1 Einführendes Beispiel

Eine Firma möchte (beliebig teilbare) Waren von einer Stadt A in andere Städte B, C und D transportieren und dort verkaufen. Dazu stehen ihr jedoch nur begrenzte Transportkapazitäten zur Verfügung. Für die Waren kann man in den Städten unterschiedlich viel Geld verlangen, allerdings kann man überall nur eine begrenzte Menge von jeder Ware absetzen. Die Städte B und C liegen im Ausland und unterliegen gewissen Einfuhrbeschränkungen, so dass insgesamt ein gewisses Kontingent nicht überschritten werden darf. Die verantwortlichen Manager der Firma suchen nun nach einer Möglichkeit, die Waren mit Hilfe der Verkehrsmittel so auf die verschiedenen Städte aufzuteilen, dass ein maximaler Gewinn erzielt wird.

Dieses ist ein typisches Problem, das zu einer linearen Funktion führt, die unter den gegebenen (ebenfalls linearen) Nebenbedingungen (Ungleichungen) maximiert werden soll.

2 Lineare Programme

Auch viele andere Optimierungsprobleme führen auf ein System von *linearen Nebenbedingungen (linear constraints)*, also Ungleichungen und u.U. Gleichungen, sowie eine *lineare Zielfunktion (objective function)*. Solche Probleme nennen wir *lineare Programme*. Lineare Programme mit n Variablen und $m+n$ Nebenbedingungen der Form

$$\begin{array}{llcll} \text{maximiere} & z & = & \sum_{j=1}^{n} c_j x_j & \\ \text{unter} & \sum_{j=1}^{n} a_{ij} x_j & \leq & b_i & (i = 1, 2, \ldots, m) \\ & x_j & \geq & 0 & (j = 1, 2, \ldots, n) \end{array}$$

die also die Maximierung der Zielfunktion fordern und deren Nebenbedingungen lediglich Ungleichungen (davon m in der $\leq$-Form und die restlichen n als Nichtnegativitätsbedingungen) nennen wir LP-Probleme in Standardform (auch die Bezeichnungen *kanonisch* bzw. *symmetrisch* sind üblich[1]).
In der Matrixschreibweise lautet das Problem dann also

$$\begin{array}{llcl} \text{maximiere} & z & = & c^T x \\ \text{unter} & Ax & \leq & b \\ & x & \geq & 0 \end{array}$$

Dabei sei

z	der Zielfunktionswert,
c^T	der transponierte Vektor der *Koeffizienten* c_j der Zielfunktion,
A	die Matrix der Koeffizienten a_{ij} der Nebenbedingungen,
b	der Vektor der oberen Schranken b_i aus den nichttrivialen Nebenbedingungen,
x	der Vektor der Variablen x_j

[1] Allerdings werden in der Literatur die Bezeichnungen nicht einheitlich gebraucht.

mit $i \in \{1, \ldots, m\}$ und $j \in \{1, \ldots, n\}$.

Die Menge aller Punkte x, die die Ungleichungen erfüllen (*feasible solutions*), bildet ein *Polyeder*.

LP-Probleme können bezüglich ihrer Lösungsmenge in vier verschiedene Kategorien eingeteilt werden:

1. Es gibt genau eine optimale Lösung.
2. Es gibt unendlich viele optimale Lösungen.
3. Das LP-Problem hat keine Lösung (infeasible LP problem).
4. Das LP-Problem ist nach oben unbeschränkt und hat deshalb keine endliche Optimallösung (unbounded LP problem).

3 Lineare Optimierung mit der Simplex-Methode

Für die Erläuterung des 1947 von G.B. Dantzig vorgeschlagenen Simplex-Verfahrens stützen wir uns auf folgendes Beispiel aus [Chvátal83]:

$$\begin{array}{lrcrcrcr}
\textit{maximiere} & 5x_1 & + & 4x_2 & + & 3x_3 & & \\
\textit{unter} & 2x_1 & + & 3x_2 & + & x_3 & \leq & 5 \\
 & 4x_1 & + & x_2 & + & 2x_3 & \leq & 11 \\
 & 3x_1 & + & 4x_2 & + & 2x_3 & \leq & 8 \\
 & x_1, & & x_2, & & x_3 & \geq & 0
\end{array}$$

Zunächst überführen wir die ersten Ungleichungen durch die Einführung von neuen Variablen in Gleichungen. Wir ersetzen z.B. die Ungleichung

$$\begin{array}{rlclcl}
 & 2x_1 + 3x_2 + x_3 & & \leq & 5 & \\
\text{durch} & 2x_1 + 3x_2 + x_3 & +x_4 & = & 5 & \text{mit } x_4 \geq 0.
\end{array}$$

Dadurch erhalten wir ein System mit n (unabhängigen / freien) *Entscheidungsvariablen (decision variables)* und m neuen (abhängigen) *Schlupfvariablen (slack variables)*, die sozusagen den Überschuss aufnehmen und die ebenfalls alle nicht negativ sein dürfen.

Wir werden nun versuchen, ausgehend von einer offensichtlichen Lösung des Gleichungssystems durch sukzessive Verbesserungen des Zielfunktionswertes eine optimale Lösung zu finden. Am Anfang setzen wir alle Entscheidungsvariablen auf Null und berechnen die Werte der Schlupfvariablen nach den Gleichungen, die wir aus den Nebenbedingungen abgeleitet haben. Dadurch erhalten wir eine zulässige Lösung, bei der die Zielfunktion den Wert Null annimmt.

Wir fragen uns nun, wie weit wir den Wert der ersten Entscheidungsvariablen x_1 (und damit auch den Zielfunktionswert) erhöhen können, ohne die Bedingung zu verletzen, dass keine Schlupfvariable negativ werden darf. (Die Werte der anderen Entscheidungsvariablen lassen wir dabei konstant auf Null.) Für jede Nebenbedingung, in der die Entscheidungsvariable x_1 vorkommt, berechnen wir also ihren Maximalwert, indem wir die jeweilige Schlupfvariable auf Null setzen. In unserem Beispiel wird die Variable

x_1 also durch die erste Nebenbedingung $x_4 = 5 - 2x_1 - 3x_2 - x_3$ ($x_4 \geq 0$) auf einen Maximalwert von $x_1 \leq \frac{5}{2}$ beschränkt. Die zu x_5 und x_6 gehörenden Nebenbedingungen implizieren $x_1 \leq \frac{11}{4}$ bzw. $x_1 \leq \frac{8}{3}$. Da alle Beschränkungen gleichzeitig gelten müssen, ist der resultierende Maximalwert das Minimum aller Schranken, also setzen wir $x_1 = \frac{5}{2}$ und damit gilt nun $x_4 = 0$.

Auf der Suche nach einem grösseren Zielfunktionswert tauschen x_1 und x_4 nun die Rollen. x_1 hat nunmehr einen positiven Wert und wechselt auf die Seite, auf der am Anfang die Schlupfvariablen standen. Wir nennen die Variablen auf dieser Seite *Basisvariablen (basic variables)*, denn alle zusammen bilden die sogenannte *Basis* des Gleichungssystems, das auch als *Dictionary* bezeichnet wird. Umgekehrt hat x_4 nun den Wert Null und wechselt auf die Seite, auf der zu Beginn die Entscheidungsvariablen standen. Die Variablen auf dieser Seite nennen wir logischerweise *Nichtbasisvariablen (nonbasic variables)*. Um feststellen zu können, welche der Nichtbasisvariablen wir noch erhöhen könnten, um die Zielfunktion zu maximieren, müssen wir diese natürlich noch so umformen, dass sie die Abhängigkeit von x_2, x_3 und x_4 widerspiegelt. Dazu stellen wir die Gleichung, aus der wir die minimale Schranke erhielten ($2x_1 + 3x_2 + x_3 + x_4 = 5$) nach x_1 um ($x_1 = \frac{5}{2} - \frac{3}{2}x_2 - \frac{1}{2}x_3 - \frac{1}{2}x_4$) und setzen in die Zielfunktion ($z = 5x_1 + 4x_2 + 3x_3$) ein. Wir erhalten

$$\begin{aligned} z &= 5(\frac{5}{2} - \frac{3}{2}x_2 - \frac{1}{2}x_3 - \frac{1}{2}x_4) + 4x_2 + 3x_3 \\ &= \frac{25}{2} - \frac{7}{2}x_2 + \frac{1}{2}x_3 - \frac{5}{2}x_4. \end{aligned}$$

Ebenso müssen wir unsere Gleichungen, die die Basisvariablen in Abhängigkeit von den Nichtbasisvariablen darstellen, durch Einsetzen aktualisieren. Wir erhalten

$$\begin{array}{ccclll} x_1 & = & \frac{5}{2} & -\frac{3}{2}x_2 & -\frac{1}{2}x_3 & -\frac{1}{2}x_4 \\ x_5 & = & 1 & +5x_2 & & +2x_4 \\ x_6 & = & \frac{1}{2} & +\frac{1}{2}x_2 & -\frac{1}{2}x_3 & +\frac{3}{2}x_4 \\ z & = & \frac{25}{2} & -\frac{7}{2}x_2 & +\frac{1}{2}x_3 & -\frac{5}{2}x_4. \end{array}$$

Nun beginnen wir von vorn und versuchen erneut, eine Nichtbasisvariable zu vergrößern, um so den Wert der Zielfunktion zu erhöhen. Diesmal kommt jedoch nur noch x_3 dafür in Frage, weil dies die einzige Variable mit positivem Koeffizienten in der Zielfunktion ist. Eine Erhöhung von x_2 oder x_4 hätte eine Verkleinerung des Zielfunktionswertes – also einen unerwünschten Rückschritt – zur Folge. Aus der ersten Gleichung folgt (mit $x_1 \geq 0$) $x_3 \leq 5$. Die zweite Gleichung beinhaltet keinerlei Einschränkung für x_3. Die dritte Gleichung führt mit $x_6 \geq 0$ zu $x_3 \leq 1$. Das heißt, wir können x_3 auf $\min(\{5; 1\}) = 1$ anheben, der Wert der Zielfunktion erhöht sich dabei von 12.5 auf 13.

Wenn wir nun wieder die Gleichungen so umformen, dass die Basisvariablen (x_3, x_1, x_5) und die Zielfunktion in Abhängigkeit von den Nichtbasisvariablen (x_2, x_4, x_6) dargestellt werden, erhalten wir als Koeffizienten in der Zielfunktion nur negative Faktoren, so dass eine Erhöhung einer beliebigen Nichtbasisvariablen zu einer Verkleinerung des Zielfunktionswertes führen würde. Das bedeutet: wir sind fertig. Das Optimum $z = 13$ wird für $x_1 = 2$, $x_2 = 0$ und $x_3 = 1$ angenommen.

Eine alternative Möglichkeit der Darstellung ist die *Tableau-Form* (tabellarisch, ohne die Angabe der Variablennamen in den Dictionaries).

Unser ersten beiden Tableaus sehen also wie folgt aus:

$$
\begin{array}{cccccc|c}
\boxed{2} & 3 & 1 & 1 & 0 & 0 & 5 \\
4 & 1 & 2 & 0 & 1 & 0 & 11 \\
3 & 4 & 2 & 0 & 0 & 1 & 8 \\
\hline
5 & 4 & 3 & 0 & 0 & 0 & 0
\end{array}
\qquad\qquad
\begin{array}{cccccc|c}
1 & \frac{3}{2} & \frac{1}{2} & \frac{1}{2} & 0 & 0 & \frac{5}{2} \\
0 & -5 & 0 & -2 & 1 & 0 & 1 \\
0 & -\frac{1}{2} & \frac{1}{2} & -\frac{3}{2} & 0 & 1 & \frac{1}{2} \\
\hline
0 & -\frac{7}{2} & \frac{1}{2} & -\frac{5}{2} & 0 & 0 & -\frac{25}{2}
\end{array}
$$

Eine Iteration des Simplex-Verfahrens läuft demnach wie folgt ab:

1. Betrachte alle Werte in der letzten Zeile des Tableaus (außer dem letzten Wert, der den momentanen negierten Wert der Zielfunktion z angibt). Sind alle Werte negativ oder gleich Null, sind wir fertig und haben eine optimale Lösung gefunden. Ansonsten wähle die größte Zahl aus. Die Spalte, in der dieses Element steht, nennen wir *Pivot-Spalte*. Sie entspricht der Nichtbasis-Variablen, die nun in die Basis wechselt.

2. Betrachte für alle positiven Elemente r dieser Spalte die zugehörigen Einträge s in der letzten Spalte. (Falls kein Wert größer als Null ist, ist das Problem unbegrenzt.) Die Zeile mit dem kleinsten Verhältnis $\frac{s}{r}$ nennt man *Pivot-Zeile*. Diese Zeile enthält genau eine Variable, die momentan zur Basis gehört, und die nun die Basis verläßt. Der Eintrag, der sowohl zur Pivot-Spalte als auch zur Pivot-Zeile gehört, heißt *Pivot-Element*.

3. Dividiere jeden Eintrag der Pivot-Zeile durch das Pivot-Element.

4. Subtrahiere von den restlichen Zeilen ein passendes Vielfaches der aktualisierten Pivot-Zeile, so dass alle Einträge in der Pivot-Spalte (außer das Pivot-Element) zu Null werden.

In seltenen Fällen kann es passieren, dass das Simplex-Verfahren nicht terminiert, sondern immer wieder die gleichen Dictionaries durchläuft. Man bezeichnet dieses Verhalten als *Cycling*. Das kann natürlich nur dann passieren, wenn sich in einer Iteration der Zielfunktionswert nicht verbessert. Solch eine Situation kann dann entstehen, wenn in einer oder mehr Nebenbedingung(en) das konstante Glied gleich Null ist. Solche Dictionaries heissen *entartet (degenerate)*. Es können dann *entartete Iterationen* auftreten, in denen sich der Wert der Basis-Variablen (und der Wert der Zielfunktion) nicht verändert. Meist endet eine Serie solcher entarteter Iterationen mit einer Iteration, in der sich der Zielfunktionswert wieder verbessert. Aber wie schon gesagt, gibt es auch Ausnahmen.

Es gibt jedoch auch Möglichkeiten, um Cycling zu vermeiden: die lexikographische Methode (eine Ausprägung der klassischen Perturbationsmethode), und die etwas neuere *Smallest-subscript rule*.

Theorem 1 (Smallest-subscript rule: BLAND**, 1977)** *Wählt man in jeder Iteration für die Variable x_k, die die Basis verlässt bzw. betritt, diejenige mit dem kleinsten Index k, so terminiert das Simplex-Verfahren garantiert.*

3.1 Komplexität

DANTZIG berichtete 1963, dass die Simplex-Methode für praktische Probleme mit $m < 50$ und $m + n < 200$ normalerweise weniger als $3m/2$ und nur selten bis zu $3m$ Iterationen benötigt. Diese Beobachtung wird durch neuere Erkenntnisse auf größeren Problemen bestätigt, die zeigen, dass die typische Anzahl der Iterationen proportional mit m und nur sehr langsam (näherungsweise logarithmisch) mit n ansteigt.

Leider existieren auch Beispiele, für die die Simplex-Methode eine riesige Anzahl (nämlich $2^n - 1$, also exponentiell viele) Schritte benötigt ("pathological examples", KLEE und MINTY 1972). Diese Beispiele sind zwar äußerst selten, jedoch ist die Simplex-Methode damit aus Sicht der Komplexitätstheorie kein effizienter Algorithmus.

Umgekehrt existiert ein polynomieller Algorithmus, die *Ellipsoid-Methode*, die in der Praxis viel zu langsam ist. Die noch deutlich jüngeren *Barrier-* oder *Interior-point-Methoden* sind ebenfalls polynomiell und erreichen auch in der Praxis gute Laufzeiten für größere Probleminstanzen.

Literatur:
Chvátal, Vašek: *Linear Programming*, W.H. Freeman and Company, New York 1983.