Methoden der Verifikation

Beweiser – „Integer Linear Problems“

Oleg Iskov

Universität Bremen
SS 2005

Inhaltsverzeichnis:

1. Grundidee …………………. 2

2. Ein Beispiel zu Einführung …………………. 2

3. Formale, abstrakte Formulierung des Problems ………………….3

4. Normalformen …………………. 5

5. Spezialfall: Diskrete lineare Programmierung
(eng. : Integer linear programming) …………………. 6

6. Geometrische Interpretation …………………. 6

7. Lösbarkeit …………………. 8

8. Lösungsverfahren …………………. 8

Simplexalgorithmus …………………. 8

9. Beispiele …………………. 9

In dieser Arbeit wird ein Verfahren zur Optimierung bestimmter Prozesse vorgestellt. Man findet hier die Beschreibung der Grundidee, die man auch mit darauf folgendem Beispiel verstehen kann. Dann werden abstrakte Formulierung und Normalformen bekannt gegeben. Hier sieht man, wie lineare Optimierungsprobleme ausgeschrieben werden. Die diskrete lineare Programmierung wird als Spezialfall angeschaut und auch geometrische Interpretation des Problems. Am Ende wird es mit der Lösbarkeit und den Lösungsverfahren beschäftigt.

1. Die Grundidee

In der Informatik beschäftigt man sich unter anderem auch mit Optimierung der bestimmten Prozesse und Verfahren. Eines der Hauptverfahren dazu ist die lineare Programmierung. Grundidee der Linearen Programmierung besteht in der Optimierung einer linearen Funktion über eine Menge, die durch lineare Gleichungen und Ungleichungen eingeschränkt ist.

Lineare Programmierung stellt eine sehr allgemeine Methode zur Optimierung von Problemlösungen dar, für die keine speziell entwickelten Algorithmen bekannt sind.

2. Ein Beispiel zu Einführung

Um es verständlich zu machen, welche linearen Probleme man lösen will und was für eine Aufgabe dabei die lineare Programmierung hat, zeigt folgendes Beispiel.

Zum **Beispiel** hat man einen Politiker, der eine Wahl gewinnen will. Dafür muss er Geld in eine Wahlkampagne investieren. Sein Wahlkampfsfeld ist eine Stadt mit 100.000 Einwohnern, Vorstadt mit 200.000 Einwohnern und das Land mit 50.000 Einwohnern. Das Problem dabei ist, das Wahlkampfgelder möglichst effizient einsetzen. Also so wenig wie möglich ausgeben und dabei in der Stadt, der Vorstadt und auf dem Land jeweils absolute Mehrheit zu erringen.

Je nach dem in welches Thema er investieren würde, steigt oder sinkt seine Beliebtheit in diesen drei Bevölkerungsgruppen, wovon auch sein Wahlerfolg abhängt.

Das zeigt uns folgende Tabelle:

Thema	Variable	Stadt	Vorstadt	Land
Straßenbau	X_1	-2	5	3
Waffenkontrolle	X_2	8	2	-5
Subventionen	X_3	0	0	10
Mineralölsteuer	X_4	10	0	-2

Gewinne o. Verluste in Tausend Wahlstimmen pro $1000

Die Tabelle lässt sich folgendermaßen interpretieren. Investiert man $1000 in Waffenkontrolle, dann steigt die Beliebtheit in der Stadt um 8 in der Vorstadt um 2 Tausend Stimmen, dabei sinkt sie auf dem Land um 5000 Stimmen.

Die erwarteten Gewinne oder Verluste lassen sich in eine lineare Funktion zusammenfassen, so dass man folgendes lineares Problem hat:

Minimiere $x_1+x_2+x_3+x_4$ unter der Bedienungen der Mindestzahl der Wahlstimmen. Nach dem man alle Wahlthemen für jede Region zusammenfasst, bekommt man folgende Nebenbedienungen:

$-2x_1+8x_2+0x_3+10x_4 >= 50$ (50% von 100, also mindestens die Hälfte aller Stimmen)
$5x_1+2x_2+0x_3+ 0x_4 >= 100$ (50% von 200)
$3x_1 -5x_2+10x_3 - 2x_4 >= 25$ (50% von 50)

$x_1, x_2, x_3, x_4 >= 0$

Das Problem der linearen Programmierung ist es, diese Aufgabe zu lösen.

3. Formale, Abstrakte Formulierung

Ein lineares Problem wird beschrieben durch die Angabe:

1. der linearen Zielfunktion, die dann optimiert werden soll,
2. eines zulässigen Bereiches, aus dem die Argumente der Zielfunktion stammen dürfen. Definition dieses Bereiches erfolgt durch so genannte Nebenbedienungen.

Ziel der linearen Programmierung ist es, eine lineare Funktion in Abhängigkeit von einer Menge der linearen Ungleichungen zu optimieren. Man bezeichnet dabei das Minimierungsproblem oder Maximierungsproblem.

Sei $c_1, c_2, \dots c_n$ die reale Zahlen und $x_1, \dots x_n$ eine Menge der Variablen, dann haben wir eine lineare Funktion:

$$f(x_1, \dots, x_n) = c_1x_1 + c_2x_2 + \cdots + c_nx_n = \sum_{i=1}^{n} c_ix_i$$

Sei b eine reale Zahl und f eine lineare Funktion dann ist:

$f(x_1, \dots, x_n) = b$ eine lineare Gleichung

und $f(x_1, \dots, x_n) \leq b$ $f(x_1, \dots, x_n) \geq b$

lineare Ungleichungen.
Die Aufgabe der linearen Programmierung ist dann folgendermaßen definiert:

$$\text{maximiere } f(x_1, \dots, x_n) = \sum_{i=1}^{n} c_ix_i$$

unter Einhaltung der linearen Bedingungen

$$\sum_{j=1}^{n} a_{ij}x_j \leq b_i \qquad \text{für } i = 1, 2, \dots, m$$

$$x_j \geq 0 \qquad \text{für } j = 1, 2, \dots, n$$

Dieses lineare Problem lässt sich auch anders Aufschreiben und zwar in einer Matrixschreibweise:

Sei $A = a_{ij}$
$c = c_j$
$x = x_j$
$b = b_i$ $i = 1, 2, \dots, m$ $j = 1, 2, \dots, n$

Dann wird das lineare Problem wie folgt definiert:

Maximiere $f(x) = cx$
unter der Einhaltung folgender Nebenbedienungen $Ax \leq b$
$x \geq 0$
Das alles kann man noch kompakter zusammenfassen (A, b, c).

4. Formale, Abstrakte Formulierung

Es gibt zwei Normalformen für die Formulierung der linearen Probleme. Das sind:

Standartform – alle Bedienungen sind durch Ungleichungen definiert
Slackform – alle Bedienungen sind durch Gleichungen definiert.

Man kann ein lineares Problem durch geeignete Umformungen in beiden Formen aufschreiben. In der Standartform ist ein lineares Problem als Maximierungsproblem definiert. Nebenbedienungen sind dabei Ungleichungen und zwar in Form - „kleiner als“, Variablen werden unbedingt auf Nichtnegativität geprüft.

Mögliche Abweichungen von der Standartform:

1. Minimierungsproblem statt Maximierungsproblem
2. Variablen ohne Nichtnegativitätsbedienung
3. Gleichheitsbedienungen statt Ungleichheitsbedienungen
4. „Größer als“ Bedienungen statt „Kleiner als“ Bedienungen

Man kann ein lineares Problem wieder in Standartform bringen, indem man einige einfache Operationen anwendet.

Wenn wir statt Maximierungsproblem Minimierung haben, dann lässt es sich durch Negierung der Koeffizienten c in der Zielfunktion, die man optimieren will. Wenn eine Variable x_j nicht auf Nichtnegativität geprüft wird, wird sie durch $x'_j - x''_j$ mit $x'_j, x''_j \geq 0$ ersetzt. Falls die Nebenbedienungen, die unsere Funktion beschränken, Gleichungen $f(x) = b$ sind, werden die durch Ungleichungen $f(x) \leq b$ und $f(x) \geq b$ ersetzt. Wenn die Bedienungen in Form „Größer als“ statt „Kleiner als“ aufgeschrieben, werden Koeffizienten a_{ij}, b_i in der entsprechenden Bedienung negiert.

Also alle Umformungen entsprechend der oben geschriebenen Abweichungen:

1. Negierung der Koeffizienten c in der Zielfunktion,
2. Ersetzung von x_j durch $x'_j - x''_j$ mit $x'_j, x''_j \geq 0$
3. Ersetzung von $f(x) = b$ durch $f(x) \leq b$, $f(x) \geq b$
4. Negierung der Koeffizienten a_{ij}, b_i in der betreffenden Bedienung i

Beispiel: minimiere $3x_1 - 5x_2$

Unter Einhaltung der Nebenbedienungen:

$$\begin{aligned} x_1 + x_2 &= 8 \\ x_1 - x_2 &\leq 4 \\ x_1 &\geq 0 \end{aligned}$$

Durch Schrittweise Anwendung der oben genannten Umformungsregeln kann man dieses lineare Problem in eine Standartform überführen. Nach der Umformung enthält man das äquivalente lineare Problem:

Maximiere $-3x_1 + 5x_2 - 5x_3$

$$
\begin{array}{rcl}
x_1 + x_2 - x_3 & \leq & 8 \\
-x_1 - x_2 + x_3 & \leq & -8 \\
x_1 - x_2 + x_3 & \leq & 4 \quad , x_1 , x_2 , x_3 \geq 0
\end{array}
$$

5. Spezialfall: Diskrete lineare Programmierung (eng. : Integer linear programming)

Die so genannte diskrete lineare Programmierung ist ein Spezialfall der linearen Programmierung. Auf Englisch wird es auch „Integer linear programming“ genannt. Bei dieser Art von Optimierungsproblemen müssen Variablen zusätzlich eingeschränkt werden. Und zwar mit x in Z^n . Das heißt, dass unser Ergebnis nur aus dem Bereich der ganzen Zahlen stammen darf.

Diese Einschränkung führt dazu, dass der Lösungsraum weiter Eingeschränkt wird. Die Extremwerte unseres Polytops sind aber immer noch realwertig, deswegen werden die bekannten Algorithmen für die Lösung der linearen Probleme keine gültige Lösung finden. Um das zu umgehen, versucht man das Ungleichungssystem so erweitern, bis die Lösung eine ganze Zahl ist.

Seinerseits ein Spezialfall der diskreten linearen Programmierung ist binäre diskrete lineare Programmierung. Da kommt noch zu Einschränkung, dass Variablen nur Werte 0 und 1 annehmen dürfen. „traveling salesman problem“ lässt sich als solches binäre diskrete lineare Problem darstellen.

6. Geometrische Interpretation:

Ein lineares Optimierungsproblem lässt sich auch geometrisch interpretieren. Und zwar wenn man eine Gleichung mit Variablen $x_1, \ldots, x_n$ hat, dann beschreibt diese eine Hyperebene in n-dimensionalem Raum. Also alle Punkte ($a_1, \ldots, a_n$), die diese Gleichung erfüllen werden, sind in der Hyperebene (Diese Hyperebene ist in 2D – Raum eine Gerade, in 3D – Raum eine Ebene). Die Ungleichungen in den Nebenbedienungen, die unsere Funktion einschränken, beschreiben einen Halbraum mit allen Punkten, die auf einer Seite der Hyperebene liegen.

Dieser Halbraum ist dann eine Menge der gültigen Lösungen. Kombiniert man mehrere Ungleichungen in ein Ungleichungssystem, dann enthält man die Punkte, die alle Ungleichungen erfüllen, als Lösungsmenge. Dabei entsteht ein konvexer Polytop oder auch Simplex genannt.

Die Funktion, die wir maximieren wollen, wird auch als eine Hyperebene dargestellt. Wir wissen nur nicht den Zielwert, also Abstand von dem Ursprung. Man findet eine Lösung, indem man unsere Ebene vom Unendlichen auf Ursprung zuwandern lässt. Der Punkt, in dem unsere Ebene Simplex zum ersten Mal berührt, ist eine gültige Lösung.

Beispiel:

Maximiere $x1 + x2$

Nebenbedienungen:

$4x1 - x2 \leq 8$

$2x1 + x2 \leq 10$

$5x1 - 2x2 \leq -2$, $\quad$ $x1, x2 \geq 0$

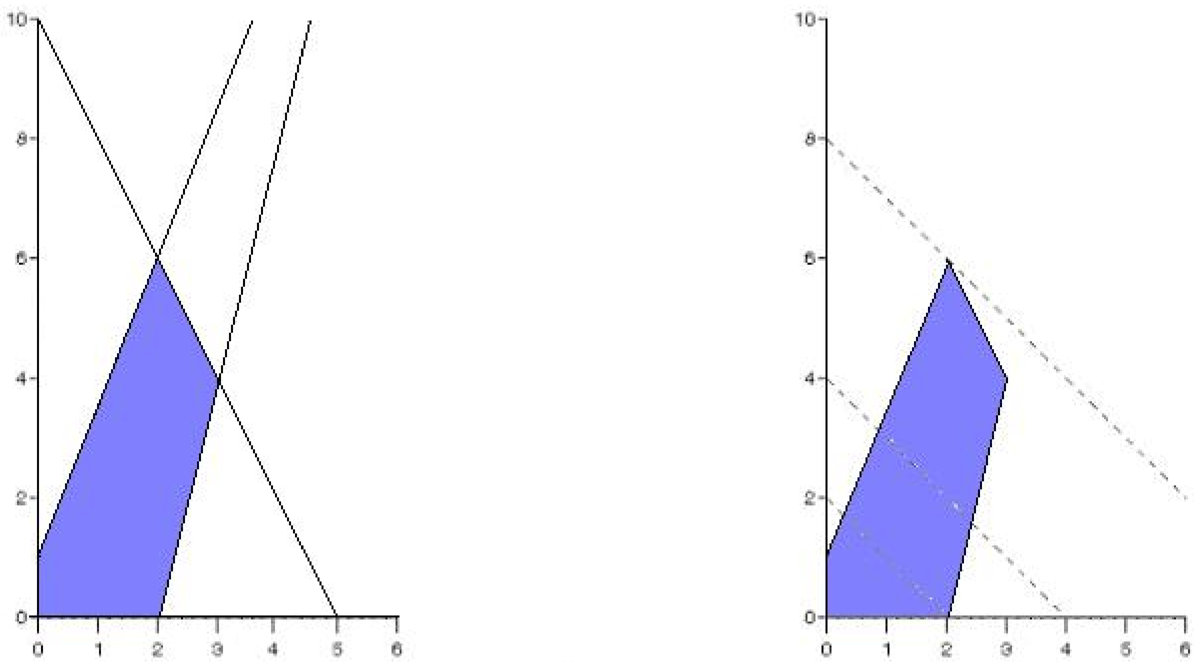

Abb. 1

7. Lösbarkeit:

Ein lineares Problem muss nicht lösbar sein. Dieser Fall tritt unter folgenden Bedienungen ein:

1. Der zulässige Bereich x: $Ax \leq b$, $x > 0$ ist leer.
2. Die Zielfunktion ist auf dem zulässigen Bereich nicht nach Oben beschränkt.

Im zweiten Fall kann durchaus eine (aufgrund der 1.Bedienungen) Lösung vorhanden sein.

8. Lösungsverfahren:

Zur Lösung von linearen Programmen wird meist der Simplexalgorithmus verwendet. Dieser Algorithmus ist in den meisten Fällen effizient, kann jedoch exponentielle Laufzeit besitzen, daher hielt man lange Zeit lineare Probleme für nicht effizient lösbar. In den letzten Jahren wurden jedoch Inner-Punkt-Verfahren zur linearen Programmierung entwickelt, deren Laufzeit polynomial ist. Ein Algorithmus mit polynominaler Laufzeit ist auch die Ellipsoid-Methode.

Will man lineare diskrete Programme lösen, werden häufig Cutting-Plane-Methoden eingesetzt: dabei wird mit einem der bekannten Verfahren eine optimale Lösung gesucht. Ist diese Lösung nicht diskret, so erweitert man das Ungleichungssystem um eine so genannte cutting plane, eine Ungleichung, die das Polytop beschneidet, ohne gültige ganzzahlige Lösungen zu verwerfen. Dieses Verfahren wird iterativ angewendet, bis als optimale Lösung ein Punkt gefunden wird, der nur aus diskreten Komponenten besteht.

Simplex – Verfahren:

Zur Lösung von der linearen Optimierungsproblemen wird in der Praxis am häufigsten der Simplex-Algorithmus eingesetzt. Zwar gibt es Anwendungsfälle, diese dieses Verfahren in keine polynomielle Laufzeit mehr löst, doch in der Praxis löst es die Mehrzahl aller Aufgaben effizient. Der Algorithmus wurde ursprünglich 1947 von Dantzig entwickelt und im Laufe der Zeit deutlich verbessert.

Wie der Name auch sagt, nutzt der Simplex-Algorithmus die Simplex-Form des Lösungsraums aus, um seine Aufgabe zu bearbeiten. Der Algorithmus erhält als Eingabe ein lineares Programm in Slackform und berechnet dann eine optimale Lösung.

Das kann man so vorstellen, dass Verfahren an einer Ecke des Polytops beginnt und eine Kette von Iterationen durchführt. Bei diesen Iterationen springt der Algorithmus jeweils zu einer benachbarten Ecke, sofern deren Wert nicht kleiner als der aktuelle ist. Der Wert wird durch Einsetzen der Ecke als Lösung in die Zielfunktion berechnet.

Der Algorithmus wird gestoppt, sobald er ein Maximum findet, das bedeutet, dass keine der benachbarten Ecke einen höheren Zielwert hat. Dieses Maximum ist global wegen der Konvexität des Polytops und der Linearität der Zielfunktion.

9. Beispiele:

Jetzt wird man ein der Beispiele des Optimierungsproblems sehen, was dann mit dem Simplex – Verfahren gelöst wird. Man hat folgende *Aufgabestellung*:

Eine Firma stellt 2 verschiedene Produkte her. Es stehen 3 Maschinen A, B, C zur Verfügung. Maschine A hat eine maximale monatliche Laufzeit (Kapazität) von 170 Stunden, Maschine B von 150 Stunden und Maschine von 180 Stunden. Eine Mengeneinheit (ME) von Produkt 1 liefert einen Deckungsbeitrag von 300 Euro, eine ME von Produkt 2 dagegen 500 Euro. Die Fixkosten betragen 36.000 Euro pro Monat. Fertigt man 1 ME von Produkt 1, dann benötigt man dafür zunächst 1 Stunde die Maschine A und danach 1 Stunde die Maschine B. 1 ME von Produkt 2 belegt nacheinander 2 Stunden Maschine A, 1 Stunde Maschine B und 3 Stunden Maschine C.

Dieses Problem schreiben wir jetzt in einer Standartform.

Maximiere $300x_1 + 500x_2 = 36.000$

Unter der Einhaltung folgender Nebenbedienungen:

$1x_1 + 2x_2 \leq 170$ Maschine A
$1x_1 + 1x_2 \leq 150$ Maschine B
$0x_1 + 3x_2 \leq 180$ Maschine C , $x_1 \geq 0$, $x_2 \geq 0$

Für den Simplex – Algorithmus muss ein Problem in einer Slackform aufgeschrieben werden, weil es sehr unhandlich ist, es in Standartform zu lösen.

Dieses Problem schreiben wir jetzt in einer Standartform. Dazu führt man so genannte *Schlupfvariablen* y_A, y_B und y_C ein, welche die nicht genutzten Zeiten der einzelnen Maschinen darstellen.

Maximiere $- 300x_1 - 500x_2 = - 36.000$

Unter der Einhaltung folgender Nebenbedienungen:

$$y_A + x_1 + 2x_2 = 170$$
$$y_B + x_1 + x_2 = 150$$
$$y_C + 3x_2 = 180\ , \quad y_A, y_B, y_C, x_1, x_2 \geq 0$$

Die Gleichungen überträgt man in so genannte Simplex – Tabelle:

	x_1	x_2	rechte Seite
G	-300	-500	-36000
Y_A	1	2	$170 = b_1$
Y_B	1	1	$150 = b_2$
Y_C	0	3	$180 = b_3$

Die Variablen in der Kopfzeile heißen Nichtbasisvariable, die Variablen in der 1.Spalte Basisvariable. Die Zahlen in der "Zeile G" - der Gleichung für die Zielfunktion - heißen Zielfunktionskoeffizienten. Die Variablen b_1, b_2 und b_3 bezeichnen die Werte der rechten Seite.

Man kann sofort eine *triviale Lösung* sagen und zwar $x_1 = 0$ und $x_2 = 0$, Variablen x_1 und x_2 sind nicht Basisvariablen und diese haben immer Wert „0“. Bei dieser Lösung hätte man einen Verlust -36000 €. Deswegen müssen wir eine bessere Lösung finden. Das macht man mittels einer *Simplex – Iteration*.

Man macht einen Austauschschritt, indem in einer Simplex-Iteration eine Basisvariable gegen eine Nichtbasisvariable ausgetauscht wird. Es werden Nichtbasisvariable mit negativem Zielfunktionskoeffizienten genommen. Jetzt sucht man diejenige Basisvariable, bei dessen Austausch man den größten Zuwachs für die Zielfunktion bekommt. Sei s die Nummer der Spalte der auszutauschenden Nichtbasisvariable und r die Nummer der Zeile der auszutauschenden Basisvariablen. Ein Austauschschritt entspricht exakt einem Schritt beim Lösen eines linearen Gleichungssystems, bei dem man die Zeile r nach der Variablen x_s auflöst und dann x_s in die restlichen Gleichungen einsetzt. Seien a_{ij} die (Matrix)Elemente der Simplex - Tabelle. Dann heißt a_{rs} das Pivotelement der Simplex - Tabelle. Die Spalte s heißt Pivotspalte, die Zeile r heißt Pivotzeile.

Bei einem Austauschschritt berechnet sich das neue Simplex – Tabelle folgendermaßen: Pivotelement, Pivotzeile für j ungleich s, Pivotspalte für i ungleich r, restliche Elemente. Man braucht ein Pivotelement. Man sucht sich diejenigen Werte r, s, für die der Zielfunktionskoeffizient s negativ ist und für die der Wert von $G := G - a_{0s}\, b_j / a_{rs}$

am größten wird. Es muss darauf geachtet werden, dass die Lösung, die nach dem Austausch entstanden ist, zulässig ist. r und s werden so bestimmt, dass der Quotient

b_j / a_{rs} mit $a_{rs} \neq 0$

in einer Spalte minimal wird. Es gibt einen negativen Zielfunktionskoeffizienten in der 1.Spalte -300. Die kann man als Pivotspalte s benutzen. Quotient für die 1.Spalte:

Reihe 1: 170 / 1 = 170
Reihe 2: 150 / 1 = 150
Reihe 3: $a_{31} = 0$, deshalb ist kein Quotient dafür berechenbar.

Es gibt den minimalen Quotient 150 in Reihe 2. Mit dem Pivotelement $a_{2,\,1}$ wird der neue Wert der Zielfunktion berechnet: G = -36000 - (-300)×150 / 1 = 9000. In der Spalte 2 gibt es einen negativen Zielfunktionskoeffizienten – 500, kommt also als Pivotspalte s in Frage. Berechnung der Quotienten dieser Spalte 2:

Reihe 1: 170 / 2 = 85
Reihe 2: 150 / 1 = 150
Reihe 3: 180 / 3 = 60

Den minimalen Quotient 60 erhält man also in Reihe 3. Mit dem Pivotelement $a_{3,1}$ berechnet sich der neue Wert der Zielfunktion G = -36000 - (-500)×60 / 3 = -26000. Damit ist es günstiger, als Pivotelement die Zahl 1 aus Spalte s = 1 und Reihe r = 2 zu wählen, weil damit die Zielfunktion am meisten anwächst, nämlich auf 9000. Jetzt muss man nur noch Austauschschritt durchführen. Mit dem Austauschschritt wird die Basisvariable Y_B mit der Nichtbasisvariablen x_1 ausgetauscht und die Simplex –Tabelle wird nach den obigen Regeln umgerechnet. Das Pivotelement ist $a_{2,1} = 1$. Umrechnung des Pivotelementes: $a_{2,1} = 1 / 1 = 1$
Umrechnung der Pivotzeile r = 2: Jedes Element wird durch das Pivotelement geteilt. Spalte 1 ist Pivotelement und wurde eben bereits berechnet.

Spalte 2: $a_{2,2} = 1 / 1 = 1$
rechte Seite: b2 = 150 / 1 = 150

Umrechnung der Pivotspalte s = 1: Jedes Element wird durch das Pivotelement geteilt, das Vorzeichen kehrt sich um.

Zielfunktion: - (-300 / 1) = 300
Zeile 1: $a_{1,1} = -(1 / 1) = -1$
Zeile 2 ist Pivotzeile und bereits berechnet
Zeile 3: $a_{3,1} = (-(0 / 1) = 0$

Umrechnung der restlichen Werte:

Zielfunktion Spalte 2: -500 - (-300 × 1) /1 = -200
Zeile 1 Spalte 2: $a_{1,1} = 2 - 1\times1 / 1 = 1$
Zeile 2 Spalte 2 gehört zur Pivotzeile und ist bereits berechnet.
Zeile 3 Spalte 2: $a_{3,1} = 3 - 0\times1 / 1 = 3$

Zielfunktion: G = -36000 - (-300 × 150) / 1 = 9000 rechte Seite:

Zeile 1: b1 = 170 - 1×150 / 1 = 20

Zeile 2 gehört zur Pivotzeile und ist bereits berechnet.

Zeile 3: b3 = 180 - 0×150 / 1 = 180

Die neue Simplex-Tabelle. Nach der Umrechnung ergibt sich eine neue Simplex-Tabelle:

	Y_B	x_2	rechte Seite
G	300	-200	9000
Y_A	-1	1	$20 = b_1$
x_1	1	1	$150 = b_2$
Y_C	0	3	$180 = b_3$

Unsere Zielfunktion in der Simplex-Tabelle besitzt noch einen negativen Koeffizienten, deswegen kann man die Lösung noch verbessern. Dafür wird noch eine Simplex – Iteration ausgeführt. Danach haben wir folgende neue Simplex – Tabelle:

	Y_B	Y_A	rechte Seite
G	100	200	13000
x_2	-1	1	$20 = b_1$
x_1	2	-1	$130 = b_2$
Y_C	3	-3	$120 = b_3$

Wir haben optimale Lösung erreicht, da wir keinen negativen Koeffizienten mehr haben. Es werden 130 ME von Produkt 1 und 20 ME von Produkt 2 hergestellt. Damit erzielt die Firma einen Gewinn von 13.000 Euro. Maschine A und Maschine B sind voll ausgelastet. Maschine C hat noch eine ungenutzte Kapazität von 120 Stunden.

Aus dieser Arbeit hat man jetzt gesehen, wie man mit Hilfe der linearen Programmierung bestimmte Prozesse und auch Verfahren optimieren kann.

Informationsquellen:

1. Freie Enzyklopädie „wikipedia.de“ (Beispiele und Definitionen)

2. Internetseite „informatikfuchs.de“

3. Skript www.techfak.uni-bielefeld.de/ags/wbski/lehre/digiSA/WS0304/IntAlg/Ausarbeitungen/lp.pdf
   Frank Schönman

4. http://ls2-www.cs.uni-dortmund.de/~voecking/EA-SS2004/skript/lp.pdf
   Berthold Vöcking, Universität Dortmund, 13. Mai 2004

5. http://www.wiwi.uni-frankfurt.de/Professoren/bartels/Skripte/Lp.pdf
   Prof. Dr. Hans G. Bartels, Universität Frankfurt am Main