Vorlesung Optimierung II
SS 2000

Inhaltsverzeichnis

1 Kombinatorische Optimierung **1**

1.1 Begriffe aus der Graphentheorie 1
1.2 Das Problem des kürzesten Wegs 6
1.3 Minimal spannende Bäume 12
1.4 Maximale Flüsse und der Algorithmus von Ford–Fulkerson 17
1.5 Kostenminimale Flüsse 22
1.6 Branch-and-Bound-Verfahren 27
1.7 Dynamische Programmierung 32

2 Spieltheorie **34**

2.1 Allgemeines und Historie 34
2.2 Grundlegende Begriffe 34
2.3 Gemischte Strategien 36
2.4 Gleichgewichtspunkte 37
2.5 Zwei-Personen-Nullsummenspiele 39
2.6 Matrixspiele 40
2.7 Existenz von Gleichgewichtspunkten 44
2.8 Endliche Spiele 45
2.9 Bimatrixspiele 48
2.10 Spiele über dem Einheitsquadrat 50
2.11 Das Oligopolspiel 54

Literatur

[1] Bauer: Wahrscheinlichkeitstheorie, Walter de Gruyter & Co, Berlin 1978.

[2] Bollobas, B.: Graph Theory, Springer, New York 1979.

[3] Burger: Einführung in die Theorie der Spiele, Walter de Gruyter & Co, Berlin 1966.

[4] Burkhard, R.E.: Methoden der ganzzahligen Optimierung, Springer, Wien-New York 1972.

[5] Collatz, L., Wetterling, W.: Optimierungsaufgaben, Springer, Berlin-New York 1971.

[6] Dantzig, G.B.: Lineare Programmierung und Erweiterungen, Springer, Berlin-New York 1966.

[7] Franklin, J.N.: Methods of mathematical economics : linear and nonlinear programming; fixed point theorems, Springer, New York, 1980.

[8] Fudenberg, Tirole: Game Theory, MIT Press, Camebridge 1991.

[9] Gaede, K.W., Heinhold,J.: Grundzüge des Operations Research, Carl-Hanser-Verlag, München-Wien 1976.

[10] Gass, S.I.: Linear Programming, Methods and Applications, Mac Graw-Hill, New York 1958.

[11] Grötschel, M., Lovasz, L., Schrijver, A.: Geometric algorithms and Combinatorial Optimization, Springer, Berlin-New York 1988.

[12] Hadley, G.: Linear Programming, Addison Wesley Publ. Comp., London 1972.

[13] Holler, Illing: Einführung in die Spieltheorie, Springer, Berlin 1990.

[14] Hu, T.Chiang: Ganzzahlige Programmierung und Netzwerkflüsse, Oldenburg, München 1972.

[15] Kall, P.: Mathematische Methoden des Operations Research, Teubner, Stuttgart 1976.

[16] Marti, J.: Konvexe Analysis, Birkhäuser, Basel 1977.

[17] Myerson: Game Theory, Harvard University Press, Camebridge 1991.

[18] v. Neumann, Morgenstern: Theory of Games and Economic Behavior, Princton Univ.Press, Princton 1944; deutsche Übersetzung: Spieltheorie und wirtschaftliches Verhalten, Physica, Würzburg 1961.

[19] Neumann, K.: Operations-Research-Verfahren I-III, Carl-Hanser-Verlag, München-Wien 1975.

[20] Nozicka, F.: Geometrie konvexer Mengen und konvexe Analysis, Akadem. Verl., Berlin 1988.

[21] Owen, G.: Spieltheorie, Springer, Berlin-New York 1971.

[22] Papadimitriou, C.H., Steiglitz, K.: Combinatorial Optimization, Prentice-Hall Inc., New Jerssey 1982.

[23] Rauhut, B., Schmitz, N., Zachow, E.-W.: Spieltheorie, Teubner, Stuttgart 1979.

[24] Roberts, A.W., Verberg, D.E.: Convex Functions, Academic Press, New York 1973.

[25] Rockafellar, R.: Convex Analysis, Princeton University Press, Princeton 1972.

[26] Schrijver, A.: Theory of Linear and Integer Programming, Wiley, Chichester 1986.

[27] Schwödiauer: Spieltheorie, in: Tomas Gal (Hrsg.), Grundlagen des Operations Research 3, Springer, Heidelberg 1987.

[28] Stoer, Witzgall: Convexity and Optimization in finite dimensions, Springer 1970.

[29] Suchowitzki, S.I., Andejewa, L.I.: Lineare und konvexe Optimierung, Oldenburg, München 1969.

[30] Szép, J., Foró, F.: Einführung in die Spieltheorie, Harri Deutsch, Thun, Frankfurt, 1983.

[31] Turau, V.: Algorithmische Graphentheorie, Addison-Wesley, 1996.

[32] van Tiel, J.: Convex Analysis, Wiley, Chichester 1984.

[33] Valentine, F.A.: Konvexe Mengen, Bibliograph. Inst., Mannheim 1968.

[34] Vogel, W.: Lineares Optimieren, Akad. Verlagsges., Leipzig 1967.

[35] Vorob'ev, N.N.: Game Theory, Springer, New York 1977.

[36] Walter: Analysis II, Springer, Berlin 1990.

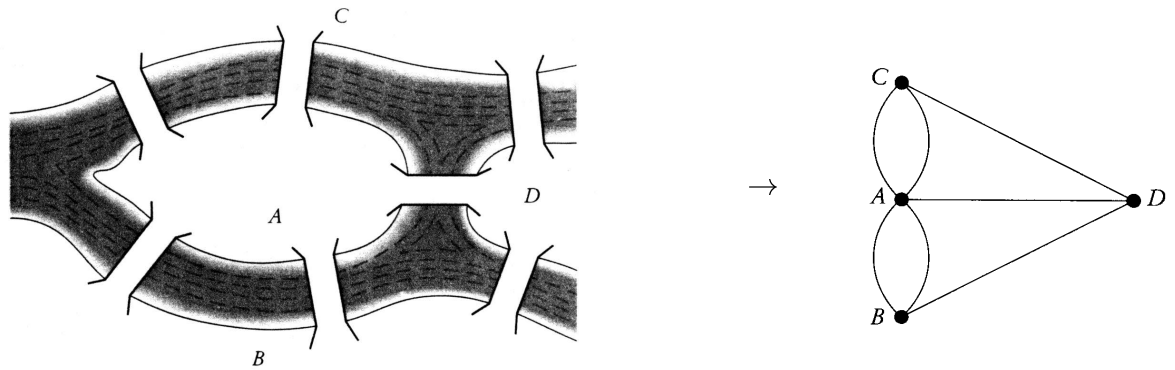
[37] Worobjow: Entwicklung der Spieltheorie, VEB Deutscher Verlag der Wissenschaften, Berlin 1975.

[38] Yang: Computing Equilibria and Fixed Points, Kluwer Academic Pub., 1999.

1 Kombinatorische Optimierung

1.1 Begriffe aus der Graphentheorie

Zur Beschreibung vieler Optimierungsprobleme eignet sich besonders die Sprache der Graphentheorie. Das erste graphentheoretisch beschriebene Problem stammt von Euler (Königsberger Brückenproblem, 1736): Es sollte ein Rundweg durch Königsberg gefunden werden, der jede Brücke über den Pregel genau einmal überquert. Euler reduzierte die Inseln und die durch den Fluß getrennten Festlandteile auf Punkte und stellte die Brücken durch Verbindungskurven dar.

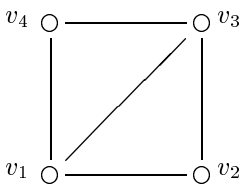


Man kann nun zeigen, daß es keinen solchen Rundweg gibt, denn sonst müsste von jeder Ecke eine gerade Anzahl von Kanten ausgehen.

Definition 1.1 *Sei V eine endliche nichtleere Menge und E eine Menge von zweielementigen Teilmengen von V. Das Paar $G = (V, E)$ heißt* Graph. *Die Elemente von V heißen* Knoten *oder* Ecken *von G, die Elemente von E* Kanten *von G.*

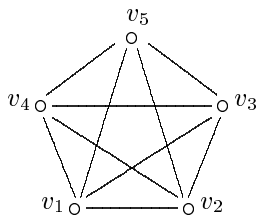
Beispiel 1.1

$$G = (V, E) = (\underbrace{\{v_1, v_2, v_3, v_4\}}_{V}, \underbrace{\{(v_1, v_2), (v_2, v_3), (v_3, v_4), (v_4, v_1), (v_1, v_3)\}}_{E})$$

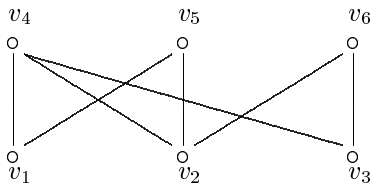


Wichtige Beispiele in der Graphentheorie sind der *vollständige Graph* einer n-elementigen Punktmenge V, dessen Kantenmenge aus allen zweielementigen Teilmengen von V besteht, und die *bipartiten Graphen*, deren Eckenmenge V so in zwei Teilmengen V_1 und V_2 zerlegt werden kann, daß Kanten nur Ecken von V_1 mit Ecken von V_2 (und umgekehrt) verbinden, aber keine Ecken von V_1 miteinander und ebenso keine Ecken von V_2 miteinander.

Beispiel 1.2 Vollständiger Graph mit 5 Ecken:



Beispiel 1.3 Bipartiter Graph mit $V_1 = \{v_1, v_2, v_3\}$, $V_2 = \{v_4, v_5, v_6\}$:



Man kann oft Graphen in der Ebene anschaulich beschreiben: Die Ecken werden durch (besonders hervorgehobene) Punkte und die Kanten durch Kurven dargestellt, die die entsprechenden Punkte verbinden. Die Kurven können sich in weiteren Punkten schneiden, die aber nicht als Ecken aufgefaßt werden.
Die Definition des Graphen schließt nicht aus, daß zwei Ecken durch mehrere Kanten verbunden sind. Wir betrachten hier nur sogenannte *schlichte Graphen* ohne Mehrfachkanten und ohne Schlingen, d.h. Kanten, bei denen die Anfangs- und Endecke übereinstimmen.

Wie üblich betrachtet man Unterstrukturen:

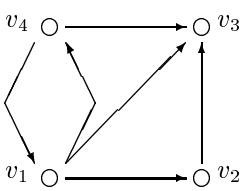
Definition 1.2 *Sei $G = (V, E)$ ein Graph. Der Graph $G_1 = (V_1, E_1)$ heißt* Teilgraph *von G, wenn $V_1 \subset V$ und $E_1 \subset E$. G_1 heißt* spannender Teilgraph *von G, wenn $V = V_1$.*

Beispielsweise durch Einbahnstraßen motiviert, kann man den Kanten eine Richtung zuweisen:

Definition 1.3 *Sei V eine endliche nichtleere Menge und E eine Menge von geordneten Paaren $(a, b) \in V \times V$ mit $a \neq b$. Dann heißt $G = (V, E)$* gerichteter Graph *(oder* Digraph*). Die Elemente von E heißen* (gerichtete) Kanten *von G, a heißt* Anfangspunkt*, b* Endpunkt *der Kante (a, b).*

Beispiel 1.4

$$G = (\{v_1, v_2, v_3, v_4\}, \{(v_1, v_2), (v_2, v_3), (v_4, v_3), (v_4, v_1), (v_1, v_3), (v_1, v_4)\})$$



Anmerkung: Wir betrachten wieder nur schlichte gerichtete Graphen. Dabei kann es zu zwei Ecken $v_1, v_2 \in V$ die beiden (entgegengesetzt gerichteten) Kanten (v_1, v_2) und (v_2, v_1) geben.

Um das „Durchlaufen“ eines Graphen zu beschreiben, benötigt man folgende

Definition 1.4 *Sei $G = (V, E)$ ein Graph.*

(a) Eine Folge $(e_1, \ldots, e_n)$ von Kanten von G heißt Kantenzug, *wenn es Ecken $v_0, v_1, \ldots, v_n \in V$ gibt mit $(v_{j-1}, v_j) = e_j$, $j = 1, \ldots, n$. v_0 heißt* Anfangspunkt, *v_n* Endpunkt *des Kantenzugs, n seine* Länge.
Der Kantenzug heißt geschlossen, *falls $v_n = v_0$ gilt.*
Sind die Kanten paarweise verschieden, dann heißt der Kantenzug Weg *bzw. im geschlossenen Fall* Kreis.

(b) G heißt zusammenhängend, *wenn es zu je zwei Ecken $a, b \in V$ einen Kantenzug gibt mit a als Anfangs- und b als End-Ecke.*

(c) G heißt Baum, *falls G zusammenhängend ist und keinen Kreis enthält.*

(d) T heißt spannender *oder* erzeugender Baum *von G, falls T spannender Teilgraph von G und ein Baum ist.*

Beispiel 1.5 In Beispiel 1.3 ist $(v_1, v_4, v_3, v_6, v_2, v_4, v_1)$ ein geschlossener Kantenzug, aber kein Kreis. $(v_1, v_5, v_2, v_4, v_1)$ ist ein Kreis. Durch Weglassen z.B. der Kanten (v_1, v_5) und (v_2, v_6) erhält man einen spannenden Baum des gegebenen Graphen.

Anmerkungen:

1. Die obigen Definitionen von Weg und Kreis lassen sich unmittelbar auf gerichtete Graphen übertragen. Man definiert in diesem Fall gerichtete Wege und gerichtete Kreise, d.h. die Richtung des Weges bzw. Kreises muß mit der Richtung der Kanten übereinstimmen.

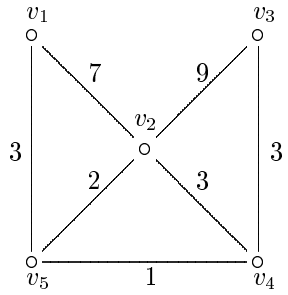
2. Ein Kantenzug (Weg, Kreis) läßt sich als Folge $(v_0, \ldots, v_n)$ von Ecken oder als Folge $(e_1, \ldots, e_n)$ von Kanten darstellen.

Für eine algorithmische Behandlung eines Graphen ist es notwendig, ihn durch eine entsprechende Datenstruktur darzustellen. Die naheliegendste Art für einen Graphen mit n Ecken ist die Darstellung durch eine *Adjazenzmatrix*, d.h. einer (n, n)-Matrix, deren Elemente a_{ij} gleich 1 sind, wenn v_i und v_j durch eine Kante verbunden sind, und 0 sonst. Bei einem ungerichteten Graphen ist die Adjazenzmatrix symmetrisch. Für den Digraph aus Beispiel 1.4 ist die Adjazenzmatrix

$$A = \begin{pmatrix} 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 \end{pmatrix}.$$

In vielen praktischen Anwendungen sind die Ecken oder die Kanten eines Graphen mit Werten (oder Gewichten) versehen. Solche Graphen heißen *ecken-* bzw. *kantenbewertet* oder *gewichtet*. Die Wertemenge ist häufig eine Teilmenge von $\mathbb{R}$. Spezielle Gewichte sind Abstände zwischen den Ecken oder Kosten, die entstehen, wenn man auf der entsprechenden Kante fährt. Das folgende Beispiel ist ein kantenbewerteter Graph mit zugehöriger bewerteter Adjazenzmatrix.

Beispiel 1.6



$$A = \begin{pmatrix} 0 & 7 & 0 & 0 & 3 \\ 7 & 0 & 9 & 3 & 2 \\ 0 & 9 & 0 & 3 & 0 \\ 0 & 3 & 3 & 0 & 1 \\ 3 & 2 & 0 & 1 & 0 \end{pmatrix}$$

Anmerkung: Für einen gewichteten Graphen $G = (V, E)$ mit Gewichtsfunktion $w : E \to \mathbb{R}$ und einen Weg $(v_1, \ldots, v_n)$ bezeichnet man $\sum_{i=1}^{n-1} w(v_i, v_{i+1})$ als Länge des (gewichteten) Weges. Die Längendefinition aus Definition 1.4 ergibt sich mit Gewichten der Größe 1.

Beispiel 1.7 (Traveling SalesmanProblem (TSP)) Gegeben seien n Städte und die Abstände zwischen je zwei Städten. Gesucht ist eine Rundreise minimaler Gesamtlänge. Die symmetrische (n, n)-Abstandsmatrix (d_{ij}) kann als gewichteter vollständiger Graph $G = (\{v_1, \ldots, v_n\}, E_n)$ mit $E_n = \{(v_i, v_j), 1 \leq i < j \leq n\}$ aufgefaßt werden. Die Menge F der zulässigen Punkte des Optimierungsproblems ist die Menge aller Kreise mit n Kanten.

Die Planung möglichst kostengünstiger Versorgungsnetze führt auf

Beispiel 1.8 (Minimal spanning tree (MST)) Sei G z.B. der Graph aus dem vorigen Beispiel. Gesucht ist ein spannender Baum, für den die Summe der Kantenlängen minimal ist. Die Menge der zulässigen Punkte des Optimierungsproblems ist also

$$F = \{\text{alle spannenden Bäume } (V, E) \text{ mit } V = \{v_1, \ldots, v_n\}\},$$

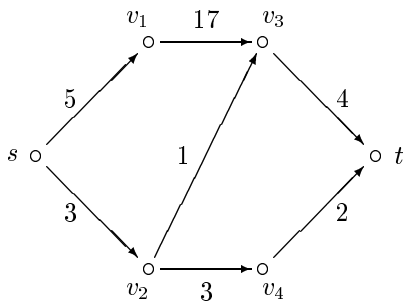
und die Kostenfunktion

$$c : (V, E) \to \sum_{(v_i, v_j) \in E} d_{ij}$$

Die Betrachtung von Transportproblemen in Leitungsnetzen mit vorgegebener Aufnahmekapazität führt auf

Definition 1.5 *Sei $G = (V, E)$ ein gerichteter Graph, $s, t \in V$ mit $(s, t) \notin E$ und es gebe einen Weg in G von s nach t. Weiter seien $c_u, c_o : E \to \mathbb{R}$ Funktionen mit $c_u \leq c_o$. Dann heißt $N = (s, t, V, E, c_u, c_o)$* Netzwerk *mit der* Quelle *s und dem* Ziel *bzw. der* Senke *t und c_u* untere Kapazität*, c_o* obere Kapazität *von N. Ist die untere Kapazität nicht angegeben, so ist sie identisch Null.*

Beispiel 1.9 Netzwerk mit unterer Kapazität $c_u \equiv 0$. Die oberen Kapazitäten c_o sind bei den Kanten angegeben.



Schickt man z.B. durch ein (Rohrleitungs-)Netzwerk von der Quelle zur Senke eine Flüssigkeit, so sollte innerhalb des Netzwerks nichts verschwinden und nichts dazukommen, d.h. die Menge, die in einen inneren Knoten hineinkommt, sollte aus ihm herausfließen. Weiter werden durch die Kapazitätsfunktionen die Mengen, die durch eine bestimmte Kante strömen, nach unten und oben beschränkt.

Definition 1.6 *Ein* Fluß *in einem Netzwerk N ist eine Abbildung $f : E \to \mathbb{R}$ (bzw. ein Vektor des $\mathbb{R}^{|E|}$) mit*

(1) $c_u(u, v) \leq f(u, v) \leq c_o(u, v)$ *für jedes* $(u, v) \in E$.

(2) $\displaystyle\sum_{(u,v)\in E} f(u, v) = \sum_{(v,u)\in E} f(v, u)$ *für alle* $v \in V \setminus \{s, t\}$.

$\displaystyle|f| = \sum_{(s,u)\in E} f(s, u) - \sum_{(u,s)\in E} f(u, s)$ *heißt* Wert $|f|$ *des Flusses.*

Anmerkungen:

(a) Es gilt $\displaystyle|f| = \sum_{(u,t)\in E} f(u, t) - \sum_{(t,u)\in E} f(t, u)$.

(b) Oft wird bei Netzwerken gefordert, daß für Kanten der Form (u, s) sowie (t, u) die Kapazitäten und damit jeder mögliche Fluß Null sind. Dann ist natürlich der Wert eines Flusses $\displaystyle|f| = \sum_{(s,u)\in E} f(s, u)$.

1.2 Das Problem des kürzesten Wegs

Definition 1.7 *Das Problem, zu jedem gerichteteten, gewichteten Graphen mit Gewichtsfunktion $w : E \to \mathbb{R}$ und mindestens zwei Ecken s und t einen gerichteten Weg minimaler Länge von s nach t zu finden, heißt* Problem des kürzesten Wegs (shortest path problem, SP). *s nennt man wieder Quellknoten, t Zielknoten. Die minimale Länge bezeichnen wir mit $d(s,t)$.*

Betrachtet man SP als Optimierungsproblem, dann ist die zulässige Menge

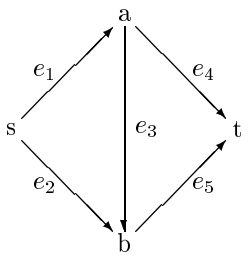
$$F = \{P = (e_{j_1}, \ldots, e_{j_k}) \mid P \text{ ist ein gerichteter Weg von } s \text{ nach } t\}$$

und die Kostenfunktion $c(P) = \sum_{i=1}^{k} w(e_{j_i})$.

Zur Formulierung von SP als LP betrachtet man die (Knoten–Kanten–) Inzidenzmatrix $A = (a_{ij})$, $i = 1, \ldots, |V|$, $j = 1, \ldots, |E|$, des Graphen G mit

$$a_{ij} = \begin{cases} +1 & \text{wenn die Kante } e_j \text{ aus dem Knoten } v_i \text{ herausführt} \\ -1 & \text{wenn die Kante } e_j \text{ in den Knoten } v_i \text{ hineinführt} \\ 0 & \text{sonst} \end{cases} .$$

Beispiel 1.10 G :



$$A = \begin{array}{c|ccccc} & e_1 & e_2 & e_3 & e_4 & e_5 \\ s & +1 & +1 & 0 & 0 & 0 \\ t & 0 & 0 & 0 & -1 & -1 \\ a & -1 & 0 & +1 & +1 & 0 \\ b & 0 & -1 & -1 & 0 & +1 \end{array}$$

$w(e_1) = w(e_5) = 1,\ w(e_2) = w(e_3) = 2,\ w(e_4) = 3.$

Wir setzen die unteren Kapazitäten Null, die oberen Kapazitäten 1 (als triviale Kapazitäten) und betrachten einen Weg von s nach t als speziellen Fluß f, der s mit dem Wert 1 verläßt und bei t mit dem Wert 1 ankommt, für den also $s^\top f = -t^\top f = 1$ gilt. Für eine andere Ecke v_i muß wegen der Fluß-Erhaltungs-Bedingung $v_i^T f = 0$ gelten, und damit ergibt sich insgesamt für einen Weg das Gleichungssystem

$$Af = (1, -1, 0, \ldots, 0)^\top,$$

wobei die „1“ zu s und die „-1“ zu t gehört. Im Allgemeinen sind die Koeffizienten von f keine ganzen Zahlen, aber für den Fluß mit den geringsten Kosten

$$\min\{c^\top f \mid Af = (1, -1, 0, \ldots, 0)^\top,\ f \geq 0\}$$

existiert eine optimale Lösung, in der jedes f_i 0 oder 1 ist und so einen kürzesten Weg von s nach t in G darstellt.

Weiter sind die $|V|$ Gleichungen des Problems redundant, da die Flußerhaltung an $|V| - 1$ Knoten die Flußerhaltung am ausgelassenen Knoten sichert. Man kann daher eine Gleichung (z.B. für t) weglassen.

Beispiel 1.10 (Forts.) Weglassen von t ergibt das Tableau:

0	1	2	2	3	1
1	1	1	0	0	0
0	−1	0	1	1	0
0	0	−1	−1	0	1

Addition der Zeile 1 zur Zeile 2 ergibt die Basis $\{1, 4, 5\}$. Das Starttableau wird so:

−4	0	−1	0	0	0
1	1	1	0	0	0
1	0	1	1	1	0
0	0	−1	−1	0	1

Eine Basis stellt eine Menge von $|V| - 1$ Kanten dar, von denen eine Teilmenge einem Weg von s nach t mit den gegebenen Kosten entspricht. Die Kanten, die nicht auf dem Weg liegen, stellen entartete Komponenten der Basis dar. Im Beispiel stellt die Basis den Weg (e_1, e_4) mit den Kosten 4 und die entartete Kante e_5 dar.

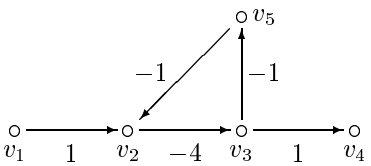
Beispiel 1.10 (Forts.) Ein Pivotschritt mit ξ_{22} ergibt das optimale Tableau

−3	0	0	1	1	0
0	1	0	−1	−1	0
1	0	1	1	1	0
1	0	0	0	1	1

Dies entspricht dem optimalen Weg (e_2, e_5) mit Kosten 3 und der entarteten Kante e_1. Eine Kante ist zur Basis hinzugefügt worden und eine andere weggenommen, um einen neuen Weg mit geringeren Kosten zu finden.

Interpretiert man einen Weg in einem gewichteten Netzwerk als Fahrroute, wobei die Länge einer Kante den Kosten entspricht, dann sind auch negative „Längen“ möglich (z.B. bei Fahrten mit Gewinn). Ein kürzester Weg von s nach t gibt dann eine Fahrtroute mit maximalem Gewinn an. Erweitert man die möglichen Fahrtrouten auf Kantenzüge und gibt es in dem Netzwerk Kreise negativer Länge, dann gibt es keine optimale Lösung.

Beispiel 1.11 In dem Graph



gibt es zwischen v_1 und v_4 nur einen Weg, und dieser hat die Länge -2, d.h. es gilt $d(v_1, v_4) = -2$. Andererseits gibt es unendlich viele Kantenzüge von v_1 nach v_4 mit den Längen -2, -8, -14,...

Wir beschränken uns hier auf gerichtete Netzwerke ohne Kreise negativer Länge.

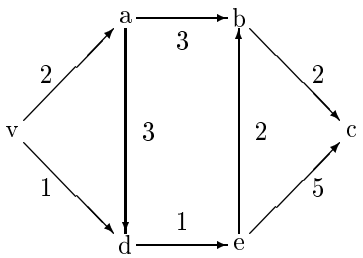
Der folgende Algorithmus behandelt nur gerichtete Graphen $G = (V, E)$ mit nichtnegativer Gewichtsfunktion $w : E \to \mathbb{R}$ und bestimmt zu einem beliebigen festen $v \in V$ die Länge eines kürzesten Weges von v zu jedem anderen $u \in V$. O.B.d.A. sei der Graph vollständig, d.h. für jedes $u, x \in V$ ist $(u, x) \in E$ (setze gegebenenfalls $w_{ux} := w(u, x) = \infty$).
Für $W \subset V$ und $x \in V$ bezeichne $\rho_W(x)$ die Länge des kürzesten Weges von v nach x, der nur Punkte aus W als Zwischenpunkte benutzt. Dies wird sukzessive bestimmt durch

Algorithmus 1.1 (Dijkstra)
Eingabe: Ein gerichteter Graph $G = (V, E)$ mit Gewichtsfunktion $w : E \to \mathbb{R}$, $v \in V$.
Ausgabe: Die Länge der kürzesten Wege von v zu den übrigen Knoten.

```
begin
   W := {v}; ρ_W(v) := 0;
   for all y ∈ V \ {v} do ρ_W(y) := w_vy;
   while W ≠ V do
   begin
      finde x mit ρ_W(x) = min{ρ_W(y) | y ∈ V \ W};
      W := W ∪ {x};
      for all y ∈ V \ W do
      begin
         ρ_W(y) := min{ρ_W(y), ρ_W(x) + w_xy};
      end;
   end;
end.
```

Satz 1.1 *Dijkstras Algorithmus bestimmt die Länge der kürzesten Wege von v zu den übrigen Knoten korrekt.*

Beispiel 1.12



Adjazenzmatrix $A = \begin{pmatrix} 0 & 2 & \infty & \infty & 1 & \infty \\ \infty & 0 & 3 & \infty & 3 & \infty \\ \infty & \infty & 0 & 2 & \infty & \infty \\ \infty & \infty & \infty & 0 & \infty & \infty \\ \infty & \infty & \infty & \infty & 0 & 1 \\ \infty & \infty & 2 & 5 & \infty & 0 \end{pmatrix}$

(0) $W = \{v\}$, $\rho(v) = 0$;
$\rho(d) = 1$, $\rho(a) = 2$, $\rho(b) = \rho(e) = \rho(c) = \infty$.

(1) Markiere d. $W = \{v, d\}$, $\rho(v) = 0$, $\rho(d) = 1$;
$\rho(e) = 1 + 1$, $\rho(a) = 2$, $\rho(b) = \rho(c) = \infty$.

(2) Markiere e. $W = \{v, d, e\}$, $\rho(v) = 0$, $\rho(d) = 1$, $\rho(e) = 2$;
$\rho(b) = 2 + 2$, $\rho(c) = 2 + 5$, $\rho(a) = 2$.

(3) Markiere a. $W = \{v, d, e, a\},\ \rho(v) = 0,\ \rho(d) = 1,\ \rho(e) = 2,\ \rho(a) = 2;$
$\rho(b) = \min\{4, 2+3\} = 4,\ \rho(c) = 7.$

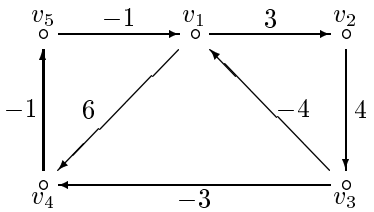
(4) Markiere b. $W = \{v, d, e, a, b\},\ \rho(v) = 0,\ \rho(d) = 1,\ \rho(e) = 2,\ \rho(a) = 2,\ \rho(b) = 4;$
$\rho(c) = \min\{7, 4+2\} = 6.$

Somit gilt $d(v,v) = 0,\ d(v,d) = 1,\ d(v,c) = d(v,a) = 2,\ d(v,b) = 4,\ d(v,c) = 6.$

Aufwand des Algorithmus: Für $|V| = n$ erfordert die Initialisierung $n+1$ Operationen. Die Schleife wird $n-1$ mal durchlaufen und erfordert jeweils $O(n)$ Operationen. Damit ist der Gesamtaufwand $O(n^2)$ Operationen.

Bei negativen Kantenbewertungen arbeitet der Algorithmus von Dijkstra i.a. nicht mehr korrekt:

Beispiel 1.13 Der folgende Graph hat zwar einige Kanten mit negativer Bewertung, aber keine Kreise negativer Länge.



Der kürzeste Weg von v_1 nach v_4 ist (v_1, v_2, v_3, v_4) mit Länge 4. Der Algorithmus von Dijkstra liefert als kürzesten Weg die Kante (v_1, v_4) mit Länge 6.

Der folgende Floyd–Warshall–Algorithmus arbeitet auch mit negativen Gewichten und erkennt Kreise negativer Länge. Er berechnet die Abstände aller Knoten gleichzeitig.

Gegeben sei ein gerichteter gewichteter Graph $G = (V, E)$ mit Knoten $v_1, \ldots, v_n$ und Gewichtsfunktion $w : E \to \mathbb{R}$. Weiter setzen wir $w(v_i, v_i) := 0$, $1 \le i \le n$, und $w(v_i, v_j) := \infty$, falls $(v_i, v_j) \notin E$. Für einen Kantenzug $\omega = (v_1, \ldots, v_s)$ in G sei $l(\omega)$ die Länge von ω.

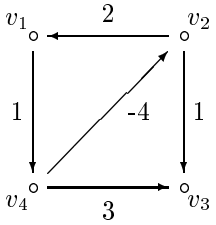
Man kann die Matrix $W = \Big(w(v_i, v_j)\Big)$ als Abstandsmatrix auffassen, wobei nur Kantenzüge zugelassen sind, die aus einer Kante bestehen. Wir geben nun eine Operation an, die zu einer gegebenen Abstandsmatrix und einem festen Knoten v_j eine Abstandsmatrix berechnet, bei der die vorherigen Kantenzüge ausgedehnt werden auf Kantenzüge, die v_j enthalten.

Definition 1.8 *Sei* $D = (d_{ij}) \in \mathbb{R}^{n \times n}$, $j \in \{1, \ldots, n\}$ *fest. Eine* Dreiecksoperation *auf* D *bezüglich* j *ist definiert durch*

$$d'_{ik} = \begin{cases} \min\{d_{ik}, d_{ij} + d_{jk}\} & \text{für alle } i, k \in \{1, \ldots, n\} \setminus \{j\} \\ d_{ik} & \text{sonst} \end{cases}$$

Beispiel 1.14

G :



mit $(d_{ij}) = \begin{pmatrix} 0 & \infty & \infty & 1 \\ 2 & 0 & 1 & \infty \\ \infty & \infty & 0 & \infty \\ \infty & -4 & 3 & 0 \end{pmatrix}$.

Dreiecksoperation für $j = 4$ ergibt $(d'_{ij}) = \begin{pmatrix} 0 & \mathbf{-3} & \mathbf{4} & 1 \\ 2 & 0 & 1 & \infty \\ \infty & \infty & 0 & \infty \\ \infty & -4 & 3 & 0 \end{pmatrix}$.

Bei dem folgenden Algorithmus wird das Minimum der Länge aller Kantenzüge von einem Knoten v zu einem Knoten u berechnet. Das stimmt mit der Länge des entsprechenden kürzesten Weges überein:

Satz 1.2 *Sei $G = (V, E)$ ein gerichteter gewichteter Graph. Existiert in G ein geschlossener Kantenzug negativer Länge, so existiert in G ein Kreis negativer Länge.*

Satz 1.3 *Sei $G = (V, E)$ ein gerichteter gewichteter Graph, der keinen Kreis negativer Länge enthält. Dann gilt für jeden Kantenzug $\omega = (v_1, \ldots, v_k)$ $\quad l(\omega) \geq d(v_1, v_k)$.*

Nach einer endlichen Zahl (n) von Dreiecksoperationen, ausgehend von der Abstandsmatrix für Kantenzüge mit einer Kante, erhält man entweder einen Kreis mit negativer Länge oder eine Matrix, deren Elemente die Länge der kürzesten Wege zwischen je zwei Knoten angeben:

Satz 1.4 *Sei $G = (V, E)$ ein gerichteter gewichteter Graph mit Knoten $v_1, \ldots, v_n$ und $D^{(0)} = (d_{ij})$ die Abstandsmatrix zu G wie vor Definition 1.8. Für $k = 1, \ldots n$ sei $D^{(k)} = (d_{ij}^{(k)})$ das Ergebnis der Dreiecksoperation bezüglich k, angewandt auf $D^{(k-1)}$. Dann gilt:*

(a) Es gibt einen Kreis negativer Länge genau dann, wenn $d_{ii}^{(k)} < 0$ für ein i und ein k gilt.

(b) Gilt $d_{ii}^{(k)} \geq 0$ für alle i und k, so ist $d_{ij}^{(k)}$ die Länge des kürzesten Weges von v_i nach v_j, wenn nur Wege zugelassen sind, die als Zwischenknoten nur Elemente aus $\{v_1, \ldots, v_k\}$ benutzen.

Zur Bestimmung der zugehörigen Wege mit kürzester Länge dienen Matrizen $(e_{ij}^{(k)})$, $1 \leq k \leq n$, in der für den kürzesten Weg von v_i nach v_j mit möglichen Zwischenknoten aus $\{v_1, \ldots, v_k\}$, falls es einen solchen gibt, die Nummer des letzten Zwischenknotens eingetragen wird.

Algorithmus 1.2 (Floyd–Warshall)
Eingabe: Eine (n, n) Abstandsmatrix $(d_{ij}^{(0)})$
Ausgabe: Feststellung eines Kreises negativer Länge oder die Abstandsmatrix d_{ij} des vollständigen Graphen, sowie für jedes v_i, v_j die höchste Nummer der Zwischenknoten auf dem kürzesten Weg von v_i nach v_j

```
begin
   for all i do
   for all j do
   begin
      d_ij := d_ij^(0); e_ij := i; d_ii := 0;
    end;
   Negativer_Kreis := 'Nein';
   j := 1;
   while (Negativer_Kreis = 'Nein') and j ≤ n do
   begin
      for k = 1 to n do
      for l = 1 to n do
      begin
         if d_kl > d_kj + d_jl then
         begin
            d_kl := d_kj + d_jl;
            e_kl := e_jl;
            if (k = l) and d_kk < 0 then Negativer_Kreis := 'Ja';
          end;
       end;
      j := j + 1;
    end;
end.
```

Beispiel 1.14 (Forts.)

$$(d_{ij}^{(0)}) = \begin{pmatrix} 0 & \infty & \infty & 1 \\ 2 & 0 & 1 & \infty \\ \infty & \infty & 0 & \infty \\ \infty & -4 & 3 & 0 \end{pmatrix} \qquad (e_{ij}^{(0)}) = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 2 & 2 & 2 & 2 \\ 3 & 3 & 3 & 3 \\ 4 & 4 & 4 & 4 \end{pmatrix}$$

$$(d_{ij}^{(1)}) = \begin{pmatrix} 0 & \infty & \infty & 1 \\ 2 & 0 & 1 & \mathbf{3} \\ \infty & \infty & 0 & \infty \\ \infty & -4 & 3 & 0 \end{pmatrix} \qquad (e_{ij}^{(1)}) = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 2 & 2 & 2 & \mathbf{1} \\ 3 & 3 & 3 & 3 \\ 4 & 4 & 4 & 4 \end{pmatrix}$$

$$(d_{ij}^{(2)}) = \begin{pmatrix} 0 & \infty & \infty & 1 \\ 2 & 0 & 1 & 3 \\ \infty & \infty & 0 & \infty \\ \mathbf{-2} & -4 & \mathbf{-3} & \mathbf{-1} \end{pmatrix} \qquad (e_{ij}^{(2)}) = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 2 & 2 & 2 & 1 \\ 3 & 3 & 3 & 3 \\ \mathbf{2} & 4 & \mathbf{2} & \mathbf{1} \end{pmatrix}$$

$d_{44}^{(2)} = -1$ zeigt an, daß es einen Kreis mit Ecke v_4 negativer Länge, nämlich -1, gibt. Den Weg selber erhält man aus $(e_{ij}^{(2)})$ durch Rückverfolgung der Ecken. Aus $e_{44}^{(2)} = 1$ ergibt sich v_1, aus $e_{41}^{(2)} = 2$ v_2, aus $e_{42}^{(2)} = 4$ wieder v_4, d.h. insgesamt der Kreis $v_4 v_2 v_1 v_4$.

Abschätzung des Aufwands für den Floyd–Warshall–Algorithmus: Die beiden „for“–Schleifen erfordern einen Aufwand von $O(n^2)$, die „while“–Schleife wird höchstens n Mal durchlaufen. Damit ist der Gesamtaufwand $O(n^3)$.

1.3 Minimal spannende Bäume

Definition 1.9 (MST) *Sei $G = (V, E)$ ein (kanten-)bewerteter zusammenhängender ungerichteter Graph, B ein spannender Baum. Die Summe der Bewertungen der Kanten von B heißt* Kosten *oder* Länge des Baums . *B heißt* minimaler spannender Baum, *falls kein anderer spannender Baum B' von G mit geringeren Kosten existiert.*

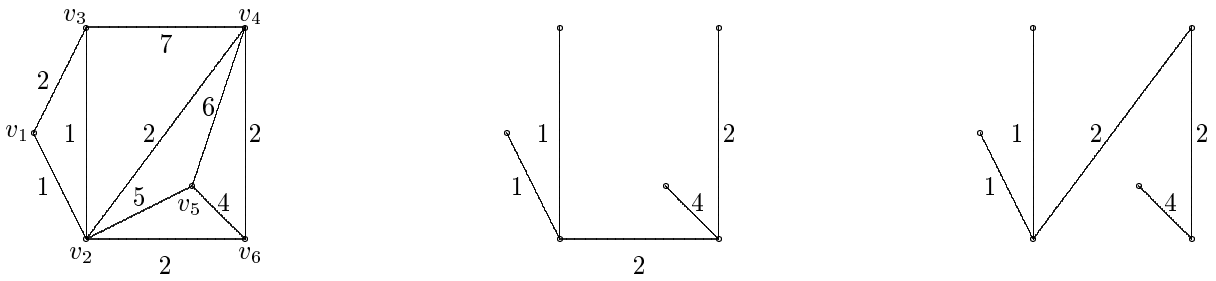
Anmerkungen:

(1) MST hat i.a. keine eindeutige Lösung, wie das folgende Beispiel zeigt. Haben in einem Graph alle Kanten gleiche Länge (d.h. ist die Gewichtsfunktion konstant), dann ist jeder spannender Baum minimal.

(2) Ein nicht zusammenhängender Graph zerfällt in endlich viele Zusammenhangskomponenten $G_i = (V_i, E_i)$, $1 \leq i \leq k$. Ist $B_i = (V_i, E_i')$ ein spannender Baum von G_i, $1 \leq i \leq k$, dann heißt $F = \bigcup\{B_i; 1 \leq i \leq k\}$ *spannender Wald* von G. Das Problem der Suche nach einem spannenden Wald für einen nichtzusammenhängenden Graphen löst man durch die Suche nach einem minimalen spannenden Baum für jede Zusammenhangskomponente. Wir werden uns zunächst auf zusammenhängende Graphen beschränken.

(3) Die Kantenlängen (bzw. -gewichte) kann man durch Einträge in eine symmetrische $(|V|, |V|)$–Matrix beschreiben. Ergänzt man wieder für fehlende Kanten bezüglich des vollständigen Graphen mit gleicher Eckenmenge die Matrix mit den Werten ∞, dann kann man sich auf Betrachtung vollständiger Graphen beschränken.

Beispiel 1.15 Graph mit zwei verschiedenen minimalen spannenden Bäumen:



Grundlage der folgenden Algorithmen zur Lösung von MST ist

Satz 1.5 *Sei $G = (V, E)$ ein kantenbewerteter zusammenhängender Graph mit Gewichtsfunktion $w : E \to \mathbb{R}$, $U \subset V$, $(u, v) \in E$ mit $u \in U, v \notin U$ und $w(u, v) = \min\{w(x, y) \mid x \in U, y \notin U\}$. Dann existiert ein minimaler spannender Baum, der (u, v) enthält.*

Der Satz ermöglicht folgenden Algorithmus von Prim:

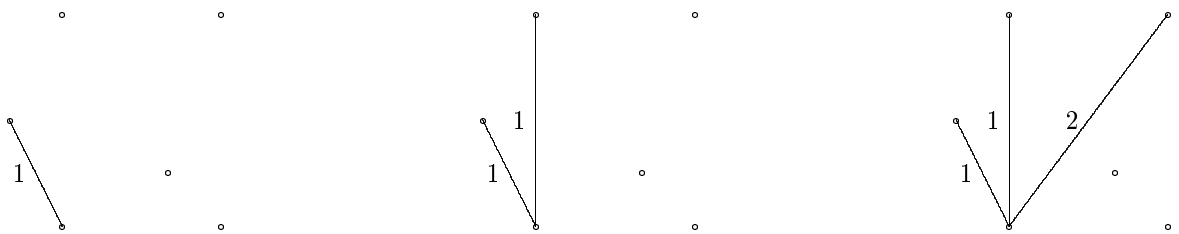
Algorithmus 1.3 (MST1)
Eingabe: $V = \{v_1, \ldots, v_n\}$ und die Abstände c_{ij};
Ausgabe: Ein minimaler spannender Baum (V, T)

```
begin
   (* Initialisierung *)
   U := {v_1};
   T := ∅;
   for all v ∈ V \ {v_1} do Nächster[v] := v_1;
   while U ≠ V do
   begin
      (* Finde die zu U benachbarte Ecke *)
      min := ∞;
      for all v ∈ V \ U do
      begin
         if w(v,Nächster[v]) < min then
         begin
            min := w(v,Nächster[v]);
            Neue_Ecke := v;
         end;
      end;
      U := U∪ Neue_Ecke;
      T := T ∪{ (Neue_Ecke, Nächster[Neue_Ecke])};
      (* Nächster aktualisieren *)
      for all v ∈ V \ U do
      begin
         if w(v, Nächster[v])> w(v, Neue_Ecke) then Nächster[v] := Neue_Ecke;
      end;
   end;
end.
```

Das Datenfeld *Nächster* dient zur Suche der kürzesten Kante, die U verläßt. Für jede Ecke $v \in V \setminus U$ gibt *Nächster*$[v]$ an, welche Ecke von U am nächsten zu v liegt. Daher muß man zum Auffinden der kürzesten Kante, die U verläßt, nur die kürzeste Kante unter den Kanten $(v,$*Nächster*$[v]), v \notin U$, bestimmen.

Satz 1.6 *Der Algorithmus MST1 löst MST in einer Zeit* $O(|V|^2)$.

Beispiel 1.15 (Forts.) Durchführung des Algorithmus ergibt:



MST1 ist i.a. bestmöglich, da für MST jede Kante betrachtet werden muß. Für Graphen mit wenigen eigentlichen Kanten, d.h. $w(u,v) = \infty$ für viele Kanten $(u,v) \in E$ des zugehörigen vollständigen Graphen, läßt sich MST1 verbessern.

Folgender Algorithmus MST2 (Kruskal,Berge) beginnt mit dem Graph $(V, \emptyset)$, d.h. mit $n = |V|$ Zusammenhangskomponenten. Bei jedem Schritt wird jeder Komponente eine Kante minimaler Länge zugeordnet, die die Komponente „verläßt“, d.h. eine Ecke liegt in dieser Komponente, die andere nicht. Zum Abschluß des Schrittes werden diese Kanten in den Graph eingefügt. Dadurch wird die Anzahl der Komponenten verringert (mindestens halbiert). Der Algorithmus bricht ab, wenn der entstandene Graph zusammenhängend ist.

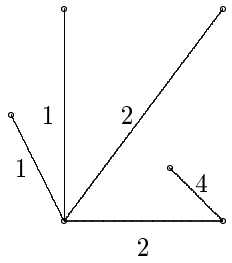
Algorithmus 1.4 (MST2)
Eingabe: Ein zusammenhängender Graph $G = (V, E)$ und für jedes $(u,v) \in E$ ein Abstand $w(u,v)$.
Ausgabe: Ein minimaler spannende Baum (V, T) von G.

```
begin
  (* Initialisierung; C ist die Menge der Zusammenhangskomponenten von (V,T) *)
  T := ∅; C := {{v_1}, ... , {v_n}};
  while |C| ≠ 1 do
  begin
    for all S ∈ C do min[S] := ∞;
    for all (u,v) ∈ E do
    begin
      S_u, S_v := Komponenten, die v und u enthalten;
      if S_u ≠ S_v then
      begin
        if w(u,v) < min[S_u] then
        begin
          min[S_u] := w(u,v);
          Kürzester[S_u] := (u,v);
        end;
        if w(u,v) < min[S_v] then
        begin
          min[S_v] := w(u,v);
          Kürzester[S_v] := (u,v);
        end;
      end;
    end;
```

for all $S \in C$ **do** $T := T \cup \{$ Kürzester[S] $\}$;
finde die Menge C der Zusammenhangskomponenten von (V, T);
end;
end.

Satz 1.7 *MST2 löst MST für (V, E) und w in einer Zeit $O(|E| \log |V|)$.*

Beispiel 1.15 (Forts.) Betrachtet man die Kanten in der Reihenfolge (v_1, v_2), (v_1, v_3), (v_2, v_3), (v_2, v_4), (v_2, v_5), (v_2, v_6), (v_3, v_4), (v_4, v_5), (v_4, v_6), (v_5, v_6), dann ergibt die Durchführung des Algorithmus:



Anmerkung: Man kann die Kanten zu Beginn des Algorithmus nach ihrer Länge sortieren. In vielen Fällen müssen dann nicht in jeder Stufe alle Kanten betrachtet werden. Enthält aber jeder minimale spannende Baum eines Graphen G eine längste Kante von G, dann wird durch die Sortierung nichts gewonnen.

Wir suchen nun spannende Bäume (d.h. Teilgraphen mit möglichst wenigen Kanten), deren Gesamtlänge maximal ist:

Definition 1.10 (MWF) *Sei $G = (V, E)$ ein (kanten-)bewerteter ungerichteter Graph mit positiver Gewichtsfunktion $w > 0$, F ein kreisfreier Teilgraph. F heißt* Wald mit maximalem Gesamtgewicht $w(F) = \sum_{e \in F} w(e)$, *falls kein anderer kreisfreier Teilgraph F' von G mit größerem Gesamtgewicht existiert.*

MWF ist eng mit MST verwandt. Sei G zunächst zusammenhängend. Wegen $w_{ij} := w(v_i, v_j) > 0$ für alle $v_i, v_j \in V$, $i \neq j$, ist ein Wald mit maximalem Gesamtgewicht dann ein spannender Baum von G. Sei jetzt

$$W := \max\{w_{ij}, 1 \leq i, j \leq |V|\} \qquad \text{und} \qquad c_{ij} = W - w_{ij} \quad \text{für alle } 1 \leq i, j \leq |V|.$$

Dann ist $c(B)$ für jeden spannenden Baum B mit dem Gesamtgewicht $w(T)$ durch

$$w(T) = (|V| - 1)W - c(T)$$

verknüpft. B hat maximales Gesamtgewicht bezüglich w genau dann, wenn B minimal bezüglich c (d. h. ein MST) ist.

Für unzusammenhängendes G betrachtet man G als Vereinigung der Zusammenhangskomponenten:

Satz 1.8 *Sei $G = (V, E)$ ein Graph mit den Zusammenhangskomponenten $S_1, \ldots, S_k$. Weiter sei für jedes $i \in \{1, \ldots, k\}$ B_i ein spannender Baum mit maximalem Gewicht. Dann ist die Vereinigung $\bigcup\{B_i; 1 \leq i \leq k\}$ ein Wald mit maximalem Gewicht.*

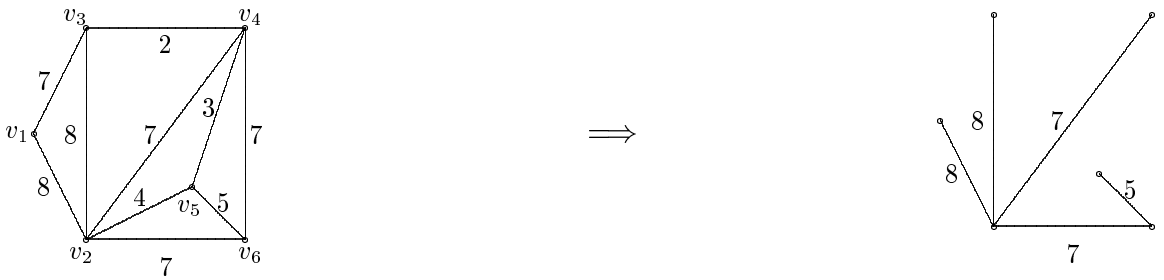
Das Gegenstück zum Satz 1.5 für das MWF–Problem ist:

Satz 1.9 *Sei $G = (V, E)$ ein kantenbewerteter Graph mit positiver Gewichtsfunktion $w : E \to \mathbb{R}$, $U \subset V$, $(u, v) \in E$ mit $u \in U, v \notin U$ und $w(u, v) = \max\{w(x, y) \mid x \in U, y \notin U\}$. Dann existiert ein maximaler spannender Wald, der (u, v) enthält.*

Damit ergibt sich für MWF der folgende „natürliche“ Algorithmus:

Algorithmus 1.5 (Der gierige Algorithmus)
Eingabe: $G = (V, E)$ mit einer positiven Gewichtsfunktion w_{ij};
Ausgabe: Ein Wald F mit maximalem Gesamtgewicht;
begin
  $F := \emptyset$;
  $C := \{\{v_1\}, \ldots, \{v_n\}\}$;
  while $E \neq \emptyset$ **do**
  begin
    (u, v) Kante maximalen Gewichts in E;
    $E := E \setminus \{(u, v)\}$;
    if u und v sind nicht in derselben Komponente von (V, F) **then**
    begin
      $F := F \cup \{(u, v)\}$;
      fasse die Komponenten von u und v zusammen;
    end;
  end;
end.

Beispiel 1.16



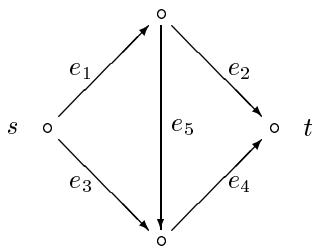
Der gierige Algorithmus schnappt also stets einfach nach dem größten erreichbaren Brocken. Trotzdem löst er das Problem:

Satz 1.10 *Der gierige Algorithmus löst das MWF–Problem.*

1.4 Maximale Flüsse und der Algorithmus von Ford–Fulkerson

Definition 1.11 *Die Aufgabe, zu jedem Netzwerk $N = (s, t, V, E, c_o)$ mit $n = |V|$ Knoten und $m = |E|$ Kanten den Fluß $f \in \mathbb{R}^m$ mit maximalem Wert zu finden, heißt* Problem des größten Flusses (max flow problem, MFP) .

Beispiel 1.17 Netzwerk mit $n = 4$ Knoten, $m = 5$ Kanten und $c_o(e_i) = 1$, $i = 1, \ldots, 5$.



Ein maximaler Fluß hat den Wert 2.

Wir haben bei der Behandlung von Beispiel 1.10 das Problem des kürzesten Weges mit Hilfe der Inzidenzmatrix in ein Fluß-Problem umgewandelt und dieses als LP-Problem gelöst. Analog ergibt sich die Formulierung des Maximalflußproblems als LP:
Gegeben sei das Netzwerk $N = (s, t, V, E, c_o)$. A sei die Inzidenzmatrix des Graphen. Dann ist ein s–t–Fluß f mit Wert $v = |f|$ gegeben durch die Restriktionen

$$Af + dv = 0, \quad 0 \le f \le c_o, \qquad \text{mit} \quad d \in \mathbb{R}^n, \quad d_i = \begin{cases} -1, & i = s \\ +1, & i = t \\ 0, & \text{sonst} \end{cases} .$$

v ist zu maximieren.

Entfernt man aus einem Fluß-Netzwerk genügend viele Kanten, dann liegen Quelle und Senke in verschiedenen Zusammenhangskomponenten, ein Fluß von der Quelle zur Senke ist also nicht mehr möglich. Existierte vorher ein Fluß, dann mußte sein Wert vollständig durch die entfernten Kanten geflossen sein, er kann also nicht größer als die Gesamt–Kapazität der entfernten Kanten sein.

Definition 1.12 *(a) Sei V eine beliebige nichtleere Menge. (S, T) heißt* Partition *von V, wenn $S, T \subset V$, $S \cup T = V$ und $S \cap T = \emptyset$.*

*(b) Sei (s, t, V, E, c_0) ein Netzwerk. Eine Partition (S, T) von V heißt (s–t–)*Schnitt *des Netzwerks, wenn $s \in S$ und $t \in T$.*

(c) $C(S,T) = \sum_{\substack{(v_i,v_j)\in E,\\ v_i\in S,\, v_j\in T}} c_o(v_i, v_j)$ *heißt* Kapazität *des Schnittes (S, T).*

Lemma 1.1 *Sei $N = (s, t, V, E, c_o)$ ein Netzwerk, (S, T) ein Schnitt und f ein Fluß. Dann gilt*

$$|f| = \sum_{\substack{(v_i,v_j)\in E,\\ v_i\in S,\, v_j\in T}} f(v_i, v_j) - \sum_{\substack{(v_i,v_j)\in E,\\ v_i\in T,\, v_j\in S}} f(v_i, v_j).$$

Insbesondere ist $|f| \leq C(S, T)$.

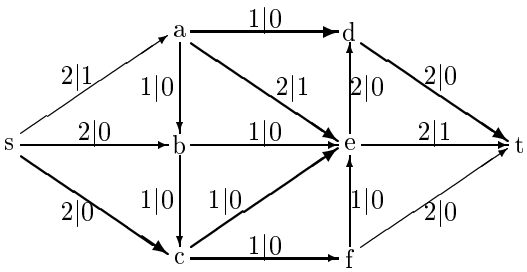
Wir wollen nun Kriterien für einen maximalen Fluß entwickeln.

Definition 1.13 *Gegeben sei ein Netzwerk $N = (s, t, V, E, c_o)$ und ein (zulässiger) s–t–Fluß f. Weiter sei $\omega = (s = v_0, v_1, \ldots, v_n = t)$ ein ungerichteter s–t–Weg.*

(a) Eine Kante (v_{i-1}, v_i) von ω mit $(v_{i-1}, v_i) \in E$ heißt Vorwärtskante *von ω, eine Kante (v_{i-1}, v_i) von ω mit $(v_i, v_{i-1}) \in E$ heißt* Rückwärtskante *von ω.*

(b) Gilt für jede Vorwärtskante e von ω $\quad f(e) < c_o(e)$ und für jede Rückwärtskante $f(e) > 0$, dann heißt ω zunehmender Weg *(bezüglich f).*

Beispiel 1.18 Bei jeder Kante e wird durch $x|y$ die (obere) Kapazität $c_o(e) = x$ und der Fluß $f(e) = y$ bezeichnet.



Ein zunehmender Weg ist (s, c, e, a, d, t).

Satz 1.11 *Ein Fluß f auf einem Netzwerk $N = (s, t, V, E, b)$ ist genau dann maximal, wenn es keinen bzgl. f zunehmenden Weg gibt.*

Beispiel 1.18 (Fortsetzung) Für das angegebene Beispiel ist $d = 1$.

Korollar 1.1 *Ein Fluß f auf einem Netzwerk N ist genau dann maximal, wenn $|f| = C(S, T)$ gilt, wobei S die Menge aller von s aus auf einem zunehmenden Weg erreichbaren Punkte sei.*

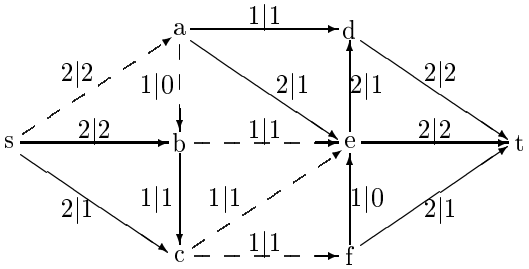
Bei Beispiel 1.10 waren die Kapazitäten ganzzahlig, nämlich 1, und es wurde ein optimaler Fluß gesucht. Dieser muß nicht unbedingt ganzzahlig sein. Es gilt aber

Satz 1.12 *Ist N ein Netzwerk mit nur ganzzahligen Kapazitäten, dann gibt es einen ganzzahligen maximalen Fluß auf N.*

Für den Zusammenhang zwischen den Schnitten und den Flüssen ergibt sich

Satz 1.13 (Maximalfluß–Minimalschnitt) *Der maximale Wert der Flüsse auf einem Netzwerk N ist gleich der minimalen Kapazität der Schnitte in N.*

Beispiel 1.18 (Forts.)



Ein minimaler Schnitt ist z.B. (S, T) mit $S = \{s, b, c\}$. Die Kapazität des Schnitts und der Wert des angegebenen Maximalflusses ist 5.

Bemerkungen:

(1) In der Praxis spielen irrationale Kapazitäten in Netzwerken keine Rolle, da im Computer nur rationale Zahlen darstellbar sind.

(2) Die Sätze 1.12 und 1.13 liefern einen Algorithmus, der einen Fluß mit maximalem Wert findet. Für ganzzahliges c_o ist dieser Fluß (und natürlich sein Wert) sogar ganzzahlig.

(3) Die Sätze machen keine Eindeutigkeitsaussage, d.h. es kann mehrere Flüsse mit maximalem Wert geben. Für ganzzahliges c_o ist mindestens einer von ihnen ganzzahlig.

Satz 1.11 legt folgenden Algorithmus nahe, bei dem für einen vorliegenden Fluß entweder gezeigt wird, daß er maximal ist, oder ein zunehmender Weg konstruiert wird, mit dessen Hilfe ein größerer Fluß gefunden werden kann. Dazu werden die Ecken sukzessive, beginnend mit s, mit einer dreistelligen Markierung versehen: Die erste Markierung $m_1(w)$ gibt die Vorgängerecke auf einem möglichen zunehmenden ungerichteten Weg an, die zweite Markierung $m_2(w) \in \{+, -\}$, ob die Kante $(w, m_1(w))$ eine Vorwärts- oder Rückwärtskante ist, und die dritte Markierung $d(w)$, um welchen Anteil der Fluß entlang dieser Kante verändert werden kann, um den Gesamtfluß zu verbessern. Weiter werden in einer „LISTE“ die Ecken aufgeführt, die selbst schon markiert wurden, aber nicht schon zur Markierung weiterer Ecken benutzt wurden.

Es ergibt sich folgende Grobstruktur:

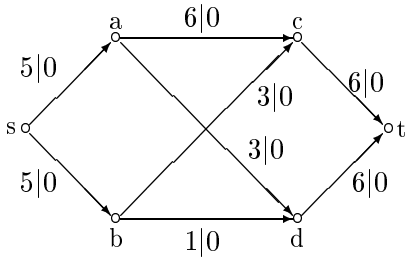
- Initialisierung mit $f \equiv 0$. Markiere s durch $m_1(s) = s$, $m_2(s) = -$, $d(s) = \infty$, LISTE= $\{s\}$.
- Lösche die Markierungen aller Ecken aus $V \setminus \{s\}$.
  Markiere alle möglichen Ecken, ausgehend von s. Lösche dabei jeweils die Ecke aus „LISTE“, von der aus markiert wird, und erweitere „LISTE“ um die Ecken, die neu markiert wurden.
  Kann keine weitere Ecke mehr markiert werden und ist t nicht markiert, dann gibt es keinen zunehmenden Weg von s nach t, d.h. der Fluß ist maximal, und der Algorithmus bricht ab.
- Rekonstruiere den zunehmenden Weg.
- Erhöhe den Fluß entlang des zunehmenden Weges und fahre mit Schritt (2) fort.

Algorithmus 1.6 (Ford–Fulkerson)
Eingabe: Ein Netzwerk $N = (s, t, V, E, c_o)$
Ausgabe: Ein maximaler Fluß f in N

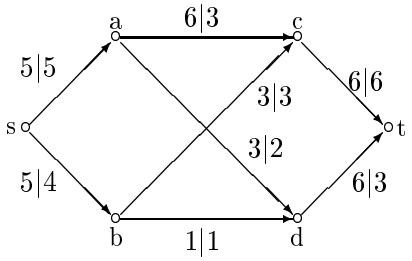
```
begin
  f := 0; (* Fluß initialisieren *)
  m1(s) := s;   m2(s) := −;   d(s) := ∞;   LISTE= {s}; (* Quelle initialisieren *)
  for v ∈ V \ {s} do
  begin
    (* Suche für zunehmenden Weg initialisieren *)
    m1(v) := 0;   m2(v) := 0;   d(v) := ∞;
  end;
  repeat
    (* Markierung der Ecken *)
    begin
      for v ∈ LISTE do
      begin
        (* Ecke markiert, aber noch nicht zur Markierung weiterer Ecken verwendet *)
        LISTE = LISTE \{v};
        for w ∈ V mit m1(w) = 0 do
        (* Suche nach neu zu markierenden Ecken *)
        begin
          if (v, w) ∈ E und f(v, w) < c0(v, w) then
          (* Vorwärtskante *)
          begin
            d(w) := min{co(v, w) − f(v, w), d(v)};   m1(w) := v;   m2(w) := +;
            LISTE = LISTE ∪{w};
          end;
          else if (w, v) ∈ E und f(v, w) > 0 then
          (* Rückwärtskante *)
          begin
```

```
                d(w) := min{f(w,v), d(v)};    m1(w) := v;    m2(w) := −;
                LISTE = LISTE ∪{w};
              end;
            end;
          if m1(t) ≠ 0 then
          (* Rekonstruktion des zunehmenden Weges und Vergrößerung des Flusses *)
          begin
            w := t;    d := d(t);
            while w ≠ s do
            begin
              v := m1(w);
              if m2(w) = + then f(v,w) := f(v,w) + d else f(w,v) := f(w,v) − d;
              w := v;
            end;
            for v ∈ V \ {s} do
            begin
              (* Suche für zunehmenden Weg initialisieren *)
              m1(v) := 0;    m2(v) := 0;    d(v) := ∞;    LISTE = {s};
            end;
          end;
        end;
    until LISTE = ∅;
end;
```

Beispiel 1.19 Für das Netzwerk



ergibt sich als maximaler Fluß

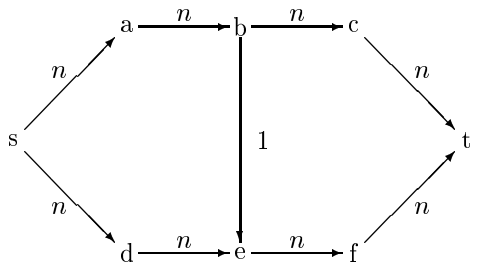


Aus dem Beweis für Satz 1.11 folgt

Satz 1.14 *Für ein Netzwerk mit rationalen Kapazitäten bricht der Ford–Fulkerson–Algorithmus mit einem zugehörigen optimalen Fluß ab.*

Der Algorithmus von Ford–Fulkerson hat in seiner ursprünglichen Version Nachteile:
Im Falle irrationaler Kapazitäten kann er bei ungeschickter Auswahl der Knoten aus *LISTE* versagen. Es gibt ein Beispiel (ohne Beweis), daß das Verfahren nicht nur nicht abbricht, sondern gegen einen Wert konvergiert, der nur 1/4 des maximal möglichen Flußwertes beträgt.
Selbst im Fall ganzzahliger Kapazitäten ist der Algorithmus nicht polynomial, da die Anzahl der vorzunehmenden Änderungen des Flusses nicht nur von $|V|$ und $|E|$, sondern auch von c_o abhängen kann:

Beispiel 1.20 Wählt man in dem Netzwerk



abwechselnd als zunehmende Wege (s, a, b, e, f, t) und (s, d, e, b, c, t), (was bei geeigneter Anordnung der Ecken bei der Auswahl aus *LISTE* möglich ist,) so wird der Flußwert in jedem Schritt nur um 1 erhöht und man benötigt insgesamt $2n$ Iterationen. Andererseits erhält man bei geschickter Wahl der Wege, z.B. mit (s, a, b, c, t) und (s, d, e, f, t), den maximalen Fluß in nur zwei Durchläufen.

Betrachtet man aber *LISTE* nicht als Menge, sondern als Warteschlange, d.h. es wird die Ecke v ausgewählt, die als erste aufgeführt ist, und die neuen Ecken werden am Schluß angefügt, dann wird immer ein Vergrößerungsweg minimaler Länge ausgewählt und der Algorithmus ist auch für irrationale Kapazitäten polynomial:

Satz 1.15 (Edmonds, Karp) *Betrachtet man* LISTE *im Algorithmus von Ford–Fulkerson als Warteschlange, dann hat der Algorithmus 1.6 Komplexität* $O(|V||E|^2)$.

Bemerkung: Für Netzwerke mit n Ecken und m Kanten, die dicht sind, d.h. für die $m = O(n^2)$ gilt, hat der Algorithmus Komplexität $O(n^5)$. Ein von Dinic eingeführter und weiterentwickelter Algorithmus betrachtet geeignete Hilfsnetzwerke und vergrößert die Flüsse nicht nur auf einzelnen Wegen, sondern auf dem gesamten Netzwerk, und hat Komplexität $O(n^3)$, ist also für dichte Netzwerke effizienter.

1.5 Kostenminimale Flüsse

Wir betrachten nun allgemeinere Netzwerke $N = (s, t, V, E, c_u, c_o)$ mit zusätzlicher unterer Kapazitätsfunktion $c_u \leq c_o$ und einer weiteren „Kostenfunktion“ $\gamma : E \to \mathbb{R}$ und suchen optimale Flüsse.

Definition 1.14 *Die Aufgabe, zu jedem Netzwerk $N = (s, t, V, E, c_u, c_o)$ mit $c_u \leq c_o$ und einer* Kostenfunktion $\gamma : E \to \mathbb{R}$ *einen Fluß f mit Wert w und minimalen Kosten $\gamma(f) := \sum_{e \in E} \gamma(e) f(e)$ (d.h. mit $\gamma(f) \leq \gamma(f')$ für alle anderen Flüsse mit Wert w,) zu finden, heißt* Problem der Bestimmung eines kostenminimalen Flusses (min cost flow problem).

Als LP-Problem formuliert, ergibt sich mit der Inzidenzmatrix A und der üblichen Funktion $d \in \mathbb{R}^n$

$$\min \gamma^\top f, \quad Af + wd = 0, \quad c_u \leq f \leq c_o, \qquad \text{mit} \quad d_i = \begin{cases} -1, & i = s \\ +1, & i = t \\ 0, & \text{sonst} \end{cases}.$$

Quelle und Senke hatten bei einem Fluß eine Sonderrolle, da für diese Ecken die Durchlaufbedingung i.a. nicht erfüllt war. Wir verallgemeinern:

Definition 1.15 *Sei $G = (V, E)$ ein gerichteter Graph mit Kapazitätsfunktionen c_u und c_o, $c_u \leq c_o$. Eine Abbildung $f : E \to \mathbb{R}$ heißt* Zirkulation, *wenn sie der Erhaltungsbedingung*

$$\sum_{(v,u) \in E} f(v, u) = \sum_{(u,v) \in E} f(u, v) \qquad \textit{für alle } v \in V$$

genügt. Die Zirkulation heißt zulässig, *wenn gilt*

$$c_u(e) \leq f(e) \leq c_o(e) \qquad \textit{für alle } e \in E.$$

Ist durch $\gamma : E \to \mathbb{R}$ eine Kostenfunktion gegeben, dann heißt die Zirkulation optimal *oder* kostenminimal, *wenn $\gamma(f) \leq \gamma(f')$ für alle zulässigen Zirkulationen f' gilt.*

Beispiele:

(1) Wir betrachten das Problem des maximalen Flusses und ein Netzwerk $N = (s, t, V, E, c_o)$ mit einem Fluß vom Wert $w(f)$. Wir erweitern G durch die gerichtete Kante (t, s) („Rückkehrkante“) zum gerichteten Graphen $G' = (V, E')$ und setzen c_o und f durch

$$c_u := 0 \quad \text{in } G', \qquad c_o(t, s) := \infty, \qquad f(t, s) := w(f)$$

auf G' fort. Mit der Kostenfunktion

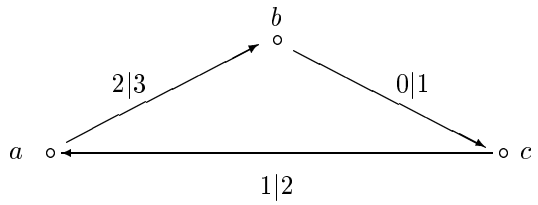
$$\gamma : E' \to \mathbb{R} \qquad \text{mit } \gamma(t, s) := -1, \quad \gamma(e) := 0 \text{ sonst}$$

ist f genau dann eine kostenminimale zulässige Zirkulation auf G', wenn der ursprüngliche Fluß auf G maximal ist.

(2) Für Netzwerke mit unterer Kapazitätsfunktion setzt wir c_u durch $c_u(t, s) := 0$ auf G' fort.

(3) Beim Problem der Bestimmung eines kostenminimalen Flusses mit Wert w setzt man $c_u(t,s) := c_o(t,s) := w$ und $\gamma(t,s) := 0$. Eine zulässige Zirkulation entspricht dann einem Fluß mit Wert w und eine optimale Zirkulation ist Fortsetzung eines kostenminimalen Flusses.

(4) Es muß nicht unbedingt eine zulässige Zirkulation existieren, wie folgendes Beispiel zeigt. Bei jeder Kante sind durch $c_u|c_o$ die Kapazitäten angegeben.



Für ein Existenzkriterium für zulässige Zirkulationen erweitern wir den zugrundeliegenden Graphen zu einem Netzwerk:

Satz 1.16 *Sei $G = (V,E)$ ein gerichteter Graph mit nichtnegativen Kapazitätsfunktionen c_u und c_o, $c_u \le c_o$. Sei $N := (s,t,V',E',c')$ folgendes Netzwerk:*

$$V' := V \cup \{s,t\}, \qquad E' := E \cup \{(s,v); v \in V\} \cup \{(v,t); v \in V\},$$
$$c'(e) := c_o(e) - c_u(e) \quad \textit{für alle } e \in E,$$
$$c'(s,v) := \sum_{(u,v)\in E} c_u(u,v) \qquad \textit{und} \qquad c'(v,t) := \sum_{(v,u)\in E} c_u(v,u) \quad \textit{für alle } v \in V.$$

Weiter sei $w := \sum_{e\in E} c_u(e)$. Dann gilt: Es gibt auf G eine zulässige Zirkulation genau dann, wenn ein maximaler Fluß auf N den Wert w hat.

Wir verallgemeinern die Begriffe „Schnitt“ und „Kapazität eines Schnittes“ auf beliebige gerichtete Graphen:

Definition 1.16 *Sei $G = (V,E)$ ein gerichteter Graph mit nichtnegativen Kapazitäten c_u und c_o, $c_u \le c_o$. Eine Partition (S,T) von V heißt* Schnitt *und*

$$c(S,T) := \sum_{\substack{(u,v)\in E \\ u\in S, v\in T}} c_o(u,v) - \sum_{\substack{(u,v)\in E \\ u\in T, v\in S}} c_u(u,v)$$

Kapazität des Schnittes.

Analog zu Satz 1.13 ergibt sich

Satz 1.17 *Sei $G = (V, E)$ ein gerichteter Graph mit nichtnegativen Kapazitäten c_u und c_o, $c_u \leq c_o$. Es gibt eine zulässige Zirkulation auf G genau dann, wenn jeder Schnitt (S, T) von G nichtnegative Kapazität hat, d.h. wenn stets*

$$\sum_{\substack{(u,v)\in E \\ u\in S, v\in T}} c_o(u,v) \geq \sum_{\substack{(u,v)\in E \\ u\in T, v\in S}} c_u(u,v).$$

Zur Konstruktion einer kostenminimalen zulässigen Zirkulation benötigen wir folgende Hilfskonstruktion.

Definition 1.17 *Sei $G = (V, E)$ ein gewichteter Graph, c_u und c_o mit $c_u \leq c_o$ Kapazitätsfunktionen, γ eine Kostenfunktion und f eine zulässige Zirkulation f in G.*
Eine Kante $(u, v) \in E$ mit $f(u, v) < c_o(u, v)$ heißt Vorwärtskante*, eine Kante $(u, v) \in E$ mit $f(u, v) > c_u(u, v)$ heißt* Rückwärtskante*. Dann heißt $G'(f) = (V, E')$ mit*

$$E' := \{(u,v); (u,v)\, Vorwärtskante\} \cup \{(v,u); (u,v)\ Rückwärtskante\}$$

heißt Vergrößerungsgraph*. Kostenfunktion und Kapazitätsfunktionen werden definiert durch*

$$\gamma'(u,v) := \begin{cases} \gamma(u,v) \\ -\gamma(v,u) \end{cases}, \quad c'_u \equiv 0, \quad c'_o(u,v) := \begin{cases} c_o(u,v) - f(u,v) & (u,v)\ Vorwärtskante \\ f(v,u) - c_u(v,u) & (u,v)\ Rückwärtskante \end{cases}.$$

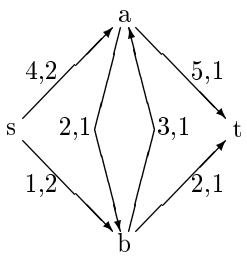
Damit ergibt sich

Satz 1.18 *Eine zulässige Zirkulation f in einem Graphen G ist genau dann kostenoptimal, wenn in dem zugehörigen Vergrößerungsgraphen $G'(f)$ keine Kreise mit negativen Kosten existieren.*

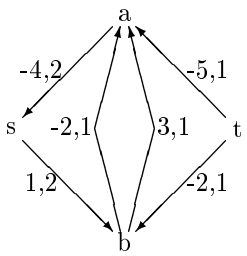
Kreise negativer Kosten lassen sich mit dem Floyd–Warshall–Algorithmus bestimmen, so daß sich folgender Algorithmus für kostenminimale Flüsse ergibt:

Algorithmus 1.7
begin
  Finde einen Fluß f mit Wert w mit dem Maxflußalgorithmus
  while Es gibt einen Kreis C negativer Kosten in (V, E') **do**
  begin
    Vergrößere f längs C
  end;
end.

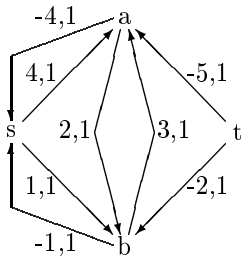
Beispiel 1.21 Gesucht ist eine kostenminimale zulässige Zirkulation in folgendem Netzwerk, in dem für jede Kante e die Kosten $\gamma(e)$, die obere Kapazität $c_0(e)$ und die Zirkulation $f(e)$ durch $\gamma(e), c_o(e), f(e)$ angegeben sind (mit $c_u = 0$.)



Eine zulässige Zirkulation f ist gegeben durch $f(s,a) = 2$, $f(a,b) = f(a,t) = f(b,t) = 1$, $f(b,a) = f(s,b) = 0$.
Das zugehörige Vergrößerungsnetzwerk N' (mit $\gamma'(e), c'_o(e)$ an jeder Kante) ist



Dieses Netzwerk enthält den Kreis (s,b,a,s) mit Kosten -5 und einer zugehörigen Zirkulation $\bar{f}$ vom Wert 1. Addition dieser Zirkulation ergibt eine neue Zirkulation $f(s,a) = f(a,t) = f(s,b) = f(b,t) = 1, f(a,b) = f(b,a) = 0$.
Das zugehörige Vergrößerungsnetzwerk N' ist jetzt



Dieses Netzwerk hat keine Kreise negativer Kosten, so daß die optimale Lösung gefunden ist.

1.6 Branch-and-Bound-Verfahren

Die „Branch-and-Bound"-Methode beruht darauf, auf eine „intelligente" Weise alle zulässigen Lösungen eines kombinatorischen Optimierungsproblems aufzulisten und mit Hilfe von bekannten Schranken Lösungen dabei auszuschließen, die als Optimum nicht in Frage kommen können.

Wir wollen dies an Hand einiger bekannter Probleme deutlich machen und betrachten zuerst das ganzzahlige Lineare Optimierungsproblem

$$\min z = c'x = c(x); \quad Ax \leq b; \quad x \geq 0; \quad x \text{ ganzzahlig.}$$

Läßt man die Forderung der Ganzzahligkeit der Lösung außer acht, dann erhält man eine Lösung x^0 des LP, die i.a. nicht ganzzahlig ist. Die Kosten $c(x^0)$ sind sicher auch eine untere Schranke für die Kosten einer optimalen Lösung x^* des ILP. Wenn x^0 schon ganzzahlig ist, hat man eine gesuchte Lösung des ILP. Im anderen Fall gibt es mindestens eine Komponente x_i^0, die nicht ganzzahlig ist. Wir könnten nun unser Ausgangsproblem aufspalten in die beiden Teilprobleme

Problem 1: $\min z = c'x = c(x); \quad Ax \leq b; \quad x \geq 0; \quad x$ ganzzahlig; $\quad x_i \leq [x_i^0]$

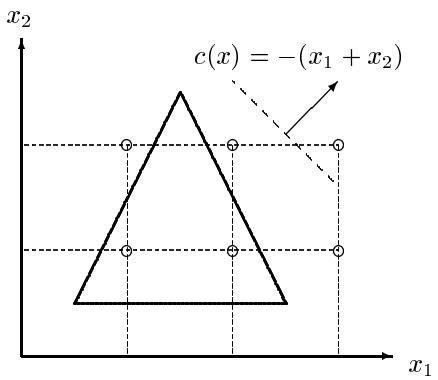
und

Problem 2: $\min z = c'x = c(x); \quad Ax \leq b; \quad x \geq 0; \quad x$ ganzzahlig; $\quad x_i \geq [x_i^0] + 1.$

Beispiel 1.22 Wir wollen die Vorgehensweise an dem Beispiel mit

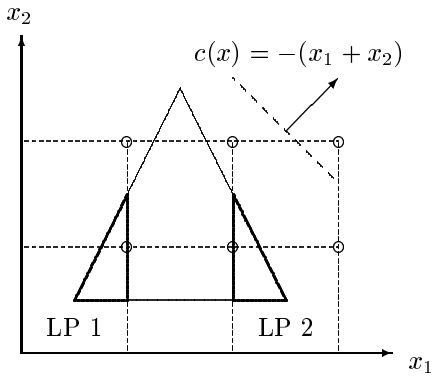
$$c(x) := -(x_1 + x_2); \quad A = \begin{pmatrix} -2 & 1 \\ 0 & -1 \\ 2 & 1 \end{pmatrix}; \quad b = \begin{pmatrix} -\frac{1}{2} \\ -\frac{1}{2} \\ \frac{11}{2} \end{pmatrix}$$

verdeutlichen.



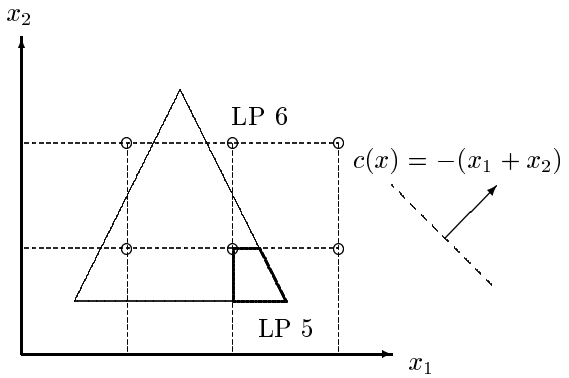
Das zugehörige LP-Problem hat die optimale Lösung $x^0 = (\frac{3}{2}, \frac{5}{2})$ mit Kosten $c(x^0) = -4$. Unterteilt man nach der 1. Komponente, dann erhält man die beiden Teilprobleme

$$LP1: \; x_1 \leq 1 \qquad \text{und} \qquad LP2: \; x_1 \geq 2.$$

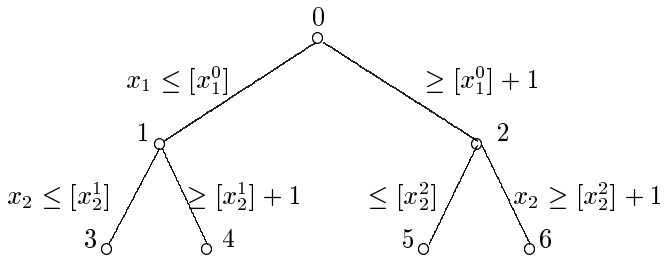


Das Problem LP2 z.B. hat die optimale Lösung $x^2 = (2, \frac{3}{2})$ mit Kosten $c(x^2) = -\frac{7}{2}$. Unterteilt man wieder (nach x_2^2), dann erhält man die beiden Teilprobleme

$$LP5 : x_2 \leq 1 \qquad \text{und} \qquad LP6 : x_2 \geq 2.$$



LP5 hat die ganzzahlige optimale Lösung $(2, 1)$ mit Kosten -3, die auch optimale Lösung für das Ausgangs-ILP ist.
Man kann diesen Auswahlprozeß mit Hilfe eines Baums beschreiben. Die Wurzel entspricht der ursprünglichen zulässigen Menge und jeder Knoten steht für ein Teilproblem.



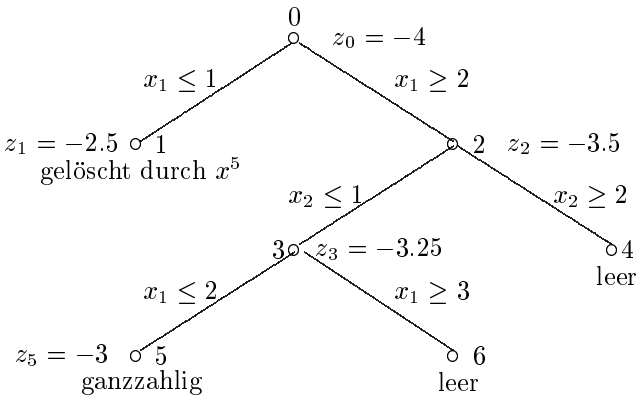
Wenn das ursprüngliche ILP eine beschränkte zulässige Lösungsmenge besitzt, muß der Verzweigungsprozess nach endlich vielen Schritten entweder mit einer ganzzahligen Lösung des entsprechenden Teilproblems oder mit einem LP ohne zulässige Lösungen enden. Die ganzzahlige Lösung mit den geringsten Kosten ist die optimale Lösung des ILP.

Ist irgendwann eine ganzzahlige Lösung z mit Kosten $c(z)$ gefunden und betrachtet man ein Teilproblem (einen Knoten des Baums), für deren Lösungen die untere Schranke der Kosten

schon größer als $c(z)$ ist, dann kommen ganzzahlige Lösungen dieses Teilproblems als optimale Lösungen nicht in Betracht. Daher muß man die Verzweigungen dieses Knotens nicht weiter untersuchen.

Für einen Algorithmus muß man nun festlegen, welchen Knoten man bei einer Verzweigung weiterverfolgt, und welche nichtganzzahlige Komponente benutzt wird, um die zusätzlichen Bedingungen einzuführen. Für beide Fragestellungen gibt es keine allgemein bestmögliche Wahl. Es erscheint sinnvoll, den Knoten weiter zu verfolgen, der die kleinste untere Schranke hat. Betreffend der Wahl der Komponenten wird die Komponente empfohlen, deren zugehörigen Zusatzbedingungen die unteren Schranken möglichst weit nach oben drückt, in der Hoffnung, im weiteren Pfad möglichst viele Knoten nicht berücksichtigen zu müssen, da die untere Schranke schon zu groß ist.

Beispiel 1.22 (Forts.) Der vollständige Baum zu unserem Beispiel ist



Zusammenfassend gilt:
Gegeben sei ein Problem A mit zulässiger Menge G. Für die Durchführung des Branch-and-Bound-Verfahrens benötigt man nur zwei allgemeine Voraussetzungen:

1. Verzweigen (Branch): Für jedes $F \subset G$ ist $F = \bigcup F_i$ darstellbar mit disjunkten F_i. Betrachtet man diese F_i als Knoten eines Graphen, so erhält man einen (gerichteten) Baum.

2. Beschränken nach unten (Bound): Es gibt ein einfaches Problem B, so daß $\min B(F) \leq \min A(F)$ für jedes $F \subset G$ gilt. (B liefert untere Schranken für die Knoten des Baums).

Bezeichnet man wie üblich die Nachfolger eines Knotens in einem gerichteten Baum als Kinder, dann hat der Algorithmus die folgende allgemeine Form: „Aktivmenge“ ist die Menge der Knoten, die noch genauer untersucht werden muß, O ist eine obere Schranke für die Lösung, „Bestlösung“ ist die beste gefundene Lösung.

Algorithmus 1.8 (Branch-and-Bound)
begin
Aktivmenge := $\{G\}$; (* G ist das ursprüngliche Problem *)

```
O := ∞;
while Aktivmenge ≠ ∅ do
begin
   wähle einen Verzweigungsknoten k ∈ Aktivmenge;
   Aktivmenge := Aktivmenge\{k};
   Erzeuge die Kinder von Knoten k, Kind[i], i = 1, ... , n_k;
   Erzeuge die unteren Schranken z_i = min B(i), i = 1, ... , n_k;
   for i = 1 to n_k do
   begin
      if z_i < O then
      begin
         if Kind[i] ist vollständige Lösung then
         begin
            if A(i) < O then
            begin
               O := A(i);
               Bestlösung := Kind[i];
            end;
         end else Aktivmenge := Aktivmenge ∪ {Kind[i]};
      end else lösche Kind[i];
   end;
end;
end.
```

Beispiel: Als realistisches Beispiel für die Anwendung des Branch-and-Bound-Verfahrens betrachten wir das TSP, für das kein Algorithmus mit polynomialer Komplexität bekannt ist, speziell das Problem der Bestimmung einer optimalen Rundreise für die folgenden 5 mit ihren Abständen gegebenen Städten

	Ac	Bi	Dr	Fr	Ha
Ac	∞	257	659	505	488
Bi	257	∞	511	618	262
Dr	659	511	∞	728	502
Fr	505	618	728	∞	759
Ha	488	262	502	759	∞

Die Diagonaleinträge sind gleich ∞ gesetzt - dies kann man als Verbieten von Schleifen interpretieren. Eine Tour entspricht nun einer Permutation der 5 Städte, d.h. einer (verallgemeinerten) Diagonalen der Abstandsmatrix $W = (d_{ij})$. Zieht man nun von allen Elementen einer festen Zeile eine positive Zahl δ ab, so daß die entstehende neue Matrix W' nichtnegative Elemente hat, dann enthält jede Tour genau ein Element der Zeile. Ihre Länge, bezogen auf die neue Matrix W' ist also um δ kürzer als die Länge bezüglich W. Dasselbe gilt für die Spalten. Reduziert man nun jede Zeile (um δ_i) und jede Spalte (um ϵ_j) so weit, daß in der schließlich gewonnenen Matrix W^* mindestens ein Element 0 ist, dann verändert sich die Länge jeder Tour um $s = \sum_{i=1}^{n} \delta_i + \sum_{j=1}^{n} \epsilon_j$. Natürlich hat jede Tour bezüglich der Matrix W^* nichtnegative Länge, d.h. s ist untere Schranke der Längen aller Touren des Ausgangsproblems.

Zeilenoperationen ergeben:

	Ac	Bi	Dr	Fr	Ha
Ac	∞	0	402	248	231
Bi	0	∞	254	361	5
Dr	157	9	∞	226	0
Fr	0	113	223	∞	254
Ha	226	0	240	497	∞

mit

δ_i
257
257
502
505
262

.

Schranke: $\sum_{i=1}^{5} \delta_i = 1783$.

Spaltenoperationen ergeben:

	Ac	Bi	Dr	Fr	Ha
Ac	∞	0	179	22	231
Bi	0	∞	31	135	5
Dr	157	9	∞	0	0
Fr	0	113	0	∞	254
Ha	226	0	17	271	∞

mit

ϵ_j	0	0	223	226	0

Schranke: $s = \sum_{i=1}^{5} \delta_i + \sum_{j=1}^{5} \epsilon_j = 1783 + 449 = 2232$.

Die Menge der möglichen Touren läßt sich nun aufspalten danach, ob eine feste Kante (v_i, v_j) enthalten ist oder nicht. Ist die Kante nicht enthalten, dann kann man den entsprechenden Eintrag in der Matrix durch ∞ ersetzen und möglicherweise für diesen Fall s erhöhen. Ist die Kante enthalten, dann kann keine Kante der entsprechenden Zeile bzw. Spalte in der Tour auftreten, man kann also diese Zeile und Spalte streichen und mit der reduzierten Matrix fortfahren. Außerdem kann die entgegengesetzt gerichtete Kante nicht in der Tour auftreten, d.h. man kann den entsprechenden Eintrag ∞ setzen.
Es ist nun sinnvoll, eine solche Kante als Verzweigungskriterium zu wählen, für die im ersten Fall (Tour enthält diese Kante nicht) man s um einen möglichst großen Wert erhöhen kann, d.h. mit

$$d_{ij} = 0 \qquad \text{und maximalem} \quad \min\{d_{ik}, k \neq j\} + \min\{d_{lj}, l \neq i\}.$$

Zum Beispiel wählt man zu Beginn die Kante (Ac, Bi) aus und erhält

im Fall (Ac, Bi)

	Ac	Dr	Fr	Ha
Bi	∞	31	135	5
Dr	157	∞	0	0
Fr	0	0	∞	254
Ha	226	17	271	∞

bzw.

s_1	Ac	Dr	Fr	Ha
Bi	∞	26	130	0
Dr	157	∞	0	0
Fr	0	0	∞	254
Ha	209	0	254	∞

mit $s_1 = 2254$

und im Fall $\neg(Ac, Bi)$

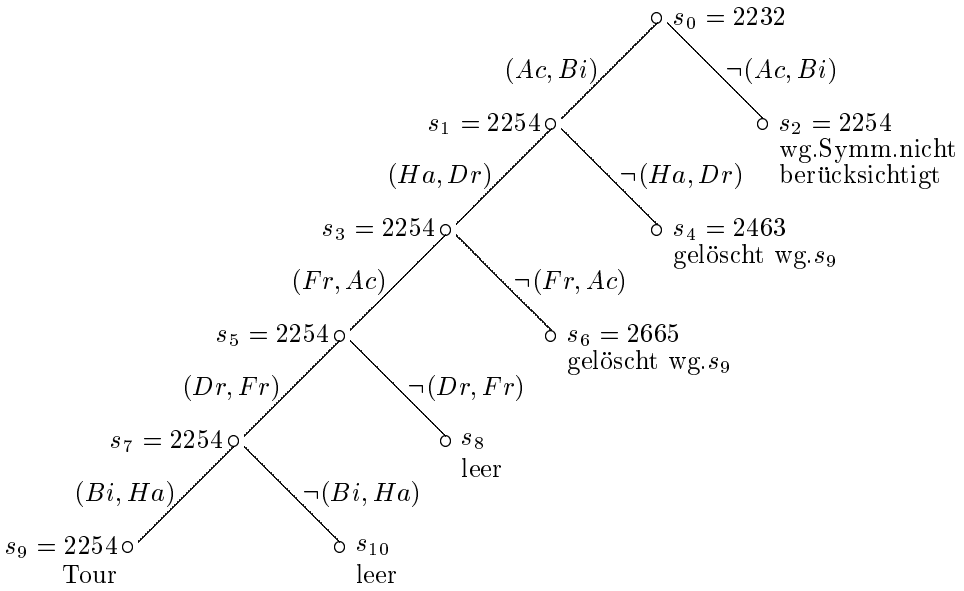
	Ac	Bi	Dr	Fr	Ha
Ac	∞	∞	179	22	231
Bi	0	∞	31	135	5
Dr	157	9	∞	0	0
Fr	0	113	0	∞	254
Ha	226	0	17	271	∞

bzw.

s_2	Ac	Bi	Dr	Fr	Ha
Ac	∞	∞	157	0	209
Bi	0	∞	31	135	5
Dr	157	9	∞	0	0
Fr	0	113	0	∞	254
Ha	226	0	17	271	∞

mit $s_2 = 2254$.

In beiden Fällen ergibt sich dieselbe Schranke, denn aus der Symmetrie des Ausgangs-TSP folgt, daß jeder Tour mit Kante (Ac, Bi) eine Tour mit $\neg(Ac, Bi)$ entspricht und umgekehrt (nämlich die Tour mit (Bi, Ac)). Man muß daher nur einen der Fälle weiterverfolgen. Als Verzweigungsbaum ergibt sich



mit den Lösungen (Ac, Bi, Ha, Dr, Fr, Ac) bzw. der Umkehrtour und der minimalen Länge 2254.

1.7 Dynamische Programmierung

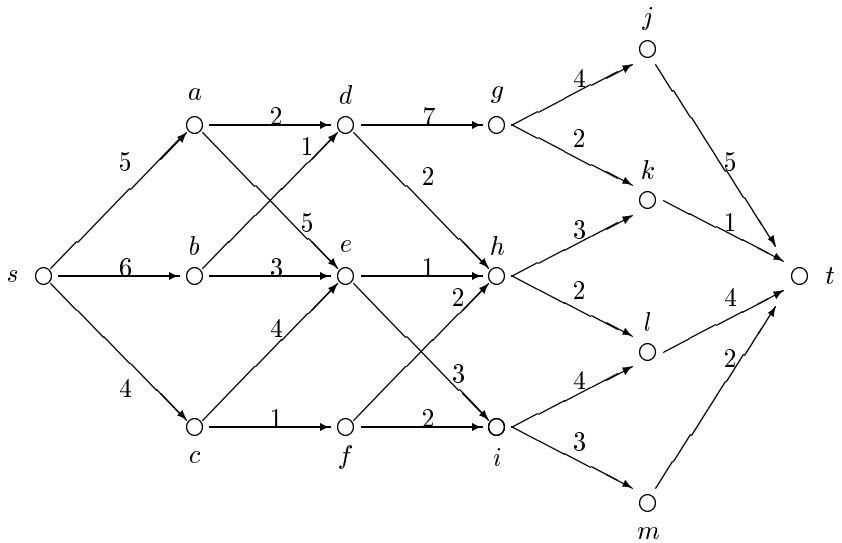
Wie beim Branch-and-Bound-Verfahren beruht das Verfahren der dynamischen Programmierung auf einer intelligenten Aufzählung aller zulässigen Punkte. Es werden aber nicht Lösungen mit Hilfe von Schranken ausgeschlossen, sondern man arbeitet sich von der letzten Entscheidung rückwärts (analog zum Beweis des Algorithmus von Dijkstra):
Muß man zur Gewinnung der optimalen Lösung eine Folge $D_1, \ldots, D_n$ von Entscheidungen treffen, dann braucht man bei der Entscheidung D_k nur die Folgen $D_1, \ldots, D_{k-1}$ zu betrachten, die zu optimalen zulässigen Lösungen vor dieser Entscheidung führten.

Das Prinzip werde an folgender Anwendungen demonstriert:

Kann man die Eckenmenge eines Netzwerks $N = (s, t, V, E)$ in disjunkte Mengen $V_0 = \{s\}$, $V_1, \ldots, V_k$, $V_{k+1} = \{t\}$ zerlegen, so daß in E nur Kanten der Form (u, v), $u \in V_i$, $v \in V_{i+1}$,

$0 \leq i \leq k$, auftreten, dann heißt N *geschichtetes* Netzwerk. Gesucht ist ein kürzester Weg in einem solchen geschichteten Netzwerk von s nach t.

Beispiel 1.23



$V_0 = \{s\}$, $V_1 = \{a, b, c\}$, $V_2 = \{d, e, f\}$, $V_3 = \{g, h, i\}$, $V_4 = \{j, k, l, m\}$, $V_5 = \{t\}$.

Beginnend bei V_k wird in einer Tabelle festgehalten, wie ein kürzester Weg von jedem $v \in V_i$ aus nach t verläuft. Dazu wird für jedes v der Nachfolger aus V_{i+1} auf einem kürzesten Weg und die zugehörige Länge festgehalten. Aus der Tabelle für V_k kann man nun *rückwärts* die Tabelle für V_{k-1} bestimmen usw. bis man bei V_0 angekommen ist. Nun kann man anhand der Tabellen *vorwärts* den kürzesten Weg rekonstruieren.

Beispiel 1.23 (Forts.) Jeder Menge V_i wird eine Tabelle zugeordnet, die zu jeder Ecke der Menge den nächsten Nachfolger und die entsprechende Länge angibt:

$$\text{Tabelle}(4) = \left\{ \begin{array}{llllll} \text{Knoten} & j & k & l & m \\ \text{Nachfolger} & t & t & t & t \\ \text{Länge} & 5 & 1 & 4 & 2 \end{array} \right. \quad \Rightarrow \quad \text{Tabelle}(3) = \left\{ \begin{array}{llll} \text{Knoten} & g & h & i \\ \text{Nachfolger} & k & k & m \\ \text{Länge} & 3 & 4 & 5 \end{array} \right.$$

$$\Rightarrow \quad \text{Tabelle}(2) = \left\{ \begin{array}{llll} \text{Knoten} & d & e & f \\ \text{Nachfolger} & h & h & h \\ \text{Länge} & 6 & 5 & 6 \end{array} \right. \quad \Rightarrow \quad \text{Tabelle}(1) = \left\{ \begin{array}{llll} \text{Knoten} & a & b & c \\ \text{Nachfolger} & d & d & f \\ \text{Länge} & 8 & 7 & 7 \end{array} \right.$$

$$\Rightarrow \quad \text{Tabelle}(0) = \left\{ \begin{array}{ll} \text{Knoten} & s \\ \text{Nachfolger} & c \\ \text{Länge} & 11 \end{array} \right.$$

Der kürzeste Weg ist dann gegeben durch (s, c, f, h, k, t) mit Länge 11.

2 Spieltheorie

2.1 Allgemeines und Historie

Gesellschaftsspiele wie Schach, Mühle, Skat, Knobeln etc. haben gemeinsam, daß mehrere Parteien teilnehmen, die entgegengesetzte oder zumindest konfligierende Ziele haben. Jeder Spieler trifft eine Entscheidung, und das Ergebnis des Spiels ist von der Gesamtheit all dieser Entscheidungen abhängig.
Den gleichen Sachverhalt findet man auch in einigen Wissenschaften und sonstigen Lebensbereichen. Einige Beispiele sind

- Entscheidungen über Preise, Produktmengen;
- Verhandlungssituationen, z. B. über Tarifverträge oder Abrüstung;
- Bietverhalten bei Auktionen;
- strategisches Verhalten in Streiks, Handelskriegen oder militärischen Auseinandersetzungen;
- Wettbewerb und Kooperation verschiedener Regierungen/Zentralbanken in einem gemeinsamen Wirtschaftsraum (Geld–, Steuer–, Außenhandelspolitik);
- evolutionäre Konfrontationen.

Aufgrund der vielen Anwendungsmöglichkeiten entwickelte sich in den 40er und 50er Jahren schnell eine Theorie (vor allem durch von Neumann, Morgenstern und Nash), die sowohl in der Mathematik als auch in den Wirtschaftswissenschaften, den Sozialwissenschaften und der Biologie von Interesse ist. So wurde z.B. in den Jahren 1994 und 1996 der Nobelpreis für Wirtschaftswissenschaften an Spieltheoretiker vergeben.

2.2 Grundlegende Begriffe

Es sollen hier nur *Strategiespiele* betrachtet werden, d. h. solche Spiele, in denen die Teilnehmer, im Gegensatz zu den reinen *Glücksspielen*, zumindest teilweise den Spielverlauf beeinflussen können. Der Zufall, z.B. in Form eines Würfels, kann hierbei als ein teilnehmender Spieler angesehen werden. Die „klassische“ Spieltheorie, die hier behandelt werden soll, setzt voraus, daß alle Spieler vollständig über die Zahl der Spieler, ihre Handlungsmöglichkeiten und Präferenzen informiert sind. Man spricht in diesem Fall von *Spielen mit vollständiger Information*. Wir beschränken uns darüberhinaus auf *nichtkooperative* Spiele. Das sind „Spiele ohne Vertrauen“, in denen Absprachen, glaubwürdige Versprechungen oder Ähnliches nicht möglich sind. Ziel jedes Spielers ist möglichst großer persönlicher Profit, im Gegensatz zu *kooperativen* Spielen in denen der Profit von Spielergruppen (*Koalitionen*) von Interesse ist.
Wir gehen im Folgenden davon aus, daß die Menge $I = \{1, \dots, n\}$ der Spieler endlich ist. Jeder Spieler i hat die Möglichkeit, zwischen (mindestens zwei) verschiedenen *Strategien* $s_i \in S_i$ zu wählen. In einem Spiel (oder einer Runde) wird hieraus eine *Situation* $s = (s_1, \dots, s_n)$ zusammengesetzt. Das Produkt $S = S_1 \times \cdots \times S_n = \prod_{i \in I} S_i$ wird als *Strategienraum* bezeichnet. In jeder Situation $s \in S$ erhält Spieler i eine Auszahlung $H_i(s) \in \mathbb{R}$, $i = 1, \dots, n$. Die Funktion H_i wird als *Auszahlungsfunktion* von Spieler i bezeichnet.

Definition 2.1 Es sei $I = \{1, \ldots, n\}$ und für $i \in I$ seien S_i Mengen und $H_i : S = S_1 \times \cdots \times S_n \to \mathbb{R}$ Funktionen. Dann heißt das System

$$\Gamma = \langle I, \{S_i\}_{i \in I}, \{H_i\}_{i \in I} \rangle$$

ein *nichtkooperatives n-Personen Spiel (in Normalform).*

Beispiel 2.1 *Stein–Schere–Papier*: Zwei Spieler ($I = \{1, 2\}$) geben gleichzeitig ein Zeichen, das Stein, Schere oder Papier ($S_i = \{St, Sch, Pa\}, i = 1, 2$) bedeutet. Die Auszahlungsfunktion H_1 für Spieler 1 sei durch folgende Tabelle (Matrix) gegeben, die für Spieler 2 durch die jeweils negativen Werte. D.h., es gilt $\sum_{i \in I} H_i(s) = 0$ für alle $s \in S$.

1\2	St	Sch	Pa
St	0	+1	-1
Sch	-1	0	+1
Pa	+1	-1	0

Definition 2.2 Ein nichtkooperatives n-Personen Spiel $\Gamma = \langle I, \{S_i\}_{i \in I}, \{H_i\}_{i \in I} \rangle$ heißt *Konstantsummenspiel*, wenn für alle $s \in S$ $\sum_{i \in I} H_i(s) = c \in \mathbb{R}$ gilt. Im Falle $c = 0$ spricht man von einem *Nullsummenspiel*.

Beispiel 2.2 *Gefangenendilemma*: Zwei Gefangene ($I = \{1, 2\}$) werden getrennt voneinander verhört. Sie werden beschuldigt, gemeinsam eine Straftat begangen zu haben. Die Indizien reichen nicht für eine Verurteilung aus. Jeder der beiden hat die Möglichkeit, die Tat zu leugnen oder ein Geständnis abzulegen ($S_i = \{L, G\}, i = 1, 2$). Gestehen beide, so werden sie zu 8 Jahren Haft verurteilt. Leugnen beide, so können sie wegen geringerer Delikte nur zu einem Jahr verurteilt werden. Gesteht einer und einer leugnet, so wird der Geständige freigesprochen und der andere zu 10 Jahren Haft verurteilt. Die Auszahlungsfunktionen H_i, $i = 1, 2$, können hier mit zwei Tabellen (oder einer Bi–Matrix) beschrieben werden:

H_1 :

1\2	L	G
L	-1	-10
G	0	-8

und

H_2 :

1\2	L	G
L	-1	0
G	-10	-8

Bemerkung 2.1 Es gehört zum Begriff eines nichtkooperativen n–Personenspiels (in Normalform), daß es nur einmal gespielt wird, und zwar so, daß jeder Spieler simultan, ohne über die tatsächlichen Entscheidungen der anderen Spieler informiert zu sein, ein Element seiner Strategiemenge auswählt. Erörterungen über Wiederholungen eines Spiels, die durch Beobachtungen des Spiels entstehenden Informationen und andere dynamische Phänomene gehören in die Theorie der *Spiele in Extensivform*. Diese können jedoch mathematisch auf Spiele in Normalform zurückgeführt werden.

Beispiel 2.3 Bei einem *Schachspiel (in Normalform)* ist eine Strategie die Menge der favorisierten Züge eines Spielers für jede mögliche Spielsituation. D.h., die Spieler wissen a-priori wie sie auf alle möglichen Züge ihres Gegners reagieren werden.

2.3 Gemischte Strategien

Häufig werden viele Runden ein und desselben Spiels gespielt. Dabei ändert man in vielen Fällen häufiger seine Strategie, denn sonst kann sich der Gegner leicht darauf einstellen. Es kommt darum nicht auf die Wahl einer einzelnen Strategie, sondern vielmehr auf die Gewichtung aller Strategien mit Wahrscheinlichkeiten an.

Definition 2.3 Eine *Wahrscheinlichkeitsverteilung* (oder *Wahrscheinlichkeitsmaß*) auf einer Menge X ist eine auf der *Potenzmenge* $\mathcal{P}(X)$ definierte Funktion $\sigma : \mathcal{P}(X) \to [0,1]$ mit den folgenden Eigenschaften:

1. $\sigma(X) = 1$
2. $\sigma(\bigcup_i X_i) = \sum_i \sigma(X_i)$ für jede abzählbare Familie $\{X_i\}$ von paarweise fremden Mengen $X_i \in \mathcal{P}(X)$.

Ist σ eine Wahrscheinlichkeitsverteilung auf einer Menge X, so kann man $\sigma(X')$ als Wahrscheinlichkeit auffassen, mit der ein Element $x \in X$ in X' liegt. Wahrscheinlichkeitsverteilungen haben folgende einfache Eigenschaften:

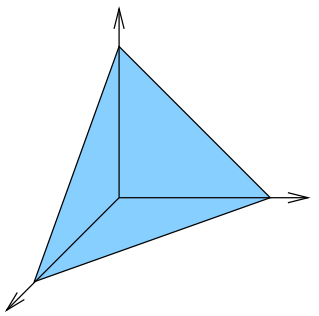
Lemma 2.1 *Für eine Wahrscheinlichkeitsverteilung σ auf einer Menge X gilt*

1. $\sigma(X_1) \leq \sigma(X_2)$ *für alle* $X_1, X_2 \in \mathcal{P}(X)$ *mit* $X_1 \subset X_2$*;*
2. $\sigma(X_1 \cup X_2) + \sigma(X_1 \cap X_2) = \sigma(X_1) + \sigma(X_2)$ *für alle* $X_1, X_2 \in \mathcal{P}(X)$*;*
3. $\sigma(\bigcup_i X_i) \leq \sum_i \sigma(X_i)$ *für jede abzählbare Familie* $\{X_i\}$ *von Mengen* $X_i \in \mathcal{P}(X)$.

Definition 2.4 Ist $\Gamma = \langle I, \{S_i\}_{i\in I}, \{H_i\}_{i\in I}\rangle$ ein nichtkooperatives n-Personen Spiel, so heißt eine Wahrscheinlichkeitsverteilung σ_i auf der Menge der Strategien S_i *gemischte Strategie* des Spielers i. Die *Menge aller gemischten Strategien* von Spieler i wird mit Σ_i bezeichnet.

Beispiel 2.4 Spiele, in denen jeder Spieler i eine endliche Auswahl *reiner Strategien* $S_i = \{s_1, \ldots, s_{m_i}\}$ zur Verfügung hat, heißen *endliche Spiele*. In diesen können die Mengen der gemischten Strategien Σ_i durch $(m_i - 1)$–dimensionale Simplizes

$$S^{(m_i)} = \{(x_1, \ldots, x_{m_i})^\top \in \mathbb{R}^{m_i} \;\Big|\; 0 \leq x_i \leq 1,\ x_1 + \ldots + x_{m_i} = 1\}$$



beschrieben werden. Eine gemischte Strategie $\sigma_i \in \Sigma_i$ wird dabei durch $\sigma_i(s_j) := \sigma_i(\{s_j\}) = x_j$, für $j = 1, \ldots, m_i$, mit einem $x \in S^{(m_i)}$ identifiziert.
Wählen die Spieler die gemischten Strategien $\sigma_1, \ldots, \sigma_i$ so wird das n–Tupel $\sigma = (\sigma_1, \ldots, \sigma_n)$ als *Situation des Spiels in gemischten Strategien* bezeichnet. $\Sigma = \Sigma_1 \times \cdots \times \Sigma_n = \prod_{i\in I} \Sigma_i$ bezeichnet die *Menge aller Situationen in gemischten Strategien*. Man beachte, daß die Menge der Situationen S (in reinen Strategien) in natürlicher Weise in Σ enthalten ist. Denn jede reine Strategie $s_i \in S_i$ wird durch die Wahrscheinlichkeitsverteilung σ_i repräsentiert, die für $S_i' \subset S_i$ definiert ist durch $\sigma_i(S_i') = 1$, falls $s_i \in S_i'$, und $\sigma_i(S_i') = 0$ sonst.

Die Auszahlung an einen Spieler wird bei Wahl gemischter Strategien durch einen *Erwartungswert* ersetzt. Im Falle endlicher Spiele ist die Wahrscheinlichkeit für das Eintreten einer Situation $s = (s_1, \ldots, s_n) \in S$ gegeben durch

$$\sigma(s) := \sigma_1(s_1) \cdot \cdots \cdot \sigma_n(s_n).$$

Der Erwartungswert E_i von Spieler i ist für eine Situation σ in gemischten Strategien

$$E_i(\sigma) := \sum_{s \in S} H_i(s)\sigma(s) = \sum_{s_1 \in S_1} \ldots \sum_{s_n \in S_n} H_i(s_1, \ldots, s_n) \prod_{j=1}^{n} \sigma_j(s_j). \tag{2.1}$$

Definition 2.5 Sei $\Gamma = \langle I, \{S_i\}_{i \in I}, \{H_i\}_{i \in I} \rangle$ ein endliches, nichtkooperatives n–Personenspiel. Dann heißt

$$\Gamma^* = \langle I, \{\Sigma_i\}_{i \in I}, \{E_i\}_{i \in I} \rangle,$$

mit E_i wie in (2.1) die *gemischte Erweiterung* von Γ.

Beispiel 2.5 (Fortsetzung) In Beispiel 2.1 (Stein–Schere–Papier) können gemischte Strategien σ_1, σ_2 der Spieler 1 und 2 mit Punkten x, y eines 2-dimensionalen Simplex im R^3 identifiziert werden. Dabei geben die Komponenten x_i, y_i, $i = 1, 2, 3$, die Wahrscheinlichkeiten an, mit denen Spieler i eine der reinen Strategien St, Sch, Pa wählt. Die Situation $s =$ (St,Pa) wird z.B. mit einer Wahrscheinlichkeit $\sigma(s) = \sigma_1(\text{St}) \cdot \sigma_2(\text{Pa}) = x_1 \cdot y_3$ eintreten. Der Erwartungswert E_1 für Spieler 1 ist für die Situation (x, y) in gemischten Strategien gegeben durch

$$E_1(x, y) = x_1(y_2 - y_3) + x_2(y_3 - y_1) + x_3(y_1 - y_2).$$

Wählen Spieler 1 und Spieler 2 etwa die gemischten Strategien $(\frac{1}{2}, \frac{1}{2}, 0)^\top$ und $(\frac{1}{2}, 0, \frac{1}{2})^\top$, so hat Spieler 1 eine Gewinnerwartung von $-\frac{1}{4}$.

Bemerkung 2.2 In Abschnitt 2.10 werden wir gemischte Erweiterungen für eine weitere Klasse von Spielen einführen. Allgemein kann man gemischte Erweiterungen immer dann definieren, wenn die Integrale

$$E_i(\sigma) = \int_{S_1} \ldots \int_{S_n} H_i(s_1, \ldots, s_n)\, d\sigma_n(s_n) \cdots d\sigma_1(s_n).$$

für alle $i \in I$ und alle $\sigma = (\sigma_1, \ldots, \sigma_n) \in \Sigma$ existieren und ihr Wert unabhängig von der Integrationsreihenfolge ist. Fragen zur Existenz und Berechnung solcher Integrale sind Gegenstand der *Wahrscheinlichkeits,-* und *Maßtheorie.*

2.4 Gleichgewichtspunkte

Das Verhalten der Spieler wird sich im Laufe des Spiels, abhängig vom Verhalten der Gegner, natürlich ändern. Ein Spieler wird seine Strategie immer dann ändern, wenn ihm dies bei gleichbleibendem Verhalten der Gegner einen Vorteil verschafft. Ist dies für keinen Spieler der Fall, so tritt ein Gleichgewicht ein. Ist $s = (s_1, \ldots, s_n) \in S$, so schreiben wir für die Situation $(s_1, \ldots, s_{i-1}, s_i', s_{i+1}, \ldots, s_n)$ im Folgenden kurz $s \| s_i'$.

Definition 2.6 Sei $\Gamma = \langle I, \{S_i\}_{i\in I}, \{H_i\}_{i\in I}\rangle$ ein nichtkooperatives n-Personenspiel. Eine Situation $s = (s_1, \ldots, s_n) \in S$ heißt *zulässig* für Spieler i, wenn gilt

$$H_i(s\|s_i') \leq H_i(s), \quad \text{für alle } s_i' \in S_i.$$

Ist eine Situation zulässig für alle Spieler, so heißt sie *Gleichgewichtspunkt*. Ein *Gleichgewichtspunkt von* Γ *in gemischten Strategien* ist ein Gleichgewichtspunkt von Γ^* (vorausgesetzt Γ^* ist definiert).

Beispiel 2.6 (Fortsetzung) In Beispiel 2.1 (Stein–Schere–Papier) sind die drei Situationen (St,Sch), (Sch,Pa), (Pa,St) zulässig für Spieler 1 und (St,Pa), (Sch,St), (Pa,Sch) zulässig für Spieler 2. Das Spiel besitzt also keinen Gleichgewichtspunkt. Dafür besitzt es aber den eindeutigen Gleichgewichtspunkt $\left((\frac{1}{3},\frac{1}{3},\frac{1}{3})^\top, (\frac{1}{3},\frac{1}{3},\frac{1}{3})^\top\right)$ in gemischten Strategien (Übung).

Gleichgewichtspunkte und die zugehörigen Auszahlungen an einen Spieler sind keineswegs immer eindeutig.

Beispiel 2.7 *Kampf der Geschlechter*: Ein Ehepaar möchte am Abend ausgehen. Es besteht die Möglichkeit entweder eine Ballettaufführung oder einen Boxkampf zu besuchen. Gehen die beiden Ehepartner getrennt zu verschiedenen Veranstaltungen, so ist in jedem Fall der Ärger über den Eigensinn des anderen so groß, daß sie von der Veranstaltung nichts haben (Nutzen -1). Gehen sie zum Boxkampf, so ist der Mann erheblich zufriedener (Nutzen 2) als die Frau (Nutzen 1). Beim Ballett ist es genau umgekehrt. Die Auszahlungsfunktionen H_1 des Mannes und H_2 der Frau lassen sich also wie folgt darstellen:

H_1 :

1\2	Ba	Bo
Ba	1	-1
Bo	-1	2

und

H_2 :

1\2	Ba	Bo
Ba	2	-1
Bo	-1	1

Hier gibt es die zwei Gleichgewichtspunkte (Ba,Ba) und (Bo,Bo), deren Qualität von beiden Ehepartnern unterschiedlich beurteilt wird. Zusätzlich gibt es einen Gleichgewichtspunkt $\left((\frac{2}{5},\frac{3}{5})^\top, (\frac{3}{5},\frac{2}{5})^\top\right)$ in gemischten Strategien mit einer Gewinnerwartung von $\frac{1}{5}$ für beide Ehepartner, also weniger als bei einem der Gleichgewichtspunkte in reinen Strategien.

Gleichgewichtspunkte sind also nicht immer wünschenswerte Situationen für die Spieler. Anderseits sind alle anderen Situationen für mindestens einen Spieler unzulässig, also „unvernünftig“. Eine wesentliche Frage der Spieltheorie ist daher, ob Gleichgewichtspunkte existieren und wie man sie bestimmen kann. Eine Vereinfachung bietet

Definition 2.7 Zwei nichtkooperative n-Personenspiele $\Gamma = \langle I, \{S_i\}_{i\in I}, \{H_i\}_{i\in I}\rangle$ und $\Gamma' = \langle I, \{S_i\}_{i\in I}, \{H_i'\}_{i\in I}\rangle$ heißen *strategisch äquivalent* (in Zeichen: $\Gamma \sim \Gamma'$), wenn es für alle $i \in I$ reelle Zahlen c_i und ein $k > 0$ gibt, so daß für jede Situation s

$$H_i(s) = k \cdot H_i'(s) + c_i \tag{2.2}$$

gilt.

Satz 2.1 *Es gelten folgende Aussagen:*

1. *Die Relation „strategisch äquivalent“ ist eine Äquivalenzrelation auf der Menge der nichtkooperativen n-Personenspiele.*

2. *Strategisch äquivalente, nichtkooperative n-Personenspiele besitzen dieselben Gleichgewichtspunkte.*

3. *Jedes Konstantsummenspiel ist strategisch äquivalent zu einem Nullsummenspiel.*

2.5 Zwei-Personen-Nullsummenspiele

In einem Nullsummenspiel für zwei Spieler genügt es die Auszahlungsfunktion eines Spielers zu kennen, da die andere durch die Beziehung

$$H_1(s) = -H_2(s) \quad \text{für alle Situationen} \quad s = (s_1, s_2) \in S_1 \times S_2 \tag{2.3}$$

bestimmt ist. Ein Zwei-Personen-Nullsummenspiel wird daher durch ein Tripel $\Gamma = \langle S_1, S_2, H \rangle$, mit $H := H_1$, beschrieben. Ein Gleichgewichtspunkt (s_1^*, s_2^*) erfüllt

$$H_1(s_1^*, s_2^*) \geq H_1(s_1, s_2^*) \quad \text{und} \quad H_2(s_1^*, s_2^*) \geq H_2(s_1^*, s_2)$$

für alle $s_1 \in S_1, s_2 \in S_2$. Wegen (2.3) kann die zweite Bedingung durch $H_1(s_1^*, s_2^*) \leq H_1(s_1^*, s_2)$ ersetzt werden. Also ist (s_1^*, s_2^*) genau dann Gleichgewichtspunkt, wenn die *Sattelpunkt-Bedingung*

$$H(s_1, s_2^*) \leq H(s_1^*, s_2^*) \leq H(s_1^*, s_2) \quad \text{für alle} \quad s_1 \in S_1, s_2 \in S_2 \tag{2.4}$$

erfüllt ist.

Definition 2.8 *Sei H eine auf dem Produkt $S_1 \times S_2$ zweier Mengen S_1, S_2 definierte Funktion. Dann heißt ein Punkt (s_1^*, s_2^*)* Sattelpunkt *von H, wenn (2.4) erfüllt ist.*

Im Folgenden soll die Existenz von Sattelpunkten untersucht werden.

Lemma 2.2 *Sei $H(x, y)$ eine reellwertige Funktion auf einer Menge $X \times Y$. Dann gilt*

$$\sup_x \inf_y H(x, y) \leq \inf_y \sup_x H(x, y).$$

Satz 2.2 *Eine reellwertige Funktion $H(x, y)$ auf einer Menge $X \times Y$ besitzt einen Sattelpunkt genau dann, wenn*

$$\max_x \inf_y H(x, y) \quad \textit{und} \quad \min_y \sup_x H(x, y) \tag{2.5}$$

existieren und darüberhinaus

$$\max_x \inf_y H(x, y) = \min_y \sup_x H(x, y)$$

gilt.

Korollar 2.2.1 *Sind $(x^1, y^1), (x^2, y^2) \in X \times Y$ Sattelpunkte einer reellwertigen Funktion $H(x, y)$ auf $X \times Y$, so auch (x^1, y^2) und (x^2, y^1). Außerdem ist der Wert von H für alle Sattelpunkte gleich dem gemeinsamen Wert der Extrema in (2.5).*

Das Korollar zeigt, daß alle Gleichgewichtspunkte in einem Zwei-Personen-Nullsummenspiel den Spielern eine, durch das Spiel bestimmte, Auszahlung garantieren. Dies ist, wie Beispiel 2.7 (Kampf der Geschlechter) zeigt, i.A. nicht der Fall.

Definition 2.9 Sei $\Gamma = \langle X, Y, H \rangle$ ein Zwei-Personen-Nullsummenspiel bei dem H einen Sattelpunkt auf $X \times Y$ besitzt. Dann sind x^* und y^* *optimale Strategien* für Spieler 1 und Spieler 2, wenn

$$\inf_y H(x^*, y) = \max_x \inf_y H(x, y) = v(\Gamma) = \min_y \sup_x H(x, y) = \sup_x H(x, y^*).$$

$v(\Gamma)$ wird als *Wert des Spiels* bezeichnet.

Bemerkung 2.3 Nicht jeder Punkt (x, y) für den $H(x, y) = v(\Gamma)$ gilt, ist Sattelpunkt von H. Darüberhinaus gibt es Funktionen H auf $X \times Y$ die keinen Sattelpunkt besitzen, obwohl

$$\sup_x \inf_y H(x, y) = v = \inf_y \sup_x H(x, y)$$

gilt (Übung). In diesem Fall lassen sich zu jedem $\epsilon > 0$ sogenannte ϵ–optimale Strategien x_ϵ und y_ϵ finden, so daß für alle $x, y \in X \times Y$

$$H(x_\epsilon, y) > v - \epsilon \quad \text{und} \quad H(x, y_\epsilon) < v + \epsilon,$$

gilt.

2.6 Matrixspiele

Definition 2.10 Ein endliches Nullsummenspiel mit zwei Spielern heißt *Matrixspiel*.

Die Bezeichnung geht darauf zurück, daß die Auszahlungsfunktion eines solchen Spiels durch eine Matrix dargestellt werden kann. Gilt $|S_1| = m$ und $|S_2| = n$, so sei a_{ij} die Auszahlung an Spieler 1 bei Eintreffen der Situation (i, j), $i = 1, \ldots, m$ und $j = 1, \ldots, n$. Dann ist das Matrixspiel eindeutig durch die $(m \times n)$–Matrix $A = (a_{ij})$ bestimmt. Eine Situation (i^*, j^*) ist ein Gleichgewichtspunkt (in reinen Strategien), bzw. ein *Sattelpunkt der Matrix*, wenn

$$a_{ij^*} \leq a_{i^*j^*} \leq a_{i^*j}, \quad \text{für } i = 1, \ldots, m,\ j = 1, \ldots, n.$$

Als direkte Konsequenz von Satz 2.2 gilt

Satz 2.3 *Die Matrix A hat genau dann einen Sattelpunkt, wenn*

$$\max_i \min_j a_{ij} = \min_j \max_i a_{ij}.$$

Die linke Seite ist der maximale Mindestgewinn für Spieler 1 und die rechte Seite der minimale Maximalverlust für Spieler 2. Nun stimmen diese beiden Werte nicht immer überein, d.h. nicht jede Matrix besitzt einen Sattelpunkt.

Beispiel 2.8 Die Matrix $\begin{pmatrix} -1 & 1 \\ 1 & -1 \end{pmatrix}$ besitzt keinen Sattelpunkt.

Anders verhält es sich bei den gemischten Erweiterungen von Matrixspielen. Hier können die Spieler aus den Mengen

$$X := \{x = (x_1, \ldots, x_m)^\top \mid 0 \leq x_i \leq 1, \sum_i x_i = 1\}$$

und

$$Y := \{y = (y_1, \ldots, y_n)^\top \mid 0 \leq y_j \leq 1, \sum_j y_j = 1\}$$

auswählen. Die Gewinnerwartung für Spieler 1 in einer gemischten Situation (x, y) ist nach (2.1)

$$E(x,y) = \sum_{i=1}^{m} \sum_{j=1}^{n} x_i y_j a_{ij} = \sum_{i=1}^{m} x_i \sum_{j=1}^{n} y_j a_{ij} = x^\top A y.$$

Wieder will Spieler 1 den Mindestgewinn maximieren und Spieler 2 den Maximalverlust minimieren. Dies führt zu den beiden Werten

$$v_1 := \max_{x \in X} \min_{y \in Y} x^\top A y \quad \text{und} \quad v_2 := \min_{y \in Y} \max_{x \in X} x^\top A y.$$

Wegen Lemma 2.2 gilt immer $v_1 \leq v_2$. Darüberhinaus gilt

Satz 2.4 (Minimax–Theorem) *Für ein Matrixspiel mit $(m \times n)$–Matrix A gilt*

$$\max_{x \in X} \min_{y \in Y} x^\top A y = v(A) = \min_{y \in Y} \max_{x \in X} x^\top A y.$$

$v(A)$ ist der Wert des Matrixspiels.

Korollar 2.4.1 *Die optimalen Strategien $X_0 \subset X$ und $Y_0 \subset Y$ von Spieler 1 und 2, in der gemischten Erweiterung eines Matrixspiels mit Matrix A, sind gegeben durch*

$$X_0 = \bigcap_{j=1,\ldots,n} \left\{x \in X \mid x^\top A^j \geq v(A)\right\} \quad \text{und} \quad Y_0 = \bigcap_{j=1,\ldots,m} \left\{y \in Y \mid A_i y \leq v(A)\right\},$$

wobei A^j die j–te Spalte und A_i die i–te Zeile von A bezeichnet.

Das Korollar zeigt, daß die beiden Mengen X_0 und Y_0 der optimalen Strategien Polytope sind. Ist der Wert des Spiels bekannt kann man die Ecken dieser Polytope, auch *Extremalstrategien* genannt, in endlich vielen Schritten berechnen.

Bemerkung 2.4 Der Wert eines Matrixspiels und ein Gleichgewichtspunkt können mit einem LP bestimmt werden. Das LP für Spieler 1 ist

$$\max\left\{\alpha \;\middle|\; x \geq 0,\ x_1 + \ldots + x_m = 1,\ x^\top A^j \geq \alpha,\ j = 1, \ldots, n\right\}.$$

Durch die Substitution $\alpha = x_{m+1} - x_{m+2}$ mit $x_{m+1}, x_{m+2} \geq 0$ und Einführung der Schlupfvariablen $x_{m+3}, \ldots, x_{m+n+2}$ für die n Ungleichungen bringt man das Problem in Standardform. Das (eventuell unzulässige) Tableau ist dann

0	0	…	0	−1	1	0	…	0
0				1	−1			
⋮		$-A^\top$		⋮	⋮		I_n	
0				1	−1			
1	1	…	1	0	0	0	…	0

Als Basis können z.B. die n Schlupfvariablen und eine (etwa die j–te) Spalte von $-A^\top$ gewählt werden. Dabei ist das resultierende Starttableau unzulässig wenn die i–te Zeile A_i von A nicht ≥ 0 ist. In diesem Fall kann man das strategisch äquivalente Problem mit Matrix $A + qE$ (Übung) betrachten, wobei E eine $(m \times n)$–Matrix ist, deren Einträge alle 1 sind.

Entsprechend dem komplementären Schlupf erhalten wir

Satz 2.5 *Sei v der Wert eines Matrixspiels mit Matrix A.*

(a) Ist $x = (x_1, \ldots, x_m)^\top$ eine optimale Lösung für Spieler 1 und gilt für eine Spalte A^j von A $x^\top A^j > v$, dann gilt für jede optimale Lösung y von Spieler 2 $y_j = 0$.

(b) Ist $y = (y_1, \ldots, y_n)^\top$ eine optimale Lösung für Spieler 2 und gilt für eine Zeile A_i von A $A_i y < v$, dann gilt für jede optimale Lösung x von Spieler 1 $x_i = 0$.

Der Satz zeigt, daß bestimmte Strategien eines Spielers ausgeschlossen werden können.

Definition 2.11 Sei A eine $(m \times n)$–Matrix.

1. Eine Strategie $x \in X$ für Spieler 1 *dominiert* eine Strategie $x' \in X \setminus \{x\}$ (*streng*), wenn gilt

$$x^\top A^j \geq x'^\top A^j \quad (x^\top A^j > x'^\top A^j) \quad \text{für } j = 1, \ldots, n.$$

2. Eine Strategie $y \in Y$ für Spieler 2 *dominiert* eine Strategie $y' \in Y \setminus \{y\}$ (*streng*), wenn gilt

$$A_i y \leq A_i y' \quad (A_i y < A_i y') \quad \text{für } i = 1, \ldots, m.$$

Insbesondere wird eine reine Strategie i von Spieler 1 von einer Strategie $x \in X$ (streng) dominiert, wenn $x^\top A^j \geq a_{ij}$ $(x^\top A^j > a_{ij})$ für alle $j = 1, \ldots, n$, bzw. $x^\top A \geq A_i$ $(x^\top A > A_i)$. In diesem Fall spricht man auch davon, daß die Zeile A_i der Matrix A (streng) dominiert wird. Analoges gilt für eine reine Strategie j von Spieler 2 und die entsprechende Spalte A^j.

Satz 2.6 *Es sei ein Matrixspiel mit Matrix A gegeben, in der eine Zeile A_i dominiert wird. A' sei die Matrix die man aus A nach Streichung der i-ten Zeile erhält. Weiter seien X_0 und Y_0, bzw. X_0' und Y_0', die optimalen Strategien von Spieler 1 und 2 in den Matrixspielen mit Matrix A, bzw. A'. Dann gilt*

1. $v(A') = v(A)$,
2. $Y_0' = Y_0$ *und* $\bar{X}_0' := \{(x_1', \ldots, x_{i-1}', 0, x_i', \ldots, x_{m-1}')^\top \mid x' \in X_0'\} \subset X_0$,
3. $\bar{X}_0' = X_0$ *bei strenger Dominanz.*

Bemerkung 2.5 Eine entsprechende Aussage gilt natürlich auch für dominierte Spalten. Dominierte Zeilen oder Spalten können somit gestrichen werden, und so läßt sich ein Problem schrittweise reduzieren.

Beispiel 2.9 In der folgenden Matrix werden dominierte Zeilen und Spalten schrittweise eliminiert:

$$\begin{pmatrix} 1 & 2 & 0 & 3 \\ 0 & 3 & 4 & 4 \\ 4 & 1 & 3 & 2 \end{pmatrix} \overset{(1)}{\Rightarrow} \begin{pmatrix} 1 & 2 & 0 \\ 0 & 3 & 4 \\ 4 & 1 & 3 \end{pmatrix} \overset{(2)}{\Rightarrow} \begin{pmatrix} 0 & 3 & 4 \\ 4 & 1 & 3 \end{pmatrix} \overset{(3)}{\Rightarrow} \begin{pmatrix} 0 & 3 \\ 4 & 1 \end{pmatrix}$$

(1) Die zweite Spalte dominiert die Vierte.

(2) Die zweite und die dritte Zeile dominieren die Erste (z.B. $\frac{2}{3}A_2 + \frac{1}{3}A_3 > A_1$).

(3) Die zweite Spalte dominiert die Dritte.

Wir wollen nun noch einige einfache Spezialfälle und deren Lösung behandeln.

Definition 2.12 Ein Matrixspiel mit einer Matrix $A = \mathrm{diag}(a_1, \ldots, a_n)$, mit $a_i > 0$ für $i = 1, \ldots, n$, heißt *Diagonalspiel.*

Satz 2.7 *In einem Diagonalspiel mit Matrix* $\mathrm{diag}(a_1, \ldots, a_n)$ *ist der Wert* $v = (\sum_{i=1}^n (1/a_i))^{-1}$, *und die optimalen Strategien x, y für Spieler 1 und 2 sind eindeutig bestimmt mit*

$$x = y = v \cdot (1/a_1, \ldots, 1/a_n)^\top.$$

Definition 2.13 Ein Matrixspiel heißt *symmetrisch*, wenn die zugehörige Matrix A eine schiefsymmetrische $(n \times n)$–Matrix ist, d.h. es gilt $A^\top = -A$.

Satz 2.8 *Der Wert eines symmetrischen Matrixspiels ist 0. Ferner sind die Mengen der optimalen Strategien für Spieler 1 und 2 identisch.*

Besonders einfach zu lösen sind (2×2)–Spiele. Besitzt die zugehörige Matrix einen Sattelpunkt, so bilden die beiden zugehörigen reinen Strategien einen Gleichgewichtspunkt. Ansonsten gilt

Satz 2.9 *Hat die (2×2)–Matrix A keinen Sattelpunkt, so sind die eindeutig bestimmten optimalen Strategien x, y und der Wert $v(A)$ des zugehörigen Matrixspiels gegeben durch*

$$x = \frac{e^\top A^*}{e^\top A^* e}, \quad y = \frac{A^* e}{e^\top A^* e}, \quad v(A) = \frac{\det(A)}{e^\top A^* e},$$

wobei $e = (1,1)^\top$ und $A^* = \det(A) \cdot A^{-1}$ die Adjungierte von A ist.

Bemerkung 2.6 Neben den (2×2)–Spielen lassen sich auch $(2 \times n)$ und $(m \times 2)$–Spiele sehr einfach lösen. Wir betrachten hier nur $(2 \times n)$–Spiele, die Behandlung der $(m \times 2)$–Spiele verläuft analog. Aufgabe des ersten Spielers ist es, $v(x) := \min_j \{a_{1j} x_1 + a_{2j} x_2\}$ zu maximieren. Dies ist wegen $x_2 = 1 - x_1 =: t$ äquivalent zur Maximierung der Funktion

$$v(t) := \min_j \{(a_{2j} - a_{1j})t + a_{1j}\}$$

für Werte $t \in [0,1]$. Die n linearen Funktionen kann man graphisch darstellen und die Funktion v als stückweise lineare Funktion bestimmen. Den Wert des Matrixspiels $v(A) = \max_{t \in [0,1]} v(t)$ kann man dann anschließend ablesen.

Abschließend geben wir ohne Beweis noch eine Lösungsmethode an, die auf alle Matrixspiele angewandt werden kann.

Bemerkung 2.7 Eine allgemeine Lösungsmethode für Matrixspiele ist das *fiktive Ausspielen*. In jeder Runde eines Spiels merkt sich jeder Spieler die Entscheidungen seines Gegners. Dabei setzt Spieler 1 die Werte $y_1, \ldots, y_n$ gemäß der relativen Häufigkeiten der Strategien von Spieler 2 bis zu diesem Zeitpunkt, und wählt seine (reine) Strategie i so, daß $A_i y$ maximal ist. Entsprechend verhält sich Spieler 2. Man kann zeigen, daß jeder Häufungspunkt der so erhaltenen Folgen von relativen Häufigkeiten ein Gleichgewichtspunkt ist.

2.7 Existenz von Gleichgewichtspunkten

In diesem Abschnitt soll für eine große Klasse von nichtkooperativen n-Personenspielen die Existenz eines Gleichgewichtspunktes gezeigt werden, insbesondere für alle endlichen Spiele. Von grundlegender Bedeutung ist dabei der *Fixpunktsatz von Brouwer*.

Satz 2.10 (Brouwer) *Es sei $K \subset \mathbb{R}^n$ ein konvexer Körper (konvexe kompakte Teilmenge des $\mathbb{R}^n$) und $f : K \to K$ stetig. Dann hat f einen Fixpunkt, d. h. es gibt ein $x \in K$ mit $f(x) = x$.*

Satz 2.11 (Nikaido–Isoda) *Sei $\Gamma = \langle I, \{S_i\}_{i \in I}, \{H_i\}_{i \in I} \rangle$ ein nichtkooperatives n–Personenspiel, das für alle $i \in I$ die folgenden Bedingungen erfüllt:*

(i) $S_i \subset \mathbb{R}^{d_i}$ *ist ein konvexer Körper.*

(ii) $H_i : S = S_1 \times \ldots \times S_n \to \mathbb{R}$ *ist stetig.*

(iii) Für jedes $x \in S$ ist die durch $x_i \mapsto H_i(x \| x_i)$ auf S_i definierte Funktion konkav.

Dann besitzt Γ mindestens einen Gleichgewichtspunkt.

Korollar 2.11.1 (Nash) *Jedes endliche n–Personen–Spiel hat mindestens einen Gleichgewichtspunkt (in gemischten Strategien).*

Bemerkung 2.8 *Ist eine der Voraussetzungen (i)-(iii) in Satz 2.11 nicht erfüllt, dann gilt die Aussage des Satzes i.A. nicht (Übung).*

2.8 Endliche Spiele

Im Falle endlicher nichtkooperativer n–Personenspiele liefert der Satz von Nash die Existenz eines Gleichgewichtspunktes. Er sagt jedoch wenig darüber aus wie diese zu finden sind. Eine erste Vereinfachung ist

Satz 2.12 *Sei* $\Gamma = \langle I, \{S_i\}_{i\in I}, \{H_i\}_{i\in I}\rangle$ *ein endliches nichtkooperatives* n*–Personenspiel. Dann ist* σ^* *Gleichgewichtspunkt von* Γ *in gemischten Strategien genau dann, wenn*

$$E_i(\sigma^* \| s_i) \leq E_i(\sigma^*) \text{ für alle } s_i \in S_i \text{ und alle } i \in I \text{ gilt.}$$

Korollar 2.12.1 *In einem endlichen nichtkooperativen* n*–Personenspiel ist jeder Gleichgewichtspunkt in reinen Strategien ein Gleichgewichtspunkt der gemischten Erweiterung.*

Korollar 2.12.2 *Sei* $\Gamma = \langle I, \{S_i\}_{i\in I}, \{H_i\}_{i\in I}\rangle$ *ein endliches nichtkooperatives* n*–Personenspiel und* σ^* *eine zulässige Situatuion in gemischten Strategien für Spieler* i*. Dann ist* $\sigma^* \| s_i$ *zulässig, d.h.* $E_i(\sigma^* \| s_i) = \max_{s_i' \in S_i} E_i(\sigma^* \| s_i')$*, für alle* $s_i \in S_i$ *mit* $\sigma_i(s_i) > 0$*.*

Wie bereits in Beispiel 2.4 gesehen, lassen sich die gemischten Strategien von Spieler $i \in I$ in einem endlichen n–Personenspiel $\Gamma = \langle I, \{S_i\}_{i\in I}, \{H_i\}_{i\in I}\rangle$ durch die Punkte eines Simplex $S^{(m_i)}$ beschreiben. Dabei gibt m_i die Anzahl der Strategien von Spieler i an; es ist also $S_i = \{s_1^{(i)}, \ldots, s_{m_i}^{(i)}\}$ für $i \in I$. Das Produkt $\mathcal{S}(\Gamma) = S^{(m_1)} \times \cdots \times S^{(m_n)}$, ein *Simplotop*, entspricht der Menge aller Situationen in gemischten Strategien. Die Gewinnerwartung E_i von Spieler i ist für

$$x = \left(\begin{pmatrix} x_1^{(1)} \\ \vdots \\ x_{m_1}^{(1)} \end{pmatrix}, \cdots, \begin{pmatrix} x_1^{(n)} \\ \vdots \\ x_{m_n}^{(n)} \end{pmatrix} \right) \in \mathcal{S}(\Gamma) \tag{2.6}$$

gegeben durch

$$E_i(x) = \sum_{(j_1,\ldots,j_n)\in\{1,\ldots,m_1\}\times\cdots\times\{1,\ldots,m_n\}} \left(\prod_{k\in I} x_{j_k}^{(k)} \right) H_i(s_{j_1}^{(1)}, \ldots, s_{j_n}^{(n)}). \tag{2.7}$$

Nach Satz 2.12 beschreibt $x^* \in \mathcal{S}(\Gamma)$ genau dann einen Gleichgewichtspunkt, wenn gilt

$$E_i(x^* \| e_j^{(i)}) \leq E_i(x^*), \quad \text{für alle } j \in \{1, \ldots, m_i\} \text{ und alle } i \in I;$$

wobei $e_j^{(i)} \in \mathbb{R}^{m_i}$ den j-ten Einheitsvektor und $x \| e_j^{(i)} = \left(x^{(1)}, \ldots, x^{(i-1)}, e_j^{(i)}, x^{(i+1)}, \ldots, x^{(n)}\right)$ bezeichnet. Dies motiviert die Betrachtung der *marginalen Erwartungsfunktion* $f : \mathcal{S}(\Gamma) \to \mathbb{R}^{m_1} \times \cdots \times \mathbb{R}^{m_n}$, die für $x \in \mathcal{S}$ definiert ist durch

$$f(x) := \left(\begin{pmatrix} E_1(x \| e_1^{(1)}) \\ \vdots \\ E_1(x \| e_{m_1}^{(1)}) \end{pmatrix}, \cdots, \begin{pmatrix} E_n(x \| e_1^{(n)}) \\ \vdots \\ E_n(x \| e_{m_n}^{(n)}) \end{pmatrix} \right).$$

Für f gilt

Satz 2.13 *Sei $\Gamma = \langle I, \{S_i\}_{i\in I}, \{H_i\}_{i\in I}\rangle$ ein endliches nichtkooperatives n–Personenspiel. Dann beschreibt $x^* \in \mathcal{S}(\Gamma)$ einen Gleichgewichtspunkt von Γ in gemischten Strategien genau dann, wenn gilt*

$$\left(x^{*(i)} - x^{(i)}\right)^\top f_i(x^*) \geq 0 \quad \textit{für alle } x \in \mathcal{S}(\Gamma) \textit{ und alle } i \in I.$$

Gilt $\left(x^{*(i)} - x^{(i)}\right)^\top f_i(x^*) \geq 0$ für ein $i \in I$ und alle $x \in \mathcal{S}(\Gamma)$, so beschreibt $x^* \in \mathcal{S}(\Gamma)$ eine zulässige Situation für Spieler i. Das ist genau dann der Fall, wenn $f_i(x^*) = 0$ oder $S^{(m_i)}$ im Halbraum $\left\{x \in \mathbb{R}^{(m_i)} \;\middle|\; x^\top f_i(x^*) \leq x^{*(i)\top} f_i(x^*)\right\}$ enthalten ist.

Beispiel 2.10 Gegeben sei ein Zwei-Personenspiel in dem jeder Spieler $i \in \{1,2\}$ drei Strategien $\{s_1^{(i)}, s_2^{(i)}, s_3^{(i)}\}$ besitzt und in dem beide Auszahlungsfunktionen gegeben sind durch

$$H_i: \quad \begin{array}{c|ccc} 1\backslash 2 & s_1^{(2)} & s_2^{(2)} & s_3^{(2)} \\ \hline s_1^{(1)} & 1 & 0 & -9 \\ s_2^{(1)} & 0 & 0 & -7 \\ s_3^{(1)} & -9 & -7 & -7 \end{array} \quad .$$

Bezeichnet A die gemeinsame symmetrische Matrix der Auszahlungsfunktionen, so ist

$$f(x,y) = (Ay, Ax) = \left(\begin{pmatrix} y_1 - 9y_3 \\ -7y_3 \\ -9y_1 - 7y_2 - 7y_3 \end{pmatrix}, \begin{pmatrix} x_1 - 9x_3 \\ -7x_3 \\ -9x_1 - 7x_2 - 7x_3 \end{pmatrix} \right).$$

Durch Prüfung der Bedingungen aus Satz 2.13 findet man, daß die gemischte Erweiterung des Spiels genau die drei Gleichgewichtspunkte $(s_1^{(1)}, s_1^{(2)})$, $(s_2^{(1)}, s_2^{(2)})$ und $(s_3^{(1)}, s_3^{(2)})$ besitzt.

Die beiden Gleichgewichtspunkte $(s_2^{(1)}, s_2^{(2)})$ und $(s_3^{(1)}, s_3^{(2)})$ sind für beide Spieler mit einer geringeren Auszahlung verbunden als $(s_1^{(1)}, s_1^{(2)})$ und können daher kaum als „Lösung des Spiels“ angesehen werden. Aufgrund solcher Probleme hat es seit den siebziger Jahren vermehrt Anstrengungen gegeben den Begriff des Gleichgewichtspunktes dahingehend zu verfeinern, daß solche unplausiblen Gleichgewichtspunkte ausgeschlossen werden. Eine, für die Spieltheorie grundlegende und allgemein anerkannte Verfeinerung stammt von Selten, der für seine Arbeiten über Gleichgewichtspunkte 1994, zusammen mit Nash und Harsanyi, den Nobelpreis für Wirtschaftswissenschaften erhielt.

Definition 2.14 Sei $\Gamma = \langle I, \{S_i\}_{i\in I}, \{H_i\}_{i\in I}\rangle$ ein endliches nichtkooperatives n–Personenspiel. Dann ist eine Situation σ in gemischten Strategien genau dann ein *perfekter Gleichgewichtspunkt* von Γ, wenn es eine Folge $\{\sigma^{(k)}\}_{k\in\mathbb{N}}$ von Situationen aus

$$\Sigma^\circ := \left\{\sigma \in \Sigma \;\middle|\; \sigma_i(s_i) > 0 \text{ für alle } s_i \in S_i \text{ und alle } i \in I\right\}$$

gibt, so daß für alle $i \in I$ folgende Bedingungen erfüllt sind:

(i) $\lim_{k\to\infty} \sigma_i^{(k)}(s_i) = \sigma_i(s_i)$ für alle $s_i \in S_i$.

(ii) $\sigma^{(k)} \| \sigma_i$ ist für i eine zulässige Situation (in gemischten Strategien).

Da die Erwartungswerte E_i stetige Funktionen auf der Menge der Situationen in gemischten Strategien sind, implizieren Bedingung (i) und (ii), daß $\sigma\|\sigma_i = \sigma$ zulässig ist für alle $i \in I$. Jeder perfekte Gleichgewichtspunkt in einem endlichen nichtkooperativen n–Personenspiel ist also ein Gleichgewichtspunkt (in gemischten Strategien).
Wenn wir σ wieder als Punkt $x \in \mathcal{S}(\Gamma)$ auffassen, beschreibt x einen perfekten Gleichgewichtspunkt genau dann, wenn es eine Folge aus $\operatorname{int}(\mathcal{S}(\Gamma))$ gibt, die gegen x konvergiert und für die $x^{(i)}$ für alle Folgenglieder eine beste Strategie für Spieler i ist. Insbesondere ist jeder Gleichgewichtspunkt, der durch ein $x \in \operatorname{int}(\mathcal{S}(\Gamma))$ beschrieben wird, perfekt.

Beispiel 2.11 (Fortsetzung) In Beispiel 2.10 gibt es genau drei Gleichgewichtspunkte, aber nur $(s_1^{(1)}, s_1^{(2)})$ und $(s_2^{(1)}, s_2^{(2)})$ sind perfekt.

Allgemein gilt

Satz 2.14 (Selten) *Jedes endliche nichtkooperative n–Personenspiel besitzt einen perfekten Gleichgewichtspunkt.*

Im Folgenden wird eine alternative Beschreibung perfekter Gleichgewichtspunkte gegeben.

Definition 2.15 Sei $\Gamma = \langle I, \{S_i\}_{i\in I}, \{H_i\}_{i\in I}\rangle$ ein endliches nichtkooperatives n–Personenspiel. Dann ist eine Situation $\sigma \in \Sigma^\circ$ in gemischten Strategien genau dann ein *ϵ-perfekter Gleichgewichtspunkt*, wenn

$$\text{aus } E_i(\sigma\|s_{i_1}) < E_i(\sigma\|s_{i_2}), \text{ für } s_{i_1}, s_{i_2} \in S_i, \quad \sigma_i(s_{i_1}) \le \epsilon \text{ folgt.}$$

Satz 2.15 *Sei $\Gamma = \langle I, \{S_i\}_{i\in I}, \{H_i\}_{i\in I}\rangle$ ein endliches nichtkooperatives n–Personenspiel. Dann ist ein Gleichgewichtspunkt σ von Γ genau dann perfekt, wenn es eine Folge ϵ_k-perfekter Gleichgewichtspunkte $\sigma^{(k)}$ gibt, so daß*

$$\lim_{k\to\infty} \epsilon_k = 0 \quad \textit{und} \quad \lim_{k\to\infty} \sigma_i^{(k)}(s_i) = \sigma_i(s_i) \textit{ für alle } s_i \in S_i \textit{ und alle } i \in I.$$

Ein perfekter Gleichgewichtspunkt ist also ein Gleichgewichtspunkt der durch gemischte Strategien approximiert werden kann, in denen jede reine Strategie mit nicht verschwindender Wahrscheinlichkeit gewählt wird, aber mit beliebig kleiner, wenn die Wahl der reinen Strategie ein Fehler wäre. Diese Charakterisierung motiviert folgende Verschärfung von Myerson.

Definition 2.16 Sei $\Gamma = \langle I, \{S_i\}_{i\in I}, \{H_i\}_{i\in I}\rangle$ ein endliches nichtkooperatives n–Personenspiel. Dann ist eine Situation $\sigma \in \Sigma^\circ$ in gemischten Strategien genau dann ein *ϵ-properer Gleichgewichtspunkt*, wenn

$$\text{aus } E_i(\sigma\|s_{i_1}) < E_i(\sigma\|s_{i_2}) \text{ für } s_{i_1}, s_{i_2} \in S_i, \quad \sigma_i(s_{i_1}) \le \epsilon \cdot \sigma_i(s_{i_2}) \text{ folgt.}$$

Ein Gleichgewichtspunkt heißt *proper*, wenn es eine Folge ϵ_k-properer Gleichgewichtspunkte $\sigma^{(k)}$ gibt, so daß

$$\lim_{k\to\infty} \epsilon_k = 0 \quad \text{und} \quad \lim_{k\to\infty} \sigma_i^{(k)}(s_i) = \sigma_i(s_i) \text{ für alle } s_i \in S_i \text{ und alle } i \in I.$$

Ein properer Gleichgewichtspunkt ist also ein Gleichgewichtspunkt der durch gemischte Strategien approximiert werden kann, in denen jede reine Strategie mit nicht verschwindender Wahrscheinlichkeit gewählt wird, aber mit wesentlich geringerer Wahrscheinlichkeit, als eine reine Strategie die größeren Gewinn verspricht. Größere Fehler werden also mit wesentlich geringerer Wahrscheinlichkeit begangen. Mit Satz 2.15 ist klar, daß jeder propere Gleichgewichtspunkt perfekt ist. Die Umkehrung gilt i.A. nicht.

Beispiel 2.12 (Fortsetzung) In Beispiel 2.10 gibt es genau einen properen Gleichgewichtspunkt, nämlich $(s_1^{(1)}, s_1^{(2)})$.

Allgemein gilt die folgende Aussage, für die wir hier keinen Beweis angeben.

Bemerkung 2.9 (Myerson) Jedes endliche nichtkooperative n–Personenspiel besitzt einen properen Gleichgewichtspunkt.

Bemerkung 2.10 Es gibt zahlreiche Arbeiten und Literatur über Verfeinerungen von Gleichgewichtspunkten endlicher Spiele. Dabei interessiert man sich vor allem für folgende Eigenschaften:

1. Es sollte einen allgemeinen Existenzsatz geben.
2. Lösungen sollten unabhängig von kleinen, „irrelevant erscheinenden“ Änderungen des Spiels sein.
3. Es sollten Gleichgewichtspunkte ausgeschlossen werden, die „intuitiv unplausibel“ sind.
4. Es sollten keine Gleichgewichtspunkte ausgeschlossen werden, die „intuitiv plausibel“ sind.

Bis heute ist keine Verfeinerung bekannt, die diesen Anforderungen umfassend genügt. Eine solche zu finden bleibt dem interessierten Leser überlassen ...

2.9 Bimatrixspiele

Definition 2.17 Ein endliches nichtkooperatives Zweipersonen-Spiel heißt *Bimatrixspiel.*

In einem Bimatrixspiel ist es möglich die Auszahlungsfunktionen in Matrizen $A, B \in \mathbb{R}^{m\times n}$ festzuhalten. Gilt $A = -B$ so handelt es sich um ein Matrixspiel. Wie bei den Matrixspielen können die Spieler ihre gemischten Strategien aus Simplizes $X \subset \mathbb{R}^m$ und $Y \subset \mathbb{R}^n$ auswählen. Im Folgenden soll eine Methode angegeben werden, mit der alle Gleichgewichtspunkte eines Bimatrixspiels bestimmt werden können.

Satz 2.16 *$(x, y) \in X \times Y$ ist genau dann ein Gleichgewichtspunkt des Bimatrixspiels mit Matrizen A, B, wenn es Zahlen α, β gibt, so daß*

$$\begin{aligned} x^\top Ay - \alpha &= 0 \\ x^\top By - \beta &= 0 \\ Ay - \alpha e &\leq 0 \\ B^\top x - \beta e &\leq 0. \end{aligned}$$

Man kann wegen Satz 2.16 auch vom Gleichgewichtspunkt (x, y, α, β) sprechen. α und β sind die zu (x, y) gehörenden Gewinnerwartungen von Spieler 1 und 2. Im Folgenden seien die Polyeder S, T definiert durch

$$S = \{(x^\top, \beta)^\top \mid B^\top x - \beta e \leq 0,\ x \in X\}$$

und

$$T = \{(y^\top, \alpha)^\top \mid Ay - \alpha e \leq 0,\ y \in Y\}.$$

Definition 2.18 Das Quadrupel $(x_0, y_0, \alpha_0, \beta_0)$ heißt *extremaler Gleichgewichtspunkt*, wenn $(x_0^\top, \beta_0)^\top$ eine Ecke von S und $(y_0^\top, \alpha_0)^\top$ eine Ecke von T ist mit $\quad x_0^\top (A + B) y_0 = \alpha_0 + \beta_0$.

Der folgende Satz zeigt, daß sich die Menge aller Gleichgewichtspunkte aus Konvexkombinationen extremaler Gleichgewichtspunkte zusammensetzt.

Satz 2.17 *Das Quadrupel (x, y, α, β) ist ein Gleichgewichtspunkt genau dann, wenn*

$$(x, \beta) = \sum_{i=1}^{r} \lambda_i (x_i, \beta_i), \quad (y, \alpha) = \sum_{j=1}^{s} \mu_j (y_j, \alpha_j), \quad \lambda_i, \mu_j \geq 0,\ \sum_{i=1}^{r} \lambda_i = \sum_{j=1}^{s} \mu_j = 1,$$

und alle Kombinationen $(x_i, y_j, \alpha_j, \beta_i)$ extremale Gleichgewichtspunkte sind.

Beispiel 2.13 Das Bimatrixspiel Γ sei gegeben durch die Matrizen

$$A = \begin{pmatrix} 2 & 1 & 2 \\ 1 & 3 & 0 \end{pmatrix}, \qquad B = \begin{pmatrix} 1 & 2 & 2 \\ 3 & 2 & 2 \end{pmatrix}.$$

Als Ecken von S ermittelt man $x^{(1)} = (1, 0, 2)^\top$, $x^{(2)} = (0, 1, 3)^\top$ und $x^{(3)} = (1/2, 1/2, 2)^\top$ und die Ecken von T sind $y^{(1)} = (1, 0, 0, 2)^\top$, $y^{(2)} = (0, 1, 0, 3)^\top$, $y^{(3)} = (0, 0, 1, 2)^\top$, $y^{(4)} = (\frac{2}{3}, \frac{1}{3}, 0, \frac{5}{3})^\top$ und $y^{(5)} = (0, \frac{1}{2}, \frac{1}{2}, \frac{3}{2})^\top$.
Durch Prüfung der Bedingung aus Definition 2.18 ergeben sich die extremalen Gleichgewichtspunkte aus den Kombinationen $(x^{(1)}, y^{(3)})$, $(x^{(1)}, y^{(5)})$, $(x^{(3)}, y^{(4)})$ und $(x^{(3)}, y^{(5)})$; die Gesamtheit der Gleichgewichtspunkte aus

$$\{x^{(1)}\} \times \{[y^{(3)}, y^{(5)}]\} \ \cup\ \{x^{(3)}\} \times \{[y^{(4)}, y^{(5)}]\} \ \cup\ \{[x^{(1)}, x^{(3)}]\} \times \{y^{(5)}\}.$$

Beschränkt man sich darauf, nur einen Gleichgewichtspunkt zu finden, so ergibt sich noch ein wirkungsvollerer Zugang. Dabei benutzt man, daß alle Bimatrixspiele mit Matrizen $A + q_1 E$, $B + q_2 E$, $q_1, q_2 \in \mathbb{R}$, strategisch äquivalent sind. Damit können wir O.B.d.A. annehmen, daß alle Einträge in A und B negativ sind.

Satz 2.18 *Sei Γ ein Bimatrixspiel mit Matrizen A, B, deren Einträge alle negativ sind. Dann bestimmt jede Lösung $(u, \hat{u}, z, \hat{z})$ von*

$$\min\{u^\top z + \hat{u}^\top \hat{z} \mid u + A\hat{z} = -e,\ u \geq 0,\ \hat{z} \geq 0, \quad \hat{u} + B^\top z = -e,\ \hat{u} \geq 0,\ z \geq 0\},$$

durch $\alpha = -(e^\top \hat{z})^{-1}$, $\beta = -(e^\top z)^{-1}$ und $x = -\beta z$, $y = \alpha \hat{z}$ eineindeutig einen Gleichgewichtspunkt (x, y, α, β) von Γ.

Damit ist das Problem des Auffindens eines Gleichgewichtspunktes zurückgeführt auf ein Optimierungsproblem, daß sich z.B. mit dem Wolfeschen Verfahren (vgl. Optimierung I) lösen läßt. Dabei ist gewährleistet, daß der Minimalwert 0 angenommen wird. In der Praxis wird allerdings in den meisten Fällen nicht das Verfahren von Wolfe, sondern eine dem speziellen Problem angepaßte Variante angewandt.

2.10 Spiele über dem Einheitsquadrat

Ein wichtiger Typ unendlicher Spiele ist der, bei dem die Spieler ein Kontinuum von reinen Strategien besitzen, die für gewöhnlich als Intervall $[0,1]$ dargestellt werden.

Definition 2.19 Ein *Spiel über dem Einheitsquadrat* ist ein Nullsummenspiel für zwei Spieler mit Strategiemengen $[0,1]$.

Die Auszahlung (an Spieler 1) ist eine auf dem Einheitsquadrat $[0,1] \times [0,1]$ definierte Funktion $H(x,y)$, durch die ein Spiel Γ über dem Einheitsquadrat eindeutig bestimmt ist. Aus der *Wahrscheinlichkeitstheorie* ist bekannt, daß eine gemischte Strategie (ein Wahrscheinlichkeitsmaß) durch eine *Verteilungsfunktion* $F : \mathbb{R} \to \mathbb{R}$ beschrieben werden kann. Diese hat die Eigenschaften

(i) F ist monoton wachsend;

(ii) $F(x) = 0$ für $x < 0$ und $F(x) = 1$ für $x \geq 1$;

(iii) F ist rechtsseitig stetig, d.h. $F_+(x_0) := \lim_{x \to x_0, x > x_0} F(x) = F(x_0)$ für alle $x_0 \in \mathbb{R}$.

$F(x)$ entspricht der Wahrscheinlichkeit, daß die Zahl aus $[0,x]$ gegriffen wird. Die Wahrscheinlichkeit, daß die Zahl aus dem Intervall $(a,b] \subset [0,1]$ gegriffen wird, ist dann $F(b) - F(a) \geq 0$. Eine reine Strategie $x_0 \in [0,1]$ kann dargestellt werden durch die Verteilungsfunktion $F_{x_0}(x) = 0$, für $x < x_0$ und $F_{x_0}(x) = 1$, für $x \geq x_0$. Eine wichtige Eigenschaft von Verteilungsfunktionen ist

Lemma 2.3 *Sei $F : \mathbb{R} \to \mathbb{R}$ eine Verteilungsfunktion. Dann existieren die links– und rechtsseitigen Grenzwerte $F_-(x)$ und $F_+(x)$ für alle $x \in \mathbb{R}$ und F ist bis auf höchstens abzählbar viele Punkte stetig.*

Wählt Spieler 1 eine reine Strategie $x \in [0,1]$ und Spieler 2 eine gemischte Strategie (Verteilungsfunktion) G, so kann die *erwartete Auszahlung* approximiert werden durch

$$\sum_{j=1}^{n} H(x, \eta_j)(G(y_j) - G(y_{j-1})),$$

mit einer Partition $\{y_0 = -\epsilon, \ldots, y_n = 1\}$ von $[-\epsilon, 1]$, $\epsilon > 0$, und Zwischenpunkten $\eta_j \in [y_{j-1}, y_j]$. Konvergieren diese Summen gegen einen festen Wert, wenn man die Feinheit der Partition $(\max_{j=1,\ldots,n} y_j - y_{j-1})$ gegen 0 gehen läßt, so wird dieser Wert das *Riemann–Stieltjes–Integral* von $H(x,\cdot)$ bezüglich G genannt. Wir schreiben für die erwartete Auszahlung im Falle der Existenz

$$E(x,G) = \int_{-\epsilon}^{1} H(x,y)\, dG(y) = \int_{\mathbb{R}} H(x,y)\, dG(y).$$

Man beachte, daß das Riemann–Integral ein Spezialfall mit $G(y) = y$ ist. Wählen Spieler 1 und 2 die gemischten Strategien F und G, so ist die erwartete Auszahlung das *Riemann–Stieltjes Doppelintegral*

$$E(F,G) = \int_{\mathbb{R}^2} H(x,y)\, d(F(x), G(y)), \tag{2.8}$$

das eine Verallgemeinerung des Riemannschen Mehrfachintegrals ist. Im Falle der Existenz, die wir hier stets voraussetzen, gilt analog zum Satz von Fubini für Riemann–Integrale

$$\begin{aligned} E(F,G) &= \int_{\mathbb{R}} E(F,y)\,dG(y) = \int_{\mathbb{R}}\int_{\mathbb{R}} H(x,y)\,dF(x)\,dG(y) \\ &= \int_{\mathbb{R}}\int_{\mathbb{R}} H(x,y)\,dG(y)\,dF(x) = \int_{\mathbb{R}} E(x,G)\,dF(x). \end{aligned}$$

Wenn die Integrale für alle F und G aus der Menge $\mathcal{V}$ aller Verteilungsfunktionen existieren, so ist die *gemischte Erweiterung* Γ^* (vgl. Bemerkung 2.2) von Γ durch die Funktion $E : \mathcal{V}\times\mathcal{V} \mapsto \mathbb{R}$ eindeutig bestimmt.

Definition 2.20 Sei Γ ein, durch die Auszahlungsfunktion $H : [0,1]^2 \mapsto \mathbb{R}$ definiertes Spiel über dem Einheitsquadrat, für das die Integrale in (2.8) für alle $F, G \in \mathcal{V}$ existieren. Dann ist die Erwartungsfunktion $E : \mathcal{V} \times \mathcal{V} \mapsto \mathbb{R}$ eine *Beschreibung der gemischten Erweiterung* von Γ.

Bemerkung 2.11 Analog lassen sich gemischten Erweiterungen von n-Personenspielen beschreiben, in denen jeder Spieler i ein abgeschlossenes Intervall $[a_i, b_i]$ an reinen Strategien besitzt.

Im Folgenden werden einige Aussagen über Riemann–Stieltjes–Integrale benötigt. Diese fassen wir zusammen in

Bemerkung 2.12 Aus der Analysis weiß man:

1. Existieren die Integrale $\int_a^b f_i(x)\,dg(x)$ und $\int_a^b f(x)\,dg_i(x)$ für $i = 1, 2$, so existieren auch die folgenden Integrale und für $\lambda_1, \lambda_2 \in \mathbb{R}$ gilt

$$\int_a^b (\lambda_1 f_1(x) + \lambda_2 f_2(x))\,dg(x) = \lambda_1 \int_a^b f_1(x)\,dg(x) + \lambda_2 \int_a^b f_2(x)\,dg(x),$$

$$\int_a^b f(x)\,d(\lambda_1 g_1 + \lambda_2 g_2)(x) = \lambda_1 \int_a^b f(x)\,dg_1(x) + \lambda_2 \int_a^b f(x)\,dg_2(x).$$

2. Existiert $\int_a^b f(x)\,dg(x)$ und ist $a < c < b$, so existieren auch die folgenden Integrale und es gilt

$$\int_a^b f(x)\,dg(x) = \int_a^c f(x)\,dg(x) + \int_c^b f(x)\,dg(x).$$

3. Ist f auf $[a,b]$ Riemann–integrierbar und g auf $[a,b]$ differenzierbar und g' Riemann–integrierbar, so existiert $\int_a^b f(x)\,dg(x)$ und es gilt

$$\int_a^b f(x)\,dg(x) = \int_a^b f(t)g'(t)\,dt.$$

4. Eine Funktion $g : [a, b] \to \mathbb{R}$ heißt *von beschränkter Variation*, wenn

$$V(g) := \sup_Z \sum_{i=1}^{n} |g(x_i) - g(x_{i-1})| \quad < \infty,$$

wobei das Supremum über alle Partitionen $Z = \{x_0 = a, \ldots, x_n = b\}$ von $[a, b]$ genommen wird. Insbesondere sind monotone Funktionen und damit auch Verteilungsfunktionen F (mit $V(F) = 1$) von beschränkter Variation.

Ist f auf $[a, b]$ beschränkt, g auf $[a, b]$ von beschränkter Variation und existiert das Integral $\int_a^b f(x)\, dg(x)$, so gilt

$$\left| \int_a^b f(x)\, dg(x) \right| \leq V(g) \cdot \sup_{x \in [a,b]} f(x).$$

Die Existenz des Integrals ist z. B. gewährleistet, wenn f auf $[a, b]$ stetig und g auf $[a, b]$ von beschränkter Variation ist.

5. Es gilt die folgende partielle Integrationsregel: Mit $\int_a^b f(x)\, dg(x)$ existiert auch $\int_a^b g(x)\, df(x)$ und es gilt

$$\int_a^b f(x)\, dg(x) + \int_a^b g(x)\, df(x) = f(b)g(b) - f(a)g(a).$$

6. Ist g eine Treppenfunktion auf $[a, b]$, die genau an den Stellen $x_1, \ldots, x_m$ Sprünge der Größe $h_1, \ldots, h_m$ besitzt, dann gilt für Funktionen f, die in $x_1, \ldots, x_m$ stetig sind,

$$\int_a^b f(x)\, dg(x) = \sum_{i=1}^{m} f(x_i) h_i.$$

7. Es sei $f_1(x) \leq f_2(x)$, für alle $x \in [a, b]$ und g monoton wachsend auf $[a, b]$. Existieren die Integrale $\int_a^b f_i(x)\, dg(x)$ für $i = 1, 2$, dann gilt

$$\int_a^b f_1(x)\, dg(x) \leq \int_a^b f_2(x)\, dg(x).$$

Im Folgenden sei $E : \mathcal{V} \times \mathcal{V} \mapsto \mathbb{R}$ immer die Beschreibung der gemischten Erweiterung des Spiels über dem Einheitsquadrat mit Auszahlungsfunktion $H : [0, 1]^2 \mapsto \mathbb{R}$. Wie bereits in Satz 2.2 gesehen, besitzt das Spiel genau dann einen einen Gleichgewichtspunkt in gemischten Strategien, wenn

$$v_1 := \sup_F \inf_G E(F, G) \qquad \text{und} \qquad v_2 := \inf_G \sup_F E(F, G).$$

angenommen werden und darüberhinaus $v_1 = v_2$ gilt. Wegen Lemma 2.2 gilt immer $v_1 \leq v_2$. Das folgende Lemma zeigt, daß es genügt, wie im Falle der Matrixspiele, die inneren Extrema über die reinen Strategien zu bilden:

Lemma 2.4 *Es gilt* $v_1 = \sup_F \inf_y E(F,y)$ *und* $v_2 = \inf_G \sup_x E(x,G)$.

Wir wollen nun zeigen, daß Spiele über dem Einheitsquadrat mit stetiger Auszahlungsfunktion $H(x,y)$ (auch *Kern* genannt) einen Gleichgewichtspunkt besitzen.

Lemma 2.5 *Ist* $H : [0,1]^2 \to \mathbb{R}$ *stetig, so gilt für alle Verteilungsfunktionen* F *und* G*, daß die Funktionen*

$$y \mapsto E(F,y) = \int_{\mathbb{R}} H(x,y)\, dF(x)$$

und

$$x \mapsto E(x,G) = \int_{\mathbb{R}} H(x,y)\, dG(y)$$

stetig sind in $[0,1]$.

Definition 2.21 Eine Folge $\{F_n\}_{n\in\mathbb{N}}$ von Verteilungsfunktionen heißt *im wesentlichen konvergent* gegen eine Verteilungsfunktion F, wenn $\lim_{n\to\infty} F_n(x) = F(x)$ für alle Stetigkeitsstellen x von F gilt.

Es gelten die folgenden beiden Sätze von Helly.

Satz 2.19 (Helly) *Jede Folge von Verteilungsfunktionen* $\{F_n\}_{n\in\mathbb{N}}$ *enthält eine Teilfolge, die im wesentlichen gegen eine Verteilungsfunktion* F *konvergiert.*

Satz 2.20 (Helly) *Sei* $\{F_n\}_{n\in\mathbb{N}}$ *eine Folge von Verteilungsfunktionen, die im wesentlichen gegen eine Verteilungsfunktion* F *konvergiert. Ferner sei* $f : \mathbb{R} \to \mathbb{R}$ *in* $[0,1]$ *stetig. Dann gilt*

$$\lim_{n\to\infty} \int_{\mathbb{R}} f(x)\, dF_n(x) = \int_{\mathbb{R}} f(x)\, dF(x).$$

Satz 2.21 *Sei* Γ *ein Spiel über dem Einheitsquadrat mit stetigem Kern* $H(x,y)$*. Dann gilt*

$$v_1 = \max_F \min_y E(F,y) = v(\Gamma) = \min_G \max_x E(x,G) = v_2.$$

Bemerkung 2.13 Der Beweis zeigt, daß sich der Wert und die optimalen Lösungen des Spiels durch Lösungen bestimmter Matrixspiele annähern lassen. Dabei hängt die Güte der Approximation vom Kern ab.

Satz 2.21 zeigt, daß die Spieler in einem Spiel über dem Einheitsquadrat mit stetigem Kern optimale gemischte Strategien besitzen. Für eine große Klasse von Spielen besitzen sie sogar optimale reine Strategien.

Definition 2.22 Ein Spiel über dem Einheitsquadrat heißt *konkav–konvex*, wenn sein Kern für jeden Wert von y konkav bzgl. x und für jedes x konvex bzgl. y ist.

Satz 2.22 *Ein konkav–konvexe Spiel über dem Einheitsquadrat mit stetigem Kern $H(x,y)$ besitzt optimale reine Strategien.*

Beispiel 2.14 Man betrachte das Spiel über $[0,1]^2$ mit Kern $H(x,y) = -2x^2+y^2+3xy-x-2y$. Wegen $H_{xx}(x,y) = -4$ und $H_{yy}(x,y) = 2$ ist das Spiel (streng) konkav–konvex. Für festes x stellt man durch eine einfache Kurvendiskussion fest, daß $H(x,y)$ genau dann minimal wird, wenn

$$y = \varphi(x) := \begin{cases} (2-3x)/2, & \text{für } x \in [0, 2/3] \\ 0, & \text{für } x \in [2/3, 1] \end{cases}$$

Entsprechend wird für festes y $H(x,y)$ genau dann maximal, wenn

$$x = \psi(y) := \begin{cases} 0, & \text{für } y \in [0, 1/3] \\ (3y-1)/4, & \text{für } y \in [1/3, 1] \end{cases}$$

Für optimale Strategien x^*, y^* gilt $H(x^*, \varphi(x^*)) = H(\psi(y^*), y^*)$. Aufgrund von $H(x^*, \varphi(x^*)) \leq H(x^*, y^*) \leq H(\psi(y^*), y^*)$ und wegen der strengen Konvexität bzw. Konkavität gilt somit $x^* = \psi(y^*)$ und $y^* = \varphi(x^*)$. Als Gleichgewichtspunkt ermittelt man $x^* = 4/17$, $y^* = 11/17$ und der Wert des Spiels ist $v = H(x^*, y^*) = -13/17$.

2.11 Das Oligopolspiel

In der Ökonomie bezeichnet man als *Oligopol* eine Wettbewerbssituation, bei der $n \geq 2$ Produzenten versuchen, in einem Markt ein bestimmtes Produkt abzusetzen. Als Modell für die Konkurrenz unter n Oligopolisten soll im Folgenden ein strategisches n–Personenspiel verwendet werden, bei dem zwar einige vereinfachende Annahmen gemacht werden, das sich aber dennoch gut zur exemplarischen Beschreibung der Anwendungsmöglichkeiten der Spieltheorie in den Wirtschaftswissenschaften eignet.
Es wird vorausgesetzt, daß die n Produzenten ein beliebig teilbares Produkt (z. B. Benzin, Heizöl, Bier etc.) herstellen. Dabei möge der Produzent i über eine Produktionskapazität K_i verfügen, $i = 1, \ldots, n$. Seine reinen Strategien bestehen nun darin, eine bestimmte Menge $x_i \in [0, K_i]$ seines Produktes herzustellen und auf dem Markt anzubieten, so daß insgesamt eine Menge von $\sum_{i=1}^n x_i$ angeboten wird. Der Preis pro Einheit richtet sich nach der insgesamt angebotenen Menge und wird mit

$$p = f\left(\sum_{i=1}^n x_i\right)$$

bezeichnet. Die dadurch gegebene Funktion $f : [0, \sum_{i=1}^n K_i] \to \mathbb{R}_{\geq 0}$ gibt also zu einer angebotenen Menge denjenigen Preis pro Einheit an, bei dem gerade das Angebot mit der Nachfrage übereinstimmt. Deshalb wird f auch als *Nachfragefunktion* bezeichnet. Wenn der Produzent i dann x_i Einheiten seines Produktes anbietet, kann er dafür am Markt einen Gesamtbetrag von

$$x_i \cdot f\left(\sum_{j=1}^n x_j\right)$$

erlösen. Um seinen unternehmerischen Gewinn zu errechnen, sind von diesem Erlös noch die Kosten $k_i(x_i)$ für die Herstellung der x_i Einheiten abzuziehen.
Wir machen die folgenden, vereinfachenden Annahmen:

(i) f ist eine stetige, in einem Intervall $[0,s] \subset [0, \min_i K_i]$ streng monoton fallende, zweimal stetig differenzierbare Funktion.

(ii) $f(0) > 0$ und $f(x) = 0$, für alle $x \geq s$.

(iii) $f''(x) \leq 0$, für alle $x \in [0,s]$.

(iv) $k_1 = \ldots = k_n =: k \ : \ [0, \max_i K_i] \to \mathbb{R}_{>0}$.

(v) k ist eine streng monoton wachsende, zweimal stetig differenzierbare Funktion mit $k''(x) \geq 0$, für alle $x \in [0, \max_i K_i]$.

Wenn die Produzenten nun die Mengen $x_1, \ldots, x_n$ ihres Produktes herstellen, macht der i–te Hersteller einen Gewinn von

$$H_i(x_1, \ldots, x_n) := x_i f\left(\sum_{j=1}^{n} x_j\right) - k(x_i), \quad i = 1, \ldots, n. \tag{2.9}$$

Definition 2.23 Das nichtkooperative n-Personenspiel $\langle I, \ \{[0,K_i]\}_{i\in I}, \ \{H_i\}_{i\in I}\rangle$ mit Auszahlungsfunktionen H_i wie in (2.9) heißt *Oligopolspiel*.

Die ökonomische Entscheidungssituation, welche Mengen $x_i \in [0,K_i]$ die einzelnen Produzenten unabhängig voneinander auf dem Markt anbieten sollen ist eine klassische Fragestellung der Ökonomie und kann mit Hilfe des entsprechenden Oligopolspiels analysiert werden. Für den Spezialfall $n = 2$ des *Duopols* ist der Begriff des Gleichgewichtspunktes seit langem (1838) als *Cournotscher Duopolpunkt* bekannt. Allgemein gilt

Satz 2.23 *Das n-Personen Oligopolspiel $\langle I, \ \{[0,K_i]\}_{i\in I}, \ \{H_i\}_{i\in I}\rangle$ besitzt unter den Voraussetzungen (i)–(v) genau einen Gleichgewichtspunkt $(x_1^*, \ldots, x_n^*)$, mit $x_1^* = \ldots = x_n^*$. Dabei gilt $x_i^* = 0$ falls $k'(0) \geq f(0)$, und im Falle $k'(0) < f(0)$ ist x_i^* die einzige Lösung der Gleichung*

$$f(nx) + xf'(nx) - k'(x) = 0,$$

die im Intervall $[0, \frac{s}{n}]$ liegt.

Beispiel 2.15 Bei einem aus drei Produzenten bestehenden Oligopol mit der Nachfragefunktion

$$f(x) := \begin{cases} 124 - x, & \text{für } x \in [0,124] \\ 0, & \text{für } x \geq 124 \end{cases}$$

und der Kostenfunktion

$$k(x) := 4x$$

erhält man die *Gleichgewichtsproduktion* $x_i^* = 30$ als Lösung von

$$(124 - 3x) + x(-1) - 4 = 0.$$

Bei dieser Gleichgewichtsproduktion erzielt jeder der drei Produzenten einen Gewinn von

$$H_i(30, 30, 30) = 30 \cdot (124 - 90) - 4 \cdot 30 = 900, \quad i = 1, 2, 3.$$

Wie in Beispiel 2.2 (Gefangenendilemma) lohnt es sich zu kooperieren, also ein Kartell zu bilden. Die Absprache, daß jeder von ihnen nur 20 Einheiten produziert führt nämlich dazu, daß jeder einen Gewinn von $20 \cdot 64 - 4 \cdot 20 = 1200$ erzielt, also ein Drittel mehr als bei der errechneten Gleichgewichtsproduktion. Allerdings ist diese Situation nicht stabil: Verdoppelt einer der Spieler seine Produktion, so erhält dieser einen Gewinn von $40 \cdot 44 - 4 \cdot 40 = 1600$, die beiden anderen jedoch nur $20 \cdot 44 - 4 \cdot 20 = 800$.