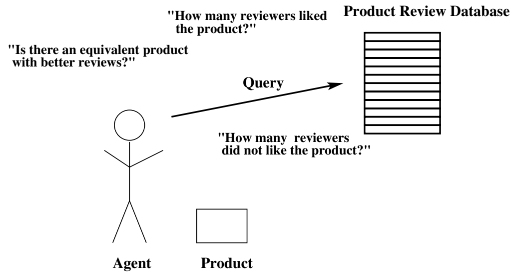


Fig. 1. Agents may need to gather information in order to determine their value for a certain item. In this figure, an agent gathers information about a product by querying a product review database. Each query returns an answer which is used by the agent to update its value for the product. However, each query costs some fixed amount, and so the agent must decide how much information it needs about the product, given the cost of gathering it.

the values that the other agents in the auction have for the product. By doing some initial deliberating on a competitor's problem, an agent can gather information that may be useful in formulating its own deliberating and bidding strategies.

3 Mechanism Design for Rational Agents

In this section we present an overview of pertinent mechanism design concepts. We assume that the reader has a basic background in game theory. The mechanism design problem is to implement an optimal system-wide solution to a decentralized optimization problem with self-interested agents with private information about their preferences for different outcomes. One of the most exciting applications of mechanism design has been in the area of auction design.

We assume that there is a set of agents, I, $|I| = n$. Each agent, i, has a *type*, $\theta_i \in \Theta_i$, which represents the private information of the agent that is relevant to the agent's decision making. In particular, an agent's type determines its preferences over different outcomes. We use the notation $u_i(o, \theta_i)$ to denote the utility of agent i with type θ_i for outcome $o \in \mathcal{O}$ ($\mathcal{O}$ is the space of possible outcomes). As mentioned in the first paragraph, the goal of mechanism design is to implement some system-wide solution. This is defined in terms of a *social choice function*.

Definition 1 (Social Choice Function). *A* social choice function *is a function $f : \Theta_1 \times \ldots \times \Theta_n \mapsto \mathcal{O}$, such that, for each possible profile of agents' types $\theta = (\theta_1, \ldots, \theta_n)$ assigns an outcome $f(\theta) \in \mathcal{O}$.*

The mechanism design problem is to implement a set of "rules" so that the solution to the social choice function is implemented despite agents' acting in their own self-interest.

Definition 2 (Mechanism). *A mechanism $M = (S_1, \ldots, S_n, g(\cdot))$ defines the set of strategies S_i available to each agent and an outcome rule $g : S_1 \times \ldots \times S_n :\mapsto \mathcal{O}$, such that $g(s)$ is the outcome implemented by the mechanism for strategy profile $s = (s_1, \ldots, s_n)$.*

A mechanism *implements* a social choice function $f(\cdot)$ if there is an equilibrium of the game induced by the mechanism which results in the same outcomes as $f(\cdot)$ for every profile of types, θ.

Definition 3 (Implementation). *A mechanism* $M = (S_1, \ldots, S_n, g(\cdot))$ implements *social choice function* $f(\cdot)$ *if there is an equilibrium strategy profile* $s^* = (s_1^*, \ldots, s_n^*)$ *such that* $g(s^*(\theta)) = f(\theta)$ *for all* θ.

An important class of mechanisms are *direct revelation mechanisms*.

Definition 4 (Direct revelation mechanism). *A* direct revelation mechanism *is a mechanism in which* $s_i = \Theta_i$ *for all* i *and has outcomes rule* $g(\hat{\theta})$ *based on reported types* $\hat{\theta} = (\hat{\theta_1}, \ldots, \hat{\theta_n})$.

One of the most important results in mechanism design is the *Revelation Principle*. It states that any mechanism can be transformed into an equivalent direct mechanism where, in equilibrium, all agents truthfully reveal their types (that is, the mechanism is *incentive compatible*).

Theorem 1 (Revelation Principle). *Suppose there exists a mechanism* M *that implements the social choice function* $f(\cdot)$ *in dominant strategies (Bayesian-Nash equilibrium). Then* $f(\cdot)$ *is truthfully implementable in a dominant strategies (Bayesian-Nash) incentive compatible direct-revelation mechanism.*

The Revelation Principle suggests that mechanism designers need only be concerned with direct-revelation mechanisms. Later in the paper we will discuss the Revelation Principle in more detail.

In this paper we restrict ourselves to settings where agents have *quasilinear preferences*. That is, the utility function of an agent i has the form $u_i(o, \theta_i) = v_i(x, \theta_i) + p_i$, where outcome o defines an allocation x and a transfer p_i for the agent.

In general, mechanisms for quasilinear preferences take a certain form.

Definition 5 (Mechanisms for quasilinear environments). *A mechanism for quasilinear environments is a mechanism* $M = (S_1, \ldots, S_n, (k(\cdot), t_1(\cdot), \ldots, t_n(\cdot)))$ *such that the outcome function* $g(\cdot) = (k(\cdot), p_1(\cdot), \ldots, p_n(\cdot))$ *where* $k : S_1 \times \ldots \times S_n :\mapsto \mathcal{K}$ *is a choice rule which selects some choice from choice set* $\mathcal{K}$*, and transfer rules* $p_i : S_1 \times \ldots \times S_n :\mapsto \mathbb{R}$*. one for each agent, compute the payment* $t_i(s)$ *made by agent* i.

In addition to quasilinear environments, we also assume that agents' have private values. This means that an agent's utility depends only on its own type. Many ecommerce applications are set in the quasilinear private-value environment. For example, many auctions belong to this set of mechanisms. The choice rule $k(\cdot)$ specifies which agents are allocated which items, and the transfers, $t_i(\cdot)$, specify the amount each agent must pay.

4 Computationally-Limited Agents

In this section we introduce our model of bounded rationality, in the form of computationally-limited agents. We describe a computationally-limited agent and then explain how we incorporate these agents into a game-theoretic framework.

4.1 A Model of a Computationally-Limited Agent

We define a computationally-limited agent to be any agent who does not know its valuations for items (i.e. its preferences) *a priori* but instead must use its computational resources in order to determine them. We define a computationally-limited agent by its computing resources and the tools that it has to effectively use them.

We assume that a computationally-limited agent has some set of computing resources T_i. While these resources may take on many forms, for the sake of expository ease we will use *computing time* as the canonical example of a resource. An agent is able to apply its computing resources on any problem j in which it is interested in. If there are m possible problems that an agent may compute on, then we let $(t_1, \ldots, t_m) \in T_i^m$ denote that agent i has allocated t_j resources to problem j.

Agents do not have infinite computing resources. We model this limitation by a *cost function*, $\mathrm{cost}_i : T_i^m \mapsto \{x | x \in R \text{ and } x \geq 0\}$. The only restrictions on the cost function of an agent is that it is non-decreasing and additive. That is, given vectors $t = (t_1, \ldots, t_m)$ and $t' = (t'_1, \ldots, t'_m)$, then $\mathrm{cost}_i(t + t') = \mathrm{cost}_i(t) + \mathrm{cost}_i(t')$ and if $t \leq t'$ then $\mathrm{cost}_i(t) \leq \mathrm{cost}_i(t')$.

We assume that computationally-limited agents are equipped with a set of algorithms $\mathcal{A}_i = \{A_i^j\}$ where A_i^j is the algorithm that agent i can use for problem j.[1] These algorithms are processes which the agent runs in order to determine its valuations or preferences. In particular, we assume that all the algorithms have the *anytime property*; they can be stopped at any point in time and will return a solution, and if given additional computational resources, the algorithm will return a better solution. This allows agents to make an explicit tradeoff between the solution quality and the cost to obtain that solution. Many algorithms have the anytime property. For example, most iterative refinement algorithms are anytime since they always return a solution, and improve it if allowed to run longer. Similarly, many search and information gathering applications can be viewed as anytime algorithms. As more information about an item is obtained, the knowledge about its true valuation improves.

While anytime algorithms are models that allow for the trading off of computing resources (for example, computing time) for solution quality, they do not provide a complete solution for agents as they do not specify how this tradeoff should be made. Instead, anytime algorithms are paired with a meta-level deliberation-control procedure the aids in determining how long to run an algorithm, and when to stop computing and act with the solution obtained. There are two components to the procedure; the *performance profile* which describes how computing affects the output of the algorithm, and a process for using the information in the performance profile to make decisions about how much resources to allocate to a problem. For the rest of the paper we will use the term performance profile to refer to both the descriptive and procedural aspects. We use the notation PP_i^j to refer to the performance profile for anytime algorithm A_i^j and let $\mathcal{PP}_i = \{PP_i^j\}$. We will additionally assume that the agents have fully normative performance profiles which allow them to make online decisions about whether to continue computing on a certain problem, whether to stop computing, or whether to switch

[1] In general, an agent can use the same algorithm on multiple problems, or can have multiple algorithms for the same problem.

to computing on a different problem. Such deliberation control procedures do exist. An example is the performance profile tree [4].

To summarize, a computationally-limited agent i is defined as

$$\langle T_i, \mathrm{cost}_i(\cdot), \mathcal{A}_i, \mathcal{PP}_i \rangle.$$

As mentioned at the start of this section, computationally-limited agents do not know their valuations *a priori*. Instead they must compute or gather information to determine them. We assume that the valuation function of an agent ($v(\cdot, \cdot)$) is determined by the amount of resources allocated ($t = (t_1, \ldots, t_m)$) and the allocation (x). That is, the valuation function of agent i is $v_i(t, x)$. The utility of the agent depends on the agent's valuation function, the payment specified by the auction mechanism, and the cost the agent has incurred via computing. That is

$$u_i(t, (x, p)) = v_i(t, x) - p_i - \mathrm{cost}_i(t).$$

For example, in a single item auction, if an agent i has allocated computing resources $(t_1, \ldots, t_m)$ where t_i is the amount of resources allocated to its own valuation problem, then

$$u_i(t, (x, p)) = \begin{cases} v_i(t_i) - p_i - \mathrm{cost}_i(t) & \text{if } i \text{ is allocated the item and has to pay price } p_i \\ -\mathrm{cost}_i(t) & \text{if } i \text{ is not allocated the item} \end{cases}$$

4.2 Strategic Behavior of Deliberative Agents

Effectively using available computing resources in single-agent settings is difficult enough. Having to interact with other agents complicates the problem further. Agents must take into account the other agents' actions in determining both how to act in the auction and also how to compute.

Let C_i be the set of computing actions for agent i. A computing action $c_j^i \in C_i$ is the act of agent i allocating one step of computation on problem j, where a step is

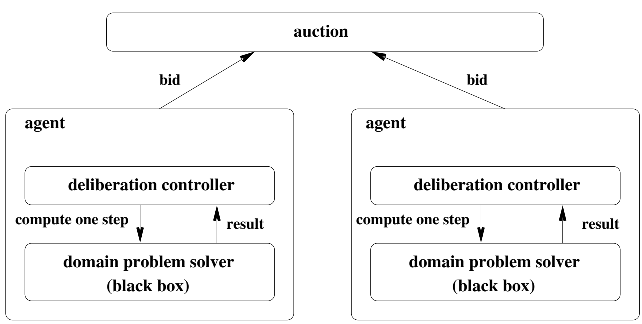


Fig. 2. An auction with two computationally-limited agents. In order to submit a reasonable bid, each agent needs to first (approximately) compute its valuations for the item that is up for auction.

defined exogenously. The action of *not* taking a computing step is also included in this set and is denoted by $\emptyset_C$. The vector $(t_1, \ldots, t_m) \in T_i^m$, introduced in the previous subsection, corresponds to agent i taking t_1 steps on problem 1, etc. As agents compute they change their knowledge about how future computing will change the solutions of the problems. This information comes from their set of performance profiles, given their current *state of deliberation*. We define a state of deliberation at $t = (t_1, \ldots, t_m)$ to be

$$\phi_i(t) = \langle n_1(t_1), \ldots, n_m(t_m) \rangle$$

where $n_j(t_j)$ stores the current valuation for problem j after t_j computing steps were taken on the problem, as well as the path followed to reach the valuation and any other features deemed to be of importance for the decision making of the agent.[2]

We define the set A_i to be the set of non-deliberative actions that an agent i can take. The set is defined by the auction. For example, in a sealed-bid auction, the set A_i is simply the set of bids that the agent may submit to the auctioneer, while in an ascending auction the set A_i is the set of messages that an agent can send to the mechanism whenever the price increases (i.e. $A_i = \{\text{in}, \text{out}\}$). We denote the action of *not* taking a non-computing action as $\emptyset_A \in A_i$.

A strategy for a computationally-limited agent is a plan which specifies which actions to execute (computing and other) at every point in the game. A *history* at time point r , $H(r) \in \mathcal{H}(r)$, when it is the turn of agent i to take an action, is defined to be a tuple consisting of the current state of deliberation of the agent, all non-computing actions the agent has taken, as well as all actions it has observed the other agents take. It is now possible to define a strategy.

Definition 6 (Strategy). *A strategy for a computationally limited agent i is*

$$S_i = (\sigma_i^r)_{r=0}^{\infty}$$

where

$$\sigma_i^r : \mathcal{H}_i(r) \mapsto C_i \times A_i.$$

To make this definition more clear, we present an example. A strategy $S_i = (\sigma_i^r)_{r=0}^{\infty}$ for a computationally-limited agent participating in a Vickrey auction, where bids are collected at time point R (and it is assumed that all computing must stop also) is defined in the following way:

$$\sigma_i^r(H(r)) = \begin{cases} (c^j, \emptyset_A) & \text{when } r < R \\ (c^j, b_i)\ b_i \in \mathbb{R} & \text{when } r = R \\ (\emptyset_C, \emptyset_A) & \text{when } r > R \end{cases}$$

That is, before the auction the agent can take any computing action it wishes, at the auction bid collection time the agent submits a bid, and then after the agent no longer takes any actions.[3]

[2] In the performance profile tree, for example, path information and feature information at automatically stored in the structure.

[3] In some situations an agent may be permitted to compute after the auction has closes.

In this new, enlarged, strategy space we look for equilibria. We call these equilibria *deliberation equilibria* [4]. We have noted in previous work that a new type of strategic behavior arises when agents are computationally-limited [5]. We have coined this new behavior *strategic deliberation*.

Definition 7 (Strategic Deliberation). *If an agent i uses any of its computing resources to determine another agent's valuation, then agent i is* strategically deliberating.

In earlier work we showed that the first-price sealed-bid auction, the ascending auction, the Vickrey auction and the generalized Vickrey auction all produce equilibria where agents have incentive to strategically deliberate [5].

In the rest of the paper we make several assumptions, some of which have already been described. First, we assume that agents have quasi-linear utilities. Second, we assume that agents have private values. Finally, we assume that the agent definitions are common knowledge. That is, we assume that the performance profiles and cost functions of the agents are common knowledge. We do not assume that agents are able to observe which computing actions other agents are taking during a game.

5 Designing Auctions for Deliberative Agents

In this section we study the problem of designing auctions specifically for computationally-limited agents. We first argue that directly porting mechanism design concepts for rational agents is not always desirable as they overlook the computational limitations which define the agents. We then propose a set of properties which we believe auctions for computationally-limited agents should exhibit. We show that most "interesting" auctions fail to have all these properties.

5.1 A Revelation Principle

In settings where the bidding agents are fully rational, the revelation principle states that given any mechanism (auction) that implements some social choice function, it is possible to construct a second incentive-compatible direct-revelation mechanism (auction) which implements the same social choice function. That is, it is possible to construct useful auctions where the dominant strategies of the agents is to truthfully reveal their valuations (types, θ_i) to the auctioneer.

If agents are computationally-limited, then they do not know their valuations, without first exerting some computational effort. Agents are provided with tools (anytime algorithms and performance profiles) which they can use to determine their valuations. We can define the *type* of agent i to be the set of all computing tools that the agent has, along with the current problem instance being computed on. That is,

$$\theta_i = [\langle T_i, \mathrm{cost}_i(\cdot), \mathcal{A}_i, \mathcal{PP}_i\rangle, \mathrm{insts}]$$

where insts is the set of problem instances.[4]

[4] For example, a problem instance in a traveling-salesman domain is the set of delivery jobs in a traveling-salesman application.

Using this definition of a *type* it is possible to formulate a revelation principle for computationally-limited agents.

Theorem 2. *Suppose there exists a mechanism* $M = (S_1, \ldots, S_I, g())$ *that implements the social choice function* $f(\cdot)$ *in dominant strategies for deliberative agents. Then* $f(\cdot)$ *is truthfully implementable in dominant strategies for computationally-limited agents.*

Proof. (**Sketch.**) The proof follows a similar argument to the proof of the revelation principle for rational agents. Given any mechanism, where in equilibrium an agent i plays strategy S_i, it is possible to construct another mechanism, where, when agent i announces type θ_i strategy s_i is executed. Agent i has incentive to truthfully announce its type in equilibrium. □

While this theorem at first appears to suggest that the tools of mechanism-design for rational agents can be directly used in settings where agents are computationally-limited, at second glance the reader will notice that some rather disturbing assumptions have been made. First, the proof relies on agents being able to submit *all* their computing tools and information about the problems they wish to compute on. In practice this is highly infeasible. Second, it assumes that the mechanism center is capable of determining all relevant valuation information from the information given to it. Again, it is unrealistic to assume that the mechanism center will have enough computing resources of its own to actually do this.

5.2 Properties of the Auctions

We believe that auctions for computationally-limited agents should have good deliberative properties in addition to good economic properties. In this section we propose a set of properties which we believe are desirable for computationally-limited agents.

In the previous section we showed that an obvious formulation of a revelation principle simply moves the computing problems of the agents to the auctioneer. We argue that this is not a desirable property since agents should be responsible for solving their own valuation problems. Second, we believe that it is unreasonable to assume that agents are capable of providing detailed enough information to the auctioneer so that the auctioneer could properly solve the problems of the agents. Finally, in many settings the auctioneer is unlikely to have adequate computing resources of its own to solve the computational problems of the bidding agents as well as determine the optimal outcome once all the bidders' computing problems have been solved. Therefore, we propose that auctions should be *non-deliberative*.

Property 1 (Non-Deliberative). An auction is *non-deliberative* if the auctioneer does not solve the agents' individual deliberation problems.

The second property we wish to have is for agents to have no incentive to strategically deliberate (Definition 7). Recall that most common auctions are susceptible to strategic deliberation. We believe that strategic deliberation is an undesirable property for several reasons. First, the use of agents' limited computing resources on problems which do not directly lead to improved valuations seems like a waste of resources which could have been directed towards improving their own valuations, and possibly the social welfare of the entire market. Secondly, effectively using computing resources to determine valuations is a challenging problem for an agent even when it is done in isolation. Having agents being concerned about computing on the valuation problems of competitors seems to place a too high strategic load on the bidding agents. Thus, we believe auctions for computationally-limited agents should be *deliberation-proof.*

Property 2 (Deliberation-Proof). An auction is *deliberation-proof* if, in equilibrium, no agent has incentive to strategically deliberate.

Finally, we believe that auctions should be designed in such a way so that the bidding agents have no incentive to further strategize. That is, agents should have incentive to not misrepresent what problems they have computed on, and what solutions they have obtained.

Property 3 (Non-Deceiving). Assume that v_i is the true (expected) computed value of agent i. A auction is non-deceiving if the agent never has incentive to send a report to the auctioneer such that if any other agent had observed the report, their belief that agent i's value is v_i would be 0.

A non-deceiving auction does not require that an agent directly reveal its computed valuations. It just ensures that an agent does not lead all participants to believe that a valuation it has obtained is not possible. For example, assume that an agent has secretly computed a value v. A mechanism would be deceiving if the agent had incentive to report that its value was strictly greater than v. The mechanism would not be deceiving if the agent had incentive to report that its value was greater than some w, $w < v$.

5.3 Results

In this section we present our results concerning auctions for computationally-limited agents. We will insist that the auctions must be non-deliberative, and so will focus our attention on *value-based* auctions, where agents do not report information about the algorithms or performance profiles being used.

Definition 8 (Value-based Auction). *A value-based auction,* $M = (S_1, \ldots, S_n, x(), t_1(), \ldots, t_n())$*, is a mechanism where the strategies of each agent are restricted so that they are functions only of (partially) determined valuations. Agents do not reveal other information about performance profiles, algorithms, cost functions or problem instances.*

Value-based auctions are non-deliberative. The auctioneer is not given any of the tools required for it to actively compute on agents' problems. Examples of value-based auctions include sealed-bid auctions where agents submit numerical bids, and ascending and posted-price auctions where agents answer yes or no to the query of whether they would be willing to buy an item at a specified price.

It is trivially easy to design value-based auctions which are deliberation-proof and non-deceiving.

Note 1. There exist value-based auctions which are both deliberation-proof and non-deceiving.

Unfortunately, the obvious candidates which satisfy both of these properties are undesirable for other reasons. For example, any auction which randomly determines a winner while ignoring the bids, is both deliberation-proof and non-deceiving. Similarly, any dictatorial auction is also deliberation-proof and non-deceiving.

We place an additional restriction on the auctions for computationally-limited agents. We insist that they must be *sensitive*.

Definition 9 (Sensitive). *An auction is* sensitive *to agents' strategies if the outcome, $o() = (x(), p())$ of the auction depends on the agents' strategies. That is, there exists some agent i and strategies s_i', s_i'', $s_i' \neq s_i''$ such that for strategy profiles $s' = (s_1, \ldots, s_{i-1}, s_i', s_{i+1}, \ldots, s_I)$ and $s'' = (s_1 \ldots, s_{i-1}, s_i'', s_{i+1}, \ldots, s_I)$,*

$$x(s') \neq x(s'') \text{ and } p(s') \neq p(s'')$$

In the rest of this paper we focus our attention to sensitive, value-based mechanisms. We start by studying sealed-bid auctions to see how computationally-limited agents behave in equilibrium. The first result states that it is impossible to design a sealed-bid auction which is also deliberation-proof.

Theorem 3. *There exists no value-based sensitive direct auction that is deliberation-proof across all instances (where an instance is defined by the agents and the current valuation problems).*

Proof. Due to space constraints we only provide an intuition of the proof. Given any allocation and payment rules of a sealed-bid auction, it is possible to construct performance profiles and cost functions such that one agent is best off incurring a small cost by computing on a competitor's valuation problem, and thus gathering information as to the likelihood of it being in the final allocation, before deciding whether to compute at a cost on its own valuation problems. □

To get rid of strategic deliberation, computationally-limited agents must be provided with enough information by the auction so that they can determine whether to devote computing resources to their own problems or not. Many multi-stage auctions reveal information. This information can be used by the agents to help determine which are the best computing and non-computing actions to take. For example, in a single item ascending auction, as the price rises the auction may reveal the number of agents remaining in the auction at a given price. The agents can use this information to deduce useful valuation information about their competitors.

However, value-based multi-stage auctions still do not result in auctions with all our proposed properties. In particular, for any multi-stage auction, there exist instances, defined by the agents' performance profiles and cost functions, where strategic deliberation occurs in equilibrium, or where agents will try to deceive their competitors by taking actions which lead their competitors to believe that they have better valuations than they really do.

Theorem 4. *There exists no sensitive value-based auction that is deliberation-proof and non-deceiving across all problem instances. (An instance is defined by agents' performance profiles, cost functions).*

Proof. Due to space limitations we provide only an overview of the proof. The proof is constructive. Starting with a direct revelation mechanism (auction) which implements some social choice function $f(\cdot)$, it is possible to find performance profiles and cost functions such that strategic deliberation occurs in equilibrium. Now take any multi-stage mechanism which implements the social choice function $f(\cdot)$. It is still advantageous for at least one agent to learn information about the values another agent has. If this information is not revealed at the right time through the auction, then strategic deliberation will still occur. If, by the rules of the auction, the actions of an agent reveal information about its possible values, then due to the structure of the performance profiles and cost functions, it is possible to show that one agent will misreport its value information in order to force the second agent (ie. the agent who would have strategic deliberated in a direct mechanism) to believe that it can not win the auction. That is, deceiving occurs. □

To summarize, we have proposed that an auction for computationally-limited agents should be non-deliberative, deliberation-proof, and non-deceiving. While these properties appear to be intuitive and reasonable, we have shown that it is not possible to design interesting auctions for deliberative agents which exhibit all three properties.

6 Related Research

Both the the game theory and the computer science research communities are interested in mechanism design issues where computation and information gathering are issues. In this section we review the literature which is most closely related to the work presented in this paper.

From the computer science research community there has been work on both bounded-rational bidding agents and mechanisms. Sandholm noted that under a model of costly computing, the dominant strategy property of Vickrey auctions fails to hold, even with simple agent models [12], while Parkes has looked at using auction design to simplify the meta-deliberation problems of the agents [9]. This earlier work, however, focused only on settings where agents were concerned only about computing on their own value - ignoring the possibility of using computational resources to gather information on competing bidders. There has also been recent work on computationally limited mechanisms. In particular, research has focused on the generalized Vickrey auction and investigates ways of introducing approximate algorithms to compute outcomes without loosing the incentive compatibility property [7,3,6]. These methods still require that the bidding agents compute and submit their valuations.

In the economics and game theory literature there has been some recent work on information acquisition and mechanism design. This work has mainly focused on studying the incentives to acquire information at different times in different auction mechanisms and usually assumes that an agent can gather information only on its value problem [10,1,2]. Rasmusen's work is the most similar to ours [11]. It assumes that agents do not know their valuations but must invest to learn them and are also able to invest in competitors value problems, however his focus is on understanding behavior such as sniping that is commonly seen in different online auctions like eBay, and he does not discuss the possibility of designing auctions for information gathering settings.

7 Conclusions and Future Research

In many settings, agents participating in mechanisms do not know their preferences (valuations) *a priori*. Instead, they must actively determine them through deliberation (e.g., information processing or information gathering). Agents are faced not only with the problem of deciding how to reveal their preferences to the mechanism but also how to deliberate in order to determine their preferences. For such settings, we have introduced the *deliberation equilibrium* as the game-theoretic solution concept where the agents' deliberation actions are modeled as part of their strategies.

In this paper, we laid out mechanism design principles for such deliberative agents. We first showed that the revelation principle applies to such settings in a trivial sense by having the mechanism carry out all the deliberation for the agents. This is impractical, and we propose that mechanisms should be *non-deliberative*: the mechanism should not be solving the deliberation problems for the agents. Second, mechanisms should be *deliberation-proof*: agents should not deliberate on others' valuations in equilibrium. Third, the mechanism should be *non-deceiving*: agents do not strategically misrepresent. Finally, the mechanism should be *sensitive*: the agents' actions should affect the outcome. We showed that no direct-revelation mechanism satisfies these four intuitively desirable weak properties. This is the first impossibility result in mechanism design for deliberative agents. Moving beyond direct-revelation mechanisms, we showed that no *value-based* mechanism (that is, mechanism where the agents are only asked to report valuations - either partially or fully determined ones) satisfies these four properties.

This result is negative. It states that either we must have mechanisms which do the deliberating for the agents, or complex strategic (and costly) counterspeculation can occur in equilibrium. However, there is some hope. It may be possible to weaken one of the properties slightly, while still achieving the others. For example, it may be possible to design multi-stage mechanisms that are not value based; the mechanism could help each agent decide when to hold off on deliberating during the auction (and when to deliberate on one's own valuation on different bundles of items in a combinatorial auction). In another direction, by relaxing strategic deliberation and compensating agents appropriately, it may be possible to design mechanisms where agents who can deliberate cheaply and efficiently deliberate for all agents. These are areas which we plan to pursue in future work.

Acknowledgements

The material in this paper is based upon work supported by the National Science Foundation under CAREER Award IRI-9703122, Grant IIS-9800994, ITR IIS-0081246, and ITR IIS-0121678.

References

1. Olivier Compte and Philippe Jehiel. On the virtues of the ascending price auction: New insights in the private value setting. working paper, December 2000.
2. Olivier Compte and Philippe Jehiel. Auctions and information acquisition: Sealed-bid or dynamic formats?, 2001. Working Paper.

3. Noa Kfir-Dahav, Dov Monderer, and Moshe Tennenholtz. Mechanism design for resource bounded agents. In *Proceedings of the Fourth International Conference on Multi-Agent Systems (ICMAS)*, pages 309–315, Boston, MA, 2000.
4. Kate Larson and Tuomas Sandholm. Bargaining with limited computation: Deliberation equilibrium. *Artificial Intelligence*, 132(2):183–217, 2001. Short early version appeared in the Proceedings of the National Conference on Artificial Intelligence (AAAI), pp. 48–55, Austin, TX, 2000.
5. Kate Larson and Tuomas Sandholm. Costly valuation computation in auctions. In *Theoretical Aspects of Rationality and Knowledge (TARK VIII)*, pages 169–182, Siena, Italy, July 2001.
6. Daniel Lehmann, Lidian Ita O'Callaghan, and Yoav Shoham. Truth revelation in rapid, approximately efficient combinatorial auctions. *Journal of the ACM*, 49(5):577–602, 2002. Early version appeared in ACMEC-99.
7. Noam Nisan and Amir Ronen. Algorithmic mechanism design. *Games and Economic Behavior*, 35:166–196, 2001. Early version in STOC-99.
8. Christos Papadimitriou. Algorithms, games and the Internet. In *STOC*, pages 749–753, 2001.
9. David C. Parkes. Optimal auction design for agents with hard valuation problems. In *Agent-Mediated Electronic Commerce Workshop at the International Joint Conference on Artificial Intelligence*, Stockholm, Sweden, 1999.
10. Nicola Perisco. Information acquisition in auctions. *Econometrica*, 68(1):135–148, January 2000.
11. Eric Rasmusen. Strategic implications of uncertainty over one's own private value in auctions. working paper, January 2003.
12. Tuomas Sandholm. Issues in computational Vickrey auctions. *International Journal of Electronic Commerce*, 4(3):107–129, 2000. Special Issue on Applying Intelligent Agents for Electronic Commerce. A short, early version appeared at the Second International Conference on Multi–Agent Systems (ICMAS), pages 299–306, 1996.

An Evolutionary Game-Theoretic Comparison of Two Double-Auction Market Designs

Steve Phelps[1], Simon Parsons[2], and Peter McBurney[1]

[1] Department of Computer Science, University of Liverpool,
Liverpool L69 7ZF UK
{sphelps, p.j.mcburney}@csc.liv.ac.uk
[2] Department of Computer and Information Science, Brooklyn College,
City University of New York, Brooklyn NY 11210 USA
parsons@sci.brooklyn.cuny.edu

Abstract. In this paper we describe an analysis of two double auction markets—the clearing house auction and the continuous double auction. The complexity of these institutions is such that they defy analysis using traditional game-theoretic techniques, and so we use heuristic-strategy approximation to provide an approximated game-theoretic analysis. As well as finding heuristic-strategy equilibria for these mechanisms, we subject them to an evolutionary game-theoretic analysis which allows us to quantify which equilibria are more likely to occur. We then weight the design objectives for each mechanism according to the probability distribution over equilibria, which allows us to provide more realistic estimates for the efficiency of each mechanism.

1 Introduction

A double-auction mechanism is a generalization of an auction in which both buyers and sellers are allowed to exchange offers simultaneously. Since double-auctions allow dynamic pricing on both the supply side and the demand side of the marketplace, their study is of great importance, both to theoretical economists, and those seeking to implement real-world market places. On the one hand, economists who are interested in theories of price formation in idealized models of general markets have often turned to exchange-like models such as Walrasian tâtonnement, to describe and understand the price-formation process [2], and on the other, variants of the double-auction are used in large real-world exchanges to trade commodities in marketplaces where supply and demand fluctuate rapidly, such as markets for stocks, futures, and their derivatives [7].

However, the models of exchanges traditionally used by economists in general equilibrium theory are often simplified for the purposes of analytical tractability to such an extent that they are of scant relevance to the designers of real-world exchanges, and even, it is sometimes argued, of scant relevance to the theoretical modelling of markets. For example, one important simplification often made is that the number of agents participating in a market is very large; this simplification allows relative market power and consequent *strategic effects* to be ignored. However, in many real-world marketplaces, such as deregulated wholesale electricity markets, there may be relatively few competitors on one or both sides of the market. With small numbers of participants, general

P. Faratin and J.A. Rodríguez-Aguilar (Eds.): AMEC 2004, LNAI 3435, pp. 101–114, 2005.

equilibrium models break down because they fail to allow for market power, and the potential gains of strategic behavior, of participants.

An alternative approach is a sophisticated micro-theory of marketplaces called *auction theory*, in which the rational behavior of individual agents faced with different pricing institutions is analyzed using game theoretic techniques. Whereas neoclassical equilibrium theory often abstracts away from the details of individual agents, game-theoretic models allow economists to build sophisticated micro-models of individual agents' reasoning and preferences. In many scenarios, especially in analyzing single-sided monopoly markets, these models have been spectacularly successful to the extent where they have been directly applied to the design of real-world auctions for high-value government and corporate assets [9]. However, in other practical scenarios, especially when it comes to analyzing and designing double-sided markets, such as exchanges, there are still a number of problems with the theory, which we shall briefly review.

Auction-theorists typically analyze a proposed market institution by defining a set of design objectives, and then proceed to show that these design objectives are brought about when rational agents follow their best strategies according to a game-theoretic analysis. The typical design objectives considered by auction-theorists are:

Allocative efficiency: The outcome of using the mechanism should be optimal in some defined sense, for example, the total surplus generated should equal the available surplus in competetive equilibrium.
Budget balance: No outside subsidy inwards or transfers outwards are required for a deal to be reached.
Individual rationality: The expected net benefit to each participant from using the mechanism should be no less than the net benefit of any alternative.
Strategy-proofness: Participants should not be able to gain an advantage from non-truthtelling behavior.

In many applications, auction-theory demonstrates the existence of market mechanisms that satisfy all of these properties when agents follow rationally prescribed bidding strategies. However, the impossibility result of [13] demonstrates that no *double-sided* auction mechanism can be simultaneously efficient, budget-balanced and individually-rational. Moreover, many of the underpinnings of the theory assume that designers' interests are restricted to only the aforementioned properties. For example, the revelation principle states that, without loss of generality, we may safely restrict attention to mechanisms in which agents reveal their types truthfully. However, this result does not take into account the potential cost or other impracticalities of polling agents for their type information. Once minimizing the cost of revelation is introduced as a design objective, the revelation principle ceases to hold, because there may exist partial-revelation mechanisms with non-truthful equilibria which sacrifice strategy-proofness for expedience of revelation. This is of more than academic interest, since in real-world electronic exchanges it is rarely possible to poll *all* agents for their valuations before clearing the market; hence the *continuous* double-auction, in which we execute the clearing operation as new offers arrive, thus increasing transaction throughput at the expense of strategy-proofness.

In designing market places, as with any other engineering problem, we often need to make such tradeoffs between different objectives depending on the exact requirements and scenario at hand. We can often satisfactorily solve such multi-objective optimisation problems, provided that we have some kind of quantative assessment of each objective, yet classical auction-theory provides only a binary yes or no indication of whether each of its limited design objectives is achievable, making it extremely difficult to compare the different trade-offs.

Further complications arise when we attempt to use auction-theory to analyze existing ("legacy") market institutions. Exchanges such as the London Stock Exchange have been in existence far longer than game-theory and auction-theory, thus, unsurprisingly, the original rules of the institution were not necessarily based on sound game-theoretic or auction-theoretic principles. Moreover, it is unrealistic to expect that core financial institutions such as these radically alter their rules overnight in response to the latest fashionable developments in auction-theory or game-theory. Rather, it may be more salient to view financial institutions *evolving* gradually and incrementally in response to a changing environment [12]. Similarly, agents participating in these institutions do not necessarily instantaneously and simultaneously adjust their trading behavior to the theoretical optimum strategy; for example, adoption of a new trading strategy may spread through a population of traders as word of its efficacy diffuses in a manner akin to mimetic evolution.[1] Thus, we may think of the institutions we see today as the outcome of a *co-evolutionary* adaptation between financial institutions on the one hand, and trading strategies on the other.

The issue of legacy institutions has ramifications for auction-design; in these contexts the choice of adjustments to the auction rules may be tightly constrained by existing infrastructure, both physical and social, thus it may be necessary to examine the *attainability* of equilibria under the new design given existing strategic behavior in the legacy design. Classical auction theory relies on classical game-theory which in turn says nothing about the dynamics of adjustment to equilibrium.

For such applications, we need to turn to models of evolution and learning in strategic environments; models that we collectively categorize under the banner of *evolutionary game theory*. Models of learning and evolution as applied to agents' strategies are not new. Where our approach differs, however, is in the application of models of learning and evolution to the market mechanism itself, a new field we call *evolutionary mechanism design* [16,3].

In this paper, we extend our previous work on evolutionary mechanism design by describing a more sophisticated means of analyzing the performance of a mechanism. Previously we have either evolved trading strategies along with the mechanism [16], or used a single heuristic bidding strategy [17]. Here we use a mix of heuristic strategies, and describe a rigorous and fully automated way of evaluating a mechanism using this mixture. We start in Section 2 with a description of the mechanisms we are studying here, and, in Section 3 with use of several heuristic strategies. Then, in Section 4, we describe our experimental set up, and in Section 5 how we use these results to establish the evolutionary behavior of the markets. Section 6 gives results, and Section 7 analyses them before Section 8 describes the work that we will pursue next.

---

[1] The adoption by derivatives traders of the Black-Scholes equation for option pricing provides an example [10].

2 The Continuous Double Auction Verses the Clearing-House

In a typical exchange, the market institution attempts to match offers to buy with offers to sell in such a way that the overall surplus extracted from the market is maximized. If offers are considered as signals of agents' valuations for a resource, and assuming agents signal truthfully, then an auctioneer can maximize allocative efficiency by matching the highest buy offers with the lowest sell offers. In this paper we compare two types of exchange:

- a market in which trades are executed as new offers arrive, and
- a market in which we wait for all traders to place offers before clearing the market.

Following the terminology of [6], we refer to the former as the continuous double-auction (CDA) and the to the latter as the clearing-house (CH).

On casual inspection of the CDA, we might expect that it is designed according to the revelation principle, and so should maximize allocative efficiency when agents signal truthfully. Surprisingly, however, it turns out that surplus extraction in a CDA is extremely *poor* under direct revelation—typical values are approximately 80 percent, which is extremely low compared with outcomes of almost 100 percent which are observed with the non-truthful strategies that are actually adopted by real traders.

The reason for this poor efficiency is easy to spot; the continuous clearing rule results in myopic matching; when the clearing operation is performed the auctioneer has only a partial view of the aggregate supply and demand in the market place. In order to maintain a high throughput of actual transactions, the auctioneer impatiently clears the market before every trader has the opportunity to place their bid. The extremely surprising thing about this institution, however, is that rational agents acting locally to maximize their own profit are able to compensate for this efficiency loss by placing extra-marginal, non-truthful bids, which collectively result in high-efficiency outcomes.

Much analysis of the CDA has focused on showing that although the CDA is not a direct revelation mechanism (DRM), it can be considered an almost-DRM by virtue of the fact that trading strategies with only a minimal amount of intelligence are able to extract high surpluses from the market [4]. However, such approaches are unsatisfactory because they fail to demonstrate that such minimalist strategies are *dominant* against more sophisticated strategies.

Ideally, we would like to find the game-theoretic solution for the CDA, and show that although truth-telling or other minimalist strategies are not dominant, we can still find the theoretical mix of strategies that are best-responses to each other, and demonstrate that the institution performs well in game-theoretic equilibria. However, even at this point, the CDA along with other variants of the double-auction market, confounds auction theorists by admitting of no clear equilibrium solution [19].[2] Hence in the absence of robust analytical tools, much analysis of this institution has used an ad-hoc mixture of computer simulation and laboratory experiments [7]. These techniques are invaluable, since they are able to faithfully incorporate many of the complex details of

[2] Though this reference is dated, to the best of our knowledge it is still the case that the CDA has no such solution.

the market institution which lead to intractability under conventional analysis. However, the results thus obtained are often critised for being difficult to generalize in the absence of compelling models that explain the observed outcomes.

Recently, however, techniques have been developed that combine simulation-based approaches with an approximated game-theoretic analysis. In the following sections, we describe and then adopt the technique proposed by Walsh and colleagues [23]. However, our work extends the scope of Walsh *et al.*'s use of their technique. Whereas the original work focuses on designing *strategies* for a given institution, specifically the CDA institution, we build on this work by applying the same technique for *mechanism* design issues, using it to compare the CDA and CH institutions.

3 Heuristic-Strategy Approximation

Walsh *et al.* introduce an approximation technique for analyzing games such as the CDA where the sheer size of the strategy and player-type spaces makes an exhaustive game-theoretic solution impractical [23].

3.1 Basic Approach

The central idea is simple. Rather than considering every possible pure strategy and type in the multi-stage game, Walsh *et al.* simplify the analysis by considering a limited number of high-level *heuristic* strategies, such as Cliff's *Zero-Intelligence Plus* (ZIP) strategy [4], and treat these high-level strategies as if they were simple pure strategies in a normal form game. For small numbers of players and high-level strategies, this gives rise to a relatively small normal form game payoff matrix which is amenable to game-theoretic solution. This *heuristic* payoff matrix is calibrated by running many simulations of the market game; variations in payoffs due to different player-types are averaged over many samples of type information resulting in a single mean payoff to each player for each entry in the payoff matrix. Players' types are assumed to be drawn independently from the same distribution, and an agent's choice of strategy is assumed to be independent of its type, which allows the payoff matrix to be further compressed, since we simply need to specify the number of agents playing each strategy to determine the expected payoff to each agent. Thus for a game with k strategies, we represent entries in the heuristic payoff matrix as vectors of the form

$$\boldsymbol{p} = (p_1, \ldots p_k)$$

where p_i specifies the number of agents who are playing the ith strategy. Each entry $p \in P$ is mapped onto an outcome vector $q \in Q$ of the form

$$\boldsymbol{q} = (q_1, \ldots q_k)$$

where q_i specifies the expected payoff to the ith strategy. For a game with n agents, the number of entries in the payoff matrix is given by

$$s = \frac{n^k - 1}{(k-1)!} \tag{1}$$

For small n and small k this results in payoff matrices of manageable size; for $k = 3$ and $n = 6, 8$, and 10 we have $s = 28, 45$, and 66 respectively. For very large n the game becomes intractable, but this is not of major concern since our interest is specifically in markets with small numbers of traders where strategic effects are likely to be prominent.

3.2 Choice of Heuristic Strategies

For moderately large values of k, combinatorial explosion rapidly leads to intractability. This constraint is of more concern than the constraint on n, since in any realistic trading environment we might, a priori, expect agents to be confronted with a vast number of high-level strategies from which to choose. There are, for example, many automated strategies that have been proposed in the literature [4,5,8,18,22]. However, there is evidence to show that in many real-life market scenarios traders choose from a limited number of heuristic strategies. For example, [15] discusses the observed strategic interaction between human agents and two predominant automated bidding strategies commonly used on two real-world auction institutions with different auction designs (Amazon and eBay).

Following these results, we base our work in this paper on the premise that we are modeling the effect of the adoption of automated trading agents in the CDA and CH markets. Thus we compare the behavior of traders using a well known automated strategy for the double-auction [18], and one that has been developed to emulate human strategic behavior in market settings [5]. By comparing these representative heuristic strategies we hope to gain insight into whether non-homogeneous populations of human and agent-based traders are strategically stable, and the likely market outcomes when human and agent-based traders interact. In addition, because we are interested in the strategy-proofness of the mechanisms themselves under different conditions, we also introduce the truth-telling strategy. If a mechanism is strategy proof, it should not be possible to do better than when truthfully report one's limit price. Thus we have $k = 3$ heuristic strategies, which is well within the limits of tractability for the Walsh approximation technique.

4 Experimental Setup

In order to compare the CDA and CH, we must first generate a heuristic payoff matrix for each institution by sampling many simulations of the market game. We made use of the JASA auction simulator[3] which implements a CDA marketplace as described in [22], as well as a CH marketplace where the market is not cleared until offers from all agents have been received.

In order to model human-like trading behavior, we adopt a trading strategy based on a modified version of the Roth-Erev learning algorithm [5] as described in [14], which we abbreviate RE. This is the same version of the Roth-Erev algorithm that we have used in our previous work [16,17]—basically a reinforcement learning approach that builds up a probability distribution over the space of possible bids. We pit this against a strategy based on ZIP, but modified for persistent-shout markets as described by Preist

[3] `http://www.csc.liv.ac.uk/~sphelps/jasa`

and Van Tol [18], which we abbreviate PvT, and the truth-telling strategy which simply bids at the agent's limit price, which we abbreviate TT.

As in [23], at the start of each game half the agents are randomly assigned as buyers and the remainder are chosen as sellers. For each run of the game, we choose limit prices from the same uniform distribution as [23], but limit prices remain fixed across periods in order to allow agents to attempt to learn to exploit any market-power advantage in the supply and demand curves defined by the limit prices for that game (This is common in much experimental work in this area [4,21], and makes it possible for both artificial traders and humans to exploit memory to quickly converge on trade prices). Additionally, although we discard limit-prices which do not yield an equilibrium price, we do not ensure that a minimum quantity exists in competitive equilibrium as this introduces a floor effect which fails to expose the inferior efficiency of a CDA. We use the Mersenne Twister random number generator [11] to draw all random values used in the simulation. Each entry in the heuristic payoff matrix is computed by averaging the payoff to each strategy across 2000 simulations.

5 Dynamic Analysis

Once the heuristic payoff matrix has been computed, we can subject it to a game-theoretic analysis. In conventional mechanism design, we solve the game by finding either a dominant strategy or the Nash equilibria: the sets of strategies that are best-responses to each other. However, because classical game-theory is a static analysis, it is not able to make any predictions about which equilibria are more likely to occur in practice. Such predictions are of vital importance in mechanism design problems. Since our design objectives depend on outcomes, we should give more consideration to outcomes that are more likely than low probability outcomes. For example, if there is a Nash equilibrium for our mechanism which yields very low allocative efficiency, we should not worry too much if this equilibria is extremely unlikely to occur in practice. On the other hand, we should give more weight to equilibria with high probability.

As in [23], we use *evolutionary* game-theory [20] to model how agents might gradually adjust their strategies over time as they learn to improve their behavior in response to their payoffs. We use the replicator dynamics equation

$$\dot{m}_j = [u(e_j, \boldsymbol{m}) - u(\boldsymbol{m}, \boldsymbol{m})]\, m_j \quad (2)$$

where $\boldsymbol{m}$ is a mixed-strategy vector, $u(\boldsymbol{m}, \boldsymbol{m})$ is the mean payoff when all players play $\boldsymbol{m}$, and $u(e_j, \boldsymbol{m})$ is the average payoff to pure strategy j when all players play $\boldsymbol{m}$, and $\dot{m}_j$ is the first derivative of m_j with respect to time. Strategies that gain above-average payoff become more likely to be played, and this equation models a simple *co-evolutionary* process of mimicry learning, in which agents switch to strategies that appear to be more successful. For the three heuristic strategies that we have chosen to analyze, we can interpret this process as modeling the potential uptake of ZIP-like automated trading agent technology; for example, managers bidding using human-like trading strategies may switch to a ZIP-like strategy if they observe a rival firm obtaining better than average profits by using automated trading agents.

For any initial mixed-strategy we can find the eventual outcome from this coevolutionary process by solving $\dot{m}_j = 0$ for all j to find the final mixed-strategy of the converged population. Unlike co-evolutionary approaches that use evolutionary computing to do the search, for instance [16,1], this model has the attractive properties that:

- all Nash equilibria of the game are stationary points under the replicator dynamics; and
- all focal points of the replicator dynamics are Nash equilibria of the evolutionary game.

Thus the Nash equilibrium solutions are embedded in the stationary points of the direction field of the dynamics specified by equation 2. Although not all stationary points are Nash equilibria, by overlaying a dynamic model of learning on the equilibria we can see which solutions are more likely to be discovered by *boundedly-rational* agents. Those Nash equilibria that are stationary points at which a larger range of initial states will end up, are equilibria that are more likely to be reached (assuming an initial distribution that is uniform).

We capture this idea of "range of initial states" with the notion of a *basin of attraction*. The basin of attraction for a stationary point is the range of mixed strategies within which all strategies will, under the replicator dynamics, lead to the stationary point. The bigger the basin, the bigger the region of strategy-space which leads to the attractor, and hence the stronger the attractor.

6 Results

Since $\sum m_i = 1$, each vector $\boldsymbol{m}$ lies in the unit-simplex. For $k = 3$ strategies we can project the unit-simplex onto a two dimensional space and then identify the switching between strategies. We plot this switching in Figures 1—4 which show plots of the *direction field* defined by equation 2 for each institution. The direction field gives us a map which shows the trajectories of strategies of learning agents engaged in repeated interactions, from a random starting position. Thus, for Figure 1, each agent participant has a starting choice of 3 pure strategies (TT, RE and PvT) and any mixed (probabilistic) combination of these three. The pure strategies are indicated by the 3 vertices of the simplex (triangle), while mixed strategies are indicated by points on the boundaries or in the middle of the simplex.

An agent is assigned a random (mixed or pure) strategy to start, and then progressively adjusts this strategy over time in repeated interactions as a result of the learning mechanism described by Equation 2. The paths shown in Figure 1 trace this sequence of adjustments. In order not to overload the display, we have not placed arrows on these paths, but the overwhelming majority of paths start inside the simplex and head outwards, towards the edges and the three vertices. This indicates that the three pure strategies act as attractors for randomly-selected mixed starting strategies. The set of oriented paths leading to each vertex indicates the basin of attraction of the corresponding pure strategy. We can assess the relative likelihood of one strategy relative to another by comparing the size of their respective basins of attraction.

Each plot shows trajectories generated from 250 randomly sampled initial $\boldsymbol{m}$ vectors. For now, we assume that every initial mixed-strategy is equally likely to be adopted

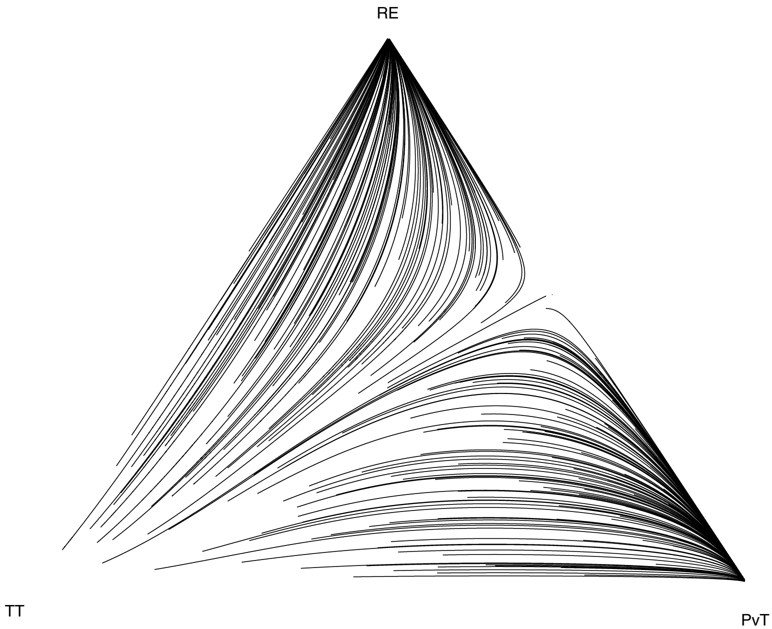


Fig. 1. Replicator dynamics direction field for CH with 6 agents

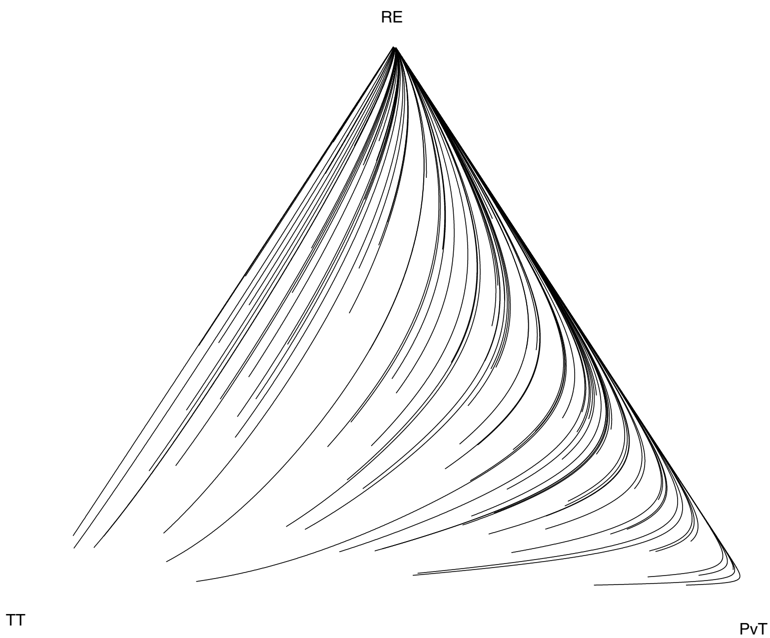


Fig. 2. Replicator dynamics direction field for CDA with 6 agents

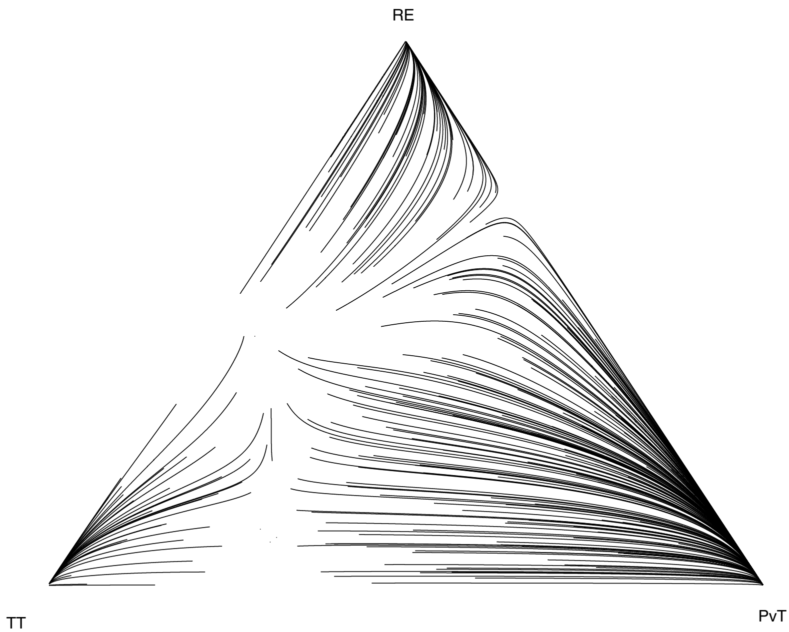


Fig. 3. Replicator dynamics direction field for CH with 8 agents

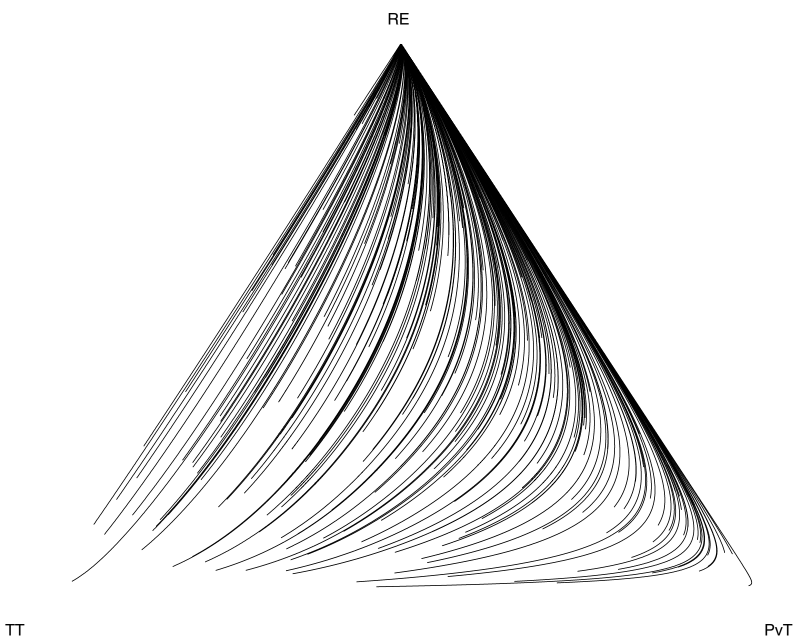


Fig. 4. Replicator dynamics direction field for CDA with 8 agents

Table 1. Pure-strategy equilibria probability distribution for 6 agents

Equilibrium	CH probability	payoff	CDA probability	payoff
TT	0.00	1.00	0.00	0.87
RE	0.50	1.00	1.00	0.98
PvT	0.50	1.00	0.00	0.93

Table 2. Pure-strategy equilibria probability distribution for 8 agents

Equilibrium	CH probability	payoff	CDA probability	payoff
TT	0.17	1.00	0.00	0.85
RE	0.18	1.00	1.00	0.98
PvT	0.65	1.00	0.00	0.92

as a starting-point for the co-evolutionary process, and so we randomly sample the initial values of m from a uniform distribution and plot their trajectories as they evolve according to equation 2.

To automate the analysis of institutions, we need to be able to provide some metric that allows us to quantify their performance in this kind of analysis. In other words, we would like to measure the size of the basin of attraction of each equilibria, in order to arrive at a probability of the equilibria actually occuring. Different equilibria will yield different outcomes and different values of our design objectives, such as market efficiency, and we would like to weight these according to their likelyhood.

Tables 2 and 1 show the stationary points of 1000 randomly sampled trajectories together with the proportion of trajectories that terminate at that point. Given the random start, this probability is an estimate of the probability of each equilibrium. In the absence of a static analysis, we discount the stationary-points that occur with less than 1% probability. Since we know the payoffs of the various points from the heuristic payoff matrix, we can then compute expected payoffs, which are also shown in the table. With probabilities over outcomes, we are in a position to assess the design of each mechanism.

7 Discussion

First of all it is clear that TT is not dominant, and hence neither the CH or CDA mechanism is strategy-proof. However, it is interesting to note that when we increase the number of agents from 6 to 8 in a CH, we see the emergence of a truth-telling equilibrium. This agrees with the approximate analysis presented in [19], and suggests that truth-telling may become a strategy adopted by more traders as the market grows larger.

In a CH market, we see that the most likely strategy to be played is the ZIP-like trading agent strategy, whereas in a CDA, the human-like RE strategy is dominant.

As expected from our discussion above, we see that payoffs under truthful bidding in a CDA are relatively low; 86% in this case. This might suggest that the CDA itself has a rather low efficiency. However, in order to assess the efficiency of the CDA we must take into account the fact that truthtelling is not an equilibrium. Since the RE strategy is dominant we should assume that all agents will eventually use this strategy, and so our efficiency will be equal to the pure-strategy payoff for RE, which in this case is 98%. In order to calculate the efficiency for the 8-agent CH market, we need to take into account that there are three possible pure-strategy equilibria. In this particular scenario, each equilibria results in the same efficiency of 100%, so we can conclude that the CH market will yield 100% efficiency in all eventualities.

Although the CDA yields lower surplus, it is not as inefficient as we might expect had we assumed that it was designed according to the revelation principle. As [6] points out, the main reason for choosing a CDA rather than a CH is to handle larger volumes of trade, and our results here suggest that this is a reasonable trade-off. Switching to a CDA from a CH as the New York Stock Exchange did in the 1860s, does not seem likely to entail a large loss of efficiency.

The above analysis assumes that all initial points in the mixed-strategy phase-space are equally likely to be selected. However, if we are in a situation where we are proposing to make changes to an existing “legacy” exchange with existing traders, our observations of current trading behavior in the legacy mechanism may influence our beliefs about likely behavior in any proposed altered version of the mechanism. For example, we may be tasked with assessing the likely impact in switching from a CH clearing rule to an exchange with continuous clearing. If we observe that traders bid truthfully in the existing mechanism, then when we come to perform the dynamic analysis for the new design, we may decide to weight our distribution of initial mixed-strategies in favor of truth-telling to reflect current observations.

8 Further Work

What we have demonstrated in this paper is an approach that provides an approximate game-theoretic analysis, involving equilibria over multiple heuristic strategies, for mechanisms that do not admit an analytical solution. This is fully automated, and gives us a means of *analyzing* and hence comparing auction mechanisms. Our previous work has demonstrated proof-of-concept for the idea of *evolving* auction mechanisms, for example using genetic programming to evolve parts of the pricing mechanism for a double auction market [17], establishing the quality of the market using a single heuristic strategy.

Since all parts of the approach we have detailed here are fully automated, it is possible to combine the these two lines of work. This will enable us to create new auction mechanisms and then use the kind of analysis described here to rate them, thus searching the space of possible mechanisms while rigorously analyzing them. With our current implementation running on a 1.4Ghz Athlon AMD processor, it takes approximately 24-hours to generate the heuristic payoff matrix and perform the dynamic analysis for a single 10-agent mechanism. We hope to significantly reduce this evaluation cost by

– using a more selective sampling, as in [24] for example;
– further optimizing our code, and
– reducing the number of samples at the expense of accuracy whilst using an optimization algorithm that will be robust to the additional noise.

Finally, we recognize that the existence of dominant strategies, and the lack of mixed-strategy attractors are probably as a result of us not taking a representative set of heuristic strategies for these early experiments. Future work will extend the number of heuristic strategies that are are used in our analysis.

With these techniques we will move closer to our overall goal of completely automated mechanism design.

Acknowledgements

This work was partially supported by NSF IIS-0329037. The authors would like to thank William Walsh, Gerry Tesauro and Karl Tuyls for their assistance, the workshop attendees for pointing out inaccuracies in a previous draft, and Bob Brisco for his helpful comments during a talk given at BTExact.

References

1. P. J. Angeline and J. B. Pollack. Competitive environments evolve better solutions for complex tasks. In S. Forrest, editor, *Genetic Algorithms: Proceedings of the Fifth International Conference (GA93)*, 1993.
2. W.D.A. Bryant. Information, adjustment and the stability of equilibrium. Technical Report 6/2000, Department of Economics, Macquarie University, Sydney, Australia, 2000.
3. D. Cliff. Evolution of market mechanism through a continuous space of auction-types. Technical Report HPL-2001-326, HP Labs, 2001.
4. D. Cliff and J. Bruten. Minimal-intelligence agents for bargaining behaviors in market-based environments. Technical report, HP Labs, 1997. `http://www.hpl.hp.com/techreports/97/HPL-97-91.html`.
5. I. Erev and A. E. Roth. Predicting how people play games: Reinforcement learning in experimental games with unique, mixed strategy equilibria. *American Economic Review*, 88(4):848–881, 1998.
6. D. Friedman. The double auction market institution: A survey. In Friedman and Rust, editors, *The Double Auction Market: Institutions, Theories and Evidence*, Santa Fe Studies in the Sciences of Complexity, chapter 1, pages 3–26. Westview, 1993.
7. D. Friedman and J.Rust, editors. *The Double Auction Market: Institutions, Theories and Evidence*, volume XIV of *Santa Fe Institute Studies in the Sciences of Complexity*. Westview, 1993.
8. S. Gjerstad and J. Dickhaut. Price formation in double auctions. *Games and Economic Behaviour*, 22:1–29, 1998.
9. P. Klemperer. How (not) to run auctions: the European 3G telecom auctions. *European Economic Review*, 46(4–5):829–845, 2002.
10. D. A. MacKenzie. An equation and its worlds: *Bricolage*, exemplars, disunity and performativity in financial economics. *Social Studies of Science*, 33:831–868, 2003.
11. M. Matsumoto and T. Nishimura. Mersenne twister: A 623-dimensionally equidistributed uniform pseudo-random number generator. *ACM Transactions on Modeling and Computer Simulation*, 8(1):3–30, 1998.
12. P. Mirowski. *Machine Dreams: Economics Becomes a Cyborg Science*. Cambridge University Press, Cambridge, UK, 2002.
13. R. B. Myerson and M. A. Satterthwaite. Efficient mechanisms for bilateral trading. *Journal of Economic Theory*, 28:265–281, 1983.

14. J. Nicolaisen, V. Petrov, and L. Tesfatsion. Market power and efficiency in a computational electricity market with discriminatory double-auction pricing. *IEEE Transactions on Evolutionary Computation*, 5(5):504–523, October 2001.
15. A. Ockenfels and A.E. Roth. The timing of bids in internet auctions: market design, bidder behaviour, and artificial agents. *AI Magazine*, 2002.
16. S. Phelps, P. McBurney, S. Parsons, and E. Sklar. Co-evolutionary mechanism design: a preliminary report. In J. Padget, D. C. Parkes, N. M. Sadeh, O. Shehory, and W. E. Walsh, editors, *Agent Mediated Electronc Commerce IV*, Bologna, Italy, 2002. AAMAS 2002.
17. S. Phelps, S. Parsons, E. Sklar, and P. McBurney. Using genetic programming to optimise pricing rules for a double auction market. In *Proceedings of the Workshop on Agents for Electronic Commerce*, Pittsburgh, 2003.
18. C. Preist and M. van Tol. Adaptive agents in a persistent shout double auction. Technical Report HPL-2003-242, Hewlett Packard Labs, 2003.
19. M. A. Satterthwaite and S. R. Williams. The Bayseian theory of the k-double auction. In Friedman and Rust, editors, *The Double Auction Market: Institutions, Theories and Evidence*, Santa Fe Studies in the Sciences of Complexity, chapter 4, pages 99–123. Westview, 1993.
20. J. M. Smith. *Evolution and the Theory of Games*. Cambridge University Press, 1982.
21. V. L. Smith. An experimental study of competitive market behaviour. *The Journal of Political Economy*, 70(2):111–137, April 1962.
22. G. Tesauro and R. Das. High-performance bidding agents for the continuous double auction. In *Third ACM Conference on Electronic Commerce*, pages 206–209, 2001.
23. W. E. Walsh, R. Das, G. Tesauro, and J.O. Kephart. Analyzing complex strategic interactions in multi-agent games. In *AAAI-02 Workshop on Game Theoretic and Decision Theoretic Agents*, 2002.
24. W. E. Walsh, D. C. Parkes, and R. Das. Choosing samples to compute heuristic-strategy Nash equilibrium. In *Proceedings of the Fifth Workshop on Agent-Mediated Electronic Commerce*, 2003.

Auctions and Bidding with Information

John Debenham

Faculty of Information Technology, University of Technology,
Sydney, NSW, Australia
debenham@it.uts.edu.au
http://www-staff.it.uts.edu.au/~debenham/

Abstract. An approach to auctions and bidding is founded on observations and expectations of the opponents' behavior and not on assumptions concerning the opponents' motivations or internal reasoning. A bidding agent operates in an information-rich environment that includes real-time market data and data extracted from the World Wide Web. This agent employs maximum entropy inference to determine its actions on the basis of this uncertain data. Maximum entropy inference may be applied both to multi-issue and to single-issue negotiation. Multi-issue variants of the four common auction mechanisms are discussed.

1 Introduction

Game theory, dating back to the work of John von Neumann and Oscar Morgenstern, provides the basis for the analysis of auctions and bidding. There is a wealth of material in this analysis [7] originating with the work of William Vickrey. Fundamental to this analysis is the central role of the utility function, and the notion of rational behavior by which an agent aims to optimize its utility, when it is able to do so, and to optimize its expected utility otherwise. Analyses that are so founded on game theory are collectively referred to as *game theoretic*, or *GT*.

The application of *GT* to the design of auction mechanisms has been both fruitful and impressive — rational behavior provides a theoretical framework in which mechanism performance may be analyzed. A notable example being the supremely elegant *Generalized Vickrey* mechanism [14]. *GT* also leads to prescriptive results concerning agent behavior, such as the behavior of agents in the presence of hard deadlines [13]. The general value of *GT* as a foundation for a prescriptive theory of agent behavior is limited both by the extent to which an agent knows its own utility function, and by its certainty in the probability distributions of the utility functions (or, *types*) of its opponents.

In some negotiations — such as when an agent buys a hat, a car, a house or a company — she may not know her utility with certainty. Nor may she be aiming to optimize anything — she may simply want to buy it. Further, she may be even less certain of her opponents' types, or whether her opponents are even aware of the concept of utility. In such negotiations, an agent may be more driven towards establishing a feeling of personal "comfort" through a process of information acquisition, than by a desire to optimize an uncertain personal utility function.[1]

[1] After becoming CEO of Citicorp in 1967, Walter Wriston said: "Banking is not about money; it is about information". It is tempting to echo this as "Negotiation is not about utility;. . . ".

P. Faratin and J.A. Rodríguez-Aguilar (Eds.): AMEC 2004, LNAI 3435, pp. 115–128, 2005.

A negotiation agent, Π, attempts to fuse the negotiation with the information that is generated both by and because of it. To achieve this, it draws on ideas from information theory rather than game theory. Π decides what to do — such as whether to bid in an auction — on the basis of information that may be qualified by expressions of degrees of belief. Π uses this information to calculate, and continually re-calculate, probability distributions for that which it does not know. One such distribution, over the set of all possible deals, expresses Π's belief in the acceptability of a deal. Other distributions attempt to predict the behavior of its opponents — such as what they might bid in an auction. These distribution are calculated from Π's knowledge and beliefs using maximum entropy inference. Π makes no assumptions about the internals of its opponents, including whether they have, or are even aware of the concept of, utility functions. Π is purely concerned with its opponents' behavior — what they do — and not with assumptions about their motivations.

Maximum entropy inference is chosen because it enables inferences to be drawn from incomplete and uncertain information, and because of its encapsulation of common sense reasoning [10]. Unknown probability distributions are inferred using *maximum entropy inference* [8] that is based on random worlds [4]. The maximum entropy probability distribution is "the least biased estimate possible on the given information; i.e. it is maximally noncommittal with regard to missing information" [6]. As applied to the analysis of auctions, maximum entropy inference presents four difficulties. First, it assumes that what the agent knows is "the sum total of the agent's knowledge, it is not a summary of the agent's knowledge, it is all there is" [10]. This assumption referred to as Watt's Assumption [5]. So if knowledge is absent it may do strange things. Second, it may only be applied to a consistent set of beliefs — this may mean that valuable information is destroyed by the the belief revision process that copes with the continuous arrival of new information. Third, its knowledge base is expressed in first-order logic. So issues that have unbounded domains — such as price — can only be dealt with either exactly as a large quantity of constants for each possible price, or approximately as price intervals. This decision will effect the inferences drawn and is referred to as representation dependence [4]. Fourth, maximum entropy can be tricky to calculate — although here the equivalent maximum likelihood problem for the Gibbs distribution [11] was solved numerically without incident by applying the Newton-Raphson method to as many non-linear, simultaneous equations as there are beliefs in the knowledge base. Despite these four difficulties, maximum entropy inference is an elegant formulation of common sense reasoning. Maximum entropy inference is also independent of any structure on the set of all possible deals. So it copes with single-issue and multiple-issue negotiation without modification. It may also be applied to probabilistic belief logic. These properties are particularly useful in analyzing auctions and bidding.

The information-theory oriented analysis described here, which employs maximum entropy inference, is referred to as *ME* in contrast to *GT*.

2 Bidding Agent Π

The form of negotiation considered is between bidding agents and an auctioneer Υ in an information rich environment. The agent described here is called the *Bidding Agent*, or Π, it engages in auctions with a set of S opponents $\{\Omega_1, \cdots, \Omega_S\}$. General information is extracted from the World Wide Web using special purpose bots that import and continually confirm information that is then represented in pre-specified predicates. Π receives information by observing its opponents $\{\Omega_i\}$ and from these bots.

The integrity of information decays in time. Little appears to be known about how the integrity of information, such as news-feeds, decays. One source of information is the signals received by observing the behavior of the opponent agents both prior to a negotiation and during it. For example, if an opponent bid \$8 in an auction for an identical good two days ago then my belief that she will bid \$8 now could be 0.8. When the probability of a decaying belief approaches 0.5 the belief is discarded.

2.1 Agent Architecture

The agents communicate using the following predicate: *Bid*(.), where *Bid*(δ) means "the sender bids a deal δ". A *deal* is a pair of commitments $\delta_{\Pi:\Omega}(\pi, \omega)$ between an agent Π and an opponent agent Ω, where π is Π's commitment and ω is Ω's commitment. $\mathcal{D} = \{\delta_i\}_{i=1}^{D}$ is the deal set — ie: the set of all possible deals. If the discussion is from the point of view of a particular agent then the subscript "Π:" may be omitted, and if that agent has just one opponent that the "Ω" may be omitted as well. These commitments may involve multiple issues and not simply a single issue such as trading price. The set of *terms*, $\mathcal{T}$, is the set of all possible commitments that could occur in deals in the deal set. An agent may have a real-valued *utility* function: $\mathbf{U} : \mathcal{T} \to \Re$, that induces a total ordering on $\mathcal{T}$. For any deal $\delta = (\pi, \omega)$ the expression $\mathbf{U}(\omega) - \mathbf{U}(\pi)$ is called the *surplus* of δ, and is denoted by $\mathbf{L}(\delta)$ where $\mathbf{L} : \mathcal{T} \times \mathcal{T} \to \Re$. For example, the values of the function $\mathbf{U}$ may expressed in units of money. It may not be possible to specify the utility function either precisely or with certainty.[2] This is addressed in Sec. 4 where a predicate *Accept*(.) represents the acceptability of a deal.

Π has a knowledge base $\mathcal{K}$ and a belief set $\mathcal{B}$. Each of these two sets contains statements in a first-order language $\mathcal{L}$. $\mathcal{K}$ contains statements that are generally true. The belief set $\mathcal{B} = \{\beta_i\}$ contains statements, β_i, that are each qualified with a *given sentence probability*, $\mathbf{B}(\beta_i)$ that represents the agent's belief in the truth of the statement. The integrity of the statements in $\mathbf{B}$ may decay in time. The distinction between the knowledge base $\mathcal{K}$ and the belief set $\mathcal{B}$ is simply that $\mathcal{K}$ contains unqualified statements and $\mathcal{B}$ contains statements that are qualified with sentence probabilities. This apparently odd distinction is made because $\mathcal{K}$ and $\mathcal{B}$ play different roles in the method described in Sec. 2.2.

Π's actions are determined by its "strategy". A *strategy* is a function $\mathbf{S} : \mathcal{K} \times \mathcal{B} \to \mathcal{A}$ where $\mathcal{A}$ is the set of actions. The idea is that at certain distinct times the function $\mathbf{S}$ is applied to $\mathcal{K}$ and $\mathcal{B}$ and the agent does something. The set of actions, $\mathcal{A}$, is limited to sending *Bid*(.) messages to the auctioneer Υ. The way in which $\mathbf{S}$ works is described in Sec. 5. In between the discrete times at which $\mathbf{S}$ is activated, information may arrive. Incoming information from all sources is time-stamped and placed in an "In Box", $\mathcal{X}$, as it arrives. Then, momentarily before the $\mathbf{S}$ function is activated, a "revision function"

[2] The often-quoted oxymoron "I paid too much for it, but its worth it." attributed to Samuel Goldwyn, movie producer, illustrates that intelligent agents may negotiate with uncertain utility.

$\mathbf{R}$ is activated: $\mathbf{R} : (\mathcal{X} \times \mathcal{K} \times \mathcal{B}) \rightarrow (\mathcal{K} \times \mathcal{B})$. $\mathbf{R}$ clears the "In Box", and updates $\mathcal{K}$ and $\mathcal{B}$ to ensure consistency. It is not described here.

2.2 Maximum Entropy Inference

Π uses maximum entropy inference. Let $\mathcal{G}$ be the set of all positive ground literals that can be constructed using the predicate and function symbols in $\mathcal{L}$.[3] A *possible world* is a valuation function $\mathbf{v} : \mathcal{G} \rightarrow \{\top, \bot\}$. That is, a possible world assigns either true ($\top$) or false ($\bot$) to each ground literal in $\mathcal{G}$. $\mathbf{V}$ denotes the set of all possible worlds, and $\mathbf{V}_{\mathcal{K}}$ denotes the set of possible worlds that are consistent with the agent's knowledge base $\mathcal{K}$ [4].

A *random world* for $\mathcal{K}$ is a probability distribution $\mathbf{W}_{\mathcal{K}} = \{p_i\}$ over $\mathbf{V}_{\mathcal{K}} = \{\mathbf{v}_i\}$, where $\mathbf{W}_{\mathcal{K}}$ expresses an agent's degree of belief that each of the possible worlds is the actual world. The *derived sentence probability* of any sentence σ in $\mathcal{L}$, *with respect to* a random world $\mathbf{W}_{\mathcal{K}}$ is:

$$\mathbf{P}_{\mathbf{W}_{\mathcal{K}}}(\sigma) \triangleq \sum_n \{\, p_n \;:\; \sigma \; is \; \top \; in \; \mathbf{v}_n \,\} \tag{1}$$

That is, we only admit those possible worlds in which σ is true. A random world $\mathbf{W}_{\mathcal{K}}$ is *consistent* with the agent's beliefs $\mathcal{B}$ if: $(\forall \beta \in \mathcal{B})(\mathbf{B}(\beta) = \mathbf{P}_{\mathbf{W}_{\mathcal{K}}}(\beta))$. That is, for each belief its derived sentence probability as calculated using Eqn. 1 is equal to its given sentence probability.

The *entropy* of a discrete random variable X with probability mass function $\{p_i\}$ is defined in the usual way [8]: $H(X) = -\sum_n p_n \log p_n$ where: $p_n \geq 0$ and $\sum_n p_n = 1$. Let $\mathbf{W}_{\{\mathcal{K},\mathcal{B}\}}$ be the "maximum entropy probability distribution over $\mathbf{V}_{\mathcal{K}}$ that is consistent with $\mathcal{B}$". Given an agent with $\mathcal{K}$ and $\mathcal{B}$, its *derived sentence probability* for any sentence, σ, in $\mathcal{L}$, is:

$$\mathbf{P}(\sigma) = \mathbf{P}_{\mathbf{W}_{\{\mathcal{K},\mathcal{B}\}}}(\sigma) \tag{2}$$

Using Eqn. 2, the derived sentence probability for any belief, β_i, is equal to its given sentence probability. So the term *sentence probability* is used from here on without ambiguity. Π uses *maximum entropy inference* which attaches the derived sentence probability to any given sentence σ.

Maximizing Entropy with Linear Constraints. If X is a discrete random variable taking a finite number of possible values $\{x_i\}$ with probabilities $\{p_i\}$ then the *entropy* is the average uncertainty removed by discovering the true value of X, and is given by $H = -\sum_n p_n \log p_n$. The direct optimization of H subject to a number, θ, of linear constraints of the form $\sum_n p_n g_k(x_n) = \overline{g}_k$ for given constants $\overline{g}_k$, where $k = 1, \cdots, \theta$, is a difficult problem. Fortunately this problem has the same unique solution as the *maximum likelihood problem* for the Gibbs distribution [11]. The solution to both problems is given by:

$$p_n = \frac{exp\Big(-\sum_{k=1}^{\theta} \lambda_k g_k(x_n)\Big)}{\sum_m exp\Big(-\sum_{k=1}^{\theta} \lambda_k g_k(x_m)\Big)}, \quad n = 1, 2, \cdots \tag{3}$$

[3] Constants are 0-ary functions.

where the constants $\{\lambda_i\}$ may be calculated using Eqn. 3 together with the three sets of constraints: $p_n \geq 0$, $\sum_n p_n = 1$ and $\sum_n p_n g_k(x_n) = \overline{g}_k$.

3 Representation Dependence

ME is criticized [4] because the way in which the knowledge is formulated in $\mathcal{K}$ and $\mathcal{B}$ determines the values derived. This property is promoted here as a strength of the method because the correct formulation of the knowledge base, using the rich expressive power of first-order probabilistic logic, encapsulates features of the application at a fine level of detail.

Price is a common issue in auction and market applications. Two ways of representing price in logic are: to establish a logical constant for each possible price, and to work instead with price intervals. Admitting the possibility of an interval containing just one value, the second generalizes the first. To represent price using price intervals, we have to specify the "width" of each interval. Suppose in an application an item will be sold in excess of \$100. Suppose the predicate *TopBid*(Ω, δ) means "δ is the highest price that agent Ω is prepared to bid". This predicate will satisfy: $\forall xy((\mathit{TopBid}(\Omega, x) \wedge \mathit{TopBid}(\Omega, y)) \rightarrow (x = y))$. A crude representation of the set of possible bids is as two logical constants in $\mathcal{L}$: $[100, 200)$ and $[200, \infty)$. Following the development in Sec. 2.2, there are two positive ground literals in $\mathcal{G}$: *TopBid*(Ω, $[100, 200)$) and *TopBid*(Ω, $[200, \infty)$), and there are three possible worlds: $\{(\perp, \perp), (\top, \perp), (\perp, \top)\}$. In the absence of any further information, the maximum entropy distribution is uniform, and, for example, the probability that Ω's highest bid $\geq$ \$200 is $\frac{1}{3}$. Now if the set of possible bids had been represented as *three* logical constants: $[100, 150)$, $[150, 200)$ and $[200, \infty]$, then the same probability is $\frac{1}{4}$. Which is correct: $\frac{1}{3}$ or $\frac{1}{4}$? That depends on Π's beliefs about Ω. In both of these examples, by using *ME*, and by specifying no further knowledge about *TopBid*(.), we have implicitly asserted that the probability of each possible world being the true world is the same. In the first example all three are $\frac{1}{3}$, and in the second all four are $\frac{1}{4}$. This is what happens when the "maximally noncommittal" distribution is chosen. Conversely, if believe that: $\forall x, y(\mathbf{P}(\mathit{TopBid}(\Omega, x)) = \mathbf{P}(\mathit{TopBid}(\Omega, y)))$ then it is not necessary to include this in $\mathcal{K}$ — it is implicitly present and we should appreciate that it is so. Sec. 1 mentioned Watt's Assumption, that assumption says more than it might at first appear to.

Following from the previous paragraph with just two logical constants, suppose the predicate *MayBid*(Ω, δ) means "Ω is prepared to make a bid of δ". Assuming the Ω will prefer to pay less than more, this predicate will satisfy: $\kappa_1 : \forall x, y((\mathit{MayBid}(\Omega, x) \wedge (x \geq y)) \rightarrow \mathit{MayBid}(\Omega, y))$, where x and y are intervals and the meaning of "$\geq$" is obvious. With just κ_1 in $\mathcal{K}$ there are three possible worlds: $\{(\perp, \perp), (\top, \perp), (\top, \top)\}$. The maximum entropy distribution is uniform, and, $\mathbf{P}(\mathit{MayBid}(\Omega, [100, 200))) = \frac{2}{3}$, and $\mathbf{P}(\mathit{MayBid}(\Omega, [200, \infty])) = \frac{1}{3}$. With no additional information, $\mathbf{P}(\mathit{TopBid}(\Omega, x))$ will be uniform and $\mathbf{P}(\mathit{MayBid}(\Omega, x))$ will be linear decreasing in x.

The conclusion to be drawn from the previous two paragraphs is that when an issue is represented using intervals there is no "right" or "wrong" choice of intervals. However, choosing the intervals so that the expected probability distribution of at least one key predicate is uniform over those intervals may simplify $\mathcal{K}$ and $\mathcal{B}$.

In *GT* each agent models its opponents by speculating on their type. Beliefs concerning an opponent's type may be represented as a probability distribution over its expected utility. In *ME* the relative truth of possible worlds are determined by statements in first-order probabilistic logic that represent beliefs concerning the opponents' behavior. So *ME* models its opponents in a fundamentally different way to *GT*, and uses a richer language to do so.

An exemplar application is used following. It concerns the purchase of a particular second-hand motor vehicle, with some period of warranty, for cash. So the two issues in this negotiation are: the period of the warranty, and the cash consideration. The meaning of the predicate *MayBid*(Ω, δ) is unchanged but δ now consists of a pair of issues and the deal set has no natural ordering. Suppose that Π wishes to apply *ME* to estimate values for: $\mathbf{P}(\mathit{MayBid}(\Omega, \delta))$ for various δ. Suppose that the warranty period is simply $0, \cdots, 4$ years, and that the cash amount for this car will certainly be at least $5,000 with no warranty, and is unlikely to be more than $7,000 with four year's warranty. In what follows all price units are in thousands of dollars. Suppose then that the deal set in this application consists of 55 individual deals in the form of pairs of warranty periods and price intervals: { (w, [5.0, 5.2)), (w, [5.2, 5.4)), (w, [5.4, 5.6)), (w, [5.6, 5.8), (w, [5.8, 6.0)), (w, [6.0, 6.2)), (w, [6.2, 6.4)), (w, [6.4, 6.6)), (w, [6.6, 6.8)), (w, [6.8, 7.0)), (w, [7.0, ∞)) }, where $w = 0, \cdots, 4$. Suppose that Π has received intelligence that agent Ω is prepared to bid 6.0 with no warranty, and to bid 6.9 with one year's warranty, and Π believes this with probability 0.8. Then this leads to two beliefs: $\beta_1 : \mathit{TopBid}(0, [6.0, 6.2));\ \mathbf{B}(\beta_1) = 0.8$, $\beta_2 : \mathit{TopBid}(1, [6.8, 7.0));\ \mathbf{B}(\beta_2) = 0.8$. Following the discussion above, before "switching on" *ME*, Π should consider whether it believes that $\mathbf{P}(\mathit{MayBid}(\Omega, \delta))$ is uniform over δ. If it does then it includes both β_1 and β_2 in $\mathcal{B}$, and calculates $\mathbf{W}_{\{\mathcal{K},\mathcal{B}\}}$ that yields estimates for $\mathbf{P}(\mathit{MayBid}(\Omega, \delta))$ for all δ. If it does not then it should include further knowledge in $\mathcal{K}$ and $\mathcal{B}$. For example, Π may believe that Ω is more likely to bid for a greater warranty period the higher her bid price. If so, then this is a multi-issue constraint, that is represented in $\mathcal{B}$, and is qualified with some sentence probability.

4 From Utility to Acceptability

One aim of this discussion is lay the foundations for a normative theory of auctions and bidding that does not rely on knowledge of an agent's utility, and does not require an agent to make assumptions about her opponents' utilities or types, including whether they are aware of their utility. Such a theory must provide some mechanism that determines the *acceptability* of a deal; ie: the probability that the deal is acceptable to an agent. Agent, Π, is attempting to buy or bid for a second-hand motor vehicle with a specific period of warranty as described in Sec. 3. Here, Π is bidding in a multi-issue auction for a vehicle, where the two issues are price and warranty period. Possible rules for this auction are described in Sec. 5.

The proposition $(\mathit{Accept}(\delta) \mid \mathcal{I}_t)$ means: "Π will be comfortable accepting the deal δ given that Π knows information $\mathcal{I}_t$ at time t". In an auction for terms ω, Π's strategy, $\mathbf{S}$, may bid one or more π for which $\mathbf{P}(\mathit{Accept}((\pi, \omega)) \mid \mathcal{I}_t) \geq \alpha$ for some threshold constant α. This section describes how Π estimates: $\mathbf{P}(\mathit{Accept}(\delta) \mid \mathcal{I}_t)$. The meaning of *Accept*(δ) is described below, it is intended to put Π in the position "looking back on it, I made the right decision at the time" — this is a vague notion but makes good sense to the author.

With the motor vehicle application in mind, $\mathbf{P}(Accept(\delta) \mid \mathcal{I}_t)$ is derived from conditional probabilities attached to four other propositions: *Suited*(ω), *Good*(Ω), *Fair*(δ), and *Me*(δ). meaning respectively: "terms ω are perfectly suited to Π's needs", "Ω will be a good agent for Π to be doing business with", "δ is generally considered to be a fair deal at least", and "on strictly subjective grounds, the deal δ is acceptable to Π". These four probabilities are: $\mathbf{P}(Suited(\omega) \mid \mathcal{I}_t)$, $\mathbf{P}(Good(\Omega) \mid \mathcal{I}_t)$, $\mathbf{P}(Fair(\delta) \mid \mathcal{I}_t \cup \{Suited(\omega), Good(\Omega)\})$ and $\mathbf{P}(Me(\delta) \mid \mathcal{I}_t \cup \{Suited(\omega), Good(\Omega)\})$. The last two of these four probabilities factor out both the suitability of ω and the appropriateness of the opponent Ω. The third captures the concept of "a fair market deal" and the fourth a strictly subjective "what ω is worth to Π". The "*Me*(.)" proposition is closely related to the concept of a private valuation in game theory. This derivation of $\mathbf{P}(Accept(\delta) \mid \mathcal{I}_t)$ from these four probabilities may not be suitable for assessing other types of deal.

To determine $\mathbf{P}(Suited(\omega) \mid \mathcal{I}_t)$, if there are sufficiently strong preference relations to establish extrema for this distribution then they may be assigned extreme values ≈ 0.0 or 1.0. Π is then repeatedly asked to provide probability estimates for the offer ω that yields the greatest reduction in entropy for the resulting distribution [8]. This continues until Π considers the distribution to be "satisfactory". This is tedious but the "preference acquisition bottleneck" appears to be an inherently costly business [2].

To determine $\mathbf{P}(Good(\Omega) \mid \mathcal{I}_t)$ involves an assessment of the reliability of the opponent Ω. For some retailers (sellers), information — of varying reliability — may be extracted from sites that rate them. For individuals, this may be done either through assessing their reputation established during prior trades [12], or through the use of some intermediate escrow service that is rated for "reliability" instead.

$\mathbf{P}(Fair(\delta) \mid \mathcal{I}_t \cup \{Suited(\omega), Good(\Omega)\})$ is determined by reference to market data. Suppose that recently a similar vehicle sold with three year's warranty for \$6,500, and another less similar was sold for \$5,500 with one year's warranty. These are fed into $\mathcal{I}_t$ and are represented as two beliefs in $\mathcal{B}$: $\beta_3 : Fair(3, [6.4, 6.6));\ \mathbf{B}(\beta_3) = 0.9$, $\beta_4 : Fair(3, [5.4, 5.6));\ \mathbf{B}(\beta_4) = 0.8$. In an open-cry auction one source of market data is the bids made by other agents. The sentence probabilities that are attached to this data may be derived from knowing the identity, and so too the reputation, of the bidding agent. In this way the acceptability value is continually adjusted as information becomes available. In addition to β_3 and β_4, there are three chunks of knowledge in $\mathcal{K}$. First, $\kappa_2 : Fair(4, 4999)$ that determines a base value for which $\mathbf{P}(Fair) = 1$, and two other chunks that represent Π's preferences concerning price and warranty:

$$\kappa_3 : \forall x, y, z((x > y) \rightarrow (Fair(z, x) \rightarrow Fair(z, y)))$$
$$\kappa_4 : \forall x, y, z((x > y) \rightarrow (Fair(y, z) \rightarrow Fair(x, z)))$$

The deal set is a 5×11 matrix with highest interval $[7.0, \infty)$. The three statements in $\mathcal{K}$ mean that there are 56 possible worlds. The two beliefs are consistent with each other and with $\mathcal{K}$. A complete matrix for $\mathbf{P}(Fair(\delta) \mid \mathcal{I}_t)$ is derived by solving two simultaneous equations of degree two using Eqn. 3. As new evidence becomes available it is represented in $\mathcal{B}$, and the inference process is re-activated. If new evidence renders $\mathcal{B}$ inconsistent then this inconsistency will be detected by the failure of the process to yield values for the probabilities in $[0, 1]$. If $\mathcal{B}$ becomes inconsistent then the revision function $\mathbf{R}$ identifies and removes inconsistencies from $\mathcal{B}$ prior to re-calculating the probability distribution. The values were calculated using a program written by Paul Bogg, a PhD student in the Faculty of IT at UTS, [1]:

	$w = 0$	$w = 1$	$w = 2$	$w = 3$	$w = 4$
$p = [7.0, \infty)$	0.0924	0.1849	0.2049	0.2250	0.2263
$p = [6.8, 7.0)$	0.1849	0.3697	0.4099	0.4500	0.4526
$p = [6.6, 6.8)$	0.2773	0.5546	0.6148	0.6750	0.6789
$p = [6.4, 6.6)$	0.3697	0.7394	0.8197	**0.9000**	0.9053
$p = [6.2, 6.4)$	0.3758	0.7516	0.8331	0.9147	0.9213
$p = [6.0, 6.2)$	0.3818	0.7637	0.8466	0.9295	0.9374
$p = [5.8, 6.0)$	0.3879	0.7758	0.8600	0.9442	0.9534
$p = [5.6, 5.8)$	0.3939	0.7879	0.8734	0.9590	0.9695
$p = [5.4, 5.6)$	0.4000	**0.8000**	0.8869	0.9737	0.9855
$p = [5.2, 5.4)$	0.4013	0.8026	0.8908	0.9790	0.9921
$p = [5.0, 5.2)$	0.4026	0.8053	0.8947	0.9842	0.9987

The two evidence values are shown above in bold face.

Determining $\mathbf{P}(\mathit{Me}(\delta) \mid \mathcal{I}_t \cup \{\mathit{Suited}(\omega), \mathit{Good}(\Omega_i)\})$ is a subjective matter. It is specified using the same device as used for *Fair* except that the data is fed in by hand "until the distribution appears satisfactory". To start this process first identify those δ that "Π would be never accept" — they are given a probability of ≈ 0.0, and second those δ that "Π would be delighted to accept" — they are given a probability of ≈ 1.0. The *Me* proposition links the *ME* approach with "private valuations" in *GT*.

The whole "accept an offer" apparatus is illustrated in Fig. 1. The in-flow of information from the Internet, the market and from the opponent agents is represented as $\mathcal{I}_t$ and is stored in the knowledge base $\mathcal{K}$ and belief set $\mathcal{B}$. In that Figure the $\Box$ symbols denote probability distributions as described above, and the $\circ$ symbol denotes a single value. The probability distributions for $\mathit{Me}(\delta)$, $\mathit{Suited}(\omega)$ and $\mathit{Fair}(\delta)$ are derived as described above. *ME* inference is then used to derive the sentence probability of the $\mathbf{P}(\mathit{Accept}(\delta) \mid \mathcal{I}_t)$ predicate from the sentence probabilities attached to the *Me*, *Suited*, *Good* and *Fair* predicates. This derivation is achieved by two chunks of knowledge and two beliefs. Suppose that Π's "principles of acceptability" require that:

$\kappa_5 : (\mathit{Me} \wedge \mathit{Suited} \wedge \mathit{Good} \wedge \mathit{Fair}) \rightarrow \mathit{Accept}$
$\kappa_6 : (\neg \mathit{Me} \vee \neg \mathit{Suited}) \rightarrow \neg \mathit{Accept}$

these two statements are represented in $\mathcal{K}$, and there are 19 possible worlds as shown in Table 1. Suppose that Π believes that:

$\beta_5 : (\mathit{Accept} \mid \mathit{Me} \wedge \mathit{Suited} \wedge \neg \mathit{Good} \wedge \mathit{Fair});\ \mathbf{B}(\beta_5) = 0.1$
$\beta_6 : (\mathit{Accept} \mid \mathit{Me} \wedge \mathit{Suited} \wedge \mathit{Good} \wedge \neg \mathit{Fair});\ \mathbf{B}(\beta_6) = 0.4$

these two beliefs are represented in $\mathcal{B}$. Then for β_5: $\mathbf{P}(\mathit{Accept} \mid \mathit{Me} \wedge \mathit{Suited} \wedge \neg \mathit{Good} \wedge \mathit{Fair}) = \frac{\mathit{Accept} \wedge \mathit{Me} \wedge \mathit{Suited} \wedge \neg \mathit{Good} \wedge \mathit{Fair}}{\mathit{Me} \wedge \mathit{Suited} \wedge \neg \mathit{Good} \wedge \mathit{Fair}} = \frac{v_{14}}{v_{14} + v_{15}}$, and so: $9 \times v_{14} - v_{15} = 0$. Likewise for β_6: $3 \times v_{16} - 2 \times v_{17} = 0$.

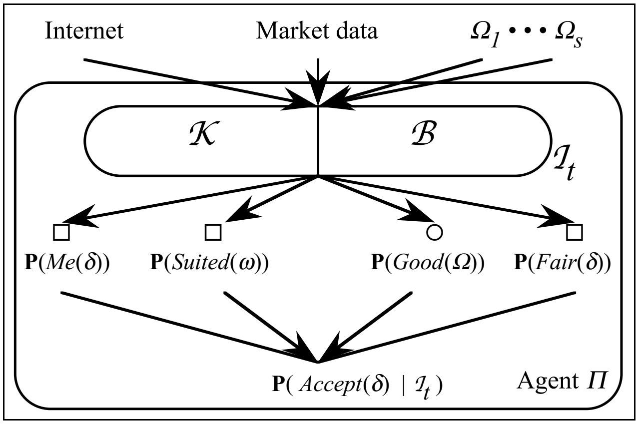


Fig. 1. Acceptability of a deal

Table 1. v_i is the agent's degree of belief that the i'th possible world is the actual world

$\mathbf{V}_{\mathcal{K}}$	*Me*	*Suited*	*Good*	*Fair*	*Accept*
v_1	$\perp$	$\perp$	$\perp$	$\perp$	$\perp$
v_2	$\perp$	$\perp$	$\top$	$\perp$	$\perp$
v_3	$\perp$	$\perp$	$\perp$	$\top$	$\perp$
v_4	$\perp$	$\perp$	$\top$	$\top$	$\perp$
v_5	$\perp$	$\top$	$\perp$	$\perp$	$\perp$
v_6	$\perp$	$\top$	$\top$	$\perp$	$\perp$
v_7	$\perp$	$\top$	$\perp$	$\top$	$\perp$
v_8	$\perp$	$\top$	$\top$	$\top$	$\perp$
v_9	$\top$	$\perp$	$\perp$	$\perp$	$\perp$
v_{10}	$\top$	$\perp$	$\top$	$\perp$	$\perp$
v_{11}	$\top$	$\perp$	$\perp$	$\top$	$\perp$
v_{12}	$\top$	$\perp$	$\top$	$\top$	$\perp$
v_{13}	$\top$	$\top$	$\top$	$\top$	$\top$
v_{14}	$\top$	$\top$	$\perp$	$\top$	$\top$
v_{15}	$\top$	$\top$	$\perp$	$\top$	$\perp$
v_{16}	$\top$	$\top$	$\top$	$\perp$	$\top$
v_{17}	$\top$	$\top$	$\top$	$\perp$	$\perp$
v_{18}	$\top$	$\top$	$\perp$	$\perp$	$\top$
v_{19}	$\top$	$\top$	$\perp$	$\perp$	$\perp$

The *ME* inference process is rather opaque — it is difficult to look at Eqn. 3 and guess the answer. In an attempt to render the inference digestible [3] uses a Bayesian net to derive $\mathbf{P}$(*Accept*). In contrast, the derivation is achieved here using *ME*. Table 2 contains some reassuring sample values. The extreme cases in which the values of the probabilities for the four evidence variables is either 1.0 or 0.0 are self-evident. The cases in which one of these values is 0.9 differ because β_5 tempers the *Good* case, and β_6 the *Fair* case rather more so. The case when the evidence probabilities are all 0.5 gives 0.129 for the probability when there is null information about the evidence. If the agent had no beliefs at all, ie: $\mathcal{I}_t$ only contained κ_5 and κ_6 then $v_i = \frac{1}{19}$, and $\mathbf{P}$(*Accept*) = $\frac{4}{19}$.

Table 2. Sample values derived for **P**(*Accept*)

P(*Me*)	**P**(*Suited*)	**P**(*Good*)	**P**(*Fair*)	**P**(*Accept*)
1.000	1.000	1.000	1.000	1.000
0.900	1.000	1.000	1.000	0.900
1.000	0.900	1.000	1.000	0.900
1.000	1.000	0.900	1.000	0.910
1.000	1.000	1.000	0.900	0.940
0.950	0.950	0.950	0.950	0.834
0.900	0.900	0.900	0.900	0.688
0.500	0.500	0.500	0.500	0.129
0.000	0.000	0.000	0.000	0.000

The *Accept* predicate generalizes the notion of utility. If $\mathcal{I}_t$ contains (*Me* $\leftrightarrow$ *Accept*) then **P**(*Accept*) = **P**(*Me*). Then define **P**(*Me*(π, ω)) to be: $0.5 \times (\frac{\mathbf{U}(\omega)-\mathbf{U}(\pi)}{\mathbf{U}(\overline{\omega})-\mathbf{U}(\pi)} + 1)$ for $\mathbf{U}(\omega) > \mathbf{U}(\pi)$ and zero otherwise, where: $\overline{\omega} = \arg\max_{\omega}\{\mathbf{U}(\omega) \mid (\pi,\omega) \in \mathcal{D}\}$.[4] An acceptability threshold α of 0.5 will then accept deals for which the surplus is non-negative. In this way *Accept* represents utility-based negotiation with a private valuation.

5 Auctions

The *ME* analysis of auctions focuses on what agents actually do rather than their reasons for doing what they do. The four common auction mechanisms are considered for an auctioneer, Υ, a single item and multi-issue bids each consisting of a set of deals. In the Dutch auction the auctioneer calls out successive sets of deals until one bidding agent shouts "mine". In the first- and second-price, sealed-bid mechanisms, bidding agents submit any number of multi-issue bids. The "Australian" mechanism is a variant of the common English mechanism in which agents alternately bid successive sets of deals until no further bids are received — as each set of deals is received the auctioneer identifies the current winning bid. So, unlike in the multi-issue English mechanism, in the Australian mechanism the auctioneer is not required to publicize fully her winner determination criterion in advance, and the bidders are not required to submit successive bids that are increasing with respect to that criterion. In the two sealed-bid mechanisms and the Australian mechanism the auctioneer determines the winner — and the runner up in the second-price mechanism — using a preference ordering on the set of all possible deals that may be made known to the bidding agents. The bids in these auctions may contain a large number of deals which is rather impractical.

Consider what happens from the auctioneer's point of view. Υ's expectation of what might happen will rely on both an understanding of the motivations and strategies of the agents taking part, and the rules of the auction mechanism. These two matters will effect

[4] The annoying introduction of $\overline{\omega}$ may be avoided completely by defining **P**(*Me*(π, ω)) = $\frac{1}{1+\exp(-\beta\times(\mathbf{U}(\omega)-\mathbf{U}(\pi))}$ for some constant β. This is the well-known sigmoid transfer function used in many neural networks. This function is near-linear for $\mathbf{U}(\omega) \approx \mathbf{U}(\pi)$, and is concave, or "risk averse", outside that region. The transition between these two behaviors is determined by the choice of β.

Υ's choice of deal set, but otherwise the analysis is the same for the four common auction mechanisms. Suppose that there are S agents, $\{\Omega_i\}_{i=1}^S$, bidding in the auction, and the value set, $\mathcal{D} = \{\delta_i\}_{i=1}^D$, contains D elements. Suppose that Υ has a total preference ordering, $\gtrsim_\Upsilon$, on the deal set, $\mathcal{D}$, and that $\mathcal{D}$ is labeled such that if $i > j$ then $\delta_i \gtrsim_\Upsilon \delta_j$. Let the predicate *TopBid*(Ω, δ) now mean "deal δ is the highest bid that Ω will make with respect to the order $\gtrsim_\Upsilon$". There are $S \times D$ ground literals in terms of this predicate. This predicate will satisfy: $\kappa_7 : \forall ixy((\mathit{TopBid}(\Omega_i, x) \wedge \mathit{TopBid}(\Omega_i, y)) \rightarrow (x = y))$. Suppose that the deal set, $\mathcal{D}$, has been chosen [see Sec. 3] so that Υ expects each of the $(D+1)^S$ possible worlds that are consistent with κ_7 to be equally probable for *TopBid*(.) for each Ω_i for $i = 1, \ldots, S$.[5] The maximum entropy distribution is uniform and $\forall ij \mathbf{P}(\mathit{TopBid}(\Omega_i, \delta_j)) = \frac{1}{D+1}$. Let the predicate *WinningBid*(δ) mean "deal δ is the highest bid that the $\{\Omega_i\}_{i=1}^S$ will make with respect to the order $\gtrsim_\Upsilon$". Then: $\kappa_8 : \forall i(\mathit{WinningBid}(\delta_i) \leftrightarrow (\neg \exists jk \mathit{TopBid}(\Omega_j, \delta_k) \wedge (k > i)) \wedge (\exists n \mathit{TopBid}(\Omega_n, \delta_i))$. There are now $(S \times D) + D$ ground literals in terms of these two predicates, but still only $(D+1)^S$ possible worlds. So:

$$\mathbf{P}(\mathit{WinningBid}(\delta_i)) = \left(1 - \frac{D-i}{D+1}\right)^S \times \left(1 - \left(\frac{i}{i+1}\right)^S\right) \tag{4}$$

For example, if $S = 2$ and $D = 3$ then the probability of the highest of the three possible deals being bid by at least one of the two agents is $\frac{7}{16}$. If the total ordering $\gtrsim_\Upsilon$ is established by a utility function then this result enables the estimation of the expected utility.[6] The analysis completed so far may be applied to any sealed-bid auction, or to any open-cry auction prior to any bids being placed. Once the bidding starts in an open-cry auction, information about what agents are, or are not, prepared to bid is available. This information may alter a bidding agent's assessment of the acceptability of a deal by feeding into the *Fair*(.) predicate — see Sec. 4. It also alters the assessments of the probabilities of what the various opponents will bid, and of any deal being the winning bid. Bids made in an Australian auction provide lower limits, and bids not made in a Dutch auction provide upper limits, to what the opponents will bid.[7] As these limits change the assessment of these probabilities are revised using Eqn. 3. A formula for $\mathbf{P}(\mathit{WinningBid}(\delta_i))$ in terms of these limits is rather messy.[8] The value derived for

[5] This is the *symmetric* case when the expected performance of each of the S bidding agents is indistinguishable.

[6] In the continuous *GT* analysis, if X_i is a random variable representing the amount bid by Ω_i, and if the distributions for the X_i are uniform on $[0, 1]$ then the expected value of the winning bid is given by the expected value of the Sth order statistic $\mathbf{E}(X_{(S)}) = \frac{S}{S+1}$.

[7] This is the *asymmetric* case.

[8] In the continuous *GT* analysis, given a sample of S non-identical, independent random variables $\{X_i\}_{i=1}^S$ where X_i is uniform on $[C_i, 1]$. For each sample, $p_i = \mathbf{P}(X_i \geq X) = \frac{1-X}{1-C_i}$ if $C_i \leq X \leq 1$ and zero otherwise. So the probability that *none* of the X_i exceed $Y \geq \max\{C_i\}$ is $\mathbf{P}(Y) = \prod_{j=1}^S (1 - p_i) = \prod_{j=1}^S \frac{Y - C_j}{1 - C_j}$ which is the probability distribution function for the largest Y. Then $\mathbf{E}(Y) = \int_{Y=\max\{C_i\}}^{Y=1} Y \times f(Y) \times dY$ where $f(Y) = (\prod_{j=1}^S \frac{Y-C_j}{1-C_j}) \times (\sum_{i=1}^S \frac{1}{Y-C_i})$. For example, for $S = 2$, $C_1 = c$, $C_2 = d$, $0 \leq c \leq d \leq 1$ then $\mathbf{E}(Y) = \frac{4-(3\times d)-d^3+(3\times c\times(d^2-1))}{6\times(1-c)\times(1-d)}$, and if $c = d = 0$ then $\mathbf{E}(Y) = \frac{2}{3}$ as we expect.

$\mathbf{P}(\textit{WinningBid}(\delta_i))$ relies on κ_7 and κ_8 in $\mathcal{K}$, together with expressions of the observed limits and the assumed expectation that each possible world is equally probable for *TopBid*(.).

Now consider the four auctions from a bidding agent's point of view. Two strategies, $\mathbf{S}$, for bidding agents are described for illustration only. First, a *keen agent* who prefers to trade on any acceptable deal to missing out — they are not primarily trying to optimize anything — although in the Australian auction they may choose to bid strategically, and may attempt to reach the most acceptable deal possible. Second, a *discerning agent* who attempts to optimize expected acceptability, and is prepared to miss out on a deal as a result.

First, consider keen agents. In a first-price, sealed-bid auction these agents will bid the entire set $\{\delta \mid \mathbf{P}(\textit{Accept}(\delta) \mid \mathcal{I}_t) \geq \alpha\}$. In an Australian, open-cry auction these agents agent may attempt to submit bids that are just "superior" to the bids already submitted by other agents. The meaning of "superior" is determined by $\gtrsim_\Upsilon$ and may be private information. If a bidding agent does not know $\gtrsim_\Upsilon$ then it will have to guess and assume it. Suppose that Δ is the set of bids submitted so far by the opponents in an Australian auction. First define the set of bids that are just superior to Δ: $\Delta^+ = \{\delta \in \mathcal{D} \mid \delta \notin \Delta, \exists \delta_1 \in \Delta, \delta \gtrsim_\Upsilon \delta_1, \forall \delta_2((\delta \gtrsim_\Upsilon \delta_2 \gtrsim_\Upsilon \delta_1) \rightarrow ((\delta_2 = \delta) \vee (\delta_2 = \delta_1)))\}$. Now bid $\{\arg\max_\delta\{\mathbf{P}(\textit{Accept}(\delta) \mid \mathcal{I}_t) \mid (\mathbf{P}(\textit{Accept}(\delta) \mid \mathcal{I}_t) \geq \alpha) \wedge (\delta \in \Delta^+)\}\}$. To avoid bidding against itself in a Vickrey auction an agent will bid a set of deals that forms a *shell*, Σ, with respect to $\gtrsim_\Upsilon$ [ie: $\forall \delta_i \delta_j \in \Sigma(\neg(\delta_i \gtrsim_\Upsilon \delta_j))$]. An agent will only bid in a Vickrey auction if $\gtrsim_\Upsilon$ is known, because that ordering will determine the "highest" non-winning bid. This uncertainty makes the Vickrey auction less attractive to keen agents than the other three forms. If keen agents do *not* feed bidding information into their acceptability mechanism in the open-cry cases, then the expected revenue will be greatest in the first-price, sealed-bid, followed by the Dutch and then by the Australian — it is not clear how the Vickrey auction fares due to the uncertainty in it. Feeding bidding information into the acceptability mechanisms of keen agents may have an inflationary effect on expected revenue in an Australian auction, and bidding non-information may have a deflationary effect in the Dutch auction. The extent to which these effects may change the expected-revenue ordering will be strategy-specific.

Second, consider discerning agents. A similar analysis to the result in Eqn. 4 may be used by a discerning agent to optimize expected acceptability in the symmetric case. This analysis follows the general pattern of the standard *GT* analysis for utility optimizing agents — see for example [15] — it is not developed here. For a discerning agent, the Vickrey mechanism has a dominant strategy to bid at, and the Australian mechanism right up to, the acceptability margin. For the Dutch and first-price mechanisms, the acceptability of the deals bid will be shaded-down from the margin. In both the Dutch and the Australian mechanisms, the margin of acceptability may move as bidding information becomes available.

5.1 Take-it-or-Leave-it

The take-it-or-leave-it mechanism is a degenerate auction in that an agent makes a single bid that stands until it is withdrawn. An opponent agent may then choose to accept a standing bit. Further, some popular auctions, such as eBay, offer vendors the facility of a "Buy Now" option. To use this option the vendor has to determine a take-it-or-leave-it price. The case of one buyer and one seller is considered here. Introduce the predicate $\textit{WillTrade}(\Omega, \delta)$ meaning "that agent Ω will accept a proposed deal δ". Then an "acceptability optimizing" Π, with information $\mathcal{I}_t$, will offer Ω the deal: $\arg\max_\delta(\mathbf{P}(\textit{WillTrade}(\Omega, \delta)) \times \mathbf{P}(\textit{Accept}(\delta) \mid \mathcal{I}_t))$. For multi-issue δ the distribution for $\mathbf{P}(\textit{WillTrade}(\Omega, \delta))$ is evaluated using Eqn. 2, and the distribution for $\mathbf{P}(\textit{Accept}(\delta) \mid \mathcal{I}_t)$ using the method in Sec. 4.

The single-issue case is analyzed to illustrate the "*ME* method" and because it gives a different value to *GT*. Suppose that a seller has a good that she values at r and wishes to determine a take-it-or-leave-it price. First assume that the single buyer Ω will prefer to pay less than more: $\kappa_9 : \forall xy((\delta_x > \delta_y) \rightarrow (\textit{WillTrade}(\Omega, \delta_x) \rightarrow \textit{WillTrade}(\Omega, \delta_y)))$. Second, following Sec. 3, choose the intervals $\mathcal{D} = \{\delta_i\}_{i=1}^{D}$ such that $\mathbf{W}_{\{\{\kappa_9\},\mathcal{B}\}}$ is uniform, where $\mathcal{D}$ is ordered naturally. Then $\mathbf{V}_\mathcal{K}$ contains $D+1$ possible worlds for the predicate $\textit{WillTrade}(\Omega, \delta)$ for $\delta \in \mathcal{D}$, and $\mathbf{P}(\textit{WillTrade}(\Omega, \delta_i)) = \frac{i}{D+1}$. Suppose that the seller knows the additional information that Ω will pay δ_y and will not pay δ_n. Then $\mathcal{K}$ now contains two further sentences: $\kappa_{10} : \neg\textit{WillTrade}(\Omega, \delta_n)$ and $\kappa_{11} : \textit{WillTrade}(\Omega, \delta_y)$. There are now $n - y$ possible worlds, the maximum entropy distribution is uniform, and using Eqn. 2: $\mathbf{P}(\textit{WillTrade}(\Omega, \delta_i)) = \frac{n-i}{n-y}, \quad y \leq i \leq n$. In general the seller's expected surplus in offering the deal δ to agent Ω is: $\mathbf{P}(\textit{WillTrade}(\Omega, \delta)) \times (\mathbf{U}(\delta) - r)$, where $\mathbf{U}(\delta)$ is the utility as in Sec. 2.1. In the continuous case, the "expected utility-optimizing price" is $\frac{\mathbf{U}(\delta_n)+r}{2}$ — this price is in terms of only the seller's valuation r and the knowledge $\neg\textit{WillTrade}(\Omega, \delta_n)$ — it is independent of the knowledge $\textit{WillTrade}(\Omega, \delta_y)$. Both *ME* and *GT* assume κ_9. In the *GT* analysis [15], the expected utility optimizing price is: $\frac{\overline{u}+r}{2}$ where $\overline{u}$ is the upper bound of an assumed uniform distribution for Ω's utility. The *GT* analysis relies on that assumption. The *ME* analysis relies on the observation $\textit{WillTrade}(\Omega, \delta_y)$ and shows that the price is: $\frac{\mathbf{U}(\delta_n)+r}{2}$. It is no surprise that these expressions have a similar structure. However they do not have the same value. Ω may be aware of her utility, u_Ω, for the good. The inherent inefficiency of bilateral bargaining [9] shows for an economically rational Ω that u_Ω, and so consequently $\overline{u}$, may be greater than $\mathbf{U}(\delta_n)$. Further, δ_n may be a "high" offer and $\overline{u}$ may be less than $\mathbf{U}(\delta_n)$. It is unlikely that they will be equal.

6 Conclusions

Auctions have been considered from the point of view of agents that bid because they feel comfortable as a result of knowledge acquisition, rather that being motivated by expected utility optimization. Information is derived generally from the World Wide Web, from market data and from observing the behavior of other agents in the market. The agents described do not make assumptions about the internals of their opponents. In competitive negotiation, an agent's motivations should be kept secret from its opponents. So speculation about an opponent's motivations necessarily leads to an endless counter-speculation spiral of questionable value. These agents require a method of uncertain reasoning that can operate on the basis of a knowledge base that contains first-order statements qualified with sentence probabilities. Maximum entropy inference is eminently suited to this requirement, and has the additional bonus of operating with logical constants and variables that represent individual deals. So the deals may be multi-issue. Four simple multi-issue auction mechanisms have been analyzed for two classes of agent: keen agents that are primarily motivated to trade, and discerning agents that are primarily motivated by the optimization of their expected acceptability. The acceptability mechanism generalizes game theoretic utility in that acceptability is expressed in terms of probabilities that are dynamically revised during a negotiation in response to both changes in the background information and the opponents' actions.

References

1. P. Bogg. http://www-staff.it.uts.edu.au/ plbogg/negotiation/demos/maxent/.
2. J. Castro-Schez, N. Jennings, X. Luo, and N. Shadbolt. Acquiring domain knowledge for negotiating agents: a case study. *Int. J. of Human-Computer Studies (to appear)*, 2004.
3. J. Debenham. Bargaining with information. In *Proceedings Third International Conference on Autonomous Agents and Multi Agent Systems AAMAS-2004*, July 2004.
4. J. Halpern. *Reasoning about Uncertainty*. MIT Press, 2003.
5. M. Jaeger. Representation independence of nonmonotonic inference relations. In *Proceedings of KR'96*, pages 461–472. Morgan Kaufmann, 1996.
6. E. Jaynes. Information theory and statistical mechanics: Part i. *Physical Review*, 106:620–630, 1957.
7. P. Klemperer. *The Economic Theory of Auctions : Vols I and II*. Edward Elgar, 2000.
8. D. MacKay. *Information Theory, Inference and Learning Algorithms*. Cambridge University Press, 2003.
9. R. Myerson and M. Satterthwaite. Efficient mechanisms for bilateral trading. *Journal of Economic Theory*, 29:1–21, 1983.
10. J. Paris. Common sense and maximum entropy. *Synthese*, 117(1):75–93, 1999.
11. S. D. Pietra, V. D. Pietra, and J. Lafferty. Inducing features of random fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(2):380–393, 1997.
12. S. Ramchurn, N. Jennings, C. Sierra, and L. Godo. A computational trust model for multi-agent interactions based on confidence and reputation. In *Proceedings 5th Int. Workshop on Deception, Fraud and Trust in Agent Societies*, 2003.
13. T. Sandholm and N. Vulkan. Bargaining with deadlines. In *Proceedings of the National Conference on Artificial Intelligence (AAAI)*, 1999.
14. H. Varian. Mechanism design for computerized agents. In *Proceedings Usenix Workshop on Electronic Commerce*, pages 13–21, July 1995.
15. E. Wolfstetter. *Topics in Microeconomics*. Cambridge UP, 1999.

Multi-attribute Bilateral Bargaining in a One-to-Many Setting

E.H. Gerding[1], D.J.A. Somefun[1], and J.A. Han La Poutré[1,2]

[1] Center for Mathematics and Computer Science (CWI),
P.O. Box 94079, 1090 GB Amsterdam, The Netherlands
[2] Eindhoven University of Technology, School of Technology Management,
P.O. Box 513, 5600 MB Eindhoven, The Netherlands
{egerding, koye, hlp}@cwi.nl

Abstract. Negotiations are an important way of reaching agreements between selfish autonomous agents. In this paper we focus on one-to-many bargaining within the context of agent-mediated electronic commerce. We consider an approach where a seller negotiates over multiple interdependent attributes with many buyers individually. Bargaining is conducted in a bilateral fashion, using an alternating-offers protocol. In such a one-to-many setting, "fairness," which corresponds to the notion of envy-freeness in auctions, may be an important business constraint. For the case of virtually unlimited supply (such as information goods), we present a number of one-to-many bargaining strategies for the seller, which take into account the fairness constraint, and consider multiple attributes simultaneously. We compare the performance of the bargaining strategies using an evolutionary simulation, especially for the case of impatient buyers and small premature bargaining break off probability. Several of the developed strategies are able to extract almost all the surplus; they utilize the fact that the setting is one-to-many, even though bargaining occurs in a bilateral fashion.

1 Introduction

It is common to characterize negotiations by their setting: bilateral, one-to-many, or many-to-many. In this paper we focus on the one-to-many bargaining setting, where a seller agent negotiates, on behalf of a seller, with many buyers individually in a *bilateral* fashion. We develop various strategies for this setting which enable a seller agent to bargain over *multiple interdependent* issues simultaneously and effectively.

In many cases, auctions can be used to effectively organize one-to-many bargaining. Depending on the setting, auctions can provide buyers with the incentive to reveal their preferences truthfully, and to allocate the goods efficiently. For various situations, however, auctions may not be the preferred protocol to bargainers. In situations of, for example, virtually unlimited supply, multiple issues, continuous sale, and/or repeated sales to the same buyers the appropriate auction protocol becomes, at best, much more complex. Consequently, businesses may opt for the intuitive and flexible bilateral bargaining protocol, where the seller agent negotiates bilaterally with one or more buyers simultaneously by exchanging offers and counter offers. In many electronic commerce domains supply is flexible and new goods can be reproduced quickly, at relatively low

P. Faratin and J.A. Rodríguez-Aguilar (Eds.): AMEC 2004, LNAI 3435, pp. 129–142, 2005.

costs; especially in these domains businesses may opt for a bilateral bargaining protocol. The sales of information goods, with its virtually unlimited supply, provide a particularly good example of an application domain where businesses may opt for bilateral bargaining.

Potentially, bargaining can lead to unsatisfied customers if buyers perceive the outcomes of the negotiations as unfair. This can occur when, for instance, two customers obtain similar goods at the same time but end up paying very different amounts. Fairness of negotiation outcomes is important for customer satisfaction, which in turn may be important for a business' long term profitability. The seller agent can prevent unfair outcomes by incorporating a fairness norm, comparable to the notion of envy-freeness in auctions [1], whereby customers are treated in a similar fashion. This fairness aspect emphasizes that bargaining is really one-to-many.

The challenge is to develop bargaining strategies for the seller agent that maximize overall revenue by utilizing differences in buyers' willingness to pay without violating the fairness constraint. In this paper we focus on strategies that can utilize differences in customers' time pressure. For the case of virtually unlimited supply, as for information goods, we present a number of one-to-many bargaining strategies for the seller, which take into account the fairness constraint and bargain over multiple attributes. In order to benefit from time pressure, these strategies specify—in addition to an actual (counter) proposal or an acceptance proposal—when to respond to an opponent's proposal. We compare the performance of the bargaining strategies using an evolutionary simulation, especially for the case of impatient buyers and small (exogenous) probability (per negotiation round) of a customer breaking off the negotiations. One set of strategies, the so-called "responsive threshold" strategies, are able to extract almost all the surplus, given sufficient time pressure. These strategies benefit from the fact that the setting is one-to-many, even though bargaining occurs in a bilateral fashion. In addition, the strategies are able to find win-win agreements (i.e., very little Pareto improvement is possible).

A number of related papers study bargaining using an evolutionary approach, e.g. [2,3,4,5]. Our approach extends previous research to multiple (types of) buyers and bilateral negotiation strategies for one-to-many multi-issue bargaining which can benefit from time pressure. In addition a growing body of literature exists on multi-issue negotiation, which focus on developing techniques a seller and/or a buyer can use to determine the relative magnitude of the various issues and consequently search for approximately win-win (or Pareto-efficient) deals [6,7,8,9]. Although we also consider the problem of how to determine the values for the various attributes of an offer, the focus of the paper lies on the development of "threshold" strategies for one-to-many negotiation. These strategies determine the desired utility level of a deal and can be used in conjunction with the techniques already developed in the literature.

The remainder of the paper is organized as follows. In Section 2 we describe the bargaining setting and introduce strategies for one-to-many bargaining. In Section 3 we discuss the simulation environment used for testing the performance of the strategies. We present the simulation results of the conducted computer experiments in Section 4. Conclusions follow in Section 5.

2 One-to-Many Bargaining

2.1 Fairness

An agent representing a business can be endowed with various bargaining strategies. The bargaining outcome should, however, be fair. Because fairness must be ensured by the seller and because buyer preferences are private, we define fairness as follows. Suppose at time t_d a buyer reaches a deal. We say that this deal is fair, relative to a fixed interval $\Delta > 0$, whenever there exist a start time t_s, with $t_d \in [t_s, t_s + \Delta]$, such that the seller is indifferent between any other deal reached within the interval $[t_s, t_s + \Delta]$. Whenever price is the only issue, a buyer does not strictly prefer any deal for which the seller is indifferent. In this case, we can give the following equivalent definition: a deal is fair, relative to a fixed interval $\Delta > 0$, whenever there exist a start time t_s, with $t_d \in [t_s, t_s + \Delta]$, such that the buyer does not strictly prefer any other deal which is reached within the interval $[t_s, t_s + \Delta]$. Note, that this definition of fairness is closely related to the notion of envy-free auctions in [1]; it adapts the notion of envy-freeness to the more continuous setting of bilateral bargaining.

2.2 Bargaining Protocol

The seller agent negotiates with many buyer agents simultaneously in a bilateral fashion by alternating offers and counter offers. An offer specifies a value for each attribute of the negotiation, such as the price, quality, quantity, and other relevant aspects. The protocol allows for multiple offers to be submitted simultaneously. Exchanging multiple offers can improve the (Pareto) efficiency of agreements made when several attributes are concerned. An offer constitutes a Pareto improvement over another offer whenever it makes one bargainer better off without making the other worse off. A bargainer proposing multiple offers can be indifferent between those offers whereas his opponent may prefer a particular offer and can improve efficiency by selecting this offer.

We call the set of offers combined with the preconditions a *proposal*. A bargainer can accept one of the submitted offers or reject all offers and place a counter proposal. Negotiations between a buyer and seller agent proceed to the next round whenever a proposal is submitted and terminates when one of the submitted offers is accepted or after a predefined period of time has elapsed. Note that a bargainer can introduce a delay before submitting a counter proposal. The duration of a round varies depending on the delay. Figure 1 depicts the alternating offer bargaining protocol.

2.3 Time Pressure

An important assumption is that buyers are impatient and prefer an early agreement. Time pressure or time impatience is a common assumption in bargaining, e.g. [10]. The seller agent is simultaneously and continuously negotiating with many buyers and is therefore less concerned with immediately reaching an agreement for a particular bargaining outcome, i.e., he is relatively patient. We model this relative time patience by assuming that the seller, unlike the buyers, has no *direct* time pressure: i.e., the seller is indifferent between selling now and later. In the experiments we do, however, consider a small exogenous probability (per negotiation round) of a customer breaking

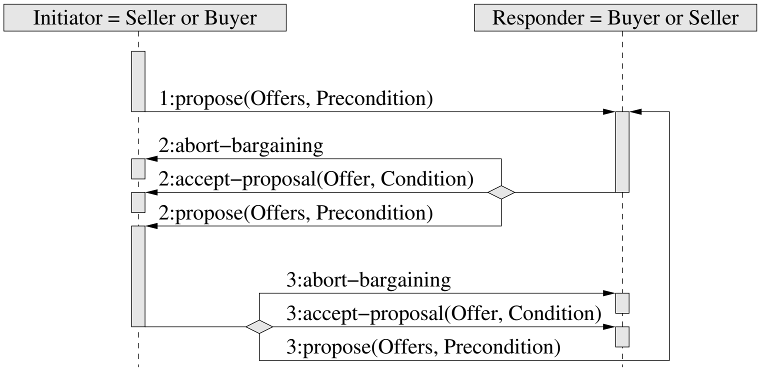


Fig. 1. The agents' bargaining protocol

off the negotiation process; therefore the seller has an indirect incentive to speed up the negotiations (all else equal).

At least in theory, the seller can benefit from buyers' time-pressure by introducing a delay before submitting a counter proposal. An important question is then which bargaining strategies can most effectively utilize these potential benefits. Experimental results discussed in Section 4 show that responsive threshold strategies, which we will discuss in the next Section, are very effective: depending on the time pressure, they are capable of extracting very large shares of the seller surplus. (Reasoned from the seller's perspective the surplus is just the maximum utility he can realize by selling the goods or services.)

2.4 One-to-Many Bargaining Strategies

The challenge is to develop bargaining strategies for the seller agent that maximize overall revenue by utilizing differences in buyers' willingness to pay without violating the fairness constraint. Instead, these strategies utilize differences indirectly through buyers' time pressure. In order to benefit from time pressure, a strategy specifies, in addition to an actual (counter) proposal or an acceptance proposal, when to respond to an opponent's proposal.

The seller strategies as developed determine the offers of a proposal in two steps. First, they specify a threshold level which sets the utility level of the offers. Second, they generate the values for the individual attributes, given the threshold. Advanced techniques for multi-issue negotiation, such as discussed in [6,7,8,9], can be applied to the latter. The focus of this paper is on effective strategies for determining the threshold; we therefore only consider the relatively simple technique of randomly determining the attribute-values given a threshold. The probability with which a value of an attribute is determined may however depend on a buyer's corresponding offer (see below).

Besides specifying the utility of a (counter) proposal, the threshold is also used to determine when to respond to an outstanding proposal. More precisely, a seller strategy responds with a fixed delay to all outstanding proposals which lie below the (current) threshold value; a proposal lies below the threshold value whenever the seller's utility for all offers in the proposal lies below the threshold value. While applying a delay to all proposals below the threshold value, the seller agent continues to negotiate with the remaining buyers by immediately responding with a counter proposal.

This negotiation without delay with the select group of buyers can, in principle, continue for several rounds. During these rounds the threshold is not adjusted. The goal at this time is to improve the Pareto-efficiency of the final agreement by finding mutually beneficial trade offs between the various attributes. In the simulation the seller agent only makes a single proposal to improve the efficiency of the deal. For a particular proposal of a buyer the seller strategy randomly generates offers within the neighborhood of the buyer's best offers, i.e., with the highest utility for the seller. This already suffices for very efficient outcomes. If a buyer does not accept one of the seller agent's offers, the seller agent will again respond with delay.

Another aspect that needs to be considered by the seller agent is the fairness of the agreements. Fairness prescribes that the seller should be indifferent between the deals made within the defined time interval. Whenever the seller agent almost simultaneously accepts two different offers a bargaining outcome may be unfair. The seller strategy ensures fairness by always making a (interesting) counter proposal, instead of accepting an offer directly. The seller agent can be equipped with a number of strategies for determining the threshold, which we introduce below.

Fixed and Time-Dependent Threshold Strategies. For purpose of comparison we introduce a fixed threshold strategy. Clearly, the fixed threshold strategy is not capable of utilizing buyers' time pressure. The purpose of the strategy is to provide some insights in the minimal extractable profit, given strategic behavior of the buyers.

The second strategy we consider is a time-dependent threshold strategy: the current utility or threshold depends on time. The threshold only changes from one period to the next. Unlike the fixed-threshold strategy the time-dependent strategy is capable of utilizing buyers' time pressure. Its success, however, depends on how much it knows about buyers' preferences, or how easily more about buyers' preferences can be learned, in relation to time-based pricing strategies.

Responsive Threshold Strategies. The fixed and time-dependent threshold strategies do not adjust the threshold based on the buyers' offers. Inspired by the first-price auction, we introduce another type of bargaining strategy with a responsive threshold. With this strategy, all offers submitted by the buyers within a certain fixed time interval are collected after the previous offers made by the seller agent. Then it determines the *current highest utility*, which is equal to the utility of the best offer from the collection of offers. The threshold is set to the current highest utility.

The success of the responsive threshold strategy does not depend on some (a priori) knowledge of buyer preferences, unlike the fixed and time-dependent strategies. Intuitively, buyers who— due to time pressure— suffer more from delay are inclined to bargain less "hard-headed" than other buyers. Consequently, these buyers may reach a deal sooner and pay a higher price. Thus, at least potentially, the strategy is capable of utilizing buyers' time pressure without requiring (a priori) knowledge of buyer preferences. Unlike auctions, actual bargaining occurs in an alternating exchange of offers and counter offers, typically initiated by a buyer. Parties bargain over the price and other relevant aspects of the provided good or service. Even though the seller agent's strategy can be auction-inspired, buyers will be unaware of this fact. They do not know the opponent's bargaining strategy on forehand; they perceive the bargaining process as bilateral. Buyers may of course suspect some relationship with other ongoing negotiations. The point is that unlike a true auction the relationship with other simultaneously submitted offers is not specified up front, through a set of rules.

Reservation Value. A drawback of the responsive threshold strategy is that it becomes vulnerable whenever groups of buyers experience very little time pressure. Without time pressure buyers have no incentive to buy soon. They may all independently decide to initially submit very low offers; consequently utility will be very low for the seller. To circumvent this we also consider responsive threshold strategies with a reservation value. A seller agent is never willing to sell below the reservation value. This means we alter the earlier definition of the current highest utility. It now becomes the maximum of the reservation value and the best offer from the offers collected within a certain time interval. An interesting advantages of introducing a reservation value occurs when some but not all buyers experience very little time-pressure. The responsive threshold strategy can then still utilize the time-pressure of the other buyers.

We consider two approaches for determining the reservation value. Either the reservation value is fixed, like the fixed-threshold strategy, or it is time dependent, like the time-dependent threshold strategy. Thus the responsive threshold strategy with a reservation value is actually a combination of the responsive threshold strategy (without reservation value) and either the fixed or time-dependent strategies.

3 Bargaining Simulation Environment

We apply a simulation environment in order to evaluate the performance and robustness of the above negotiation strategies against many learning buyers. The agents in the simulation are assumed to be boundedly rational: they can learn and adapt their strategies by a process of trial and error, and they do not know the seller's strategy. The bargaining process is repeated many times, enabling buyers and the seller to learn from past interactions. An evolutionary algorithm is used to model the learning aspect of the agents. This is a common approach within the field of agent-based computational economics (ACE) [11]. A number of related papers study bargaining using an evolutionary approach, e.g. [2,3,4,5]. Our approach extends previous research to multiple (types of) buyers and bilateral negotiation strategies for one-to-many multi-issue bargaining which can benefit from time pressure.

3.1 The Bargaining Game

The seller agent negotiates with many buyer agents simultaneously by alternating offers and counter offers as described in Section 2.2, where the buyer agents initiate the negotiations. For our simulations we set a maximum number of n discrete periods, where n is set sufficiently large such that it has no significant impact on the results. For the analysis we assume that offers consist of two interdependent attributes, e.g. the price and the quality. We note that buyer agents in the simulation may leave the negotiations prematurely (due to a bargaining break off probability) but do not enter later. We also assume that, since buyers are impatient, buyer agents in the simulation will respond to the seller agent's counter offers without delay. This is modeled by having the buyer's counter proposal or acceptance proposal occur in the same period as the seller's proposal.

3.2 Buyers and Their Agents

Buyers are interested in buying at most one good in each bargaining game. They can have different preferences regarding the time pressure and attribute value combinations, which together constitute the buyer *type*. For the analysis we assume a finite number of k types. Although k is fixed, the number of participating buyer agents of each type varies randomly for each negotiation game and is determined independently by a Poisson distribution with average λ.

To illustrate the feasibility of our approach for interdependent attributes, we use the well-known Cobb-Douglas utility function to represent a player's preferences for the two attributes. More specifically, the utility u_i for buyer type i in case of a disagreement equals zero and in case of an agreement u_i is defined as

$$u_i = (v_{1,i} - o_1)^{\alpha_i}(v_{2,i} - o_2)^{\beta_i}\delta_i^t,$$

where α_i and β_i are parameters that indicate the relative importance of the attributes; o_1 and o_2 are the negotiated values the seller receives for the attributes (and the buyer has to give in); and $v_{1,i}$ and $v_{2,i}$ represent the maximum buyer i's is willing to give in on the individual attributes. For example, let attribute 1 and 2 refer to price and quality. Then o_1 represents the price and o_2 the difference between the maximum quality and the actual quality of the good received; $v_{1,i}$ then represents the maximum price buyer i is willing to pay and $v_{2,i}$ the maximum buyer i is willing to give in on the quality. Furthermore, δ_i is the discount factor used to model the time pressure, and t is the negotiation time. In the simulation depreciation occurs at discrete time intervals. Therefore, δ is the discrete representation of time pressure and t indicates the period in which an agreement is reached. Note that discount factors are commonly used for modeling time pressure, e.g. in the Rubinstein-Ståhl alternating-offers model [10].

Buyer Agent's Strategy. The buyer agents in the simulation apply time-dependent strategies similar to the seller's time-dependent threshold strategy described in Section 2.4. The buyer agent also uses an analogous (random) strategy for determining the values of the attributes given the threshold. The time-dependent strategy consists of a piece-wise linear function to determine the threshold. The parameters that determine the function are adaptive: using an evolutionary algorithm they evolve such that the performance of the strategy increases.

We also applied an extended strategy in our experiments by using two separate piece-wise linear functions: one produces the threshold for determining the utility level of the offers and the other function determines the threshold for accepting or rejecting the seller's offers. The separation of the two functions enhances the bargaining capabilities of the buyer agent. Results using the two representations are very similar. The outcomes presented in this paper are based on the extended strategy.

3.3 Seller Agent

The seller agent bargains with a number of buyers simultaneously, without knowing the type of these buyers. The seller agent's utility in case of an agreement equals $u_s = o_1^{\alpha_s} o_2^{\beta_s}$, and is zero in case of a disagreement (recall from Section 2.3 that the we can assume the seller has no time pressure). The total utility equals the sum of utilities obtained over all buyers. Production costs are set to zero.

We consider five strategies for the seller agent: fixed threshold, time-based threshold, responsive threshold and two combined strategies (see also Section 2.4). The time-based threshold strategy is similar to the strategy used by the buyer. The first two strategies and the combined strategies have parameters which determine respectively the threshold value and the reservation value during a bargaining game. These parameters are *adaptive*: optimal values are learned using an evolutionary algorithm, explained below. The responsive threshold strategy does not have any parameters that need to be learned.

3.4 The Evolutionary Algorithm

Evolutionary algorithms (EAs) are a class of search algorithms inspired by Darwin's theory on variation and natural selection, and are becoming increasingly popular for modeling economic behavior, particularly within the field of agent-based computational economics (ACE), see e.g. [11]. We use an implementation based on "evolution strategies" [12], which is typical for real-valued encoding of the strategies (whereas the more popular branch of "genetic algorithms" is originally based on binary encoding).

The EAs are used to produce effective bargaining strategies for the buyer agents. Strategies for the agents of different buyer *types* are produced by separate EAs, which operate in parallel. This allows for heterogeneous strategies to emerge. Furthermore, in case of an adaptive seller agent, a separate EA is also used to produce strategies for the seller agent. A graphical representation of the evolutionary simulation with two buyer types and an adaptive seller agent is given in Fig. 2.

Each EA starts with a population of *parent* strategies, which are randomly generated. The EA then performs the following cycle to improve the quality or *fitness* of the strategies. First, the reproduction operator generates a population of *offspring* strategies by randomly selecting strategies from the parent population and slightly mutating the strategy to obtain variation.

In the next step, the fitness of the strategies is determined by the average utility obtained in a number of bargaining games. At the start of each bargaining game, the number of participating buyers of each type is determined randomly using a Poisson distribution as described above. Buyer agents are then generated for each buyer and are assigned a randomly selected strategy from either the parent or offspring population of the corresponding type. Similarly, a strategy is selected randomly for the seller agent (in case of an adaptive seller). The bargaining game is played for a fixed number of times, determining the number of buyers and assigning new strategies at the start of each game.

In the final stage of the cycle, a deterministic selection scheme called $(\mu + \lambda)$-selection chooses the strategies with the highest fitness from both the parents and the offspring populations as the new parents for the next generation [12]. The cycle is repeated for a fixed number of generations.

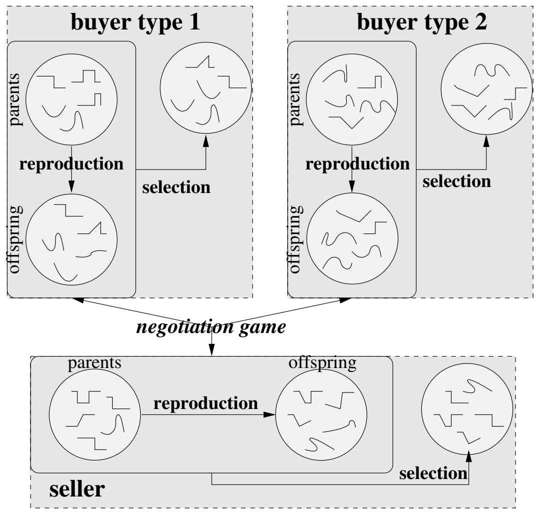


Fig. 2. The EA cycle for negotiations with two buyer types and an adaptive seller

Strategy Encoding. As mentioned in Section 3.2, the buyer agent's strategy consists of two piece-wise linear functions: an offer and a threshold function. The functions are encoded using real values, where each bending point of a function is encoded by two real values (i.e., the period and the corresponding threshold value). Additionally, two end points mark the values for the first and last period. For example, 8 real values are needed to encode a pair of functions with two line pieces each.

The same representation is used for the seller agent if he uses a time-based threshold strategy. If a fixed threshold is used, only a single real value is needed to encode this. Note that the seller agent uses the same function for both the threshold and for producing offers.

Mutation with Exponential Decay. The mutation operator changes the strategy of an agent as follows. Each real value x_i is mutated by adding a zero-mean Gaussian variable with a standard deviation σ [12]: $x_i' := x_i + \sigma N_i(0, 1)$. All resulting values larger than unity (or smaller than zero) are set to unity (respectively zero). In our simulations, we use a model of exponentially decaying standard deviations. This approach ensures convergence and is analogous to simulated annealing, where a temperature parameter determines the variation of the solution. A half-life parameter determines the number of generations that the mutation standard changes to half the value.

4 Experimental Results

4.1 Settings

In this paper we report on two series of experiments with either a 0% or a 1% exogenous customer break off probability: a break off probability of 1% means that at every new round there is a 1% probability of an active customer breaking off (or leaving) the current bargaining game. The following settings are used for both series of experiments. (We note that also experiments are carried out using other settings, e.g. with a different number of participating buyers and buyer valuations, resulting in very similar outcomes, but are omitted due to space limitations.) Buyers are grouped into three types ($k = 3$), each type having adaptive bargaining strategies evolving in separate populations. The time pressure (discount factor) for each type is set as a control parameter. The values $v_{1,i}$ and $v_{2,i}$, and the parameters α_i and β_i are randomly generated from a uniform distribution at the beginning of each experiment, such that $v_{1,i}, v_{2,i} \in [100, 300]$ and $\alpha_i, \beta_i \in [0.7, 0.9]$. A buyer furthermore has a *minimum threshold value*, which is a minimum acceptable utility and is fixed at 10% of the most favourable utility (i.e., the utility u_i when $o_1 = o_2 = 0$, see Section 3.2).

The piece-wise linear functions of the buyer agents, and of the seller agent in case of time-based threshold strategy, consist of two line pieces. The number of buyers of each type participating in a bargaining game is determined randomly by a Poisson distribution with the average $\lambda = 10$. Buyers and sellers produce 3 offers in each round, which are randomly selected given a threshold value. However, when the seller produces counter offers without delay to improve Pareto efficiency (see Section 2.4), the seller generates 5 offers in the vicinity of the buyer's best offers. The length of a bargaining game is set to 40 periods.

The EA settings are chosen such that results are robust and the EAs are able to find good solutions. All buyer types use equal settings, with 20 strategies in the parent populations and 20 offspring strategies. The mutation standard deviation (see Section3.4) is initially set to 0.2, and decays with a half-life value of 50 generations. The EA settings for the seller are the same, except that each seller population only contains 10 strategies. Buyers have larger populations because more buyers than sellers participate each game, and because in case of the extended buyer strategy (with two functions) the search space for the buyer is larger (a higher population size is often recommended for larger search spaces). The fitness of the strategies for a single generation is determined by 100 bargaining games. For these settings the EAs are able to find almost optimal solutions for simple test cases.

4.2 Results

The results reported in this Section are obtained after a process of learning, when the strategies have converged. It is important to note that, during learning, the preferences of the buyers remain unchanged, although the number and composition (i.e., number of each type) of buyers can differ in each bargaining game. Experiments are run for 40000 bargaining games (400 generations). Results are averaged over the last 1000 bargaining games of an experiment, and over 30 experiments, accounting for random settings such as the number of participating buyers and the buyer's preferences.

Figure 3 compares—for a break off probability of (a) 0% (on the left) and (b) 1% (on the right)—the obtained fraction of the total *seller surplus* for different seller threshold strategies and buyer discount factors (buyers have equal discount factors). We define the seller surplus of a bilateral negotiation as the seller's maximum feasible utility, i.e., when the buyer offers her minimum threshold value and the offer is Pareto-efficient. As shown in Fig. 3, the fixed threshold strategy (1) is able to extract around 75% of

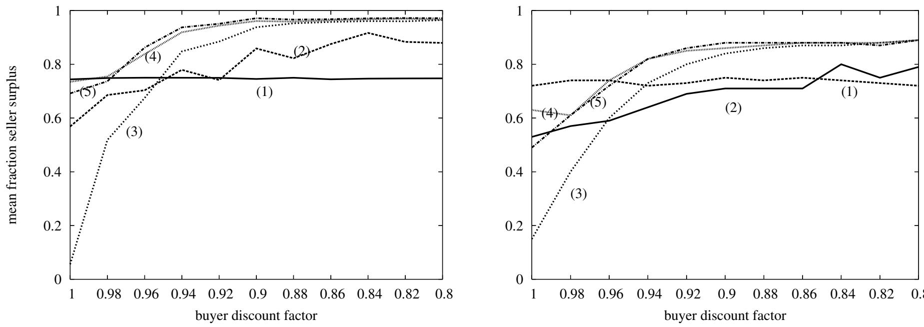


Fig. 3. Seller's obtained fraction of total surplus, with (a) 0% (on the left) and (b) 1% (on the right) break off probability, using 5 threshold strategies: (1) fixed threshold, (2) time-dependent threshold, (3) responsive threshold, (4) combined (3) and (1), and (5) combined (3) and (2)

the seller surplus. The outcomes are relatively independent of the break off probability; this is because almost all deals will be closed in the first or second round (the average round of agreement lies between 0.14 and 0.24 in case of time pressure). Note that these outcomes are independent of the discount factor. Clearly, the fixed threshold strategy is unable to benefit from the buyers' time pressure.

The time-based threshold strategy (2), on the other hand, shows that higher profits can be obtained if the threshold changes over time, see Fig. 3a. Buyers with a high valuation will purchase relatively early, since waiting for a better deal does not compensate the loss due to time discounting. Buyers with a low valuation, on the other hand, have the incentive to reach an agreement in a later stage if they can get a better price for it. This way the seller can indirectly discriminate between buyers with different valuations and time pressures. The performance of the time-based threshold strategy (2) is, however, vulnerable to an increase in the bargaining break off probability (see Fig. 3b).

Note that with no time discounting (i.e., when $\delta = 1$) the fixed threshold strategy performs better. This is due to the difference in strategy complexity: only a single value needs to be optimized in case of a fixed threshold, whereas an entire function (encoded by 4 values) needs to be learned in case of the time-based threshold. This is clearly more difficult, especially within a dynamic environment with learning buyers.

Outcomes using the responsive threshold bargaining strategies (see Fig. 3 (3),(4), and (5)) show an impressive increase in the fraction of surplus when buyers are impatient. If the time pressure becomes sufficiently high, the seller obtains almost the entire surplus. Even for lower time pressure, results are much better for the seller compared to the fixed and time-based threshold strategies. The consequence of increasing the bargaining break off probability from 0% to 1% is that buyers' time pressure needs to be a bit higher before the threshold strategies will dominated the other strategies and the maximal attainable performance drops a bit. This drop in performance is, however, mainly caused by lower sales due to the premature departure of customers; thus the seller's bargaining position does not change fundamentally.

For the case of no or very low time pressure, the results also show that simple auction-like mechanisms such as the responsive threshold strategy are not sufficient in case of unlimited supply. Without competition between buyers, the market price goes

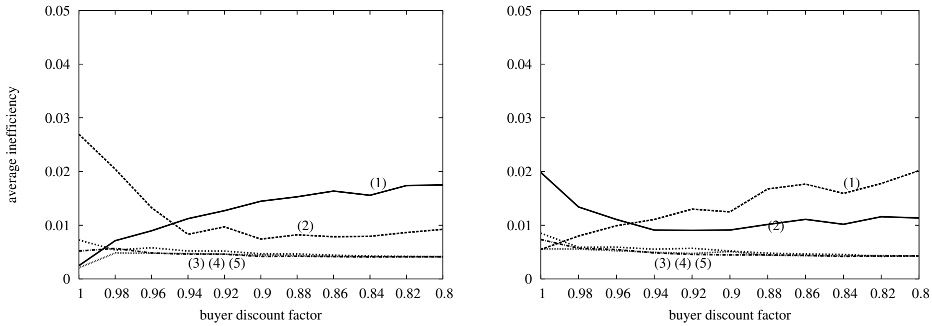


Fig. 4. Average inefficiency of the agreements using different strategies with (a) 0% (on the left) and (b) 1% (on the right) break off probability. The numbers correspond to the seller threshold strategies in Fig. 3.

to cost level, resulting in a zero surplus for the seller. This problem can be resolved by combining the responsive threshold strategy with an adaptive reserve value. As shown in Fig. 3, this results in very good outcomes, even if buyers are very patient. This makes the combined strategy very versatile. We note that these outcomes also generalize to settings where buyer types have different time preferences, assuming that buyers with higher valuation have a higher time pressure. The outcomes are not shown here due to space limitations.

The mean (in)efficiency of the obtained bargaining results is depicted in Fig. 4. The inefficiency is measured as the seller's maximum Pareto-improvement of a given outcome divided by the seller's actual utility plus the improvement. The outcomes show low inefficiencies for all strategies (the highest inefficiency is around 2.7% of the total utility). However, the responsive threshold strategies result in relatively the most Pareto-efficient deals. Unlike the other strategies, the responsive strategies set the threshold exactly to the best offer. Since this is the buyer's best offer, it is already quite efficient. In case of the other strategies, however, the utility of the offers usually exceed the seller's threshold, and the seller first needs to map the buyer's offers to the right utility level when making counter offers. This mapping results in additional inefficiencies of the outcomes. Even with a reasonably simple strategy for determining the relative magnitude of the attribute values, already good results are found. The Pareto-efficiency is expected to improve even further by incorporating more advanced strategies as described in e.g. [7,6,8,9]. This is however left for future work.

4.3 Bargaining Revisited

A possible strategy of the buyer agent is to bid very low, and then accept the counter offer of the seller. Such a strategy could be beneficial in case the seller's counter offer is influenced by the buyers' offers, as with the responsive threshold strategies. This could then result in low profits for the seller. To see if indeed buyers benefit from such a strategy, the strategy representation for buyers was extended by using two separate functions: one produces the threshold for determining the utility level of the offers and the other function determines the threshold for accepting or rejecting the seller's offers (see Section 3.2).

Even with separated function, however, the responsive threshold strategy performs very much in favor of the seller (as shown by the results). This occurs because the counter offer is delayed by the seller whenever offers fall below the (seller's) threshold, hence providing the buyers with an incentive to try and get an agreement without delay.

5 Concluding Remarks

In this paper, we consider strategies for a seller agent who negotiates with many buyers simultaneously in a bilateral fashion over multiple interdependent attributes. These strategies respect a notion of fairness such that buyers are treated similarly. An important aspect of the developed strategies is their ability to benefit from impatient buyers that prefer early agreements. Buyers can have different valuations and time preferences. A buyer's actual valuation and time preference is only known to himself (i.e., a buyer's type constitutes private information).

The strategies introduced determine three aspects: a threshold, multi-attribute offers with a utility level corresponding to the threshold, and a scheme for determining when to respond. Five different threshold strategies for the seller agent are evaluated and compared: (1) fixed threshold, (2) time-dependent threshold strategies, (3) responsive, (4) responsive with fixed reservation value, and (5) responsive with time-dependent reservation value. The last two strategies are actually a combination of the responsive threshold strategy with the first two strategies.

We use an evolutionary simulation to analyze the performance of the different strategies when the buyers and seller bargain over two interdependent attributes. The buyers' bargaining strategies adapt and learn through the use of an evolutionary algorithm (EA). The seller's strategies (1) and (2), and the combined strategies (4) and (5) also adapt and learn using an EA. The responsive threshold strategy (3), on the other hand, determines the threshold value based exclusively on the offers received by the buyers, and does not require any learning.

The outcomes show that bilaterally exchanging multiple offers combined with a random offer generation mechanism suffices for closely approximating Pareto-efficiency. Furthermore, given a small probability (per negotiation round) of a customer breaking of the negotiations the responsive threshold strategies appear to be very successful in utilizing time pressure and consequently extract a very high share of the surplus. For sufficiently high time pressure, the seller obtains almost all surplus, indicating that buyers submit and/or accept offers close to their reservation value. Thus buyers self-select to pay their valuation, while the bargaining outcomes respect our notion of fairness. The results also show superior performance of the combined strategies (4 and 5) compared to the auction-inspired strategy (3), in case some or all buyers have very little time pressure. In other words, the combined strategy is very versatile.

References

1. Goldberg, A.V., Hartline, J.D.: Envy-free auctions for digital goods. In: Proceedings of the 4th conference on Electronic commerce, ACM, ACM (2003) 29–35
2. Gerding, E., van Bragt, D., Poutré, J.L.: Multi-issue negotiation processes by evolutionary simulation: Validation and social extensions. Computational Economics **22** (2003) 39–63

3. Fatima, S., Wooldridge, M., Jennings, N.R.: Comparing equilibria for game theoretic and evolutionary bargaining models. In: Proc. 5th Int. Workshop on Agent-Mediated E-Commerce, Melbourne, Australia. (2003) 70–77
4. Phelps, S., McBurney, P., Parsons, S., Sklar, E.: Co-evolutionary auction mechanism design: A preliminary report. In Padget, J., Shehory, O., Parkes, D., Sadeh, N., Walsh, W., eds.: Agent-Mediated Electronic Commerce IV: Designing Mechanisms and Systems. Volume 2531 of Lecture Notes in Computer Science. Springer-Verlag, Berlin (2002) 123–142
5. Cliff, D.: Explorations in evolutionary design of online auction market mechanisms. Electronic Commerce Research and Applications **2** (2003) 162–175
6. Klein, M., Faratin, P., Sayama, H., Bar-Yam, Y.: Negotiating complex contracts. Group Decision and Negotiation **12** (2003) 111–125
7. Faratin, P., Sierra, C., Jennings, N.R.: Using similarity criteria to make issue trade-offs. Journal of Artificial Intelligence **142** (2003) 205–237
8. Ehtamo, H., Hämäläinen, R.P.: Interactive multiple-criteria methods for reaching pareto optimal agreements in negotiations. Group Decision and Negotiation **10** (2001) 475–491
9. Somefun, D., Gerding, E., Bohte, S., Poutré, J.L.: Automated negotiation and bundling of information goods. In Rodriguez-Aguilar, J.A., Faratin, P., Walsh, W.E., Parkes, D., eds.: Workshop on Agent Mediated Electronic Commerce V: Designing Mechanisms and Systems. Volume 3048 of Lecture Notes in AI., Heidelberg, Springer-Verlag (2004)
10. Rubinstein, A.: Perfect equilibrium in a bargaining model. Econometrica **50** (1982) 97–109
11. Tesfatsion, L.: (Guest Editor), Special issue on agent-based computational economics. Journal of Economic Dynamics and Control **25** (2001)
12. Bäck, T.: Evolutionary Algorithms in Theory and Practice. Oxford University Press, New York and Oxford (1996)

Bidding for Customer Orders in TAC SCM

David Pardoe and Peter Stone

The University of Texas at Austin, Austin TX 78712, USA
{dpardoe, pstone}@cs.utexas.edu

Abstract. Supply chains are a current, challenging problem for agent-based electronic commerce. Motivated by the Trading Agent Competition Supply Chain Management (TAC SCM) scenario, we consider an individual supply chain agent as having three major subtasks: acquiring supplies, selling products, and managing its local manufacturing process. In this paper, we focus on the sales subtask. In particular, we consider the problem of finding the set of bids to customers in simultaneous reverse auctions that maximizes the agent's expected profit. The key technical challenges we address are i) predicting the probability that a customer will accept a particular bid price, and ii) searching for the most profitable set of bids. We first compare several machine learning approaches to estimating the probability of bid acceptance. We then present a heuristic approach to searching for the optimal set of bids. Finally, we perform experiments in which we apply our learning method and bidding method during actual gameplay to measure the impact on agent performance.

1 Introduction

Supply chains are a current, challenging problem for agent-based electronic commerce. One problem commonly faced by agents acting in supply chains is that of negotiating with customers in order to sell goods. Such negotiations are often handled through reverse auctions in which sellers submit sealed bids in response to requests for quotes (RFQs) from customers. This situation becomes particularly difficult when sellers must bid in multiple auctions simultaneously, because an agent cannot await the outcome of one auction before bidding in another. When deciding which auctions to bid in and what bids to place, an agent with limited resources must be able to judge and balance the competing risks of not winning enough auctions and of winning too many. In the former case, it is unable to fully utilize its resources towards profitability; in the latter, it will be unable to meet its obligations to customers.

The Trading Agent Competition Supply Chain Management (TAC SCM) scenario [1] provides a perfect testbed for the study of this problem. In TAC SCM, agents competing as computer manufacturers must handle three basic subtasks: acquiring components, managing a local manufacturing process, and selling assembled computers to customers. Agents receive incomplete information about the state of the game and have a limited amount of time in which to make decisions, resulting in a challenging competition. The problem studied in this paper is motivated by our work on TacTex [2], the third place entry from the

P. Faratin and J.A. Rodríguez-Aguilar (Eds.): AMEC 2004, LNAI 3435, pp. 143–157, 2005.

first TAC SCM competition. From our experience, we have identified the sales subtask as the most crucial aspect of the TAC SCM scenario.

In this paper, we focus on the problem of determining the optimal set of bids for an agent to make in response to RFQs for computers received from customers. The key technical challenges we address are i) predicting the probability that a customer will accept a particular bid price, and ii) searching for the most profitable set of bids.

The remainder of this paper is organized as follows. In Sect. 2 we give a brief summary of the TAC SCM scenario and provide information on related work. We give a complete description of the problem we are solving in Sect. 3. In Sect. 4 we present a comparison of several machine learning approaches to estimating the probability of bid acceptance. We describe a heuristic approach to finding an optimal set of bids in Sect. 5. In Sect. 6 we measure the impact of learning on agent performance by performing controlled experiments involving actual TAC SCM games. Sect. 7 proposes directions for future work and concludes.

2 Background

In this section, we give a brief summary of the TAC SCM scenario, emphasizing the parts that are most relevant to the sales subtask, and provide information on related work.

2.1 The TAC SCM Game

In a TAC SCM game [3], six agents act as computer manufacturers in a simulated economy that is managed by a game server. The length of a game is 220 simulated days, with each day lasting 15 seconds of real time. At the beginning of each day, agents receive messages from the game server with information concerning the state of the game, such as the customer RFQs for that day. Agents have until the end of the day (i.e. $< 15s$) to send messages to the server indicating their actions for that day, such as bids on RFQs. The game can be divided into three parts: production and delivery, component supply, and computer demand.

In this paper, we focus on the computer demand, or sales, aspect of the TAC scenario. Customers wishing to buy computers send all six agents identical RFQs consisting of:

- the type of computer desired (1 of 16);
- the quantity of computer desired (1–20);
- the due date (3-12 days in the future);
- a reserve price indicating the maximum amount the customer will pay; and
- a penalty that must be paid for each day the delivery is late. Orders are canceled on the fifth late day.

Reserve prices range from 75% to 125% of the base price of the requested computer type, multiplied by the quantity, and penalties range from 5% to 15% of the reserve price. The base price of a computer is equal to the sum of the base prices of its parts [3]. Agents respond to the RFQs by making offers to sell at a certain price, with the agent offering the lowest bid on each RFQ winning the order. Agents are unable to see the prices offered by other agents or even the winning prices, but they do receive a report each day indicating the highest and lowest price at which each type of computer sold on the previous day.

The number of RFQs that come from customers depends on the level of customer demand, represented by a parameter D. The actual number of RFQs each day is drawn from a Poisson distribution with D as its mean. Fluctuation in demand is modeled by multiplying D by an amount representing the current trend each day. This trend follows a random walk, and D is bounded between 80 and 320, with its initial value chosen uniformly randomly from this range.

2.2 Related Work

The problem of predicting the probability of winning an auction with a particular sealed bid is commonly approached through statistical methods such as those surveyed in [4]. Such methods often require extensive historical information about competitors' past bids and assume a static environment. In TAC SCM, probabilities vary considerably throughout the game, and almost no information is available about competitors' bids while the game is running. A machine learning approach similar to that used in this paper is developed by [5], which uses a naive Bayes classifier to predict the probability of a bid winning based on the bid price, features of the RFQ , and available information about other bidders.

A solution to the TAC SCM bidding problem similar to the one used in this paper is presented in [6], which uses linear regression on recent bidding results to form predictions of bid acceptance and then uses stochastic programming to determine optimal bids. Additional approaches are described in [7] and [8].

3 Problem Specification

We now specify the problem we are addressing in this paper. We consider the problem of an agent participating in a TAC SCM game that must decide what bids to place on the RFQs it has received from customers on a given day. The inputs to the agent's decision process are the following:

- The set of customer RFQs;
- The agent's available resources (components and assembled computers in inventory along with the future production cycles); and
- Information about past auctions (the agent's knowledge of its own bids and the reported highest and lowest prices at which each type of computer sold)

Because there are many more computers requested each day than one agent can produce, the goal of an agent is not to win every auction, but to find the set of bids that maximizes the agent's expected profit without committing the agent to produce more computers than it possibly can. (Viewing TAC as a game, an agent's goal should be to maximize its profit relative to the profits of competing agents, but due to the difficulty of determining the effect an agent's bids will have on other agents, we will assume our agent is only concerned with its own profit. In a real supply chain, this profit maximization would be the true goal.) A simple approach to this problem is to predict the highest price at which each auction could be won and to bid this price on several of the more profitable auctions, expecting to win each one. A more sophisticated approach involves considering the possibility of placing high bids on many auctions in hopes of winning some fraction of them.

This second approach is the one used by TacTex, our agent in the first TAC SCM competition, and is the approach that is considered in this paper. An agent implementing this approach has two requirements: the ability to form estimates of the probability of winning an auction as a function of the bid price, and a means of using these estimates to find the set of bids that maximizes the agent's expected profit. We consider these two agent components separately in the next two sections. In Sect. 4, we experiment with different machine learning approaches to predicting the probability of bid acceptance, and in Sect. 5, we present a heuristic approach to the problem of bid selection.

4 Learning Auction-Winning Probabilities

Predicting the probability of winning an auction in TAC SCM is a challenging problem for three main reasons: (i) agents receive very limited information on auction results, (ii) no two auctions are the same due to the differing attributes of each RFQ, and (iii) winning prices can fluctuate rapidly due to changing game conditions. As a result, an approach based on analyzing past auction results from the current game is unlikely to yield accurate predictions. We therefore turn to machine learning methods using training data from many past games.

The problem we are trying to solve can be viewed as a multiple regression problem. This could be solved by using a regression learning algorithm to learn the probability of winning an auction as a function of factors including the bid price. We instead follow a modified approach used by [9] to solve a similar conditional density estimation problem from a different TAC scenario. This approach involves dividing the price range into several bins and estimating the probability of winning the auction at each bin endpoint. A post-processing step converts the learned set of probabilities to a probability density function by interpolating between bin endpoints and enforcing a monotonicity constraint that ensures that probabilities decrease as prices increase. In this method, a separate predictor is trained for each endpoint to predict the probability of winning at that point. The concept to be learned by each predictor is therefore simpler than the concept that would be learned if we used a single predictor for all prices. We leave an empirical comparison with the latter approach for future work.

In this section we focus on the task of training these individual predictors. We describe the format of the training data, compare the effectiveness of several learning algorithms, and then look at the impact that the choice of training data has on the predictions. It is important to note that training is done off-line, so the game's time constraints are not a factor.

4.1 Training Data Format

The data for our experiments is taken from the results of the semifinal and final rounds of the first TAC SCM competition held in August 2003. Winning bids for customer RFQs can be obtained from game logs made available immediately after each game terminates. Several hundred thousand RFQs were issued over the course of the games, providing ample data for training and testing.

A training instance is created for each RFQ. The 23 attributes included in each instance reflect the details of the RFQ it represents, along with the information available to agents at the time about the level of demand in the game and the recent prices for which the requested type of computer has been sold. Each instance contains the current date; the quantity, penalty, due date, and reserve price for the RFQ; and the highest and lowest prices at which the requested computer type was sold over the past five days. The additional attributes provided about customer demand give a picture of how the daily number of RFQs has varied over the course of the game. All monetary values are expressed as a fraction of the computer's base price.

A separate predictor is trained for each price point x at which we want to predict the probability of winning an auction, where x is expressed as a fraction of a computer's base price. For a given value of x, each auction is labeled with a 1 if the winning bid was greater than x and with a 0 otherwise. Instances representing RFQs receiving no bids are labeled with a 1 if x is less than or equal to the reserve price.[1]

4.2 Algorithm Comparison

We first performed an experiment comparing the effectiveness of using several different regression learning algorithms to train predictors: neural networks (with a single hidden layer and using backpropagation), M5 regression trees, M5 regression trees boosted with additive regression (which successively fits a new base learner to the residuals left from the previous step), decision stumps (single-level decision trees) boosted with additive regression, J48 decision trees, J48 decision trees boosted with AdaBoost, and BoosTexter. BoosTexter [10] is a boosting method that was originally designed for text classification and is the algorithm used in [9]. It uses decision stumps as the base learner and a form of AdaBoost to weight each training instance between rounds of training, outputting a weighted sum of the learned decision stumps. Other algorithms we considered were support vector machines, naive Bayes, and k-nearest neighbors, but these did poorly in initial testing. For all algorithms other than BoosTexter, we used the implementations provided in the WEKA machine learning package [11], using default parameters. Informal attempts at tuning these parameters did not appear to significantly affect performance.

For comparison, we also include the results obtained from using a simple heuristic predictor that gives reasonably good results. For each of the past five

[1] Note that this formulation represents the standpoint of a seventh agent wanting to know the probability that none of the other six agents would place a bid below x.

days, the predictor forms a uniform density function on the interval between the highest and lowest prices reported for the requested computer type. A weighted sum of these density functions, with those from more recent days receiving more weight, is then used as a probability density function from which estimates of bid acceptance are taken.

In our experiment we evaluated each of the learning algorithms on a data set taken from the final round of the competition. We used cross-validation, meaning that training and test data came from the same games. While the true value of a probability prediction is the utility gained from using it, determining this in the context of a TAC SCM agent is not feasible, and so we instead use root mean squared error between the predicted probabilities and actual outcomes as the measure of comparison. We ran separate tests predicting the probability of winning an auction at several different values of x. The results of three 10-fold cross-validations with $x = 0.7$ are presented in Fig. 1 and are similar to results for other values of x.

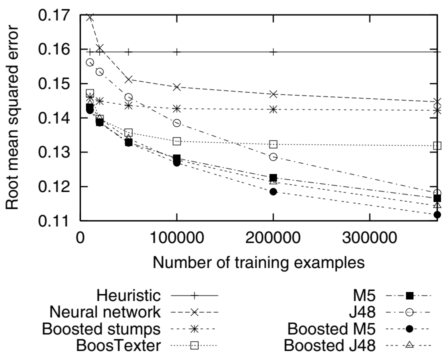


Fig. 1. Results for $x = 0.7$

With a large number of training instances, the tree-based methods clearly had the best performance, followed by BoosTexter. The errors of the non-tree-based methods level off after a limited number of training instances, while the errors of the tree-based methods continue to decrease until the point at which all available training data is used. For training sets of size 200,000 and 370,000, the difference observed between each pair of algorithms in Fig. 1 is statistically significant at the 95% confidence level.

4.3 Choice of Training Data

In the previous experiment, the training data and test data were taken from the same set of games, with the same agents participating in each game. This raises the possibility that the algorithms learned concepts that pertained to specific games and set of agents but were not applicable in general. This is also unrealistic in the TAC setting, as an agent could not have predictors trained on data from the game it is currently participating in. In practice, it is important to know whether a predictor trained for one set of agents will be reliable in games with a different set of agents. Our next experiment addresses these issues.

We consider the case of an agent participating in the final round of the competition. The agent would want to train predictors on the data most relevant to the situation. At the beginning of the final round, the most relevant data would likely come from the results of the semifinal round, which contained two brackets of six agents each. As a result, this data would reflect on both the agents in the final round and the agents defeated in the semifinal round. After the finals begin, the agent would be able to analyze the results of completed games from the finals and would have the option of retraining its predictors with this new data, either by itself or in combination with the data from the semifinals.

We performed an experiment comparing the results of training with these choices of training data. First, we divided the games from the final round into two halves, labeled finals1 and finals2. We then used finals2 as the test data for predictors trained on data from different sources: the semifinals, finals1, the semifinals combined with finals1, and finals2 (using cross validation). The results for M5 trees and BoosTexter, the top two performing algorithms, are shown in Figures 2 and 3. Again, $x = 0.7$. The learning curves are labeled with the source of data used for training.

When the predictors were trained on data other than finals2, the performance gap between M5 trees and BoosTexter disappeared, and the performance of the other tree-based methods, even boosted M5 trees, fell behind. The errors of the tree-based methods no longer continued to decrease as more training instances were used, and sometimes the error increased, as observed in Fig. 2 when data from the semifinals was used for training. This suggests that the strong performance of the tree-based methods in the first experiment was largely due to their ability to learn game-specific factors that do not generalize well. While BoosTexter appears to achieve somewhat lower errors than M5 trees in this experiment, further testing on different game scenarios would need to be done to determine whether this is the case in general.

As we expected, the predictors trained on data from finals2 outperformed the predictors trained on data from different games. Still, the performances of the latter were better than that of the heuristic. The predictors trained on finals1 performed better than those trained on the semifinals, confirming that more relevant training data produces better results. Somewhat surprisingly, the predictors trained on the combination of finals1 and the semifinals performed better than the predictors trained on finals1 only. It may be that a predictor trained on data from a variety of sources will generalize the best to a new situation, even if some of the training data is less relevant for the new situation.

The results of these experiments suggest that with the right choice of learning algorithm and training data, we can learn the probability of winning an auction reasonably well. However, to measure the value of our predictions, we need to

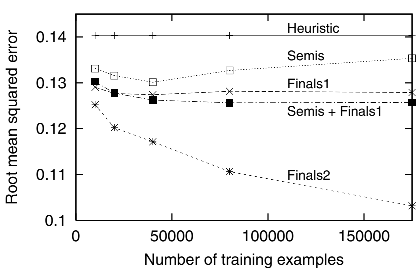


Fig. 2. M5 Trees

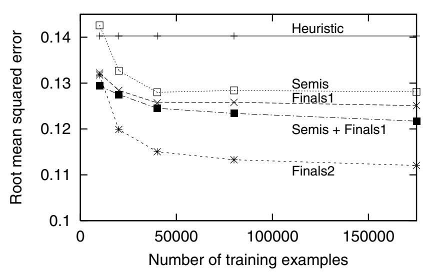


Fig. 3. BoosTexter

use them as the input to a method of selecting bids in actual TAC games. We present such a method in the next section, and then experimentally evaluate its performance in Sect. 6.

5 Bid Selection

We now consider the problem of bid selection. Recall that each day, customers send roughly 80-320 RFQs to all agents simultaneously, with each RFQ requesting a specific type and quantity of computer by a certain date. For each RFQ, the agent that bids lowest wins the order. In this section, we cast the bidding problem as an optimization problem and describe a heuristic approach to finding the optimal set of bids to offer in response to a single day's customer RFQs.

5.1 Problem Formulation

An agent's goal in selecting bids should be to maximize its total expected future profit. This value depends on the unknown strategies of competing agents, however, and the exact computation of this value would likely be intractable even if these strategies were known. As a result, we present a somewhat myopic agent that aims to maximize its profit only on computers due over the next 12 days (the range of due dates for computers requested on a given day) and that makes some simplifying assumptions. These assumptions are:

1. All computers delivered over the next 12 days will come from computers and components that are already in inventory or expected to be delivered during that period.
2. After the next 12 days, the average number of each computer type ordered from the agent per day will remain the same as the average over the past few days.
3. For the rest of the game, the agent will purchase only enough components to meet the need from the expected production in Assumption 2. The prices of these components will be the same as recently observed prices.
4. Computers held in inventory at the end of 12 days are of no value.
5. The agent is able to accurately predict the probability of winning an order given the bid price.

Assumption 1 means that our agent has a fixed set of resources to work with when selecting bids. This is a reasonable assumption due to the fact that our agent (and many other agents from the competition) tries to carry a large component inventory and tends to place long-term rather than short-term component orders. Because the prices paid for components in inventory are sunk costs, our agent will only consider replacement costs when determining the cost of producing a computer. These costs can be determined from Assumptions 2 and 3 by projecting future component use, deciding whether the components used over the next 12 days will need to be replaced, and determining how much this will cost per component. If the current inventory of a component exceeds the projected use, then that component's replacement cost will be zero. Assumption 4 means that our agent should be willing to use all computers in inventory and all upcoming production cycles for computers that will be delivered over the next 12 days. Assumption 5 simply tells us that our agent has access to the bid acceptance functions we tried to learn in the previous section.

As a result of these assumptions, our agent is essentially pretending that it is acting in a static environment, and this could lead to suboptimal behavior if future game circumstances change. For example, if computer prices are currently low due to low demand, but the trend of customer demand is increasing, then it might be wise to hold on to computers in inventory for later sale, violating assumption 4. The ability to more accurately predict the future values of components and computers would be valuable, but we leave this to future work.

The profit our agent obtains over the next 12 days depends not only on the RFQs being bid on on the current day, but also on RFQs that will be received on later days for computers due during the period. If we were to ignore these future RFQs when selecting the current day's bids, our agent might plan to use up all available production resources on the current RFQs, leaving it unable to bid on future RFQs. One way to address this issue would be to restrict the resources available to the agent for production of the computers being bid on currently. This is the method used by [6]. We instead take the approach of predicting the RFQs that our agent will receive for computers due during the period, and coming up with bids for these RFQs at the same time as the actual RFQs from the current day. Future RFQs are randomly generated according to the parameters given in the game specification and our current estimate of the level of customer demand and its trend. This has the effect of causing our agent to decide which resources to reserve for future RFQs, and limited testing suggests that our agent performs better when using this method than when we explicitly restrict the resources available.

5.2 Optimization Method

We now have an optimization problem with the following inputs:

- The agent's current computer orders
- The resources available to the agent over the next 12 days: production cycles, computers and components currently in inventory, and expected future deliveries of components
- A cost associated with each computer representing the expected replacement costs of its components
- A set of RFQs for computers due over the next 12 days, including both the current day's actual RFQs and predicted future RFQs.

Our goal is to find the set of bids that maximizes expected profit on these RFQs and existing orders. Those bids representing actual RFQs for the current day will then be offered to customers.

We make the assumption that we will always want our agent to fill existing orders if possible, and so the agent begins by scheduling the production necessary to fill these orders. This leaves our agent with a reduced set of resources and means that it only needs to concern itself with the expected profit from RFQs.

If we were considering only a single auction and had no resource constraints, the expected profit resulting from a particular bid price would be:

$$Expected\,profit = P(order|price) * (price - cost) \tag{1}$$

The optimal bid would be the value that maximized this quantity.

Computing the expected profit from a set of bids when resource constraints are considered is much more difficult, however, because the profit from each auction cannot be computed independently. For each possible outcome of the auctions in which it is not possible to fill all orders, the profit obtained depends on the agent's production and delivery strategy. For any nontrivial production and delivery strategy, precise calculation of the expected profit would likely require separate consideration of a number of possible auction outcomes that is exponential in the number of auctions. If we were guaranteed that we would be able to fill all orders, we would not have this problem. The expected profit from each auction could be computed independently, and we would have:

$$Expected\,profit = \sum_{i\ \epsilon\ all\ auctions} P(order_i|price_i) * (price_i - cost_i) \tag{2}$$

Our bidding heuristic is based on the assumption that the expected number of computers ordered for each RFQ will be the actual number ordered. In other words, we pretend that it is possible to win a part of an order, so that instead of winning an entire order with probability p, we win a fraction p of an order with probability 1. This assumption greatly simplifies the consideration of filling orders, since we now have only one auction outcome to consider, while leaving the formulation of expected profit unchanged. As long as it is possible to fill the partial orders, (2) will hold, where the probability term now refers to the fraction of the order won. It would appear that this approach could lead to unfilled orders when the agent wins more orders than expected, but in practice, this is not generally a problem. Most of the RFQs being bid on are the predicted RFQs that will be received on future days, and so the agent can modify its future bidding behavior to correct for an unexpectedly high number of orders resulting from the current day's RFQs. The agent can also set aside a small number of completed computers in inventory to serve as a buffer to prevent penalties in case any problems remain. When using this bidding strategy, our agent indeed tends to have very few late or missed deliveries.

By using this notion of partial orders, we can transform the problem of bid selection into the problem of finding the most profitable set of partial orders that can be filled with the resources available. Although this problem lends itself to standard optimization methods, our choice of method is constrained by the limit of 15 seconds per simulated game day and by the size of the problem (there may be over a thousand RFQs to consider). We use a greedy production scheduler that tries to utilize production resources as profitably as possible. All bids are initially set to be just above the reserve price, which means we begin with no orders. The production scheduler then chooses an RFQ and an amount to lower its bid by, resulting in an increased partial order for that RFQ. The scheduler

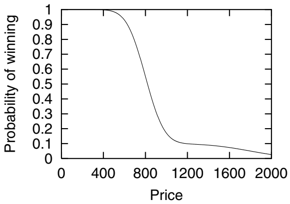


Fig. 4.

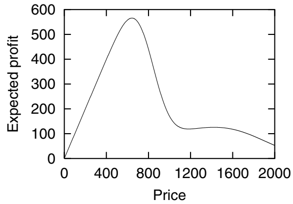


Fig. 5.

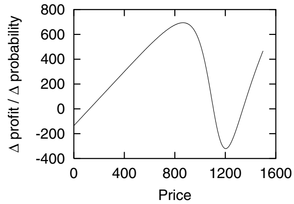


Fig. 6.

simulates filling this increase by scheduling its production as late as possible, taking completed computers from inventory if production is not possible. This process is repeated until no more production is possible or no bid can be reduced without reducing the expected profit.

Because we are working with resource constraints, the goal of the greedy production scheduler at each step is to obtain as large an increase in profit as possible while expending as few production resources as possible. To illustrate how this is done, consider Figures 4, 5, and 6. Fig. 4 represents the predicted probability of winning an auction as a function of the bid price, or alternatively, the fraction of the auction we assume we will win at each bid price. Fig. 5 shows the expected profit at each price, found using (1). Now suppose that our current bid is 1500, and we are considering lowering this bid. We would like to find the bid decrease that produces the largest increase in profit per additional computer ordered. This quantity is equal to $(Profit(x) - Profit(1500))/(Probability(x) - Probability(1500))$ for $x < 1500$ and is graphed in Fig. 6. From the graph, we can see that the optimal decision is to lower the bid to about 850. At each step, the production scheduler performs this analysis to find the bid reduction that will produce the largest increase in profit per additional computer for each RFQ, and chooses the RFQ for which this value is the largest.

In many cases, the most limited resource is production cycles. In such cases, the increase in profit per cycle used is a better measure of the desirability of a partial order than the increase in profit per additional computer, so we divide the latter quantity by the number of cycles required to produce the type of computer

Table 1. The bidding heuristic

- For each RFQ, compute both the probability of winning and the expected profit as a function of price
- Set the bid for each RFQ to be just above the reserve price
- Repeat until no RFQs are left in the list of RFQs to be considered:
  - For each RFQ, find the bid lower than the current bid that produces the largest increase in profit per additional computer ordered (or per additional cycle required during periods of high factory utilization)
  - Choose the RFQ and bid that produce the largest increase.
  - Try to schedule production of the partial order resulting from lowering the bid. If it cannot be scheduled, remove the RFQ from the list.
  - If the production was scheduled, but no further decrease in the bid will lead to an increase in profit, remove the RFQ from the list.
- Return the final bid for each RFQ.

requested by the RFQ and use the resulting values to choose which RFQ should be considered next. We consider cycles to be the limiting factor whenever the previous day's production used more than 90% of the available cycles to produce computers used to fill orders (as opposed to computers produced using spare cycles in order to build up inventory).

The range of possible bid prices is discretized for the sake of efficiency. Even with fairly fine granularity, this bidding heuristic produces a set of bids in significantly less time than the 15 seconds allowed per simulated game day. Attempts to use local search methods to improve the bids found yielded almost no increase in profit, suggesting at the very least that our greedy method tends to find local minima. The complete bidding heuristic is summarized in Table 1.

6 Agent Performance

In this section, we evaluate the effectiveness of our learning approach and bidding method when used as part of a complete agent in TAC SCM gameplay. We do this through a series of experiments in which agents using different combinations of bidding methods and prediction methods play against each other repeatedly.

6.1 Agent Design

For each agent, production and delivery are handled by a greedy production scheduler that gives near-optimal performance in practice. In order to isolate the effects of bidding, we modified the game settings to allow each agent to receive an effectively unlimited quantity of each component on the 15th game day at no cost, eliminating the need for a strategy for purchasing components from suppliers. This is not entirely unrealistic, as many agents in the competition actually ordered the majority of their components on the first game day [12]. Agents were only allowed to carry up to 200 of each type of computer in inventory, to prevent them from using their limitless components to build up large computer inventories during periods of low customer demand. The effect of this limitation was to increase the responsiveness of computer prices to changes in demand, creating a more dynamic and interesting game scenario.

The agents differ in the bidding methods used and the predictions of bid acceptance probability. The bidding methods used are

- Bidder: The bidding method presented in Sect. 5.
- OldBidder: A previously developed hill climbing bidding method [2].

and the bid acceptance prediction methods are

- Learning: Learning as described in Sect. 4.
- Heuristic: The heuristic described in Sect. 4.2.
- OldHeuristic: A previously developed prediction heuristic [2]. Testing on game data has shown this heuristic to be less accurate than Heuristic.

OldBidder and OldHeuristic are taken from TacTex, our entry in the 2003 TAC SCM competition, and are described in detail in [2]. Each of the six combinations of bidding methods and prediction methods is used by one agent.

6.2 Experimental Setup

Three rounds of 30 games were played between the six agents. During the first round, the agents labeled as using Learning actually used the same heuristic as Heuristic. The game logs from the first round were then used to train sets of predictors to be used by the Learning agents in the second round. For both agents, we trained a separate predictor for each of the 26 price points between 0 and 1.25 times the base price spaced at an interval of 0.05, using BoosTexter as the learning algorithm and using 10% of the available data (about 100,000 instances). Because each learning agent is trying to outbid only the other agents and not itself, its own bids were ignored when determining the winning bid for each training instance. The functions mapping bids to probabilities of acceptance are created from the 26 predictions by enforcing a monotonicity constraint as described in [9], with the added step of setting all probabilities for bids above the reserve price to 0. A second round of games was then played.

In the third round, the Learning agents used a set of predictors that had been trained on the logs from the semifinal and final rounds of the 2003 TAC SCM competition. The purpose of this was to determine how well the predictors would generalize to a different set of agents.

6.3 Results

The results are presented in Table 2. The average relative score of each agent is given along with the standard deviation. An agent's relative score in a game is its score minus the average score of all agents for that game. The average score over all agents and games in each round was around $80 million. Because all agents were initially given sufficient components to last the whole game, no component costs are included in any of the scores presented.

Table 2. Average relative score (in millions)

Agent	*Relative Score*		
	Round 1	*Round 2*	*Round 3*
Bidder/Learning	6.10 ± .28	9.04 ± .3	6.49 ± .73
Bidder/Heuristic	6.13 ± .28	2.95 ± .42	5.20 ± .57
Bidder/OldHeuristic	2.2 ± .30	-.31 ± .42	1.37 ± .40
OldBidder/Learning	-4.17 ± .22	1.60 ± .55	-1.80 ± .70
OldBidder/Heuristic	-4.21 ± .24	-5.87 ± .30	-4.54 ± .53
OldBidder/OldHeuristic	-6.09 ± .39	-7.41 ± .48	-6.72 ± .46

From the results of the first round, we can see that agents using Bidder outperform the agents using OldBidder, and that for each bidding method, agents using Heuristic outperform agents using OldHeuristic. This is the expected result.

The results of the second round show exactly what we had hoped to see: using learning significantly improved agent performance. Bidder/Learner outscored Bidder/Heuristic in all but one game and by an average margin of $6 million.

In the third round, the agents using Learning still showed a performance improvement, but by a smaller margin. Considering that the predictors used by Learning were trained on games involving a completely different set of agents, and a somewhat different game scenario (i.e., a limited component supply), this result is very promising. In actual competition, we might not have access to games involving only the agents we are competing against, and this experiment suggests that learning could still be successfully applied in such a case.

7 Future Work and Conclusion

In this paper, we considered the problem faced by an agent acting in a supply chain that must bid in simultaneous reverse auctions to win orders from customers. Using TAC SCM as a test domain, we presented a learning approach to the task of predicting the probability of bid acceptance, and we presented a heuristic bidding method that uses these predictions. A comparison of learning algorithms showed that M5 regression trees and BoosTexter result in similar prediction accuracy when testing and training data come from separate games. When used as part of a complete agent, learned predictors were shown to provide a significant improvement in performance over heuristic predictors.

One important result demonstrated was that the learned predictors generalize well to new situations, both in terms of prediction accuracy and of agent performance. This gives us hope that our learning approach can be used successfully in competition when facing different sets of agents or agents that change their behavior over time.

There are several possible ways in which predictions could be improved. The results of Sect. 4.3 suggest that acquiring data from a variety of situations might aid in training a more robust predictor. Further experiments could determine the best combinations of data for an agent to use. Also, additional information available to an agent could be included as features, such as knowledge of the availability and prices of components. Finally, an agent could make use of its knowledge of auction results during a game to make on-line improvements to its predictors. Boosting-based predictors would lend themselves well to this approach, since making incremental modifications to the existing predictors would be straightforward.

The heuristic bidding method presented appears to work well, but needs to be made less myopic. This could be done by developing better estimates of the future values of components and computers in inventory, in order to allow more informed decisions of whether to hold on to them for later use.

Acknowledgments

This research was supported in part by NSF CAREER award IIS-0237699.

References

1. Sadeh, N., Arunachalam, R., Eriksson, J., Finne, N., Janson, S.: TAC-03 a supply-chain trading competition. AI Magazine (2003)
2. Pardoe, D., Stone, P.: TacTex-03: A supply chain management agent. SIGecom Exchanges **4** (2004) 19–28

3. Arunachalam, R., Eriksson, J., Finne, N., Janson, S., Sadeh, N.: The TAC supply chain management game. Technical report, Swedish Institute of Computer Science (2003) Draft version 0.62.
4. Papaioannou, V., Cassaigne, N.: A critical analysis of bid pricing models and support tool. In: IEEE International Conference on Systems, Man and Cybernetics, Piscataway, NJ (2000)
5. Lawrence, R.D.: A machine-learning approach to optimal bid pricing. In: Proceedings of the Eighth INFORMS Computing Society Conference on Optimization and Computation in the Network Era, Arizona (2003)
6. Benisch, M., Greenwald, A., Grypari, I., Lederman, R., Naroditskiy, V., Tschantz, M.: Botticelli: A supply chain management agent designed to optimize under uncertainty. SIGecom Exchanges **4** (2004) 29–37
7. Kiekintveld, C., Wellman, M., Singh, S., Estelle, J., Vorobeychik, Y., Soni, V., Rudary, M.: Distributed feedback control for decision making on supply chains. In: International Conference on Automated Planning and Scheduling. (2004)
8. Dahlgren, E., Wurman, P.: PackaTAC: A conservative trading agent. SIGecom Exchanges **4** (2004) 38–45
9. Schapire, R.E., Stone, P., McAllester, D., Littman, M.L., Csirik, J.A.: Modeling auction price uncertainty using boosting-based conditional density estimation. In: Proceedings of the Nineteenth International Conference on Machine Learning. (2002)
10. Schapire, R.E., Singer, Y.: BoosTexter: A boosting-based system for text categorization. Machine Learning **39** (2000) 135–168
11. Witten, I.H., Frank, E.: Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations. Morgan Kaufmann (1999)
12. Estelle, J., Vorobeychik, Y., Wellman, M., Singh, S., Kiekintveld, C., Soni, V.: Strategic interactions in a supply chain game. Technical report, University of Michigan (2003)

Agents' Strategies for the Dual Parallel Search in Partnership Formation Applications*

David Sarne[1] and Sarit Kraus[1,2]

[1] Department of Computer Science, Bar-Ilan University, Ramat-Gan, 52900, Israel
[2] Institute for Advanced Computer Studies, University of Maryland, College Park, MD 20742

Abstract. In many two-sided search applications, autonomous agents can enjoy the advantage of parallel search, powered by their ability to handle an enormous amount of information, in a short time, and the capability to maintain interaction with several other agents in parallel. The adoption of the new technique by an agent suggests a reduction in the average cost per interaction with other agents, resulting in an improved overall utility. Nevertheless, when all agents use parallel search in Multi-Agent Systems (MAS) applications, the analysis must take into consideration mainly equilibrium dynamics which shape their strategies. In this paper we introduce a dual parallel two-sided search model and supply the appropriate analysis for finding the agents' equilibrium strategies. As a framework application for our analysis we suggest and utilize the classic voice communication partnerships application in an electronic marketplace. By identifying the specific characteristics of the equilibria, we manage to supply efficient means for the agents to calculate their distributed equilibrium strategies. We show that in some cases equilibrium dynamics might eventually drive the agents into strategies by which all of them end up with a smaller expected utility. Nonetheless, in most environments the technique has many advantages in improving the agents expected utility.

1 Introduction

Agents' search for partners is an inherent process in many MAS applications [15]. A common scenario in partnerships models is where the searching agent is satisfied with only one partner for forming partnership. Typical applications of this type include buyer-seller [2], peer-to-peer media exchange, dual-backup services [13], etc. In these applications each potential partnership suggests a different utility for the agent. The agent can't a-priori evaluate the expected utility from a partnership with any of the other agents, though it can evaluate the overall utility distribution. Learning about the expected utility from a partnership with a specific agent is possible by interacting with this agent. This process involves the consumption of some of the agent's resources (e.g. cost of search). Thus the agent's main challenge in such an application is to find a strategy for determining, at each stage of its search, whether to try and partner with one of the formerly interacted agents, or resume its search. Partnering with an agent will yield an

* This research was supported in part by NSF under grant #IIS-0208608.

P. Faratin and J.A. Rodríguez-Aguilar (Eds.): AMEC 2004, LNAI 3435, pp. 158–172, 2005.

immediate benefit, while resuming the search might result in a better partnership (though inducing a further search cost). The search process is considered to be two-sided, as all agents are engaged in search, and a partnership between two agents will be formed only if both agents commit to it.

Traditionally, two-sided search models [3] consider humans engaged in dual search activities (marriage market, for instance) with pure sequential search strategies (where each party samples and evaluates one potential partner at a time). However, the advantages of agents in filtering and processing information, as well as their improved parallel interaction capabilities (in comparison to humans), suggests an improved two-sided search technique, by which the agents sample and assess several potential partner agents, during a search stage, in parallel. This is mostly beneficial when parts of the search costs do not necessarily depend on the number of potential partner agents sampled. In this case the parallel search reduces the average search cost per sampled potential partner.

Our goal in this paper is to present and analyze a dual parallel two-sided search model, where all agents in the electronic marketplace (or any other MAS environment) use the parallel search. The transition from one agent using the new technique into a dual usage requires a complete understanding of the dynamics which drive the agents' strategies towards a stable equilibrium. An important output of the analysis is an algorithm that significantly simplifies the process of extracting the equilibrium strategies. This type of algorithm is extremely valuable for both agents and their developers. As part of the discussion we show that in many cases the dual parallel search may yield a better utility for the agents, though in some cases the effect of the proposed search technology is an equilibrium, with lower expected utility for each agent. These are important inputs for market makers, when considering the integration of the parallel search techniques in the agents they supply to their customers.

As a framework application for our analysis we use an eCommerce environment where agents represent telephony service providers. Consider, for example, a service provider, serving a specific geographic location, having long term formal partners (other service providers) in various geographical destinations, with whom it has termination agreements[1]. At any time the service provider can produce a short term forecast for its unused bandwidth. In most cases, the amount of such bandwidth, and the relatively short period the service provider can commit to, makes the option of selling this remaining bandwidth non-beneficial. Alternatively, the service provider can use this bandwidth in an exchange process for reducing its termination costs. In the latter case, the service provider will create an agent operating in a designated market, where the other agents represents different service providers, serving different locations. The agent will seek to create ad-hoc short term partnerships with these agents, to make use of the unexploited bandwidth on both sides. Once a partnership is formed, each service provider will route some of its traffic to the remote destinations via the other service provider's network, instead of using its costly formal long term termination partners. Similar application of exchanging unexploited resources, can be found in [13,1].

[1] An agreement defining a service provided by an interconnection provider, whereby it connects a call from a point of interconnection to a network termination point.

In the telephony communication domain, the benefit that can be obtained from a given partnership is a function of the Quality of Service (QoS) that can be guaranteed by such a connection - service providers offer their customers different tariffs for different levels of services, as defined in a Service Level Agreement (SLA). The key measure for evaluating the Quality of Service is speech quality. While traditional methods of determining speech quality were based on subjective tests with panels of human listeners (ITU-T P.800/P.830, [9]), recent ITU standards [10] suggest automatic prediction of voice quality that would be given in typical subjective tests. This is done by intrusive end-to-end test calls (i.e. generating test traffic) and passive monitoring of traffic in strategic locations along the call route. It is notable, however, that speech quality is not linear, and as the call traverses the two networks, the quality can not be evaluated simply as a function of both networks' separate performances. Thus for the purpose of evaluating the perceived utility from each potential partnership, an agent needs to perform a full set of tests (and can't rely on former tests performed for the connection of its network with other networks). The costs of these tests represents the search cost in our model.

At this point the model diverges into two important variants, differing in the expected utility each partnership member obtains from the partnership. In the first variant, the utility each of the two agents forming a partnership obtain is equal. The second variant suggests different utilities for both agents, in a way that the utility gained by each of the agents will be considered as randomly drawn from a general distribution (see for example [3]). In the suggested telephony application there are many factors supporting each of the two variants, such as the SLAs structure (committing to the quality of both or just one of the inbound and outbound channels), different tariffs, attractiveness of different geographical locations, etc. We support both variants throughout the paper.

As we add the ability to interact with several potential partner agents in parallel, on top of the traditional sequential search model, we need to consider the search cost structure. The integration of passive monitoring and intrusive test calls, suggests a fixed and variable components in the cost structure. The fixed component can be associated with passive testing devices, monitoring all traffic simultaneously. The variable component is associated with the intrusive end-to-end test calls. The number of test calls performed is derived from the number of potential partners evaluated over each search round. Similar applications in which such dual parallel search can be used include secondary markets for exchanging remaining resources in those cases where selling them is not the core business of the organization, or when the overhead for selling them makes it non-beneficial, and thus an exchange mechanism is used (see [13]).

In the next section we address relevant multi-agent and matching literature. In section 3 we present the model. An equilibrium analysis and an efficient algorithm for finding the equilibrium strategy are provided in section 4. The incentive to deviate from the sequential two-sided search towards the parallel search is presented in section 5. Section 6 compares the two model variants. We conclude and suggest directions for future research in section 7.

2 Related Work

The application of agents seeking coalitions with other agents is quite common in MAS environments and in electronic markets in particular [5,6,2]. A specific case of such coalitions is the partnerships model where each agent seeks a single partner [12,15,13]. Different mechanisms for partnering suggest different assumptions regarding the agent's scanning capabilities, ranging from the option to scan as many agents as needed, through making use of a central matcher or middle agents [4] and up to a complete distributed process, without the help of a predefined organization or a central facilitator [11]. Our model is of the latter type, and assumes a distributed environment where each agent needs to invest resources for interacting with other agents and evaluate the potential partnerships.

The basic concepts of search, can be found in the classical search theory. Here, various models of one-sided search (assuming no mutual search activities) and two-sided search were suggested [7,14,3]. While the concept of parallel search (also known as variable sample sizes) was suggested for the one-sided search [8], two-sided models always assume a sequential search. Thus the uniqueness of the suggested model is in allowing all agents to use the parallel method. It is notable that the transition from the one-sided to two-sided models, when considering parallel search techniques, involves many complexities, as the equilibrium considerations become the main issue of the analysis[2]. Additionally, in search theory, search "costs" are usually modelled by the discounting of the future flow of gains, while in MAS environments the total search period is relatively short and utilities are immediate.

In a recent paper [12] we present an initial analysis of equilibrium in a two sided search model, where there are two types of searchers (buyers and sellers), and only one type may use parallel search. We have shown that the adoption of parallel search by buyers in C2C markets, leads to new equilibrium strategies, which can significantly improve the utility of the buyers (thus reducing the utility of the sellers using the sequential search). Nevertheless, our analysis was limited to the usage of the technique by one side of the interaction, and for the case where the perceived utilities are different. When all agents can use the parallel search technique, the problem of evaluating the stability of suspected equilibrium strategies becomes significantly more complex. The main challenge is in handling the enormous expansion of the strategy space. In sequential two-sided search the strategy space is bi-dimensional (having each agent's reservation value[3] on each

[2] In one-sided search, the problem can be seen as a simple optimization problem, with no equilibrium concerns, since the focus is on a single searcher's decision.

[3] As in most search models, in the following sections we show that the agents will use a reservation value based strategy. The agent will accept all offers that yield a utility greater than or equal to a reservation value, and reject all those that yield a utility less than this value. Notice the reservation value of the search strategy is different than a reservation price usually associated with a buyer or a seller that are not involved in a search. While the reservation price denotes an agent's true evaluation of a specific potential partnership, the reservation value of a search strategy is mainly a lower bound for accepted partnership, derived from the expected utility optimization considerations.

axis), thus each agent's incentive to deviate from a given strategy, can be checked along its own reservation value. In our dual parallel search model, each potential strategy must be compared with all possible combinations of reservation values and the number of partners sampled over a search round.

3 The Dual Parallel Two-Sided Search Model

Consider an environment populated with numerous agents, seeking to form partnerships for their benefit throughout random interactions with other agents. As suggested in the introduction, the perceived utility for an agent from any suggested partnership with any specific agent, denoted by U, can be seen as randomly drawn from a population with p.d.f. $f(U)$ and c.d.f. $F(U)$, $(0 \le U < \infty)$. We assume that agents, while ignorant of the utility obtained by partnering with specific agents, are acquainted with the overall possible partnership's utility distribution. This assumption is common in search models (see [3,14,13,11]).

Integrating the above into the service providers application, we consider each agent as suggesting a termination service for a standard unit of time and volume[4]. Any random interaction between two agents, may yield a partnership for terminating calls, for the benefit of the two represented service providers. The utility from such a partnership is expressed in monetary units, as the service providers' main concern is revenue.

When using the parallel search, at any stage of its search the agent encounters N potential partner agents, interested in forming a partnership. This is in comparison to the traditional sequential two-sided search, where the agent samples only one other agent at any stage of its search [3]. For each encounter both agents evaluate the utility from such a partnership (in the telephony application this will mean testing the perceived connection between the two networks). We assume this utility is randomly drawn from a similar distribution function for all agents (either with a similar value or as two different values, according to the model variant), due to the high number of potential partners and the similar operational cost structure.

We denote, α and β as the fixed and variable cost components of a search stage. In the telephony application, these costs are associated with intrusive test calls and passive monitoring for testing the perceived connections with the N networks represented by the potential partnering agents. Thus the total search cost per a search round is $\alpha + \beta N$. Notice the values α and β are standard for all agents, as testing the quality of a connection between two service providers is conducted in similar methods based on standard testing devices. After evaluating each of the potential partnerships, each agent will make a decision whether to commit to one of them. Obviously, each agent is interested in partnering with the agent with whom the potential partnership will yield the maximum utility. A partnership will eventually be formed only if both agents are willing to commit to it. Otherwise the agents will resume their search according to the same process as

[4] Notice the agent can consider numerous potential partners as these include service providers from any geographical location, IXCs, and ISPs supporting VoIP.

described above. Notice that since each new interaction suggest a new potential utility, and since the agent may commit only to the best in its sample, deadlocks will never occur. Since the agents are not limited by a decision horizon and can control the intensity of their search, and the interaction with other agents does not imply any new information about the market structure, their strategy is stationary - an agent will not accept an opportunity it has rejected beforehand. Thus agents will use a reservation value strategy. The fast parallel interactions between agents, ensure finding a partner within reasonable time. This, along with the fact that utilities can be seen as immediate (due to the short partnership duration) allow us to ignore the influence of a discounting factor when considering expected utilities. Lastly, notice that as all agents are subject to a similar search cost, and the perceived utility can be seen as randomly drawn from the same population, all agents share the same reservation value.

As the search process is two-sided, the main challenge is in finding the equilibrium strategies. Any set of strategies that can't guarantee equilibrium stability will not hold. As suggested in the introduction, we distinguish between two variants of the above model. The first assumes both agents in a partnership will yield the same utility, while in the second each agent diversely evaluates the utility it can gain from a given partnership. Each variant will yield different equilibrium strategies and thus will differ with the expected utility gained by the agents.

For analysis purposes, we'll use several notations in the following sections. A strategy of sampling N other agents over each search round, and acting according to a reservation value x_N will be denoted (N, x_N). The expected utility of an agent when using strategy (N, x_N), will be denoted $V_N(x_N)$. As the agent is mainly concerned at each search round with the partnership offering the maximum utility in its sample, we will use the random variable U^N to denote the partnership with the maximal utility in an N size sample. The p.d.f. and c.d.f. of the variable U^N will be denoted $f_N(x)$ and $F_N(x)$, accordingly.

4 Agents Strategies and Equilibrium Dynamics

We start by formulating the expected utility for the agent when using a strategy (N, x_N), given the strategy (k, x_k) used by the other agents in the environment. In the variant where both agents obtain the same utility from a given partnership the expected future utility $V_N(x_N)$ is:[5]

$$V_N(x_N) = \frac{\int_{y=max(x_N,x_k)}^{\infty} y f_N(y) F(y)^{k-1} dy - \alpha - \beta N}{\int_{y=max(x_N,x_k)}^{\infty} f_N(y) F(y)^{k-1} dy} \tag{1}$$

This can be decomposed into two parts: the conditional expected utility from a partnership that will be eventually formed, and the aggregated search cost derived by the number of search cycles. The number of search cycles is geometric and the probability of success is $\int_{y=max(x_N,x_k)}^{\infty} f_N(y) F(y)^{k-1} dy$. In addition, notice that:

[5] Due to space considerations, we include the basic formulation and sketch of proofs. The extended version of the paper, including the detailed formulation, proofs and discussions can be found at www.cs.biu.ac.il/ sarne/FullPapers/Dualparallel.pdf.

$$F_N(x) = F^N(x) \qquad f_N(x) = \frac{dF_N(x)}{dx} = Nf(x)F(x)^{N-1} \tag{2}$$

and thus, substituting (2) in (1) we eliminate the sample maximum notations from the equation and obtain a function which depends only on the cost structure α and β and the general distribution function $F(x)$:

$$V_N(x_N) = \frac{N\int_{y=max(x_N,x_k)}^{\infty} yf(y)F(y)^{N+k-2}dy - \alpha - \beta N}{N\int_{y=max(x_N,x_k)}^{\infty} f(y)F(y)^{N+k-2}dy} \tag{3}$$

Notice that any usage of $x_N < x_k$ will yield the same utility as using x_k. This is simply because any partnership suggesting a utility lower than this value will always result in a rejection from the other agent and consequently the continuance of the search. Similarly, for the variant where both agents obtain different utilities:

$$V_N(x_N) = \frac{(1 - F_k(x_k))\int_{y=x_N}^{\infty} yf_N(y)dy - \alpha k - \beta Nk}{(1 - F_k(x_k))(1 - F_N(x_N))} \tag{4}$$

and by substituting (2) in the above equation we obtain a simpler function. Notice that unlike the expected utility in (1), here, any change in the reservation value, x_N, including a reduction to a value beyond x_k will affect the expected utility. This is because of the absence of any correlation between the utility an agent gains from a potential partnership and the probability of being accepted by the other agent. Thus:

$$\lim_{x_N \to 0} V_N(x_N) = E[U^N] - \frac{\alpha k + \beta Nk}{(1 - F_k(x_k))} \tag{5}$$

Except for the above difference, the expected utility curve is quite similar in its structure for both variants of the model. As the agent increases its reservation value, and becomes more selective, the utility improvement gained by partnering with a better partner significantly decreases in comparison to the additional cost incurred by the increase in the number of search rounds. Thus both variants satisfy:

$$\lim_{x_N \to \infty} V_N(x_N) = -\infty \tag{6}$$

4.1 Reservation Value and Expected Utility Function

Having established the behavior of the expected utility function for upper and lower values of the reservation value, we now show that for both variants of the model, the expected utility function has a unique maxima.

Theorem 1. *When all other agents use strategy (k, x_k), an agent's expected utility function, $V_N(x_N)$, when using strategy (N, x_N) is quasi concave in x_N with a unique maxima satisfying:*

$$V_N(x_N) = x_N \tag{7}$$

Sketch of Proof: We present the proof for the variant where both agents obtain equal utilities from a partnership. The proof for the second variant is similar. Deriving (3) we obtain:

$$\frac{dV_N(x_N)}{dx_N} = \frac{f_N(x_N)F^{N+k-2}(x_N)\big(V_N(x_N) - x_N\big)}{N\int_{y=max(x_N,x_k)}^{\infty} f(y)F(y)^{N+k-2}dy} \equiv r(x_N)(V_N(x_N) - x_N) \;\;,\;\; x_N \geq x_k \tag{8}$$

A solution for (8) will require $V_N(x_N) = x_N$. Note that $f_N(x_N) > 0$ implies $r(x_N) > 0$, hence for x_N satisfying $V_N(x_N) = x_N$:

$$\frac{d^2V_N(x_N)}{dx_N{}^2} = r^{'}(x_N)(V_N(x_N) - x_N) + r(x_N)(V_N{}^{'}(x_N) - 1) < 0 \tag{9}$$

Thus $V_N(x_N)$ is quasi concave with a unique maxima. □

From (1-6) and theorem 1 we can sketch the graph $V_N(x_N)$. The basic structure of the curve is given in Figure 1, using the uniform distribution function and the parameters: $N = 3$, $k = 2$, $\alpha = 0.05$, $\beta = 0.005$ and $x_k = 0.55$ (substituted in equations (3) and (4)). Notice the difference between the two variants for reservation values $x_N \leq x_k$, as described above.

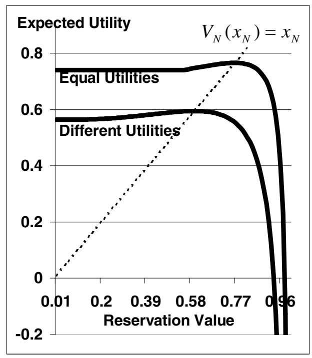


Fig. 1. Agent's expected utility

Once we have established the expected utility structure, we can suggest a simple algorithm for calculating the agent's optimal reservation value (and thus its expected utility, according to (7)) given the number of potential partners sampled, N, and the other agents' strategy (k, x_k). The algorithm makes use of a binary search and will always reach the agent's optimal reservation value, in a finite number of steps. Since the algorithm is very similar to the one we have suggested in [12] for the case where only one of the agents uses the parallel search technique, it will not be detailed in the current context.

4.2 Dual Parallel Two-Sided Search Strategy

Having at hand an efficient means for finding the agent's optimal reservation value, given the strategy of the other agents (k, x_k), we move on to explore the dynamics affecting agents' strategies in the dual parallel search. We start by introducing two important equations that can be used for finding an agent's optimal reservation value, given the number of partners it samples over a search round, N, and the other agents' strategy (k, x_k).

Theorem 2. *When an agent samples N potential partners over a search round and the other agents use strategy (k, x_k): (a) The agent's optimal reservation value x_N in the variant with equal utilities, satisfies:*

$$(N+k-1)(\alpha+\beta N) = N\Big((max(x_k,x_N)-x_N)(F^{k+N-1}(x_k)-1) + \int_{y=max(x_N,x_k)}^{\infty}(1-F^{k+N-1}(y))dy\Big) \tag{10}$$

(b) The agent's optimal reservation value x_N in the different utilities variant, satisfies:

$$\alpha k + \beta N k = (1 - F_k(x_k)) \int_{y=x_N}^{\infty} (1 - F_N(y))dy \tag{11}$$

Sketch of Proof:
(a) Set the first derivative of (3) to 0 and use integration by parts, substituting: $dv = f(y)F(y)^{N+k-2}$ and $u = y$ in (3). Manipulating and rearranging the result we obtain (10).
(b) Use a similar methodology as in (a) to obtain (11). □

The suggested equalities in theorem 2 are useful computational methods for calculating x_N. This can be particularly important when checking the incentive to deviate from a given strategy. The following theorem 3 captures another important characteristic of the agent's strategy that will aid us in finding the equilibrium.

Theorem 3. *(a) In both variants of the model, when all other agents sample k potential partners over a search round, if an agent's expected utility of sampling $k+1$ potential partners, $V_{k+1}(x_{k+1})$ is smaller than $V_k(x_k)$, then the expected utility when sampling N potential partners, $V_N(x_N)$, where $N > k+1$, is also smaller than $V_k(x_k)$. (b) Similarly, when all other agents sample k potential partners over a search round, if an agent's expected utility of using $k-1$ potential partners, $V_{k-1}(x_{k-1})$, is smaller than the expected utility when using k potential partners, $V_k(x_k)$, then the expected utility when using N potential partners, where $N < k-1$ is also smaller than $V_k(x_k)$.*

Sketch of Proof:
(a.1) For the variant with equal utilities - assume otherwise (e.g. the expected utility when using N, when $N > k+1$ and $V_k(x_k) \leq V_N(x_N)$, is greater than $V_k(x_k)$). Then subtract two instances of equation (10) using N and k. Substituting β as obtained from manipulating the inequality $x_N - V_{N+1}(x_{N+1}) > 0$, using (1), and analytically exploring both sides of the inequality expression, we obtain a contradiction, given the incorrectness assumption. The detailed proof, depicted on several pages, can be found in the full version of the paper.
(a.2) For the variant with different utilities - assume otherwise ($x_N > x_k$). Create 3 instances of (11) for k, $k+1$ and N. Then subtract the first two and the last two equations. Set all integrals' lower bounds to x_N and notice the obtained inequality is strictly negative (by deriving the term inside the integral), which leads to a contradiction, given the incorrectness assumption.
(b) In a similar methodology as used for the proof of part (a), making use of the appropriate modifications of the inequalities. □

The above theorem is an important milestone towards the formation of an algorithm for finding the equilibrium, as it allows us to limit the number of potential partners that needs to be considered, when checking the stability of a potential equilibrium strategy (only the former and next subsequent numbers need to be considered).

4.3 Finding the Equilibrium

Having the results and proofs given in previous subsections, we now suggest an efficient method for finding the equilibrium in the dual parallel two-sided search model. From theorem 3, we conclude that in order to check the stability of strategy (N, x_N), one only needs to check the expected utility of an agent when deviating to strategy $(N+1, x_{N+1})$ and $(N-1, x_{N-1})$. This can be simply calculated using equations (10) and (11). If the expected utility for strategy (N, x_N) is greater than the other two, then this is an equilibrium.

As the agents are identical, they will all use the same equilibrium strategy (if an equilibrium exists), thus their expected utility will be identical. This resolves the uncertainty in case of a multiple equilibria scenario - all agents will use the equilibrium strategy with the highest expected utility.

An important consideration is the upper bound for N, when seeking the equilibrium strategy. An equilibrium doesn't necessarily exist, and while using the proposed method one may wonder when to stop as N grows and the equilibrium conditions are not satisfied. We propose a simple upper bound that can be used with both variants of the model.

Theorem 4. *(a) An upper bound value for the number of partners to be considered over a search round, in the variant with equal utilities, is the solution $N = N_{max}$ of the equation:*

$$E[U^N] = \alpha + \beta N \tag{12}$$

(b)An upper bound value for the number partners to be considered over a search round, in the variant with different utilities, is the solution $N = N_{max}$ of the equation:

$$E[U^N] = \alpha N + \beta N^2 \tag{13}$$

Sketch of Proof:
(a) + (b) - by substituting (12-13) in (3-4), we attain a negative expected utility. The expected utility will remain negative for any $k \geq N_{max}$. Though the agents will unavoidably abandon search activity for these k values, and if no equilibrium was found up to this point then the problem with the current search cost structure (α and β values) doesn't have a pure equilibrium solution. Such an N_{max} value can always be found as the left hand side of equations (12-13) is concave and the right hand side is convex (except for the case where the agents would have initially abandoned the search, e.g. where the left hand side of the equations is smaller than the right hand term for $N = 1$).

To summarize the methodology for finding the equilibrium (if any exists), we suggest the following algorithm.

Algorithm 1. *An algorithm for finding the equilibrium strategy (N, x_N) for the dual parallel search model.*
Input: *α, β - cost structure coefficients; F(x) - the utility c.d.f.*
Output: *$(V_N(x_N), x_N, N)$ - Equilibrium strategy, if one exists, otherwise a "no equilibrium" message.*

```
Set N_max according to equation (12).
Set List[]=null;
for (N=1;N≤ N_max;N++) {
    if (N>1) calculate V_{N−1}(x_{N−1}) using (10), where k = N;
    calculate V_N(x_N) and V_{N+1}(x_{N+1}) using (10), where k = N;
    if (V_N(x_N) ≥ V_{N+1}(x_{N+1}))
        if (N=1) or (V_N(x_N) ≥ V_{N−1}(x_{N−1}))
            add (V_N(x_N), x_N, N) to List[].
}
If (List[]!=null) return member with highest V_N(x_N)
else return("no equilibrium");
```

The above algorithm is applicable for the variant with equal utilities. In order to use it with the other variant, one needs to replace equations (12) and (10) with equations (13) and (11).

If there is an equilibrium solution to the problem, the algorithm will find the equilibrium strategy in $o(N_{max})$ stages, where N_{max} is the upper bound, calculated according to (12-13), and its value is mostly influenced by the utility distribution function. The innovation of the proposed algorithm is in bounding the space of possible strategies which needs to be compared for any suspected equilibrium strategy. The complexity of the solution, in the absence of such a bound is discussed in the next section.

5 The Incentive to Use Parallel Search

Considering equations (1) and (4), it is clear that the traditional sequential search model is a specific case of the general dual parallel search as described in the model section. For example, by substituting $N = 1$ in (4), we obtain a similar expected utility function for the sequential search model as described in [3]. Nevertheless, the sequential two-sided search will not be stable in many cases, since single agents have an incentive to deviate from the sequential strategy for many plausible combinations of α and β values. Figure 2 demonstrates this phenomena for the uniform distribution function. As the utility varies from 0 to 1, the bottom triangular area represents all plausible α and β combinations where the agents will consider a sequential two-sided search (e.g. where the expected utility for the agents in a sequential equilibrium strategy is positive). Out of this area, we have isolated (on the left side) all combinations of α and β where an agent can increase its expected utility by deviating from such a sequential strategy (assuming all other agents' strategies are sequential). For calculation purposes we used equations (10) and (11). Notice that for a large portion of the cost structures, any single agent has an incentive to deviate from its sequential strategy. Furthermore, the incentive to deviate from the sequential strategy is mainly for the combinations of α and β with small values (in comparison to the average utility from a partnership), which characterizes most MAS applications.

As each agent has the incentive to use the parallel search technique, the traditional sequential two-sided search model transforms into a dual parallel search.

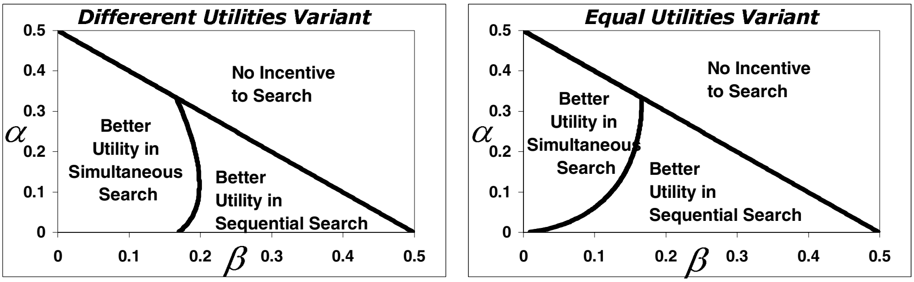


Fig. 2. The incentive for using parallel search

This also suggests a possible improvement in the expected utility of the agents, in comparison to the traditional sequential two-sided search models. Again, the improvement is mostly noticeable for values where α and β are relatively small in comparison to the utilities gained from the partnerships.

The transition into a dual parallel search model doesn't necessarily guarantee an expected utility improvement. In some cases, the dual parallel usage may result, inevitably, in an equilibrium where the agents worsen their expected utility. In such a scenario, all agents could gain more by using a sequential strategy, but each, separately has an incentive to deviate towards a parallel strategy, resulting eventually in a non-optimal result. This is demonstrated in figure 3, again for the uniform distribution function, with $\alpha = 0.1$ and $\beta = 0.05$. The middle curve represents the expected utility when sampling N potential partners over a search round according to the horizontal scale. The upper and lower curves represent the expected utility when a single agent deviates to $N+1$ and $N-1$, respectively. Here, the equilibrium utility is obtained when using $N = 5$ over a search stage (taking the variant of different utilities). Even though the sequential two-sided search utility is greater, none of the agents will maintain such a strategy as both agents have an incentive to deviate towards a higher number of sampled partners in a search round. The example given for the variant with equal utilities demonstrates a scenario where the expected equilibrium utility when using a

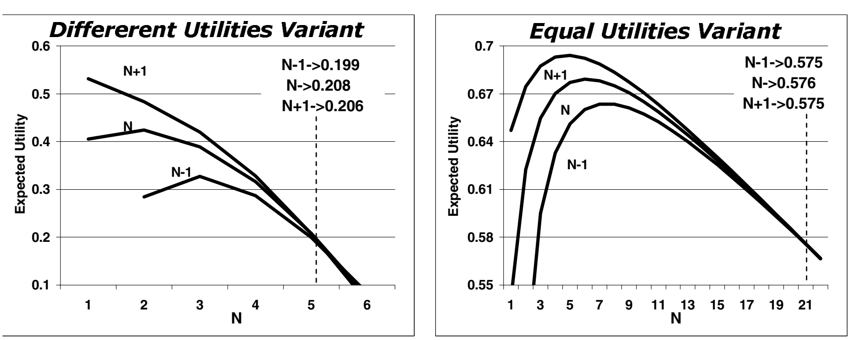


Fig. 3. Deviating from equilibrium

parallel ($N = 21$) search is higher than when using a sequential search, though it is far below the possible expected utility that could have been obtained by using $N = 6$.

Thus, extra care should be taken in the analysis of the dual parallel search equilibrium. This can be extremely important for market makers for understanding the consequences of allowing the agents to sample more than a single potential partner over each search stage, or even for actually limiting the number of partners that can be sampled by the agents at each turn.

6 Model Variants Comparison

Throughout the examples given in the previous section, one might notice from figure 2 that there is a stronger incentive to use the parallel search technique in the variant where the utilities for both parties are different. This doesn't necessarily mean the expected utility in this variant is greater. In fact, as notable from figures 1 and 3, the agents' expected utility is greater in the variant where both agents gain the same utility. The explanation for this phenomena can be found in the strong correlation between both agents acceptance decision, when utilities are equal, in comparison to no correlation at all in the second variant. If both agents gain a similar utility from a given partnership, then the probability that each of them is the highest in the other agent's sample is relatively high. On the other hand, when the expected utility from a given partnership is random, the probability of being the agent with the highest utility to a given potential partner is $1/N$. This insight can be formally proven, as the following theorem states.

Theorem 5. *When all agents use a dual parallel search with N potential partners over a search round, the variant with equal utilities will yield the agents using the equilibrium strategy a higher expected utility in comparison to the equilibrium utility that can be gained in the other variant.*

Sketch of Proof:
Set $k = N$ in equations (10) and (11) to obtain equilibrium reservation values, and isolate the term $\alpha + \beta N$. Then subtract the two equations, to obtain:

$$\frac{N}{2N-1}\int_{y=x_N^I}^{\infty}(1 - F^{2N-1}(y))dy = \frac{1}{N}\int_{y=x_N^{II}}^{\infty}(1 - F^N(x_N^{II}))(1 - F^N(y))dy \tag{14}$$

Where x_N^I is the reservation value of the equal utilities variant and x_N^{II} is the reservation value of the different utilities variant. Obviously the equation can hold only if $x_N^I \geq x_N^{II}$, and from theorem 2 we obtain $V_N(x_N^I) \geq V_N(x_N^{II})$. □

7 Conclusions

As demonstrated throughout this paper, in many cases, an agent engaged in search has an incentive to adopt the parallel search technique. Nowadays, as agents' technology is a reality and traditional processing and communication limitations were removed, it is high time to consider the dual parallel search model in MAS domains and in particular the two-sided search application for the electronic marketplace. We manage to present a complete equilibrium analysis, and suggest an efficient algorithm for calculating the agents' equilibrium strategies, given the environment parameters (utilities distribution and search cost coefficients). Deriving the equilibrium strategy is a complex task, as all agents can control both the number of partners they sample and their acceptance criteria; thus the challenge of finding a stable set of strategies becomes significantly complex. The novelty of the proposed algorithm is in the capability to bound the relevant strategy space and quickly eliminate non-equilibrium strategies. The adoption of the method can significantly improve the expected utility either when used one-sidedly or simultaneously by all agents. Nevertheless, as part of the discussion, we show that in some cases equilibrium dynamics might drive the agents into a strategy where the number of partners sampled results in a non-optimal expected utility. In some rare cases this could even worsen the expected utility in comparison to the sequential search. The later scenario further emphasizes the importance of the analysis given and the proposed algorithm, as market makers can use the results to understand and evaluate the influence of the decision to allow agents in their marketplace to use parallel search. The proposed analysis was followed by a plausible eCommerce application from the telephony call termination partnering domain. The division into two specific variants of the model, differing by the perceived utility from a given partnership, extends the variety of applications this model can be integrated in.

Notice that throughout the paper we assumed that the agent commits only to the potential partnership with the agent associated with the highest utility in the sample (assuming it is above its reservation value). Nevertheless, in a given sample, there might be several agents with a utility that might be greater than the reservation value being used. Thus the agent can improve its expected utility by considering committing also to the next best agent in the sample, upon receiving a rejection from the current potential partner agent. It is notable, however, that this technique has some setbacks. First, all agents need to wait for a rejection/acceptance message from their best sampled agent before considering committing to their next best candidate. This creates many constraints, and necessitate a protocol in which all the agents conduct each search round simultaneously. This might also require the introduction of a discounting factor into the model. Second, because of the significant amount of time that needs to be allocated for each search round (because of the expected bottlenecks), a failure or malfunction of one of the agents might, in extreme cases, drive the entire system into a "hold" position. Thus, prior to considering such a model for future research, a substantial research effort should be made to build the infrastructure and protocols for handling the additional dependencies and constraints involved.

References

1. R. Azoulay-Schwartz and S. Kraus. Stable strategies for collaborative behavior of agents in answering queries. *Artificial Intelligence journal*, 154(1-2):43–93, 2004.
2. S. Breban and J. Vassileva. Long-term coalitions for the electronic marketplace. In *B. Spencer, ed., Proc. of E-Commerce Applications Workshop*, 2001.
3. K. Burdett and R. Wright. Two-sided search with nontransferable utility. *Review of Economic Dynamics*, 1:220–245, 1998.
4. K. Decker, K. Sycara, and M. Williamson. Middle-agents for the internet. In *Proc. of IJCAI*, pages 578–583, 1997.
5. T. Ito, H. Ochi, and T. Shintani. A group-buy protocol based on coalition formation for agent-mediated e-commerce. *IJCIS*, 3(1):11–20, 2002.
6. K. Lermann and O. Shehory. Coalition formation for large scale electronic markets. In *Proc. of ICMAS'2000*, pages 216–222, Boston, 2000.
7. J. McMillan and M. Rothschild. Search. In Robert J. Aumann and Amsterdam Sergiu Hart, editors, *Handbook of Game Theory with Economic Applications*, pages 905–927. 1994.
8. P. Morgan. Search and optimal sample size. *Review of Economic Studies*, 50(4):659–675, 1983.
9. ITU-T Recommendation P.800. Methods for subjective determination of transmission quality. August 1996.
10. ITU-T Recommendation P.862. Perceptual evaluation of speech quality (pesq), an objective method for end-to-end speech quality assessment. February 2001.
11. D. Sarne and S. Kraus. The search for coalition formation in costly environments. In *Proc. of CIA2003*, pages 117–136, 2003.
12. D. Sarne and S. Kraus. Agent's multiple inquiries for enhancing the partnership formation process. In *Proc. of CIA2004*, pages 286–302, 2004.
13. D. Sarne and S. Kraus. Time-variant distributed agent matching applications. In *Proc. of AAMAS-2004*, pages 168–175, 2004.
14. R. Shimer and L. Smith. Assortative matching and search. *Econometrica*, 68(2):343–370, 2000.
15. K. Sycara, S. Widoff, M. Klusch, and J. Lu. Larks: Dynamic matchmaking among heterogeneous software agents in cyberspace. *Autonomous Agents and Multi-Agent Systems*, 5:173–203, 2002.

Three Automated Stock-Trading Agents: A Comparative Study

Alexander A. Sherstov and Peter Stone

The University of Texas at Austin, Department of Computer Sciences,
Austin, TX 78712 USA
{sherstov, pstone}@cs.utexas.edu

Abstract. This paper documents the development of three autonomous stock-trading agents within the framework of the Penn Exchange Simulator (PXS), a novel stock-trading simulator that takes advantage of electronic crossing networks to realistically mix agent bids with bids from the real stock market [1]. The three approaches presented take inspiration from reinforcement learning, myopic trading using regression-based price prediction, and market making. These approaches are fully implemented and tested with results reported here, including individual evaluations using a fixed opponent strategy and a comparative analysis of the strategies in a joint simulation. The market-making strategy described in this paper was the winner in the fall 2003 PLAT live competition and the runner-up in the spring 2004 live competition, exhibiting consistent profitability. The strategy's performance in the live competitions is presented and analyzed.

1 Introduction

Automated stock trading is a burgeoning research area with important practical applications. The advent of the Internet has radically transformed the nature of stock trading in most stock exchanges. Traders can now readily purchase and sell stock from a remote site using Internet-based order submission protocols. Additionally, traders can monitor the contents of buy and sell order books in real time using a Web-based interface. The electronic nature of the transactions and the availability of up-to-date order-book data make autonomous stock-trading applications a promising alternative to immediate human involvement.

The work reported here was conducted in the Penn Exchange Simulator (PXS), a novel stock-trading simulator that takes advantage of electronic crossing networks to realistically mix agent bids with bids from the real stock market [1]. In preparation for an open live competition, we developed three parameterizable trading agents and defined several instantiations of each strategy. We optimized each agent independently, and then conducted detailed controlled experiments to select the strongest of the three for entry in the live competition.

It is important to realize from the outset that this research is primarily an *agent study* pertaining to the interactions of particular agents in a fixed economy. Although PXS makes a strong and reasonable claim to implementing a realistic simulation of the stock market, the results and conclusions in this paper pertain to test economies including specific other stock-trading agents. In particular, we do not aim to create strategies

P. Faratin and J.A. Rodríguez-Aguilar (Eds.): AMEC 2004, LNAI 3435, pp. 173–187, 2005.

that are ready for profitable deployment in the real stock market (otherwise we would likely not be writing this paper!). Rather, this paper makes three main contributions. First, it contributes an empirical methodology for studying and comparing stock-trading agents—individually as well as jointly in a shared economy—in a controlled empirical setting. Second, it implements this methodology to compare three specific trading agents based on reinforcement learning, myopic trading using regression-based price prediction, and market making. Third, this paper contributes detailed specifications of promising strategy designs, one of which vastly outperformed competitor strategies in an open stock-trading competition and exhibited consistent profitability under a variety of market conditions.

The remainder of the paper is organized as follows. Section 2 provides the relevant technical background on the PXS simulator, our substrate domain. Section 3 characterizes prior research and points out the distinguishing features of this work. Section 4 discusses our approach to the automated stock-trading problem, explains our assumptions, and details our experimental methodology. Sections 5–7 describe our three stock-trading strategies. Sections 8–10 present and analyze the experimental results, focusing, respectively, on individual evaluations, the joint simulation, and the live competitions. Finally, Section 11 concludes with a discussion of unresolved questions and promising directions for future work.

2 Background

The Penn-Lehman Automated Trading (PLAT) project [1] is a research initiative designed to provide a realistic testbed for stock-trading strategies. PLAT provides a simulated stock-trading environment known as the Penn Exchange Simulator (PXS) that merges virtual orders submitted by computer programs with real-time orders from the Island electronic crossing network (ECN) [2]. No actual monetary transactions are conducted, and the efficacy of a trading strategy can be reliably assessed in the safety of a simulated market. Many previous stock simulators been developed that execute simulated orders at the current price in the real stock market. However, such simulators miss the effect of a simulated order on this price, an effect that becomes increasingly significant as the size of the orders increases. The main novelty of PXS is that it uses not only the current stock price, but also the whole list of pending limit orders to realistically determine the effect of simulated activity on the market [1].

PXS operates in *cycles*. During every cycle, a trading agent can place new orders and/or withdraw some of its previously placed orders. When placing a *buy order*, the agent specifies the number of shares it wishes to purchase and the highest price per share it is willing to pay. PXS sorts the buy orders by price into a *buy order book*, with the most competitive (highest-priced) orders at the top of the book. Likewise, a *sell order* states the amount of stock being sold and the lowest price per share the seller is willing to accept. PXS sorts the sell orders into a *sell order book*, with the most competitive (lowest-priced) orders on top. When an order arrives, PXS matches it with orders in the opposite order book (starting at the top of the book) that meet the order's price requirements. Partial matches are supported. Any unmatched portion of the order is placed in the corresponding book, awaiting competitive enough counterpart orders to match fully.

Apart from complete order-book data, the simulator makes a variety of agent-specific and market-wide information available to aid in order placement. In addition to real-time operation (*live mode*), the simulator supports *historical* simulations that use archived stock-market data from the requested day. Historical mode operates on a compressed time scale, allowing the simulation of an entire trading day in minutes. As a result, the agent is able to place considerably fewer orders overall than in live mode. Aside from the lower order-placement frequency, historical mode is operationally identical to live mode.

In December 2003 and April 2004, live PLAT stock-trading competitions were held including agents from several universities. The sole performance criterion was the Sharpe ratio, defined as the average of the trader's daily score over several days divided by the standard deviation. Thus, favorable placement in the competition required not only sizable daily earnings but also consistent day-to-day performance. The trader's score on a given trading day was its total profit and loss ("value") at the end of the day *plus* total "rebate value" (computed as $0.002 per share that added liquidity to the simulator) *minus* total "fee value" (computed as $0.003 per share that removed liquidity from the simulator). These rebates and fees are the same as those used by the Island ECN. Arbitrarily large positive or negative intra-day share holdings were allowed. However, the entrants were to completely unwind their share positions before the end of the day (i.e., sell any owned shares and buy back any owed shares) or face severe monetary penalties.

As a benchmark strategy for the experiments reported in this paper, we used the Static Order-Book Imbalance (SOBI) strategy [1], provided to participants in the PLAT competition as an example trading agent. We used default settings for all SOBI parameters. SOBI sells stock when the volume-weighted average price (VWAP) of the buy-book orders is further from the last price than the sell-book VWAP, interpreting this as weaker support for the current price on the buyers' part and a likely depreciation of the stock in the near future. In the symmetric scenario, SOBI places buy orders.

3 Related Work

Prior research features a variety of approaches to stock trading, including those presented here. Automated market making has been studied in [3,4,5]. Reinforcement learning has been previously used to adjust the parameters of a market-making strategy in response to market behavior [3]. Other approaches to automated stock trading include the reverse strategy and VWAP trading [5,6]. A brief overview of these common approaches can be found in [7].

To our knowledge, there have been no empirical studies of the interactions of heterogeneous strategies in a joint economy, yet such simulations would likely be more revealing of a strategy's earning potential than a study of the strategy in isolation. As a result, this work combines detailed individual evaluations of the strategies with a principled study of their performance in a joint economy. Another distinguishing feature of this research is the use of a highly realistic stock simulator. Furthermore, this research bases performance evaluations on the Sharpe ratio, a reliable measure of "the statistical significance of earnings and the trade-off between risk and return" [1]. The Sharpe ratio is "the most widely-used measure of risk-adjusted return," a quantity most modern fund managers seek to maximize (rather than raw profits) [8]. Unfortunately, the Sharpe ratio has seen little use in the automated stock-trading literature.

The strategies themselves certainly do set this work apart from previous research. Specifically, we know of no other research applying reinforcement learning to the complete stock-trading task. Moreover, the exact design and parameterization of the trend-following and market-making strategies used in this paper have likely not been tried elsewhere. However, what truly makes this work original are the principled comparisons of the strategies in a novel, more realistic setting, with a relatively uncommon and valuable performance metric.

4 Approach and Assumptions

The generic stock-trading agent architecture used throughout this paper is illustrated in Figure 1. The PLAT competition does not allow share/cash carryover from one trading day to the next. This algorithm is therefore designed to run from 9:30 a.m. to 4 p.m., the normal trading hours, maximizing profits on a single day (lines 1–4) and completely unwinding the share position before market close (lines 5–9). The actual trading strategy is abstracted into the COMPUTE-ACTION routine. Given the system's state, this routine prescribes the withdrawal of some of the previously submitted orders and/or the placement of new orders, each given by a type (BUY or SELL), volume, and price. Sections 5, 6, and 7 explore distinct implementations of this routine.

```
GENERIC-TRADING-AGENT
1  while current-time < 3 p.m.
2      do state ← updated trader, market stats;      action ← COMPUTE-ACTION(state)
3         if action ≠ VOID
4            then place/withdraw orders per action
5  withdraw all unmatched orders
6  while market open                                  ▷ unwind share position
7      do state ← updated trader, market stats
8         if share-position ≠ 0
9            then match up to |share-position| shares of top order in opposite book
```

Fig. 1. Generic agent architecture

Position unwinding (lines 5–9 in Figure 1) works as follows. The agent starts by withdrawing all its unmatched orders. Then, if the agent *owes* shares (has sold more than it has purchased), it places a buy order, one per order placement cycle, for s shares at price p, where s and p are the volume and price of the top order in the sell book. The liquidation of any *owned* shares proceeds likewise. This unwinding method allows for rapid unwinding at a tolerable cost. By spacing the unwinding over multiple cycles, this scheme avoids eating too far into the books. By contrast, a scheme that simply places a single liquidating order is unable to take advantage of future liquidity and possibly better prices.

The fundamental assumption underlying the generic agent design of Figure 1 is that profit maximization and position unwinding are two distinct objectives that the automated trading application can treat separately. Although this task decomposition may be suboptimal, it greatly simplifies automated trader design. Moreover, the profit-maximization strategies proposed in this paper, perhaps with the exception of the approach based on reinforcement learning, hold very reasonable share positions throughout the day, making unwinding feasible at a nominal cost. The time of the phase shift between profit maximization and position unwinding (3 p.m.) was heuristically chosen so as to leave more than enough time for fully unwinding the agent's position.

We have adopted the following experimental methodology in this paper. First, we developed three parameterizable strategies (implementations of the COMPUTE-ACTION routine) and defined several instantiations of each strategy. Next, we evaluated each instantiation separately, in a controlled setting, using SOBI as a fixed opponent strategy. In what follows, we describe only the most successful instantiation of each strategy. Finally, we identified the most successful strategy among these by means of a joint simulation. The live competitions, albeit not controlled experiments, have offered additional empirical feedback.

5 The Reinforcement Learning Agent

Reinforcement learning [9] is a machine-learning methodology for achieving good long-term performance in poorly understood and possibly non-stationary environments. Given the seemingly random nature of market fluctuations, it is tempting to resort to a model-free technique designed to optimize performance given minimal domain expertise and a reasonable measure of progress. A machine-learning approach to this problem is further motivated by the need to *adapt* to the economy (particular mix of opponents, market performance, etc.). A fixed, hand-coded strategy can hardly account for all contingencies.

In its simplest form, a reinforcement learning problem is given by a 4-tuple $\{\mathcal{S}, \mathcal{A}, T, R\}$, where $\mathcal{S}$ is a finite set of the environment's states; $\mathcal{A}$ is a finite set of actions available to the agent as a means of extracting an economic benefit from the environment, referred to as *reward*, and possibly of altering the environment state; $T : \mathcal{S} \times \mathcal{A} \to \mathcal{S}$ is a state transition function; and $R : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ is a reward function. The state transition and reward functions T and R are possibly stochastic and unknown to the agent. The objective is to develop a *policy*, i.e., a mapping from environment states to actions, that maximizes the long-term return. A common definition of return, and one used in this work, is the discounted sum of rewards: $\sum_{t=0}^{\infty} \gamma^t r_t$, where $0 < \gamma < 1$ is a discount factor and r_t is the reward received at time t.

The original RL framework was designed for discrete state-action spaces. In order to accommodate the continuous nature of the problem, we used *tile coding*, a linear function-approximation method, to allow for generalization to unseen instances of the continuous state-action space.

5.1 Strategy Design

Since the transition function T is an unknown feature of the environment meant to be learned by the agent, the design of a trading strategy reduces to the specification of the state-action space and the reward function. After exploring several formulations of the stock-trading problem as a reinforcement-learning task, we adopted the following design:

State Space. The state space is given by a single variable, the price parameter $\Delta p_t = p_t - \overline{p_t}$, computed as the difference between the current last price and an exponential average of previous last prices: $\overline{p}_t = \beta\overline{p}_{t-1} + (1 - \beta)p_t$. The effect of β is to focus the agent on short-term or long-term trends (see Section 5.2 for an experimental study of this effect). The definition of the price parameter as a difference serves a twofold purpose. On the one hand, it gives an indication of the latest market trend: $\Delta p \approx 0$ corresponds to a stationary market, $\Delta p < 0$ corresponds to a decline in price, and $\Delta p > 0$ indicates that the stock price is on the rise. On the other hand, this definition makes the learned policy more general by eliminating the dependency on the absolute values of the prices.

We limited the state space to the price parameter for the following reasons. First of all, share and cash holdings are of no use as state variables: the "right" trading decision is never contingent on these parameters because the agents are allowed to have an arbitrarily large positive or negative share/cash position, and position unwinding is no part of the profit-maximization strategy. For the same reason, a "remaining time" parameter would not be helpful either. Although additional state variables might have been useful, we decided to avoid the corresponding increase in complexity.

Action Space. The action space is likewise given by a single variable, the volume of shares to purchase or sell. We limited the range of this variable to $[-900, 900]$, with negative values corresponding to sell orders and positive values, to buy orders. This trade size is a very generous leash, allowing rapid accumulation of share positions as large as 150,000 shares and beyond. To save a dimension of complexity, we decided against extending the action space to include order price. Instead, we always set the price of an order to the last price, leaving it up to the agent to adjust the demanded volume accordingly.

Reward Function. Ideally, the reward should be computed only once, at the end of the trading day, with zero intermediate rewards assigned at each time step along the way; otherwise, there is a danger that the trader will learn to optimize the sum of local rewards without optimizing the final position ([9], Chapter 3). There are two important complications with this approach. First, it rules out on-line adjustment to the economy. Second, given the complexity of the state-action space (2 continuous variables) and the duration of a simulation (≈50,000 order placement cycles in live mode), the training time requirements of this method seem excessive. Instead, we use a localized reward function, given by the difference in present value (cash holdings plus shares valued at the last price) from the last time step.

5.2 Parameter Choices

We used the Sarsa algorithm [10,11] with the following parameters: $\alpha = 0.04$, $\gamma = 0.8$, $\epsilon = 0.1$, and $\lambda = 0.7$. We have not experimented with varying these values and used them as reasonable general settings. A final parameter that played a substantial role was β, the update rate for computing the exponential average of past prices:

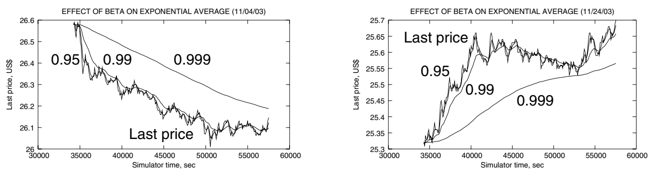


Fig. 2. Effect of β on price average

$\overline{p}_t = \beta\overline{p}_{t-1} + (1-\beta)p_t$. Figure 2 demonstrates the behavior of the average price on two trading days with different price dynamics and β settings. As the graphs indicate, the closer β is to 1, the more "inert" the exponential average, i.e., the less responsive to changes in the price trend. On the one extreme, $\beta = 0.95$ essentially duplicates the last price graph, yielding little information about past price dynamics. On the other extreme, $\beta = 0.999$ yields an average that is not at all representative of the changes in price dynamics. The graphs indicate that a choice of $\beta = 0.99$ offers a nice balance, responding sufficiently quickly to genuine trend reversals and ignoring random fluctuations. We use this informed heuristic choice for β in our experiments, leaving a more detailed optimization with respect to actual performance for future work.

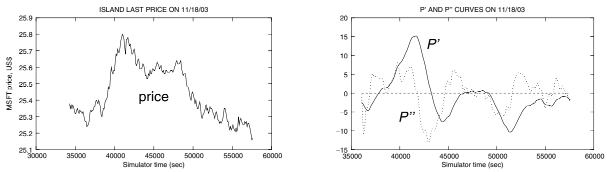
The strategy was trained on 250 historical simulations, each encompassing over 15,000 order placement cycles, for a total of nearly 4 million Bellman backups. This amount of training effort was deemed to provide the agent with sufficient experience. Each simulation involved SOBI as the agent's only opponent. The trading days were a random mix of trading days in October 2003, similar in composition to the handpicked collection of days on which performance was measured. The agent functioned in learning mode (i.e., using the original settings of the learning and exploration rates) during evaluative simulations to allow on-line adjustment to the economy.

6 The Trend-Following Agent

Our second agent uses a trend-following (TF) approach based on price prediction. Unlike reinforcement learning, this approach constructs an explicit model of market dynamics, based on linear regression, to guide order placement. Roughly, the strategy is as follows. If the price is rising (i.e., the slope of the regression line is positive), the agent places buy orders, confident that it will be able to sell the purchased shares back at a higher price. If, on the other hand, the price is falling, the agent will place sell orders. In either case, the agent attempts to unwind its share position just before the price starts to drop (if it is currently on the rise) or just before the price starts to rise (if it is currently on the decline).

The details of the TF approach are best illustrated through an example. Figures 3a and 3b show, respectively, the Island last price on November 18, 2003, and the first and second derivatives[1] P' and P'' of the price function (scaled differently to permit display

[1] As explained below, P' and P'' extract growth information from *series* of price data, to account for its noisy nature. Therefore, P' and P'' are not derivatives in the strict sense of the term since they do not capture instantaneous change, and we abuse this mathematical concept slightly here.



a *b*

Fig. 3. Island last price on 11/18/2003 (*a*) and the corresponding P' and P'' curves (*b*)

on the same set of axes). The value of the P' curve at time t is the slope of the linear regression line computed using the price data for the past hour, i.e., for the time interval $[t - 3600, t]$, where t is expressed in seconds. The length of the time interval presents a trade-off between currency (shorter time intervals generate P' curves that are more responsive to price fluctuations) and stability (longer time intervals generate P' curves that are more "inert" and thus less susceptible to random fluctuations). We used an interval width of 1 hour, the duration of a typical medium-term trend, to balance these desirable characteristics. The purpose of the P' curve is to distill growth and decrease information from the price graph, detecting genuine long-term price trends and ignoring short-term random price fluctuations.

The value of the P'' curve at time t is the slope of the linear regression line computed using the P' curve data for the past 400 seconds, i.e., over the time interval $[t - 400, t]$. The width of the time interval over which the regression line is computed offers the same trade-off between responsiveness and stability; our experiments suggest that the value of 400 seconds offers a good balance. The P'' curve is above the x-axis whenever the P' curve is exhibits growth, and below the x-axis whenever the P' curve is on the decline. Therefore, the $P''(t)$ value changes sign whenever the P' curve reaches a local extremum, signaling a likely trend reversal in the near future. The purpose of the $P''(t)$ is to alert the agent when the price trend is reversed.

Figure 4 presents the trend-following strategy in pseudo-code. We used a trade size of 75 shares, a rather generous limit leading to share positions as large as 150,000 shares. Further increasing the trade size may complicate unwinding. Another essential component of the strategy is the order pricing scheme (lines 1–2). In the pseudo-code,

COMPUTE-ACTION($state$)

1 $sell\text{-}price \leftarrow \max\{last\text{-}price, predicted\text{-}last\text{-}price\}$
2 $buy\text{-}price \leftarrow \min\{last\text{-}price, predicted\text{-}last\text{-}price\}$
3 **if** $P' > 0$ **and** $P'' > 0$
4 **then return** "BUY 75 shares at *buy-price*"
5 **elseif** $P' < 0$ **and** $P'' < 0$
6 **then return** "SELL 75 shares at *sell-price*"
7 **elseif** $share\text{-}position \neq 0$ ▷ reversal, so unwind
8 **then** withdraw unmatched orders
9 **return** "match up to $|share\text{-}position|$ shares of top order in opposite book"
10 **else return** "VOID"

Fig. 4. The trend-following strategy

we defined $predicted\text{-}last\text{-}price = a \cdot t_{curr} + b$, where t_{curr} is current time and $a = P'(t_{curr})$ and b are the parameters of the linear regression line fitted to the price data for the past hour. Our original implementation always stepped a fractional amount in front of the current top order, ensuring rapid matching of placed orders. The current strategy design uses a more cautious pricing scheme that experiments show results in systematically better performance.

7 The Market-Making Agent

As discussed above, the objective of the trend-following strategy is to look for long-term trends in price fluctuations, buying stock when the price is low and later selling stock when the price has gone up (and vice versa with the price going in the opposite direction). As a result, the performance of the strategy is highly dependent on the price dynamics of a particular trading day. If more consistency is desired, an approach based on market making (MM) may be more useful. Unlike the trend-following strategy, the MM strategy capitalizes on small fluctuations rather than long-term trends and is likely to produce a smaller variance in profit.

Our final approach to the stock-trading problem combines the regression-based price prediction model presented in Section 6 with elements of market making. The strategy (Figure 5) still buys stock when the price is increasing at an increasing rate and sells stock when the price is decreasing at an increasing rate. However, rather than wait for a trend reversal to unwind the accumulated share position, the agent places buy and sell orders in pairs. When the price is increasing at an increasing rate, the agent places a buy order. As soon as this *primary* order is matched, the agent places a sell order at price $p + PM$ (the *conditional* order, so called because its placement is conditional on the matching of the primary order), confident that the latter will be matched shortly when the price has gone up enough. The PM (profit margin) parameter is the per-share profit the agent expects to make on this transaction. Our implementation uses $PM = \$0.01$

```
COMPUTE-ACTION(state)
S ← price of top sell order + $0.001 ▷ sell price
B ← price of top buy order - $0.001 ▷ buy price
place qualifying conditional orders
if P' > 0 and P'' > 0
    then create conditional order
            "SELL 75 @ B + PM"
        return "BUY 75 shares at B"
elseif P' < 0 and P'' < 0
    then create conditional order
            "BUY 75 @ S − PM"
        return "SELL 75 shares at S"
else return "VOID"
```

Fig. 5. The market-making strategy

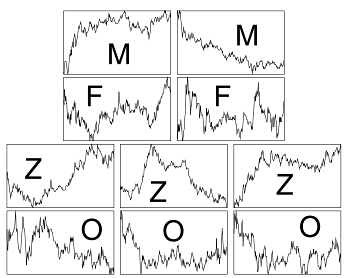


Fig. 6. Price dynamics (stock price vs. time) on the 10 days used for strategy evaluation. Dates (left to right and top to bottom): 11/3, 4, 6, 19, 12, 18, 24, 21, 13, 26. (All dates in Oct. 2003.) Label legend: M="monotonic," F="substantial fluctuation," Z="zigzag behavior," O="mixed (other)."

as a sufficiently profitable yet safe choice. The situation is symmetric when the price is decreasing at an increasing rate. Finally, the agent takes no action during periods designated as "price reversal" by the prediction module (with price increasing/decreasing at a decreasing rate): since the orders are placed in pairs at what is deemed a "safe" time, no additional effort is called for to unwind the share position. The pricing scheme (stepping just behind the top order) is designed to avoid fees for removing liquidity, as discussed in Section 2.

8 Individual Assessment

Our controlled experiments in this section and Section 9 use a set of 10 trading days carefully selected to represent typical price dynamics (Figure 6), namely, monotonic decrease/increase, substantial fluctuation, and zigzag and mixed behaviors. The graphs in Figure 6 are scaled differently and convey only the shape of the price curves. Each graph is labeled by a symbol denoting the price behavior, with a legend given in the caption. Table 1 displays the raw profit/loss of the RL, TF, and MM strategies in individual simulations against SOBI, with days labeled by price behavior (the labels are taken from Figure 6). The bottom row gives each strategy's average profit/loss over the 10 days, a measure of overall efficacy. In this section, the strategies were allowed to run through 4 p.m., i.e., the unwinding code (lines 5–9 of Figure 1) was omitted and the final score was computed as present value (cash holdings plus shares valued at the closing price).

Table 1. Individual assessment of RL, HC, and MM vs. SOBI

Price	RL vs. SOBI		TF vs. SOBI		MM vs. SOBI	
M	11134	-21935	-4015	-29686	529	-30286
M	45680	-56308	-3591	-44216	972	-52255
F	-5142	55710	-4292	108476	-471	117192
F	-50529	17464	-1533	19958	1131	24908
Z	-69683	230715	-4390	155539	-518	154082
Z	358774	96387	3163	32383	-3370	15605
Z	-284563	-11059	-479	-1964	744	-2417
O	49621	-13805	-5494	-12063	654	-22632
O	3407	25026	-4139	118016	638	85099
O	2302	29015	-4692	23098	1224	27467
Ave	6100	35121	-2946	36954	153	31676

Table 2. RL, TF, MM, and SOBI in a joint simulation

Date	RL	TF	MM	SOBI
11/03/03	-7314	-2659	692	550
11/04/03	-40712	-1623	1087	-23999
11/06/03	-10980	-2119	-13	51432
11/12/03	-160178	-1159	-1321	99489
11/13/03	-20981	-430	684	43088
11/18/03	-209277	6045	-1300	75569
11/19/03	-22747	-3469	108	15550
11/21/03	28345	-3677	735	-6216
11/24/03	-992	90	1081	-2289
11/26/03	19299	-4776	259	22295
Average	-42553	-1377	201	27546
Std. dev.	78395	3012	879	39273
Sharpe Ratio	-0.5428	-0.4573	0.2290	0.7014

8.1 Reinforcement Learning

RL by far outperforms SOBI on the two days with monotonic price behavior. On days with substantial fluctuation in price, SOBI is profitable and RL loses money. Finally, the two strategies exhibit roughly comparable performance on the days with zigzag and mixed price behavior, each finishing 4 days in the black and losing money on the 2 other days.

RL's performance under different market conditions is a direct consequence of the problem formulation as a reinforcement-learning task. The strategy is profitable on both days with steady price growth/decline, a success owed to the price difference parameter that recognizes market trends, and an indication that learning and adaptation do take place. Such a parameter is not particularly valuable on days with substantial fluctuation because trends are short-term and trend reversals are frequent. The concluding 6 days (zigzag and mixed behavior) are much more auspicious for the strategy because the market trends last significantly longer, accounting for RL's profitability on most of the days.

It is no doubt encouraging to see RL, a strategy evolved by a generic machine-learning technique with minimal domain expertise, perform overall comparably with SOBI, a hand-coded approach requiring a firm grasp of stock trading. On the other hand, the experiments reveal much room for improvement under certain price dynamics, in part due to the difficulties of adapting RL methodology to the stock-trading domain. A major problem is the exogenous nature of the transition function: when the agent places an order, it cannot control when the order will be matched, if at all. The reward function is oblivious to this fact, attributing any change in present value, which may well be due to random price fluctuation, to the last action taken. This misattribution of reward is likely to present a great impediment to learning.

A different and much more successful RL-based approach to trading in a continuous double auction setting such as the stock market is reported in [12]. That method computes a belief function (a mapping from bid and ask prices to the likelihood of a trade) based on recent market behavior and then uses dynamic programming (with the belief function serving as the market model) to compute an optimal order. An approach of this type would be readily implementable in the PXS framework, which makes complete order-book data available. This alternative formulation shifts the entire learning challenge from the RL agent to a non-RL analytical subsystem that constructs a trade probability model, leaving to the agent only a straightforward recursive computation. In contrast, we relied on the RL agent to learn the task from scratch.

Yet another research avenue to consider is direct (policy-search) RL methods. It has been argued [8] that these methods help avoid the search space explosion due to continuous variables and learn more efficiently from the incremental performance indications in financial markets (as opposed to the delayed-reward domains in which value-based methods have excelled).

8.2 Trend Following

As expected, TF beats SOBI on the days with monotonic price behavior by avoiding large positive share positions when the price is declining or large negative share positions when the price is increasing. SOBI is far more profitable on days with substantial fluctuation because it does not rely on longer-term price trends. On the days with zigzag and mixed price behavior, TF wins a third of the time. TF's strongest performance on a day with zigzag price behavior jibes well with the intuition that TF should perform best under price trends of medium duration: shorter trends diminish the value of prediction, while longer ones often contain aberrations that trigger premature unwinding.

With a single profitable day, TF's performance is disappointing. TF is the only strategy in Table 1 with a negative average profit/loss. In fact, additional analysis reveals that

TF often steadily loses value throughout the day. We have experimentally verified that this is not due to a problem with timely unwinding. Specifically, when we incorporated periodic unwinding in the above design (ensuring that the agent keeps its share holdings to a moderate amount instead of relying on an advance warning of a trend reversal from the prediction module), we observed no change in performance. Our understanding is that, on the contrary, the prediction module generates too many false alarms, triggering premature unwinding.

8.3 Market Making

MM's results are very encouraging. The agent is profitable on 70% of the days. MM performs very well on days with monotonic and mixed price behaviors. Days with zigzag price behavior seem to present a problem, however. One explanation is that the conditional orders, whose primary counterparts match just before the extremum, are placed and never matched due to the unfavorable change in price; the resulting share imbalance is never eliminated and severely affects the agent's value. In terms of raw profitability, MM wins 4 of the 10 simulations. However, MM's profits seem far more consistent, a claim we quantify in Section 9.

The market-making approach shows great promise. Neither the reinforcement learning nor trend-following approach come close to rivaling MM's profit consistency. An important extension for MM to be viable in practice would be an adaptive mechanism for setting the trade size and profit margin, both highly dependent on the economy. A further nuance is that there is an inherent trade-off between these parameters. If the agent trades large volumes, it will have to accept narrow profit margins or else see its conditional orders unmatched; if the agent trades little, it can afford to extract a more ambitious profit per share.

9 Comparative Analysis

Table 2 contains joint-simulation performance data for every strategy presented above and every trading day. This time, the strategies ran through 3 p.m., at which point control was turned over to the unwinding module (lines 5–9 of Figure 1). Each strategy finished every trading day with zero share holdings. We used the PLAT scoring policy and performance criterion (Section 2). RL and TF were largely unprofitable, finishing with a negative score on 8 of the days.[2] RL's performance was particularly poor, as the large negative scores indicate, presumably because its training experience did not incorporate key features of the joint economy. MM and SOBI, on the other hand, were consistently winning, finishing with a positive balance on 7 of the days. Of the four strategies, SOBI's scores are the most impressive.

The bottom row of Table 2 shows the Sharpe ratios for each strategy in the joint simulation. SOBI wins with the highest Sharpe ratio, followed by MM, TF, and RL. It is

[2] Given RL and TF's large losses, it has been speculated that "reversing" their trading recommendations would have yielded a profitable strategy. In general, this claim is unwarranted due to the limited liquidity provided by a handful of other agents; it is impossible to predict how the other traders would have reacted to the new orders.

noteworthy that MM, generating profits that are a tiny fraction of SOBI's, finished with a Sharpe ratio quite close to SOBI's. This fact is due to the emphasis on consistency built into the Sharpe ratio. An important lesson to learn from this comparison is that if the Sharpe ratio is the primary criterion, large profits are not strictly necessary for placing in the top ranks; a consistent strategy that generates small profits will be a strong contender. Therefore, we decided to use MM in the live competition.

It is certainly disappointing that RL, the most innovative of the three approaches, did very poorly both in an absolute sense and in comparison with the other strategies. However, the focus of this study was not a quest for a novel stock-trading algorithm but a comparative evaluation of strategies with the purpose of selecting one as a competition entry. At the same time, the results above help explain stock traders' preference for market making and similar well-tried methodologies over original machine-learning techniques.

10 Live Competition Results

The market-making strategy proved best of the three strategies in off-line experiments. But of course one of the three had to prevail. The true test of this research was how the chosen strategy would do in an open competition with agents created by many other people also trying to win.

The exact MM strategy we used in the live competition differed from the original design of Figure 5 in two respects. First, the trade size was scaled down (from 75 to 15) to account for the higher order-placement frequency in live mode. Second, the primary sell and buy orders were priced at the last price plus and minus profit margin, respectively; the corresponding conditional orders were priced at the last price. This more cautious pricing scheme gives a greater assurance that, if a primary order matches, its corresponding conditional order will match as well, avoiding costly share imbalances.

Participants in the December 2003 and April 2004 PLAT live competitions were divided into two separate economies. Tables 3 and 4 summarize the performance of MM and the 5 other strategies in its group (labeled #1 through #6, in order of final rankings). The top 10 rows show daily scores, and the bottom row shows the Sharpe ratios. MM fully justified our hopes in December 2003, exhibiting steady profitability on every single trading day and attaining the highest Sharpe ratio (no agent in the other group attained a positive Sharpe ratio). As to MM's competitors, agents #2, #3, and #4 were highly profitable, routinely realizing profits in the thousands. MM's profits were an order of magnitude smaller but far more consistent than its opponents', resulting in the highest Sharpe ratio. In April 2003, MM's performance fell short of its December victory. However, the agent achieved a satisfactory Sharpe ratio and placed second, exhibiting profitability on 8 of the days and suffering minor losses on the other two. Overall, MM has attained an 18-day record of profitable and consistent performance. Details of the competition, including complete results, are available from the PLAT website.[3]

[3] The competition results are currently at `http://www.cis.upenn.edu/~mkearns/projects/newsandnotes03.html`. The main project page is `http://www.cis.upenn.edu/~mkearns/projects/pat.html`.

Table 3. Dec. 2003 live competition results

Date	MM	#2	#3	#4	#5	#6
12/9	135	-7447	-4106	4034	-56731	-7E+5
12/10	381	3006	-3254	3625	-6E+5	-7E+5
12/11	436	1365	5971	1251	196	-6E+5
12/12	140	848	322	-986	-2E+5	-7E+5
12/13	62	2536	1334	1286	-1E+5	-5E+5
12/16	439	3716	3940	3129	18227	-7E+5
12/17	359	3501	7924	433	10873	-7E+5
12/18	411	1037	2163	1389	0	-5E+5
12/19	430	4617	-119	-9512	0	-5E+5
12/20	679	1692	-64	2148	599	-6E+5
Ave	347	1487	1411	680	-98167	-6E+5
St. dev.	185	3378	3772	3887	2E+5	84772
Sharpe	1.88	0.44	0.37	0.17	-0.48	-7.32

Table 4. Apr. 2004 live competition results

Date	#1	MM	#3	#4	#5	#6
04/26	3433	271	1045	1307	-4655	-9E+6
04/27	1374	538	4729	2891	-1370	-8E+6
04/28	2508	-242	243	-1563	2178	-7E+6
04/29	2928	-248	-6694	-1349	2820	-8E+6
04/30	3717	13	12508	-1339	2766	-8E+6
05/03	3444	636	11065	3230	2961	-7E+6
05/04	1322	386	-2377	1850	2665	-8E+6
05/05	3300	452	5708	2037	-5746	-8E+6
05/06	2199	461	9271	2465	2402	-8E+6
05/07	966	121	11755	1041	-2545	-8E+6
Ave	2519	239	4725	1057	148	-8E+6
St. dev.	1009	316	6551	1829	3413	+6E+5
Sharpe	2.50	0.76	0.72	0.58	0.04	-12.59

11 Conclusions and Future Work

This paper documents the development of three autonomous stock-trading agents. Approaches based on reinforcement learning, trend following, and market making are presented, evaluated individually against a fixed opponent strategy, and analyzed comparatively. A number of avenues remain for future work. The reinforcement learning approach calls for an improved reward function. The trend-following agent needs a more accurate price prediction module that would eliminate premature unwinding due to a perceived trend reversal. The highly successful market-making strategy would further benefit from automatic adjustment of the trade size and profit margin as a function of the economy.

This study confirms that automated stock trading is a difficult problem, with reasonable heuristics often leading to marginal performance and small-profit strategies proving competitive, according to important metrics, with highly profitable strategies. While the area of stock trading has received much attention in the past, the unique opportunities and challenges of up-to-date order book information and of electronic share exchanges, as exemplified in part by the presented approaches, merit further study.

Acknowledgments

We thank Professor Michael Kearns and his PLAT group at the University of Pennsylvania for their development of the PXS simulator, for allowing us to participate in the competitions, and for their continual and prompt support. This research was supported in part by NSF CAREER award IIS-0237699 and an MCD fellowship.

References

1. Kearns, M., Ortiz, L.: The Penn-Lehman automated trading project. In: IEEE Intelligent Systems. (2003) To appear.
2. WWW: Island ECN. (`http://www.island.com`)

3. Chan, N., Shelton, C.R.: An electronic market maker. Technical Report 200-005, MIT Artificial Intelligence Laboratory (2001)
4. Das, S.: Intelligent market-making in artificial financial markets. Technical Report 2003-005, Massachusetts Institute of Technology AI Lab/CBCL (2003)
5. Feng, Y., Yu, R., Stone, P.: Two stock-trading agents: Market making and technical analysis. In: Agent Mediated Electronic Commerce V: Designing Mechanisms and Systems. Volume 3048 of Lecture Notes in Artificial Intelligence. Springer Verlag (2004) To appear.
6. Yu, R., Stone, P.: Performance analysis of a counter-intuitive automated stock-trading agent. In: Proceedings of the 5th International Conference on Electronic Commerce, ACM Press (2003) 40–46
7. WWW: Common stock-trading strategies. (`http://www.cis.upenn.edu/~mkearns/projects/strategies.html`)
8. Moody, J., Saffell, M.: Learning to trade via direct reinforcement. IEEE Transactions on Newral Networks **12** (2001) 875–889
9. Sutton, R., Barto, A.: Reinforcement Learning: An Introduction. MIT Press, Cambridge, MA (1998)
10. Rummery, G.A., Niranjan, M.: On-line Q-learning using connectionist systems. Technical Report CUED/F-INFEG/TR66, Cambridge University Department (1994)
11. Sutton, R.S.: Generalization in reinforcement learning: Successful examples using sparse coarse coding. In Tesauro, G., Touretzky, D., Leen, T., eds.: Advances in Neural Information Processing Systems 8, Cambridge, MA, MIT Press (1996) 1038–1044
12. Tesauro, G., Bredin, J.L.: Strategic sequential bidding in auctions using dynamic programming. In: First International Joint Conference on Autonomous Agents and Multi-Agent Systems, Bologna, Italy (2002)

Specifying and Monitoring Market Mechanisms Using Rights and Obligations

Loizos Michael, David C. Parkes, and Avi Pfeffer

Division of Engineering and Applied Sciences, Harvard University,
Cambridge, MA 02138, U.S.A.
{loizos, parkes, avi}@eecs.harvard.edu

Abstract. We provide a formal scripting language to capture the semantics of market mechanisms. The language is based on a set of well-defined principles, and is designed to capture an agent's rights, as derived from property, and an agent's obligations, as derived from restrictions placed on its actions, either voluntarily or as a consequence of other actions. Rights and obligations are viewed as first-class goods, from which we define fundamental axioms about well-functioning market-oriented worlds. Coupled with the scripting language is a run-time system that is able to monitor and enforce rights and obligations. Our treatment extends to represent a variety of market mechanisms, ranging from simple two-agent single-good exchanges to complicated combinatorial auctions.

1 Introduction

Many authors have written about a future of agent-mediated electronic commerce, in which agents engage in commerce on behalf of individuals and businesses. We take this idea seriously, and provide a formal scripting language for describing economic markets that is: (i) natural and easy to understand, for humans to be able to participate, (ii) formal and unambiguous, for artificial agents to be able to participate, and (iii) amenable to automatic monitoring.

The need for a formal method to describe markets in a computer-compliant yet human-friendly way naturally arises in a variety of contexts. Most prevalent is that of online transactions between agents, including both humans and artificial bidding agents. An equally important context is the need for a platform for testing new agent designs, simulating new mechanism designs, and evaluating their properties. Our framework provides such a platform. We implement a set of well-defined design principles and enable the specification of platforms for describing and monitoring market mechanisms.

The scripting language we propose captures the essential semantics, namely *rights* and *obligations*, of market mechanisms. Rights enable agents to obtain utility by taking actions on goods that they own, while obligations allow them to engage in safe transactions and to make credible commitments to the rules of market mechanisms. We adopt rights and obligations as first class goods, from which fundamental market axioms can be derived. These axioms are enforced

P. Faratin and J.A. Rodríguez-Aguilar (Eds.): AMEC 2004, LNAI 3435, pp. 188–201, 2005.

within a monitoring environment that we couple with our formal scripting language. Given a description of a market mechanism, the monitoring environment implements the market in a prescribed way, thus giving precise semantics to the scripting language.

Agents can interact with the monitoring environment and affect, through their actions, the state of the virtual market. During such an interaction, agents themselves can initiate new market mechanisms by specifying obligations on their behavior and granting rights to participants (e.g., the right to place bids).

We take a *black-box* approach to the specification of agents and impose no restrictions to their design and internal workings. As a result of this approach, the monitoring environment is freed from complex activities such as planning for agents and winner-determination in auctions. The monitoring environment can instead *verify* whether certain goals are established by having agents state obligations and then provide sufficient information to enable their easy verification. For instance, an auctioneer can provide market-clearing prices to allow the monitoring environment to check that the outcome satisfies a competitive equilibrium, without the need for the system to compute that equilibrium itself. Thus our approach provides a middle road between a completely formal but hard to program system, and a completely open-ended but informal system.

The framework is introduced through a discussion of its main characteristics and capabilities. At the end of the paper we demonstrate its flexibility through a number of examples, including an English auction, a second-price auction, and a combinatorial auction.

1.1 Related Work

Our approach is consistent with economic theory on property rights and organization theory. Quoting Tirole [13], "*a decision right or authority granted to a party is the right for the party to pick a decision in an allowed set of decisions. A property right on an asset, i.e. its ownership, is a bundle of decision rights.*" It is standard to model a firm as a collection of assets and consider the ability of a firm to retain a specific subset of its bundle of rights while selling all other residual rights [5]. The role of obligations and commitment is recognized to be important for efficient contracts [5], and for auction and mechanism design [6].

Prior work in multi-agent systems has considered the role of rights and obligations for the specification and semantics of open systems [1,3,4,12,15], with approaches differing in whether the monitoring environment actively enforces sanctions (as in our work) [1,4], or only passively maintains the global state and informs agents of their obligations [3]. Approaches also differ as to whether obligations are state-based (as in our work) [3], or specified in terms of actions that an agent must perform in a particular state [1]. Some work [3,12] observes that agents might contract other agents to satisfy the formers' obligations, but none of this work adopts rights and obligations as first class goods that agents can explicitly trade and exchange. Similarly, we are unaware of any work that explicitly sets out to model the rights that derive from goods in economic worlds or the semantics of ownership and possession.

Our notions of conditional, limited, and disjunctive rights are shared with previous work on formal specification languages for financial contracts [7], although that work focuses on the formal description and analysis of new forms of financial contracts and not on providing frameworks for open agent societies. The π-calculus has also been used for the specification of a complex model of a Spanish fish market [10], although again the goal in that work was to assist with the development of complex institutions rather than provide semantics for participants or monitor and enforce properties of dynamic state.

The formal theory of deontic logic [8,9], the logic of rights and obligations, is concerned with performing valid inference in high-level deontic logics, seeking to establish the validity of statements such as "is every obligatory action permitted?" A duality between rights and obligations provides a cornerstone of deontic logic, with an obligation defined as an action that must be performed when no other action can be taken, due to lack of rights. Our work differs in this aspect, by defining rights on actions, but obligations in terms of properties on states. We adopt *soft obligations* with sanctions rather than hard obligations, an approach termed "contrary-to-duty" in the deontic logic literature [9]. In particular, our agents can make mistakes and take actions that lead them to dead-ends in which their obligations cannot be met.

2 Framework Overview

In this work we propose a framework comprised of a scripting language and a monitoring environment, with the former providing the necessary syntax for describing economic markets, and the latter providing the language semantics. This is analogous to the case of programming languages, where a programmer uses the language to write a program, with the semantics defined through the program's execution in a prescribed manner. The programmer in our framework is the *domain designer*, and the program is the *domain description*, a collection of laws governing the particular economic market being modeled.

As ordinary programs can import libraries that provide specific functionality, so is the case with domain descriptions. The domain designer can import libraries describing economic market laws that are commonplace in a variety of settings. We have written such libraries, such as: a library on "exchanges of goods" with laws on how goods can be traded, given, or sold between agents; a library on "handling rights and obligations" with laws on how rights can be given up, issued, or revoked, and how obligations can be taken on, imposed, or cleared.

A domain description is fed into the monitoring environment. The monitoring environment then runs a virtual market world governed by the laws specified in the domain description. The laws define the initial state of affairs of the market, the objects that populate it, and the relevant properties of these objects, whose values determine the state of the market over time. The laws also dictate how agents might join or leave the market, and the available actions through which the agents might affect and observe the market's status. The agents are not simulated as part of the virtual world, but they are instead acting independently and only communicate with the monitoring environment.

It is clear that every implementation of a given market may lead to a different sequence of states describing the evolution of the virtual world. Each such sequence is called a *scenario* and corresponds to a specific instantiation of an economic market. The actual scenario that occurs is ultimately defined by the actions taken by the participating agents.

2.1 Design Principles

Our framework implements a set of well-defined design principles, which we discuss below:

Black-Box Principle: Agents are entities that exist outside our framework, implemented in some fashion that is (possibly) independent of the proposed scripting language. They can reason based on their own beliefs and freely choose to take actions or not, within the market world they participate in.

Free-Will Principle: We cannot force agents to take specific actions, and in particular, to take actions that satisfy their obligations. Instead, we impose punitive sanctions to agents that fail to meet their obligations.

Restriction Principle: The monitoring environment is able to restrict the execution of actions for which appropriate rights are not held by the agents.

Soundness Principle: When an action is actually executed (i.e., when the appropriate right was held and the invoked action was physically executable in the current state of the virtual world), its effects are produced in accordance with the laws of the economic market being modeled.

The first two principles exemplify the generality of the framework we propose. Agents are treated as black boxes, without imposing any requirements on their internal workings other than their ability to interact with the provided interface. We cannot force agents to act in a prescribed way. This justifies the approach of using punitive sanctions, an approach that we follow in this work.

Our Restriction Principle is justified because the agents can only request that actions be taken. The final decision lies with the monitoring environment, which screens the action execution requests based on agent rights. This principle is further supported, when viewed in conjunction with the Free-Will Principle: An agent's options can be limited by the monitoring environment, but the agent still retains the choice of which (if any) option to exercise. This is the situation faced by any agent trying to devise a plan to achieve some goal. The actions available to the agent are restricted (albeit not by lack of rights, but rather by lack of physical ability). The agent itself is responsible for choosing appropriate actions that will fulfill its goal.

Finally, the Soundness Principle provides an appropriate soundness condition for our framework. It states that the monitoring environment will always respect the laws of the market, as defined by the domain designer.

2.2 eBay Example

In this section we demonstrate the parallels between eBay and our framework, illustrating how our stated principles are justified by existing virtual markets.

eBay participants freely choose to join or leave eBay's virtual world. Participants interact with the market by invoking actions (e.g., define upper bounds on bids) through the provided interface, and depending on whether certain conditions are met (e.g., if the bid was actually a valid numerical value) the actions are executed and produce their effects (e.g., the input value becomes the agent's new upper bound for the proxy bidding). The agents then observe the new state of affairs, and continue to invoke their next action (if any).

Notice how the Black-Box Principle applies here, with the agents being independent of the engine (i.e., eBay's servers) running the virtual world, and that the only requirement they need to meet is that they can interact with eBay's market through the provided interface (e.g., web page links and forms). The Free-Will Principle applies in particular, with eBay not forcing a participant to honor a transaction, but punishing violators by means of negative feedback.[1]

The Restriction Principle relates to how eBay participants can auction items, or bid on items, only when they have appropriate rights. Thus, a participant does not have the right to place a bid lower than the current highest bid; invoking such an action will result in the action being rejected and not executed.

Lastly, the Soundness Principle applies, for instance, in that a paying agent is guaranteed that if the paying action is executed, then the payment will be made, no matter what other events (e.g., the concurrent execution of some other payment, or the fact that some auction closed one hour ago) take place.

3 Rights and Obligations

Agents pursuing goals, either by choice, or as an imposed requirement, often have restrictions on their available options of how to meet these goals. These options and goals correspond precisely to the notions of rights and obligations, which play a prevalent role in our framework. We view rights and obligations as tradable goods that can be given, taken, exchanged, sold, or auctioned. As such, rights and obligations are treated as any *first-class object* with a set of predetermined properties whose values are part of the market's state.

We contend that viewing rights and obligations as goods is necessary for defining natural economic market protocols, and we illustrate the strength of such a treatment via a number of example domain descriptions later on. To give a taste of why this idea is in fact very powerful, consider the situation of an audio compact disk being sold in an eBay auction, with the winner being awarded the item as per eBay's rules. What is implicit in this transaction is that the winner is also awarded the rights and obligations accompanying the item, and in particular, the right to listen to the audio compact disk, and the obligation not to infringe the copyright of the producers of the music. That is, in

[1] From `http://pages.ebay.com/help/confidence/programs-investigations.html`: *"eBay cannot force a seller to honor their transactions. You should leave appropriate feedback for the reluctant seller..."*. The action of providing negative feedback is taken by participants, but it is ultimately provided by eBay as a sort of a punitive sanction.

reality the auctioneer was not simply selling an audio compact disk, but rather a bundle of goods that includes the item itself and certain rights and obligations.

We view rights as the options of an agent participating in a market mechanism, coming from property or possession of items, or otherwise given to the agent. Rights determine the actions that an agent can take as a means of fulfilling its own private goals, by qualifying the executability of actions. In their full generality rights are conditional, with their provisions being applicable only under certain conditions.

Definition 1 (Rights). *We let* right(#action,#condition) *denote the right to execute action* #action *whenever condition* #condition *is true.*

As an example, an agent renting a car from 10:00am to 6:00pm might be given a right of the form *right(drive_car, 10:00am* $\leq$ *Time* $\leq$ *6:00pm).* If the condition is not met (and given that the agent does not have any other rights on driving the car), then the agent cannot drive the car, virtue of the Restriction Principle. The syntax of conditional rights is sufficiently expressive to account for perpetual and expiring rights, and for more involved rights, such as the perpetual right to buy bonds, but only once every year and within a limited time span.

On the other side we have obligations, which we view as constraints on an agent's behavior, or goals an agent should fulfill as a participant in some market mechanism. An agent freely chooses when and how to satisfy its obligations by appropriately exercising its rights, in the spirit of the Free-Will Principle. Rather than enforcing (via planning) that agents meet their obligations we let the monitoring environment detect violations and appropriately impose punitive sanctions, as defined by the domain designer or the participating agents. Sanctions might include the revocation of an agent's rights, the loss of money or possessions, the enforcement of additional obligations, or the banning of an agent from participating in the market mechanism altogether.

Definition 2 (Obligations). *We let* obligation(#satisfy, #violate, #punishment) *denote the obligation of ensuring that condition* #satisfy *is satisfied no later than condition* #violate*, under penalty of executing action* #punishment*.*

The obligation is flagged as satisfied or violated according to which condition is met first, and in the case of a violation the appropriate punitive sanction is imposed through the execution of action *#punishment*. This general form allows us to represent obligations of the following forms

obligation(false, balance(alice)<1000, close_account(alice))

obligation(balance(alice)>1000, Time>12/31/2004, deduct_account(alice, 100))

Assuming that Alice holds such obligations, then in the first case she must ensure that her bank balance does not drop below 1000 dollars at any time, under penalty of her bank account being closed, while in the second case she must ensure that her bank balance goes above 1000 dollars (but not necessarily stays there) at some time before the end of the year, under penalty of 100 dollars being deducted from her account.

4 Objects, States, Actions

The dynamic model of our monitoring environment is fairly standard. The world goes through a sequence of states, with each state specifying values for the properties of certain objects that populate the state. Each object is associated with a class (like in object oriented programming), of which the object is an instance, and which defines the set of properties of the object. Every object has a unique name used by agents to reference that object. A set of basic classes are defined by our framework, but the domain designer can extend this set.

The use of objects provides a uniform treatment for both physical goods, like apples, and abstract goods, like rights and obligations. Money is supported through the use of *account* objects, with an object property corresponding to the account balance. Transferring money through payments is equivalent to changing this balance in an appropriate way. The notions of ownership and possession are also readily supported as properties of objects. Transferring an item from some agent to another reduces to simply changing the values of these properties.

The properties and existence of these objects are only affected by means of actions taken by the agents, through their interaction with the monitoring environment. In their *primitive* form, actions have preconditions and effects. When the monitoring environment attempts to execute an action, following its *invocation* by an agent, it first checks whether the agent holds an appropriate right, and whether the preconditions of the action are satisfied, and subsequently updates the state according to the action's effects. The set of effects is as follows.

Definition 3 (Effects). *We let* create(#object,#class)*,* destroy(#object)*, and* set(#object,#property,#value) *denote respectively the action effects that create object* #object *as an instance of class* #class*, destroy object* #object*, and set the property* #property *of object* #object *to the value* #value*.*

Our framework also supports more expressive conditional and quantified effects, special instances of which are the non-deterministic, or probabilistic effects.

In their *transactional* form, actions are ordered sequences of actions (which may be primitive or transactions themselves). As in databases, when executing a transaction, either all or none of the actions are successful. The execution model is to execute each of the actions in turn, updating the world state after each action. If any primitive action fails to meet its preconditions, or the agent fails to have an appropriate right at the time of each primitive action's execution, the entire execution is rolled back to its original state. Conceivably the first action can grant or revoke an agent's right to execute a subsequent action in the transaction, allowing for a really expressive set of transactions to be modeled.

The notion of transactions is very powerful and useful, with a number of applications, like that of implementing safe exchanges of goods. The transaction

sell(apple,1,alice) ≡ transaction([give(apple,alice),take_money(1,alice)])

for instance, specifies a fail-safe way for Bob to sell his apple to Alice for one dollar, without either party being vulnerable to the other's reneging. Transactions can also be used in a number of other contexts, including that of implementing disjunctive or expiring rights, where the agent with the disjunctive/expiring right essentially has the right of executing a transaction comprised of the action indented to be executed, and followed by the agent giving up the right.

A certain set of basic primitive actions and transactions are implemented by our framework, including actions for transferring ownership or possession of goods, transferring money, issuing or giving up rights, taking on obligations, etc. In addition to the built-in actions, the domain designer may define actions specific to the domain being modeled, like, for instance, the *fill_gas(#car)* action whose effect is that of setting the *gas_level* property of the *#car* object to *full*.

We capture exogenous events that are outside the agents' control by allowing the monitoring entity to execute certain actions and attributing their execution to an all-powerful "god" agent. Thus, for instance, the initial state of the system is populated by means of the god agent executing the *initialize* action once the domain description is loaded. Although the god agent can execute only a certain fixed set of actions, on a well-defined set of occasions, these actions are in general transactions, with their constituent actions being defined as part of the domain description. This allows the domain designer to essentially specify the effects of god's interventions, in situations like the arrival or departure of an agent, or the violation of some obligation, in which case the god agent executes the punitive action associated with the violated obligation.

An important special action implemented by our framework is that of querying, which serves as a way to implement private information. We treat the values of object properties as been hidden from an agent, unless the agent has an appropriate right to query an object property for its value.

Definition 4 (Query). *We let* query(#object,#property) *denote the action of querying the value of property* #property *of object* #object*. When the action is executed, the agent that invoked the action learns the queried value.*

For example, the property representing the collected bids in a sealed-bid auction is only viewable by the auctioneer, thus preserving secrecy.

5 Ownership and Possession

Property rights are a basic building block of markets and our framework takes a stand on what the rules governing these rights should look like. To start with, we make an important distinction between ownership and possession. Ownership of an object implies a bundle of rights, including the right to use the object and the right to sell it. It also includes the right to sell various rights to the object. For possession, we use the word *holding*, which we take to mean rightful possession. When one holds something, one has the ability to use it through possession, and one also has the right to use it. However, one does not have the right to sell it or to sell any rights to it. This is a common status in the real world. For example, when someone rents a car, he has possession of it and the right to use it (for a limited time), but he does not have the right to sell it. We make precise the notions of ownership and possession through the following axiomatic definitions.

Definition 5 (Ownership Axiom). *We take* ownership of a good *to be synonymous with owning the right of setting the properties of the good to any values physically possible. We call this the* Fundamental Axiom of Ownership*.*

Our framework implements the Fundamental Axiom of Ownership by issuing ownership of a right of the form

right(#action, accessible(#action, #agent))

to every agent *#agent* joining a virtual market, where *accessible(#action, #agent)* holds exactly when action *#action* only affects properties of goods owned by agent *#agent*. By exercising this right, the owner of an apple can sell or give possession of the apple, since the effects of these actions are only affecting the *owned_by* and *held_by* properties of the apple. Notice that in the latter case, the owner can actually take the apple back, since he still owns the right of setting the possessor of the apple. In particular, this implies that an agent owning a right, but not holding it, can still rightfully execute an action, since the agent can always reclaim possession of the right, execute the action, and then return the right to its previous possessor, all within a single transaction.

Definition 6 (Possession Axiom). *We take* possession of a good *to imply possession of the right to use the good in a set of prescribed ways associated with the good's class. We call this the* Fundamental Axiom of Possession*.*

As before, our framework implements the Fundamental Axiom of Possession by issuing possession of a right of the following form to all participating agents

right(#action, (object(#object), value(#object,
[(held_by, #agent), (uses, #uses)]), member(#action, #uses)))

where *object(#object)* holds exactly when object *#object* exists, and *value (#object, [(#property, #value),...])* holds exactly when the property *#property* of object *#object* has value *#value*, for every property-value pair in the list.

Notice that the rights associated with the Fundamental Axioms of Ownership and Possession are respectively owned and simply held by agents. Hence, in the former case the Fundamental Axiom of Ownership applies recursively on the associated right itself with the right being the owned object. So, an agent owning a car, not only owns the right to drive it, but the agent also owns the right to sell the right to drive the car, to some other agent. Selling the right to use an object without selling the object itself is extremely common. For example, you might sell someone the right to walk across your land without selling the land.

Rights in real life are often not given, but rather issued. When you give someone the right to walk on your land you still retain that right for yourself, exactly because you do not give that person your instance of the right, but rather you issue a new copy of the right. This is achieved through the use of an issuing action defined by our framework, and appeals to the following axiom.

Definition 7 (Rights Axiom). *We take* ownership of a right *to imply ownership of the right to issue ownership or possession of the former right (with non-weaker conditions) to others. We call this the* Fundamental Axiom of Rights*.*

Other fundamental axioms are also defined in our framework, such as axioms relating to performing transactions (e.g., giving someone the right to exchange goods with you). All axioms are implemented by issuing suitable rights to agents.

6 Implementation Issues

Both the monitoring environment and the specification language are currently implemented in Prolog, whose goal-oriented computation is a natural fit with the computational tasks of our framework (e.g., checking if conditions are met).

Agents joining the monitoring environment are assigned a private channel, through which all subsequent communication is taking place, thus associating each exchanged message with a unique agent. Communication is taking place asynchronously, while the monitoring environment employs a continuous treatment of time, with actions occurring instantaneously.

In a typical execution, an agent is sent a Prolog list containing all the object properties of the current state that are visible to the agent. Given the received message, the agent reasons and chooses to invoke some action by replying with the predicate *invoke(#action)*. The monitoring environment records the invocation event and attempts to execute the action. Success or failure of actions is recorded and the state of the virtual world is updated and stored in a database that can be later used to review the evolution of a scenario. Periodically, the monitoring environment checks whether any obligation has been satisfied or violated, recording the event and enforcing the appropriate punitive sanction.

Regarding the scalability of our framework, we note that we are not concerned with the problem of planning, but rather with that of execution monitoring; the latter remains decidable and tractable as long as the conditions of actions, rights, and obligations are not inherently undecidable or intractable to begin with. Preliminary experimental results using agents and markets we have implemented, suggest that such issues should not arise in natural market descriptions.

7 Example Representations

In this section we represent a number of different auction markets within our framework. The representations do not describe the agents participating in an auction; the agents can be implemented in some arbitrary language, and their implementation is done outside our framework. Neither do the representations define the process by which auctions determine winners; the exact process used is chosen and executed by the participating agents. For instance, the winner-determination in a combinatorial auction can be performed using combinatorial optimization, and the agent acting as the auctioneer is responsible for running the appropriate combinatorial optimization algorithm. Rather, the representations define the rules of the auctions and capture the important properties of winner-determination (such as the fact that the highest bids win).

We use boldface to indicate the main language operators and underlining to indicate action names. We have also substituted certain parentheses with curly brackets to enhance readability. Other than these cosmetic enhancements the domains are presented below in the Prolog implementation of the scripting language of our framework. The full domain descriptions can be found online at `http://www.eecs.harvard.edu/~loizos/norms.html`. The object *clock* is an instance of the *event* class, and serves as a way to hold the time at which the current state of the world was instantiated. The various predicates used are provided by our framework and were already described in previous sections. The actions *sell(#good,#price,#receiver)* and *jail(#agent)* are imported from the appropriate libraries, with the latter retracting all the rights of an agent, when executed. The action *issue_p(right(#action,#condition),#agent)* is the built-in action of issuing possession of rights to agents. We assume, and do not explicitly represent below, the fact that agents have the right to open auctions on items they own. Also, unless otherwise stated, we assume that bidders have the right to query all the properties of an auction and all the properties of the items being auctioned. Such query rights are given to the bidders at the auction opening.

7.1 Open-Cry English Auction

In a typical open-cry English auction scenario an agent owning an item invokes the action of opening an auction. This establishes the auction parameters, through the *create_auction* action not shown here. The auctioneer also gives all bidders the right to place bids, conditioned on the new price being higher than the current price. Finally, the auctioneer commits to closing the auction and selling the item to the highest bidder soon after that.

```
action(Agent, open_auction(Auction, Item, OpeningPrice)) :- transaction([
create_auction(Auction, Item, OpeningPrice),
take_on(obligation( { value(Auction, status, closed) } , { value(clock, happened_at, Time),
value(Auction, last_bid_time, LastBidTime), atleast(Time, LastBidTime+100) }
, { jail(Agent) } )),
take_on(obligation( { value(Auction, [ (status, closed), (highest_bid, HighestBid),
(highest_bidder, HighestBidder) ]), object(Event), value(Event, [ (instance_of, event),
(description, invoked(Agent, sell(Item, HighestBid, HighestBidder), successfully)) ]) }
, { value(Auction, [ (status, closed), (highest_bidder, HighestBidder), (closing_time,
ClosingTime) ]), value(clock, happened_at, Time), atleast(Time, ClosingTime+100),
HighestBidder \= Agent } , { jail(Agent) } )),
issue_p(right( { place_bid(Auction, Bid) } , { value(Auction, [ (highest_bid, HighestBid),
(status, open) ]), atleast(Bid, HighestBid+1) } ), Bidder) ]).
```

Bidders proceed to place bids by raising the current highest bid. This grants the auctioneer the right to sell them the item at that price.

```
action(Agent, place_bid(Auction, Bid)) :- transaction([ raise_bid(Auction, Bid),
issue_p(right( { sell(Item, Bid, Agent) } , { value(Auction, [ (status, closed), (highest_bid,
Bid), (highest_bidder, Agent) ]), value(clock, happened_at, Time),
atleast(ClosingTime+100, Time) } ), Auctioneer) where value(Auction, [ (auctioneer,
Auctioneer), (item, Item) ]) ]).
```

action(Agent, raise_bid(Auction, Bid)) :- **preconditions**([object(Auction), value(Auction, highest_bid, CurrentBid), atleast(Bid, CurrentBid+1)]), **effects**([set(Auction, highest_bid, Bid), set(Auction, highest_bidder, Agent), set(Auction, last_bid_time, Time) **where** value(clock, happened_at, Time)]).

At the end, the auctioneer closes the auction and continues to invoke the *sell* action, as obligated by the rules of the auction.

7.2 Sealed-Bid Second-Price Auction

In a typical sealed-bid second-price auction scenario an agent opens an auction in the same manner as in the open-cry English auction. The main difference is that when the auctioneer grants to the bidders the right to query the properties of the auction, the granted right is conditional on the queried property not being the *set_of_bids* property, preserving in this way the secrecy of the collected bids.

action(Agent, open_auction(Auction, Item, OpeningPrice)) :- **transaction**([
create_auction(Auction, Item, OpeningPrice),
take_on(obligation(**{** value(Auction, [(status, closed), (set_of_bids, SetOfBids), (winner, HighestBidder), (payment, SecondHighestBid)]), get_second_price(SetOfBids, SecondHighestBid), get_first_bidder(SetOfBids, HighestBidder) **}** , **{** value(clock, happened_at, Time), value(Auction, last_bid_time, LastBidTime), atleast(Time, LastBidTime+100) **}** , **{** jail(Agent) **}**)),
take_on(obligation(**{** value(Auction, [(status, closed), (winner, HighestBidder), (payment, SecondHighestBid)]), object(Event), value(Event, [(instance_of, event), (description, invoked(Agent, sell(Item, SecondHighestBid, HighestBidder), successfully))]) **}** , **{** value(Auction, [(status, closed), (winner, HighestBidder),]), value(clock, happened_at, Time), atleast(Time, ClosingTime+100), HighestBidder \= Agent **}** , **{** jail(Agent) **}**)),
issue_p(right(**{** query(Auction, QueriedProperty) **}** , **{** value(Auction, status, open), QueriedProperty \= set_of_bids **}**), Bidder),
issue_p(right(**{** place_bid(Auction, Bid) **}** , **{** value(Auction, status, open) **}**), Bidder)]).

action(Agent, create_auction(Auction, Item, OpeningPrice)) :- **preconditions**([\+ object(Auction)]), **effects**([create(Auction, sealed_auction), set(Auction, owned_by, Agent), set(Auction, held_by, Agent), set(Auction, auctioneer, Agent), set(Auction, status, open), set(Auction, item, Item), set(Auction, set_of_bids, [(Agent,OpeningPrice)]), set(Auction, winner, undefined), set(Auction, payment, undefined), set(Auction, last_bid_time, Time) **where** value(clock, happened_at, Time), set(Auction, closing_time, undefined)]).

Bidders proceed to submit sealed bids, by updating the *set_of_bids* property, but without ever seeing its actual contents. Each bidder can only place one bid.

action(Agent, place_bid(Auction, Bid)) :- **transaction**([submit_bid(Auction, Bid),
issue_p(right(**{** sell(Item, Bid, Agent) **}** , **{** value(Auction, [(status, closed), (payment, Bid), (winner, Agent)]), value(clock, happened_at, Time), atleast(ClosingTime+100, Time) **}**), Auctioneer) **where** value(Auction, [(auctioneer, Auctioneer), (item, Item)])]).

> **action**(Agent, submit_bid(Auction, Bid)) :- **preconditions**([object(Auction), value(Auction, set_of_bids, SetOfBids), \+ member((Agent,AnyBid), SetOfBids)]), **effects**([set(Auction, set_of_bids, [(Agent,Bid)|SetOfBids]) **where** value(Auction, set_of_bids, SetOfBids), set(Auction, last_bid_time, Time) **where** value(clock, happened_at, Time)]).

Finally, the auctioneer closes the auction by declaring a winner and a payment and continues to invoke the appropriate *sell* action.

7.3 Combinatorial Auction

The case of a combinatorial auction resembles the sealed-bid auction, and thus we only briefly discuss the main points of difference. In a Vickrey-Clarke-Groves (VCG) auction (see Jackson [6]), as the one represented below, the auctioneer opens the auction for a set of items, invoking (amongst other things) the following action for committing to an efficient outcome and VCG payments:

> take_on(obligation(**{** value(Auction, [(status, closed), (set_of_bids, SetOfBids), (allocation, Allocation), (prices, Prices), (payments, Payments), (marginal_allocations, AllocationPerMarginalMarket), (marginal_prices, PricesPerMarginalMarket)]), AllAllocations = [Allocation|AllocationPerMarginalMarket], AllPrices = [Prices|PricesPerMarginalMarket], checkOutcomeEfficiency(SetOfBids, AllAllocations, AllPrices), checkVCGPayments(SetOfBids, AllAllocations, Payments) **}** , **{** value(clock, happened_at, Time), value(Auction, last_bid_time, LastBidTime), atleast(Time, LastBidTime+100) **}** , **{** jail(Agent) **}**)),

Bids are then placed, each specifying a bundle of items. Each bidder can submit multiple bids (issuing the corresponding rights). Given the semantics of an "additive-or" bidding language, any number of bids can then be accepted. On closing the auction, the auctioneer determines the revenue-maximizing allocation and the VCG payments. The auctioneer also provides the revenue-maximizing allocations in each marginal economy (with each bidder removed in turn), and *competitive equilibrium* (CE) prices in the main and marginal economies [2]. The CE prices allow the monitoring environment to verify that the allocations are optimal by checking best-response conditions for the seller and for each bidder. These checks are taken care of by a call to the *checkOutcomeEfficiency* predicate. Once the main and marginal allocations are checked, the VCG payments are checked with a call to the *checkVCGPayments* predicate. Both predicates are implemented in Prolog and are part of the actual domain description.

8 Conclusions

We argue that rights and obligations, important in human economies and often enforced through legal remedies, will be important in agent-mediated economies. We have defined a formal language that allows the specification of market mechanisms and a monitoring environment that allows for the automatic checking of rights and the enforcement of sanctions based on failed obligations. Simulations, frequently in the form of competitions such as the Trading Agent Competition [11,14], have often been used to explore market space and drive research into agent-based reasoning within electronic markets. We hope that the formal approach taken here, in which the semantics of markets are exposed to agents, will also prove useful in the development of principled methods in agent-based reasoning within electronic markets. We feel that the design principles implemented by our framework capture the main underlying assumptions of many virtual market designs and implementations (like eBay), and thus provide an infrastructure for the specifications of future virtual markets, and simulation platforms for testing agent designs.

Acknowledgments

Useful comments and suggestions were received during seminar presentations of earlier versions of this work in DEAS and AMEC VI. This work is supported in part by NSF grants IIS-0238147 and IIS-0091815.

References

1. A. Artikis, J. Pitt, and M. Sergot. Animated specifications of computational societies. In *Proc. of Conf. on AAMAS*, 2002.
2. S. Bikhchandani and J. Ostroy. The Package Assignment Model. *Journal of Economic Theory*, 107(2), December 2002.
3. Aspassia Daskalopulu and T. S. E. Maibaum. Towards electronic contract performance. In *DEXA Workshop*, 2001.
4. C. Dellarocas, M. Klein, and J. A. Rodriguez-Aguilar. An exception-handling architecture for open electronic marketplaces of contractnet software agents. In *Proc. ACM Conf. on Electronic Commerce (EC'00)*, 2000.
5. Oliver Hart. *Firms, Contracts, and Financial Structure*. Oxford Univ. Press, 1995.
6. Matthew O. Jackson. Mechanism theory. In *The Encyclopedia of Life Support Systems*. EOLSS Publishers, 2000.
7. S. L. Peyton Jones and J-M. Eber. How to write a financial contract. In J. Gibbons and O. de Moor, editors, *The Fun of Programming*. Palgrave Macmillan, 2003.
8. L. Thorne McCarty. Permissions and obligations. In *Proc. IJCAI*, 1983.
9. J. J. Meyer and R. J. Wieringa, editors. *Deontic Logic in Computer Science*. John Wiley & Sons, 1993.
10. Julian Padget and Russell Bradford. A π-calculus model of a Spanish fish market. In *Proc. of First Int. Workshop on Agent Mediated Electronic Trading*, May 1998.
11. N. M. Sadeh, R. Arunachalam, J. Eriksson, N. Finne, and S. Janson. TAC'03: A supply chain trading competition. *AI Magazine*, 24(1), Spring 2003.
12. Yao-Hua Tan and Walter Thoen. A logical model of directed obligations and permissions to support electronic contracting. *IJEC*, 3(2), 1999.
13. Jean Tirole. Incomplete contracts: Where do we stand? *Econometrica*, 67(4), 1999.
14. M. P. Wellman, P. R. Wurman, K. O'Malley, R. Bangera, S. Lin, D. Reeves, and W. E. Walsh. Designing the market game for a trading agent competition. *IEEE Internet Computing*, 2001.
15. M. Wooldridge, N. R. Jennings, and D. Kinny. The gaia methodology for agent-oriented analysis and design. *Journal of AAMAS*, 3(3), 2000.

iAuctionMaker: A Decision Support Tool for Mixed Bundling

Antonio Reyes-Moro[1] and Juan A. Rodríguez-Aguilar[2]

[1] iSOCO Lab. Intelligent Software Componets S.A,
c\ Alcalde Barnils 64/68 A 08190 Sant Cugat del Valles, Barcelona, Spain
antonio.reyes@isoco.es
[2] Institut d'Investigació en Intel.ligència Artificial, IIIA,
Consejo Superior de Investigaciones Científicas, CSIC,
Campus de la Universitat Autònoma de Barcelona,
08193 Bellaterra, Barcelona, Spain
jar@iiia.csic.es

Abstract. This paper presents iAuctionMaker as a novel tool that serves as a decision support for e-sourcing professionals on their pursuing of auction optimisation. Given a set of items to auction, iAuctionMaker helps an auctioneer determine how to separate items into promising bundles that are likely to produce better outcomes than the bundle of items as a whole. Promising bundles are those that satisfy certain properties believed to be present in competitive sourcing scenarios. These properties are defined by e-sourcing professionals and capture their experience and knowledge in the domain. iAuctionMaker models this knowledge as constraints to be satisfied by any bundle, and implements an optimisation algorithm to find the bundles that maximize satisfaction. Experimental results are shown to demonstrate the applicability of the approach. Case studies are presented to demonstrate that iAuctionMaker improves current e-sourcing practices and provides an alternative to combinatorial scenarios whose complexity hinders their application in actual-world sourcing scenarios.

1 Introduction

The negotiation scenario considered in this paper starts out with a *buyer* requiring to acquiring a set of items (be them either products or services). The buyer will negotiate the price and conditions[1] of each item by means of one or more on-line reverse auctions [16]. A set of providers will be invited to bid under certain auction rules that include bidding and winning rules. The auction is expected to allocate to some (all) of these providers some (all) of the items at auction.

A common industrial scenario involves multiple goods or services to be purchased as a whole with the intention of benefiting from volume-based discounts. One may think, for example, of demand aggregation from different companies. The winner gets huge business and the losers get nothing, a simple and well-known strategy for lowering the price.

[1] Notice that we consider multi-unit, multi-attribute items.

P. Faratin and J.A. Rodríguez-Aguilar (Eds.): AMEC 2004, LNAI 3435, pp. 202 – 214, 2005.

Unfortunately, things are not so simple. Maybe there is just one single provider that can provide everything so, if auctioning a single bundle, he will not face difficulties in getting the business at the price he quotes. Maybe it is not acceptable that (as providers bid for the whole thing), they lower the price for company-A product at the expense of increasing the price for company-B product (and consequently deal with B complaining about why are they buying more expensive than last year).

Therefore, the question is: should a seller who wants to maximise his revenue conduct separate auctions, one for each of several objects, or should he conduct a single auction for the entire bundle, or should he group items into bundles and conduct several auctions?

Efforts and tools have been developed to answer this problem. The general procedure is to allow flexibility in bidding by allowing providers to bid over combinations of items according to their preferences. I.e., give providers a way to state that they will offer a better price if there is a guarantee that they will get a certain amount of the business. This mechanism is known as combinatorial bidding [10] and has been widely studied in literature as an optimum artifact to maximize results.

To achieve coherent and practical results from a reverse combinatorial auction it is a must to introduce constraints that sacrifices mathematical optimality of the winning set in favour of obtaining realizable and practical outcomes [8][10][17]; (it is unnatural to have 40 different winners, for example, so it will be convenient to limit the amount of winners and state a lower bound of the amount of business they can get).

Unfortunately, combinatorial bidding capabilities are rarely found on commercial systems[2] [4] [15], and yet there is a major problem that prevents the practical application of combinatorial auctions: complexity. Bidding in a combinatorial auction requires accurate knowledge and understanding of the auction's dynamics in order to decide what is the next bid to place [14], [18]. Moreover, the constraints imposed to determine winners make winning rules complex to follow.

The conclusion is that the practical application of the above methods is usually constrained to very controlled and specific environments (e.g. [5]).

To overcome this situation, e-sourcing professionals usually follow an alternate approach: based on market real data and knowledge, the whole bundle is divided into separate auctions where the appropriate providers are invited and where certain properties are satisfied. These properties model the expertise of e-sourcing specialists in the form of *rules of thumb,* and their applicability they believe can turn into interesting benefits [6].

Each auction is a simple reverse auction supported by the majority of existing commercial on-line auction platforms, and their execution present none of the complexities previously discussed. And the outcome of each auction is highly promising, as they have been designed to verify some criteria known to maximise savings.

It is interesting to state that theoretically, this methodology is expected to produce not as good results as a full combinatorial auction as it is likely that hidden synergies or interesting market situations are left out and unexploited. In spite of that, we believe that the former assert is true if and only we assume the ideal situation where everybody knows and controls the combinatorial auction mechanisms. Such a supposition is, for

[2] To develop a commercial system that allows fully flexible wining rule configuration to cope with real situations is costly, despite interesting results are being obtained.

the best of our experience, never satisfied in practice and, consequently, not only it cannot lead to the desired result, but even produce catastrophic outcomes.

Nevertheless, in this methodology, the core process that is responsible for the success of the e-sourcing event is obviously the process of determining the grouping of items into bundles to be auctioned. Expertise and market knowledge are key factors, but in many situations the number of lines (100s or more) and the number of providers (20s or more) makes the problem intractable. These difficulties make desirable to count on a tool to aid e-sourcing professionals at this stage. We have called this tool iAuctionMaker.

iAuctionMaker is a decision support tool that assists an auctioneer in defining the *ideal* bundles by declaring a list of pre-existing constraints that can be tuned and prioritised according to his preferences. iAuctionMaker solves the problem of finding the bundles that maximizes the satisfaction of these constraints.

Although the bundling problem has been previously addressed in the literature we observed that it has mostly focused on the issue of whether a seller ought to sell items separately or as a bundle and to determine the price of the bundle(s) to be sold. In general, the bundling literature has evolved from early works considering a single seller bundling two goods [1] to works considering the analysis of a monopolist bundling multiple goods [3], to more recent works spurred by the advent of the Internet that contemplate competition of multiple sellers [2] [12]. As to multi-seller markets, in [12] the authors propose and analyse a bundling model to set both price and bundle composition in which a seller is not considered in isolation but in a market scenario wherein additional sellers compete to offer their bundles.

iAuctionMaker takes a different stance. We depart from a market scenario in which a single buyer aims at acquiring a bundle of multi-attribute goods. Unlike traditional approaches, it is not our aim to decide whether the buyer ought to purchase the goods separately or as a bundle, along with an appropriate pricing strategy. Our goal is to produce a bundle composition for a buyer that leads to clusters of providers (bidders) exhibiting high degrees of competitiveness, while at the same time satisfying the buyer's preferences (modelled as a collection of constraints). Furthermore, we expect that the partitioning of the whole bundle of items also benefits bidders since they are expected to address the bid construction problem [14] [18] for smaller bundles (less goods and competitors).

2 Problem Definition

The formal formulation of the problem is the following:

- $I = \{I_1, I_2, \dots I_n\}$ is a finite set of n items representing the goods or services to be purchased.
- $P = \{P_1, P_2, \dots P_m\}$ is a finite set of m providers.
- $A = \{A_1, A_2, \dots A_o\}$ where each element of A is a function $A: 2^I \rightarrow R$ that models a property or observation of a subset of I representing a bundle. These properties might be various (number of providers, number of lines, bundle volume, etc.) and

come from different sources (previous provider behaviour, a preliminary RFQ[3], provider and item characterization, etc, etc.)

- $C = \{c_1, c_2, \ldots, c_r\}$. Each $c_i \in C$ is a *soft*-constraint defined as a tuple $<A_i, S_i, w_i>$ with the following meaning: A_i is the bundle property to be evaluated, $S_c: R \rightarrow [0\ldots1]$ is a scoring function (formally defined in section 3) that expresses the satisfaction degree for A_i (0 indicates no satisfaction at all; and 1 indicates maximum satisfaction); and $w \in R^+$ expresses the relative weight of the constraint.

The objective is to find $L=\{L_1, L_2, \ldots L_q\}$ a set partition of I that maximises the following expression:

$$S(L) = \sum_{L_k \in L} \frac{\dfrac{\sum_{c \in C} S_c(A_c(L_k)) \cdot w_c}{\sum_{c \in C} w_c}}{|L|} \qquad (1)$$

subject to: (1) $L_i \cap L_j = \emptyset \quad \forall L_i, L_j \in L; L_i \neq L_j$; and (2) $L_1 \cup L_2 \cup \ldots L_l = L$.

This problem is a particular instance of the set partition problem [13], which is known to be NP complete [7].

3 Solution

To solve the problem formulated in the previous section, we first have to define a bundle's utility theory (i.e, define S_c and design an optimisation algorithm.

3.1 MAUT-Based Bundle Evaluation

The method used in *iAuctionMaker* for scoring a bundle is based on Multi attribute utility theory [11], since bundle's utility or *goodness* can be evaluated by the degree of satisfaction of a list of attributes for a given user's preference and importance.

To model preferences and importance we have previously defined a set of constraints. Each constraint $c_i = <A_i, S_i, w_i>$ evaluates some property A_i of a bundle by means of a scoring function S_i. To define S_i, we have followed the guidelines proposed in [15] where membership functions are studied to *intuitively* model human preferences. With these considerations in mind, $Sc: R \rightarrow [0\ldots1]$ is defined as follows:

$$\begin{array}{ll} S_c(p) = \alpha+(b-p)\cdot\beta & p \in [a, b] \\ S_c(p) = \max\{S_c(a)-(a-p)\cdot\delta, 0\} & p<a \\ S_c(p) = \max\{S_c(b)-(p-b)\cdot\phi, 0\} & p>b \end{array} \qquad (2)$$

where: (i) a,b ($a<b$) define the preferred range $[a\ldots b]$ of values; (ii) $sl \in \{ANY, LIB, MIB, LIBC, MIBC\}$ defines the preference slope of S_c; (iii) $mh \in \{YES, NO\}$. if $mh = YES$ then values are not accepted out of the preference interval (they will score 0); and (iv) α, β, δ, ϕ, depend on the value of sl and mh and are calculated as shown in table 1.

[3] Request For Quotation.

Table 1. Values of α, β, δ, ϕ, as a function of *mh* and *sl*

mh \ sl	ANY	LIB	MIB	LIBC	MIBC
No	$\alpha = 1$ $\beta = 0$ $\delta = \gamma\cdot\frac{(1-\eta)}{(b-a)}$ $\phi = \delta$	$\alpha = \eta$ $\beta = \frac{(1-\eta)}{(b-a)}$ $\delta = 0$ $\phi = \gamma\cdot\beta$	$\alpha = 1$ $\beta = -\frac{(1-\eta)}{(b-a)}$ $\delta = \gamma\cdot\beta$ $\phi = 0$	$\alpha = \eta$ $\beta = \frac{(1-\eta)}{(b-a)}$ $\delta = -\gamma\cdot\beta$ $\phi = -\gamma\cdot\beta$	$\alpha = 1$ $\beta = -\frac{(1-\eta)}{(b-a)}$ $\delta = \gamma\cdot\beta$ $\phi = \gamma\cdot\beta$
Yes	$\alpha = 1$ $\beta = 0$ $\delta = +\propto$ $\phi = +\propto$	$\alpha = \eta$ $\beta = \frac{(1-\eta)}{(b-a)}$ $\delta = 0$ $\phi = +\propto$	$\alpha = 1$ $\beta = -\frac{(1-\eta)}{(b-a)}$ $\delta = +\propto$ $\phi = 0$	$\alpha = \eta$ $\beta = \frac{(1-\eta)}{(b-a)}$ $\delta = +\propto$ $\phi = +\propto$	$\alpha = 1$ $\beta = -\frac{(1-\eta)}{(b-a)}$ $\delta = +\propto$ $\phi = +\propto$

Intuitively, when a value falls within the preference limits, is given a value that is at least η, which models the limit of satisfaction. Depending on the slope *sl* the scoring progress from η to *1* as we move through *[a...b]*. When a value falls outside the preference limits, it receives a score that will be progress from η or *1* to *0* depending on how 'close enough' is the value to the preferred side. By 'close enough' we consider values within a neighbourhood of the interval (computed as a percentage $1/\gamma$ of the interval length). Figure 1 shows the scoring for $a=10$, $b= 20$, $mh=NO$, $\eta = 0.3$, $\gamma= 2$.

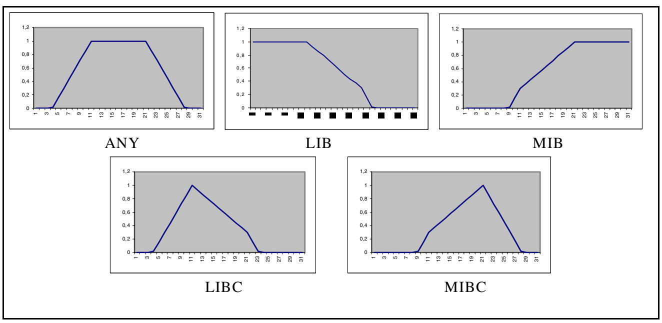


Fig. 1. Scoring functions

- ANY is used when we just want that a property *p* of the bundle to fall between [a…b].
- LIB states that we will consider the constraint satisfied if $p \leq b$, and that we will consider full satisfaction when $p \leq a$.
- MIB states that we will consider the constraint satisfied if $p \geq a$, and that we will consider full satisfaction when $p \geq b$.
- LIBC means that a constraint is satisfied if $p \in [a...b]$; full satisfaction will be considered when $p=a$; and minimum satisfaction when $p=b$.

So far we have explained how to obtain the degree of constraint satisfaction for a bundle by means of a scoring function. For a set of constraints, scoring functions are then weighted to obtain the overall bundle scoring as shown:

$$\frac{\sum_{c \in C} S_c(A_c(L_k)) \cdot w_c}{\sum_{c \in C} w_c} \tag{3}$$

By setting each w_c accordingly, the user is allowed to state satisfaction preference among constraints.

3.2 Optimisation Algorithm

The optimisation algorithm implemented for *iAuctionMaker* falls in the category of random probabilistic methods. Well-known exponents of these are genetics algorithms and neighbourhood search [9]. The main idea within these methods is to start by a number of initial solutions and implement a try and test procedure to discover better solutions. The try and test procedure, or search, is basically an iterative random change of these solutions until the best exponent converge to a local optima or a maximum number of try-and-test cycles are done. The search is directed by favouring changes that follow certain heuristics.

We have devised our own probabilistic search procedure which is mainly a neighbourhood search directed by heuristics. Whether it is a new procedure, extension, optimisation, or rediscovering of existing search algorithms is, at the current stage of the work, of no importance to us and is beyond the scope and interest of this paper. Literature is full of claimed-as-new algorithms that are actually rediscovers of existing algorithms previously applied to different problem domains.

The reason why we have applied such an algorithm is because random search methods are usually fast, perform relatively well and are easy to implement. Also, random search methods are usually independent from the objective function, and their performance do not heavily rely on exploiting problem characteristics and lower bounds identification. This is of special interest to us, as we expect the number and type of the constraints to be highly determined by the final user and his application domain (e.g. food, transportation, indirect materials, etc.). Even if we are able to model the problem as an integer program, the introduction of new constraints will force us to study the feasibility of the current model and to change it accordingly. Employing a branch-and-bound procedure would require a considerable amount of expert knowledge and effort to tune the heuristics function each time new constraints are changed or refined, in order to maintain algorithm performance.

The main criticism to approximate random search is sub-optimality. In our case, we believe this is not critical as the solution is just an intermediate phase in our process that will terminate with the execution of an auction, the real outcome of which is unpredictable. The algorithm can be outlined as follows:

```
L = ∅
for each Ii ∈ I
    create Li = {Ii}
solution = copy_of(L)
while (convergence is not reached)
  randomly pick Li ∈ L; with probability
  inversely proportional to S({Lj})
  randomly pick Lj ∈ L ∪ {∅} ; Lj ≠ Li with uniform
  probability
  randomly pick Ik ∈ Li with probability
  inversely proportional to S({Li}) - S({Li - Ik})
  Li = Li - Ik
  Lj = Lj ∪ Ik
  if Li = ∅ then L = L - Li
  if Lj ∉ L then L = L ∪ Lj
  if S(solution)<S(L) then
        solution = copy_of(L)
return solution
```

The previous procedure is explained as follows: An initial solution is built by considering each item to be auctioned in isolation. Then we enter an iterative phase where we randomly select a bundle. The bundle is selected implementing a roulette wheel [9] where the chances of each bundle are inversely proportional to its constraints satisfaction value (i.e. *bad* bundles will be selected more often, in an attempt to transform them into good ones) From this bundle, we select the item that is most probably causing the low satisfaction value of the bundle. To identify *bad* items we calculate the difference between the constraint satisfaction degree of the bundle with and without the item. Once an item has been selected, we remove it from the bundle, and next we randomly select a bundle (or a new bundle) where to add the item. The heuristic beneath this procedure is to penalise bad combinations of items. The random nature of the neighbourhood search prevents excessive falling into local optima.

The basic procedure has been enhanced with typical techniques to allow the algorithm to converge faster:

- Implement a backtracking procedure that allow to backtrack to a previous state if the current solution differs more than certain percentage from the best solution known so far.
- Avoid useless operations (i.e., changes that are known not to lead to better solutions).
- Implement an alternate *change* operator that joins two bundles. A probabilistic factor is used to randomly determine which operator to apply at each step of the algorithm.

3.3 Implementation

The core of iAuctionMaker is implemented in *java* and verifies the XML and J2EE standards, which simplify the implementation of multiple user interfaces (web-based, for example) as well as its integration with existing applications. The object model of iAuctionMaker works with *interface* declarations of *constraints* and *properties* objects. This allows the easy extension of the system by *implementations* of client-custom constraints. As mentioned in the previous section, the random search procedure implemented will not need of reformulation if new constraints are added. This will optimise product customisation, without incurring in costly AI expertise and algorithm-refactoring.

4 Results

This section will present two experimental outcomes of *iAuctionMaker.* The first results aims to demonstrate that the search procedure developed for the tool performs satisfactorily. The second results present the commercial application of *iAuctionMaker* to two real sourcing scenarios.

In order to validate the optimisation algorithm proposed we performed two experimental tests. The first one aims at measuring the optimality of the algorithm when compared to a complete search procedure. The second experiment aims at measuring the correctness and applicability of the algorithm for large instances of the problem.

For the first experiment we generated 1000 random instances of a problem consisting of 11 lines and 11 providers[4]. The problem instance considers that all providers are capable of providing all lines. The price provider *j* offers for line *j* is randomly determined with uniform probability between [0..10]. We considered four constraints: *volume aggregation, best bidder presence, bids' variability* and *number of providers.*

Each problem is solved optimally by a brute force search procedure[5], i.e., all possible solutions are generated. Then, each problem is solved with our algorithm and results are compared. 7 rounds are executed, each varying the algorithm termination condition, which controls the number of iterations (different solutions) explored by our algorithm, for each round. Table 2 shows the results obtained.

Results suggest that *iAuctionMaker* seems to perform accordingly with the objectives, i.e., optimal solutions are found with considerable less search effort and, maybe more important, the relative difference between the optimal and the sub-optimal solution found is more than acceptable.

Table 2. Results obtained for experiment 1

% of search effort	% of problems solved optimally	% variation between optimal and best solution found
0.2	23	4.11
1.05	56	2.87
1.93	69	2.3
8.36	85	1.63
16.22	89	1.4
77	93	1.53
150	96	1.36

The second experiment consisted in solving problems for which we know the best solution in advance. Problems are generated by randomly separating items into 2, 4, 5, 10, or 20 bundles. Each bundle is only provided by a certain group of providers. Two constraints were considered, *number of providers,* and *bundle volume.* Hence, the best possible solution is to find such bundles. The problem size was fixed to 100 lines and 20 providers, which is a problem size larger than the ones to be solved in our case studies. Table 3 shows the optimality evolution as we increase the search effort.

Results suggest that the performance indicators observed for experiment 1 can also be obtained for large instances of problems and at affordable cost.

[4] An affordable size to be solved by a brute force search algorithm.

[5] The total number of different solutions for this problem size is 678570 (see Bell Numbers for further information).

Table 3. Results obtained for experiment 2

% of problems solved optimally	Average mean solution time (seconds)
17%	0.16
54%	0.49
70%	0.9
87%	3.7
89%	6.8
98%	26.8
99%	65.9

In conclusion, these experiments suggest that we can be fairy confident on the *goodness* of *iAuctionMaker* random optimisation procedure to be applied in real scenarios.

4.1 Case Study: Electricity Purchase

The first scenario studied the initials offers received from 5 South-Europe electricity companies to power a total of 20 manufacturing facilities in Spain that belong to the same company. The plants are all of similar power consumption and are geographically distributed across the country. For each location, bidders decide whether to bid or not. The bid presented stands for the average price of the Kilowatt, according to last year consumption records. The scenario is therefore translated into a problem consisting of 20 lines and 5 providers. Table 4 presents the initial offer given by each provider for each location[6].

Table 4. Electricity market data

Locations/Providers	**P1**	**P2**	**P3**	**P4**	**P5**
BAD	5,545	5,836	5,415	5,493	5,329
BEZ	5,313	5,528	5,384	5,269	5,028
CAR	5,599	5,896	5,339	5,604	5,311
COD	5,495	5,91	5,247	5,489	5,195
COR	4,417	4,831	4,484	4,444	
DULC	5,883	6,296	5,761	5,978	
ESP	5,496	5,881	5,431	5,483	5,361
GEN	5,129	5,402	4,886	5,211	4,903
GRA	5,317	5,739	5,24	5,264	5,13
GUA	5,366	5,119		5,347	5,141
PER	5,219	5,583	5,112	5,265	5,151
PLANT	5,494	5,988	5,606	5,381	5,261
RAI	5,724	6,353	5,727	5,743	5,561
RIB	5,795	6,021	5,575	6,033	5,423
RON	5,803	6,204	5,774	5,869	5,498
SES	5,31	5,831	5,422	5,289	5,134
SEV	5,182		5,083	5,238	5,101
VOÑÑ	5,345	5,745	5,452	5,212	5,146
ZAM	5,312	5,634	5,067	5,439	5,093
FRI	5,258	5,428	5,005	5,209	5,035

6 Bidders are first invited to an RFQ where to place their first offer. (They do not know yet that an auction may later take place).

The company's sourcing professionals know that in order to achieve savings, it will be of interest to group facilities rather than auction each facility in isolation. However, some of the bidders are new, small companies (the Spanish electricity market was liberalized short ago) which are geographically specialized and are likely to bid aggressively for facilities in their area, whereas unable to compete for others.

To model this knowledge into *iAuctionMaker* three constraints were given:

1. The bigger the bundle (in price terms[7]), the better.
2. The best possible offer for the whole bundle[8] by a single provider must be at most 1% worse than the offer obtained by selecting the best offer per location.
3. Ideally there should be 3 providers whose offers for the whole bundle differ less than 3%.

Constraint 1 tried to make bigger bundles, where there exists place for competitiveness (constraint 3). To prevent missing very competitive offers for certain locations, constraint 2 is given.

Table 5. Modelling constraints

Constraint	A (Observable variable)	S_c[9]	w_c
c1	Total bundle price, calculated as the sum of the mean offer for each line.	a = 20 b = 100	1
c2	Number of providers that are %1 from the optimal.	A = 1 b = 1	1
C3	Number of providers capable of offer the whole bundle.	a = 2 b = 3	1

With these constraints, *iAuctionMaker* finds the 3-bundle distribution shown in *Figure 2*[10].

All three bundles are quite similar and satisfy the desired properties:

- There are 3 providers close enough to compete.
- The amount of business is interesting.

In case of no auction activity, the risk is to purchase %0.47 more expensive than the current situation.

4.2 Case Study: Transportation Purchase

The second scenario studied the initials offers received from 16 transportation companies to deliver a company range of products to 81 destinations across Europe. A first round of RFQs where conducted to obtain initial price-matrix (destination by kg).

---

7 All prices are in EURO.

8 We *induce* the value of the offer for the bundling as the addition of the known offers for the individual locations.

9 The rest of parameters are set as follows sl=MIB, mh=NO, $\eta = 0.3$, $\gamma = 2$. Refer to table 1.

10 This solution depicted is just one among others that scored identically. The company purchasing department evaluated them all and the final auction configuration (not shown here) was selected considering geographical distribution.

Based on historical data, the price matrix was reduced to a single column representing the total cost to each destination for each bidder.

The application of *iAuctionMaker* to this scenario produced an interested outcome: it was not possible to find any *promising* bundle to undergo an auction.

To obtain an explanation for this, *iAuctionMaker* was given the following two constraints:

- The bigger the bundle, the better.
- The best bidder for the bundle must be also the best bidder for each single destination in the bundle.

In other words, with these two constrains *iAuctionMaker* was configure to identify the current winning set of providers and group destinations accordingly.

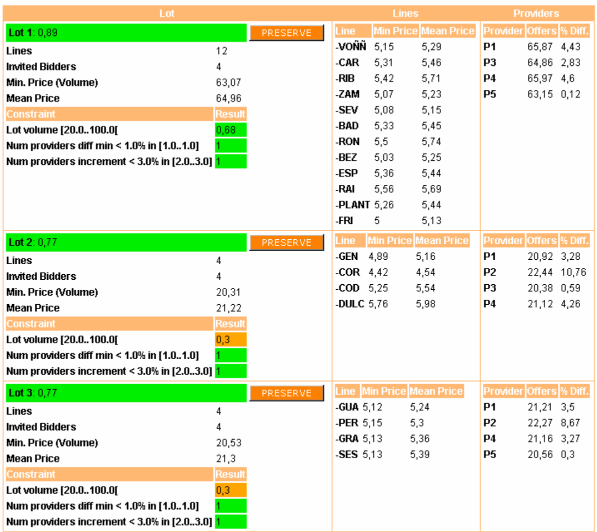


Fig. 2. iAuctionMaker results for the electricity problem

As seen in figure3, the solution obtained has 3 bundles, each corresponding to 3 winners: *P9, P13 and P4.* Notice the difference in price between the winner and the immediate competitor (a minimum of 43% for Bundle 3). This explains why there is no room for an auction. Obviously, the bundles obtained correspond to a particular geographical distribution for which each winner is clearly specialised (*Bundle 1* only contains locations in *Italy*, for example).

After obtaining these results, the company purchasing department verified that there was no mistake in the offers received and the negotiation ended after a second round of offers.

Conclusively, *iAuctionMaker* proved to be useful in assisting the user to identify scenarios where the application of an on-line auction will not produce clear benefits.

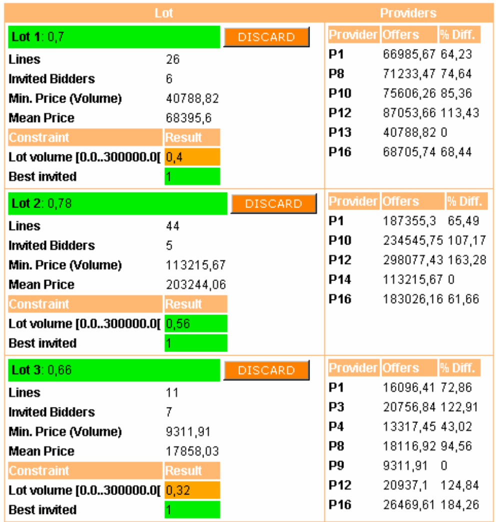


Fig. 3. iAuctionMaker results for the transportation problem

5 Conclusions

This paper has presented *iAuctionMaker as* a novel decision support tool for e-sourcing professionals. The motivation was to improve current e-sourcing procedures and provide an alternative to combinatorial or constraint bidding whose complexity prevents their application in real sourcing scenarios. The methods and algorithms developed were highly directed by software industry needs of efficiency and easy of extension and customisation. Experimental results showed promising results, which were later verified by successful application to real industry problems. Customers highly evaluated the tool and were satisfied with the results obtained.

Future work basically lays in the application of the tool to more real sourcing scenarios from various industries. This will provide us with useful feedback from e-sourcing professionals as well as to test new constraints obtained from their domains. Our goal is to provide an extensive library of *rules of thumb* that contains the expert knowledge of sourcing professionals.

References

[1] Adams, W. and Yellen, J. (1976). Commodity bundling and the burden of monopoly. Quartely Journal of Economics, 90:475-498, 1976.

[2] Bakos, Y. and Brynjolfsson. (1999). Bundling and competition on the internet: Aggregation strategies for information goods. Working paper series, MIT Sloan School of Management.

[3] Bakos, Y. and Brynjolfsson, E. (1998). Bundling information goods: Pricing, profits and efficiency. In D. Hurley, B. Kahin, and H. Varian, editors, The Economics of Digital Information Goods. MIT Press, Cambridge, Massachussetts.

[4] Combinenet. http://www.combinenet.com, 2004.

[5] Elmaghraby, W. and Keskinocak, P. (2002). Technology for Transportation Bidding at the Home Depot. To appear in C. Billington, T. Harrison, H. Lee, J. Neale, editors, The Practice of Supply Chain Management., Kluwer.

[6] Eppen, G. D., Hanson, W. A. (1991). Bundling- New Products, New Markets, Low Risk. Sloan Management Review 32(4): pp. 7-14.

[7] Garey, M. R. and Johnson, D. S. (1979). *Computers and Intractability: A Guideto the Theory of NP-Completeness*. W. H. Freeman and Company, New York.

[8] Giovannucci, A. Rodríguez, J.A. Reyes, A. Noria, F.X. and Cerquides, J. (2004). *Towards automated procurement via agent-aware negotiation support*. In Proceedings of the Third International Joint Conference on Autonomous Agents and Multi-agent Systems (to appear), New York City, July 19-24, 2004.

[9] GoldBerg, D.E. *Genetics Algorithms in Search, Optimisation & Machine Learning*. (1989) Addison-Wesley.

[10] Kalagnanam, J., Parkes D. (2003) Auctions, Bidding and Exchange Design. Chapter 10, in Supply Chain Analysis in the eBusiness Era, Edited by David Simchi-Levi, S. David Wu and Z. Max Shen,. (forthcoming)

[11] Keeny, R. L. and H. Raiffa. (1993). *Decision Making with Multiple Objectives: Preferences and Value Tradeoffs*. Cambridge, UK: Cambridge University Press.

[12] Kephart, J. O. and Fay, S. A. (2000). Competitive Bundling of Categorized Information Goods. EC'00, October 17-20, Minneapolis, Minnesota.

[13] Kreher, D.L. Stinson, D. R. (1998) *Combinatorial Algorithms: Generation, Enumeration, and Search*, CRC Press.

[14] Parkes, D. (2000) Optimal Auction Design for Agents with Hard Valuation Problems, In Agent Mediated Electronic Commerce II: Towards Next-Generation Agent-Based Electronic Commerce Systems (LNAI 1788), Moukas et al. (eds), Springer-Verlag.

[15] Reyes, A. Rodríguez J.A., López M., Cerquides J. and Gutierrez. D. (2003) *Embedding Decision Support in e-Sourcing Tools: Quotes, A Case Study.* Group Decision and Negotiation Journal, Vol. 12, pp 347-355.

[16] Sandholm, T., Suri, S., Gilpin, A., and Levine, D. (2002). Winner Determination in Combinatorial Auction Generalizations. In Proceedings of the First International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS), Bologna, Italy, July, pp. 69-76. .

[17] Sandholm, T. and Suri, S. (2001). Side Constraints and Non-Price Attributes in Markets. International Joint Conference on Artificial Intelligence (IJCAI), In Proceedings of the Workshop on Distributed Constraint Reasoning, Seattle, WA, August 4th.

[18] Song, J. and Regan, A. (2004). Approximation algorithms for the Bid Construction Problem in Combinatorial Auctions for the Procurement of Freight Transportation Contracts. In Proceedings of the 2004 TRB meeting.

Author Index