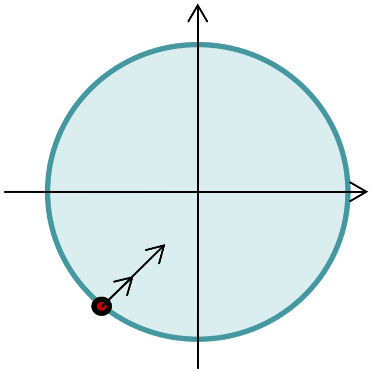
Optimierung

Vorlesung 5

Lineare Programmierung

In der letzten Vorlesung



Optimierung mit Nebenbedingungen

$$\boxed{\begin{aligned} \nabla_x \mathcal{L}(x,\lambda) &= 0 \\ c_i(x) &= 0, \quad \forall i \in \mathcal{E} \\ c_i(x) &\geq 0, \quad \forall i \in \mathcal{I} \\ \lambda_i &\geq 0, \quad \forall i \in \mathcal{I} \\ \lambda_i c_i(x) &= 0, \quad \forall i \in \mathcal{E} \cup \mathcal{I} \end{aligned}}$$

KKT Bedingungen für ein Minimum

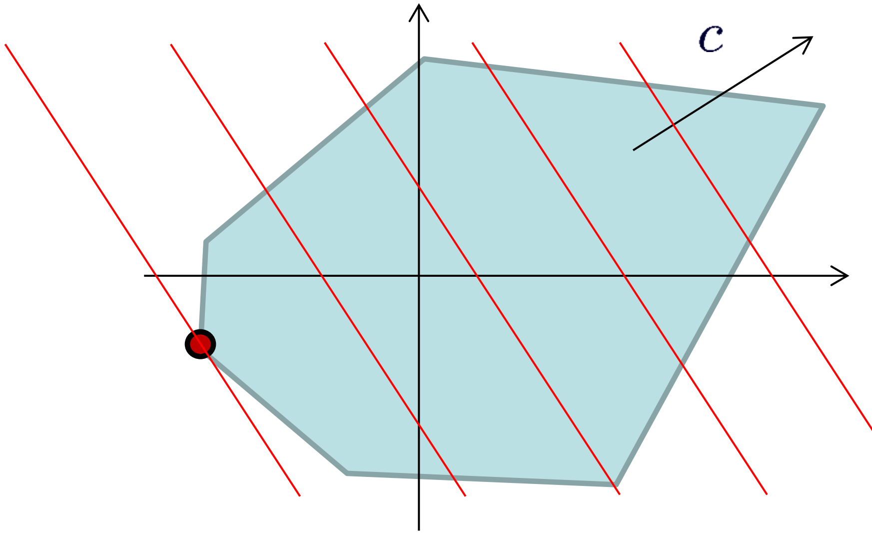
$$q(\hat{\lambda}) = \mathcal{L}(\hat{x}, \hat{\lambda}) = f(\hat{x})$$

Lagrange Dualität

Lineare Programme

- Der einfachste Fall von Optimierung mit Nebenbedingung sind lineare Programme:

$$\min_x c^\top x, \qquad Ax - b \geq 0$$



Polyeder möglicher Lösungen

- Sowohl die Zielfunktion $f(x)$ ist <u>linear</u> als auch alle Nebenbedingungen.

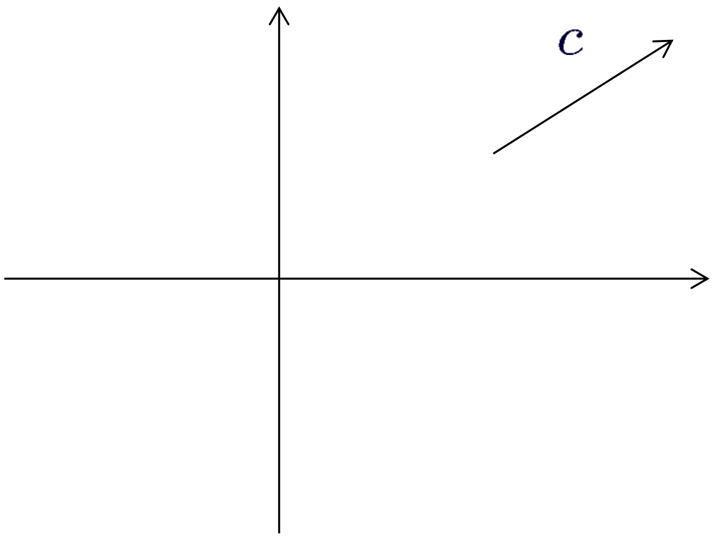
- Das Problem ist offensichtlich konvex und es gilt daher strenge Dualität (mehr dazu später)

- Die Nebenbedingungen spannen einen konvexen Polyeder auf. Falls ein Minimum existiert, liegt es auf dem Rand des Polyeders.

Lineare Programme ohne Lösung

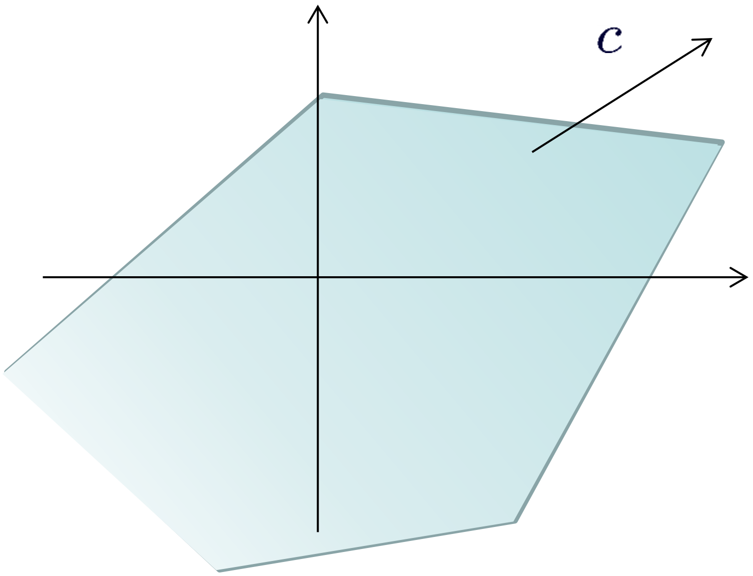
1. Ein lineares Programm ist **ungültig**, wenn der zulässige Polyeder die leere Menge darstellt.

   D.h. einige Nebenbedingungen schließen sich gegenseitig aus.



2. Ein lineares Programm ist **unbeschränkt**, wenn der Polyeder in negativer Richtung von c offen ist.

   Die Zielfunktion kann dann beliebig kleine Werte annehmen.



Standardform

- Bedingt durch das Simplex-Verfahren zur Optimierung linearer Programme hat sich die Formulierung in der **Standardform** durchgesetzt:

$$\min_x c^\top x, \qquad Ax = b,\ x \geq 0$$

- Jedes lineare Programm lässt sich in diese Form überführen.

- Um Ungleichheitsbedingungen in die geforderte Gleichheitsbedingung umzuformulieren, bedient man sich sogenannter **Schlupfvariablen** ξ

$$Ax - b \geq 0 \qquad \rightarrow \qquad Ax - \xi - b = 0,\ \xi \geq 0$$

- Durch Aufteilen von $x = x^+ - x^-$ in den negativen und den positiven Teil, lässt sich das Problem in die Standardform erweitern

$$\min_x \begin{pmatrix} c \\ -c \\ 0 \end{pmatrix}^\top \begin{pmatrix} x^+ \\ x^- \\ \xi \end{pmatrix}, \qquad \begin{pmatrix} A & -A & -I \end{pmatrix} \begin{pmatrix} x^+ \\ x^- \\ \xi \end{pmatrix} = b,\ \overset{x}{\boxed{\begin{pmatrix} x^+ \\ x^- \\ \xi \end{pmatrix}}} \geq 0$$

Optimalitätsbedingungen

- Lagrange-Funktion

  $$\mathcal{L}(x, \lambda, s) = c^\top x - \lambda^\top (Ax - b) - s^\top x$$

  mit separaten Lagrange-Multiplikatoren λ und s

- Karush-Kuhn-Tucker Bedingungen (siehe letzte Vorlesung):

  $$A^\top \lambda + s = c \qquad (\nabla_x \mathcal{L} = 0)$$

  $$Ax = b$$

  $$x \geq 0$$

  $$s \geq 0$$

  $$x^\top s = 0 \qquad \text{(Komplementaritätsbedingung)}$$

- Da das Problem konvex ist, sind diese notwendigen Bedingungen auch <u>hinreichend</u>.

Duales Problem

- Das duale Problem lautet

$$\max_{\lambda} b^\top \lambda, \quad A^\top \lambda + s = c, \; s \geq 0$$

- Für lineare Programme ist die Beziehung zwischen dem primalen und dem dualen Problem besonders stark ausgeprägt.

- Unter Ausnutzung der KKT Bedingungen können wir z.B. im Optimum folgende Umformung vornehmen:

$$\begin{aligned} c^\top x &= (A^\top \lambda + s)^\top x && \text{(Verwendung der 1. KKT Bedingung)} \\ &= (Ax)^\top \lambda + s^\top x \\ &= (Ax)^\top \lambda && \text{(verwende } x^\top s = 0 \text{)} \\ &= b^\top \lambda && \text{(verwende } Ax = b \text{)} \end{aligned}$$

d.h. die Zielfunktionen nehmen im Optimum den gleichen Wert an (starke Dualität)

Starke Dualität und Symmetrie

- Starke Dualität ist keine Überraschung, da lineare Programme die entsprechenden Bedingungen aus der letzten Vorlesung erfüllen.

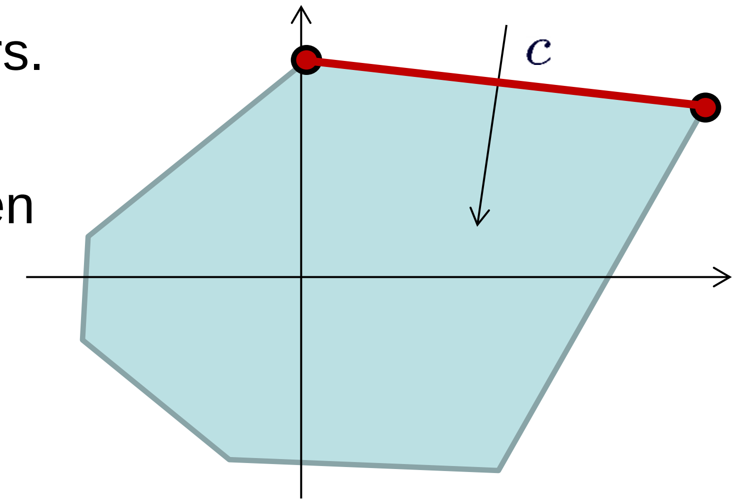
- Es können folgende Aussagen über die Lösbarkeit gemacht werden:
  - Wenn eines der Probleme lösbar ist, ist auch das andere lösbar (direkte Konsequenz der starken Dualität)
  - Wenn das eine Problem unbeschränkt ist, ist das andere ungültig.

- Außerdem ist die Primal-Dual-Beziehung bei linearen Programmen <u>symmetrisch</u>:

  Das duale Problem des dualen Problems ist wieder das primale Problem.

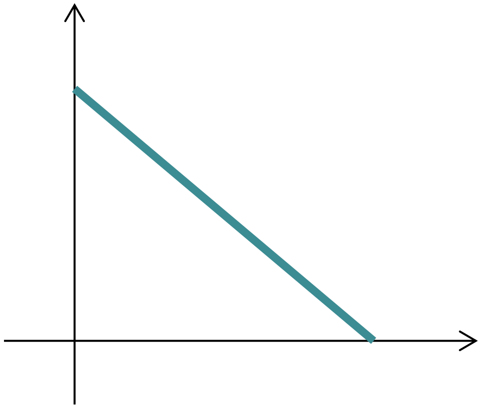
- Entsprechend sind die optimalen Variablen des einen Problems gerade die optimalen Lagrange-Multiplikatoren des anderen Problems.

Geometrie linearer Programme



- Optima liegen immer auf Ecken des Polyeders.
- Mehrere Optima sind möglich. Optimale Ecken sind dabei immer in der Menge der Optima enthalten.
- Grundlage des Simplex-Verfahrens:
  Ausgehend von einer Ecke, gehe zu einer benachbarten Ecke mit niedrigerer Energie.
- Vorsicht! Die bildliche Anschauung ändert sich durch Verwendung der Standardform.
- Polyeder → Polytop in höherdimensionalem Raum

  Jede Schlupfvariable ξ vergrößert die Dimension des Lösungsraums.



Simplex-Verfahren: Problemstruktur

- Lagrange-Funktion enthält die primalen und dualen Variablen

$$\mathcal{L}(x, \lambda, s) = c^\top x - \lambda^\top (Ax - b) - s^\top x \qquad x, s \in \mathbb{R}^n,\ \lambda \in \mathbb{R}^m$$

- Von zentraler Bedeutung ist die Matrix $A \in \mathbb{R}^{m \times n}$

- Anzahl der Zeilen m : Anzahl der Gleichheitsbedingungen
- Anzahl Spalten n : Dimension von x (inkl. Schlupfvariablen)

- Annahme: $n > m$ und der Rang von A ist m

- Ist dies nicht erfüllt, heißt das:
  - Die zulässige Menge G ist ein einzelner Punkt oder leer (warum?)

    oder

  - Nebenbedingungen sind linear abhängig. Dem können wir durch Eliminierung entsprechender Gleichungen abhelfen.

Indexmengen und Basislösungen

- Das Simplex-Verfahren gehört zu den **Active-Set-Verfahren**.

- In jedem Schritt teilt es die Variablen x_i in die **inaktive Menge (Basis)** $\mathcal{B}$ und die **aktive Menge** $\mathcal{N} = \{1, ..., n\} - \mathcal{B}$ auf.

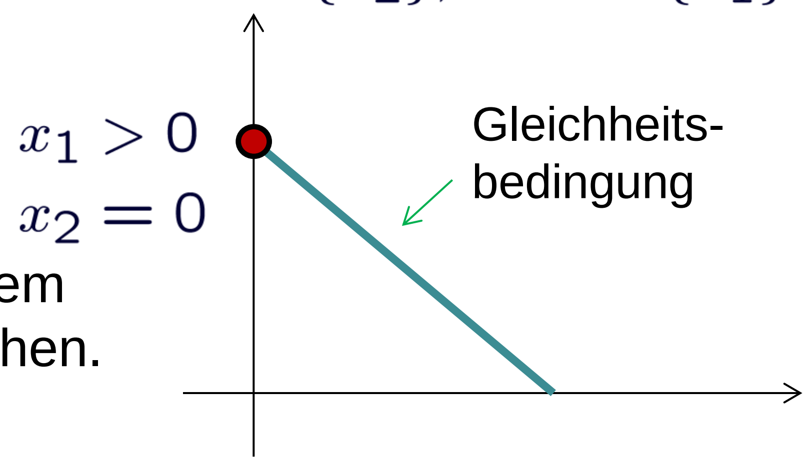
- Die Basis $\mathcal{B}$ enthält immer genau m Indizes. Für die entsprechenden x_i muss die Gleichheitsbedingung $Ax = b$ erfüllt sein.

- Aus $i \in \mathcal{N}$ folgt hingegen $x_i = 0$ (die Bedingung $x_i \geq 0$ ist aktiv)

- Die entsprechende Lösung liegt immer auf den Ecken des Polytops und nennt sich **Basislösung**.

- Wir suchen die <u>optimale</u> Basislösung, indem wir Indizes zwischen $\mathcal{B}$ und $\mathcal{N}$ austauschen.

$$\mathcal{N} = \{x_2\}, \quad \mathcal{B} = \{x_1\}$$



Zerlegung der Matrix

- Die Mengen $\mathcal{B}$ und $\mathcal{N}$ zerlegen entsprechend auch die Matrix A (Beispiel)

$$A = \begin{array}{c} \begin{array}{ccccc} \phantom{a_{11}} & \phantom{a_{12}} & x_3 = 0 & \phantom{a_{14}} & x_5 = 0 \end{array} \\ \begin{pmatrix} a_{11} & a_{12} & a_{13} & a_{14} & a_{15} \\ a_{21} & a_{22} & a_{23} & a_{24} & a_{25} \\ a_{31} & a_{32} & a_{33} & a_{34} & a_{35} \end{pmatrix} \end{array}$$

- Wir erhalten eine invertierbare Matrix $B = \begin{pmatrix} a_{11} & a_{12} & a_{14} \\ a_{21} & a_{22} & a_{24} \\ a_{31} & a_{32} & a_{34} \end{pmatrix}$

  sowie eine Matrix $N = \begin{pmatrix} a_{13} & a_{15} \\ a_{23} & a_{25} \\ a_{33} & a_{35} \end{pmatrix}$

Bestimmung der Variablen

- Bestimmung der primalen Variablen $x = (x_B, x_N)$

  Die Gleichheitsbedingungen müssen erfüllt sein:

  $$Ax = Bx_B + Nx_N = b$$

  $x_N = 0$ daher $x_B = B^{-1}b$

- Bestimmung der dualen Variablen $\lambda,\ s = (s_B, s_N)$

  Wir haben $s_B = 0$ (Komplementarität)

  Aus der KKT-Bedingung $A^\top \lambda + s = c$ erhalten wir durch Aufteilung

  $B^\top \lambda = c_B$ und $N^\top \lambda + s_N = c_N$

  Damit erhalten wir:

  $\lambda = B^{-\top} c_B$ $\qquad s_N = c_N - N^\top \lambda$

Welche der KKT-Bedingungen sind erfüllt?

- Die KKT-Bedingungen sind nur im Optimum alle erfüllt. Noch sind wir nicht im Optimum. Welche der Bedingungen sind also verletzt?

$$
\begin{aligned}
A^\top \lambda + s &= c \\
Ax &= b \\
x &\geq 0 \qquad \text{(Start mit Basislösung, daher per Definition erfüllt)} \\
\boxed{s \geq 0} \\
x^\top s &= 0
\end{aligned}
$$

- Falls alle $s_N \geq 0$, sind alle Optimalitätsbedingungen erfüllt
  → Abbruchkriterium

- Andernfalls ist jeder Index $j \in \mathcal{N}$ für den $s_j < 0$ ein Kandidat, um einen verbesserten Eckpunkt zu generieren.

- Einfachste Strategie: wähle q, so dass s_q minimal ist.

Wechsel zu neuer Basislösung

- Wir kennen nun also den Index $q \in \mathcal{N}$, den wir in die Basis neu aufnehmen möchten.

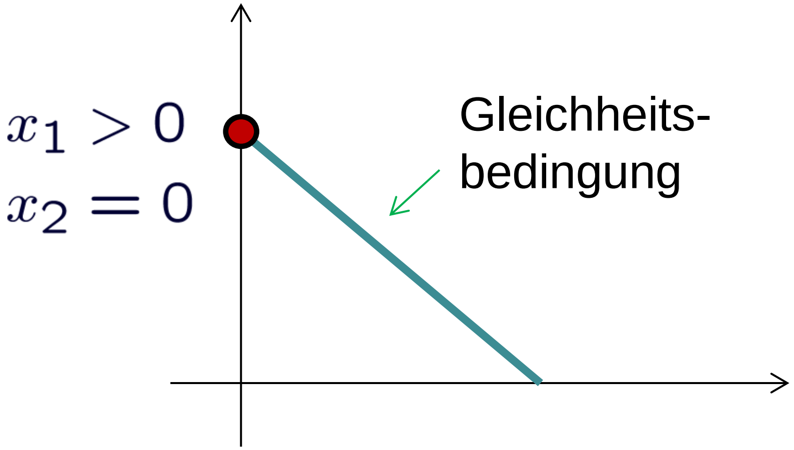
- Da die Basis immer aus genau m Elementen besteht, muss ein anderer Index $p \in \mathcal{B}$ nach $\mathcal{N}$ wechseln.

- Um herauszufinden welcher, erhöhen wir x_q (das bisher 0 war) unter Einhaltung der Gleichheitsbedingungen.

- Dabei erreicht irgendwann ein $x_i, i \in \mathcal{B}$ den Wert 0.
  → Setze $p = i$ und entferne das Element aus der Basis

- Falls dies nicht passiert, ist das Problem unbeschränkt.

$$\mathcal{N} = \{x_2\}, \quad \mathcal{B} = \{x_1\}$$



Zusammenfassung des Simplex-Verfahrens

1. Finde eine initiale Basislösung

2. Berechne alle primalen und dualen Variablen für diese Basislösung

3. Solange nicht alle $s_N \geq 0$:
   - Wähle q mit $s_q < 0$
   - Erhöhe x_q unter Einhaltung von $Ax = b$ bis $x_p = 0$ für ein $p \in \mathcal{B}$ (Pivoting)
   - Entferne p aus der Basis
   - Update aller Variablen für die neue Basis

- Noch offen:
  - Wie finden wir eine initiale Basislösung?
  - Wie funktioniert Pivoting genau?
  - Wie berechnen wir ein effizientes Update von B^{-1}?
  - Was passiert im Fall einer **degenerierten Basis** $(x_i = s_i = 0,\ i \in \mathcal{B})$?

Finden einer initialen Basislösung

- Um eine initiale Basislösung (Ecke) zu finden, müssen wir ein (einfacheres) lineares Programm lösen:

  $$\min_z e^\top z, \qquad Ax + Ez = b,\ x \geq 0,\ z \geq 0$$

  mit der Diagonalmatrix E mit Einträgen $E_{ii} = \mathrm{sign} b_i$ und $e := (1, ..., 1)^\top$

- Eine Basislösung für dieses lineare Programm ist einfach zu sehen:

  $$x = 0, \quad z_i = |b_i|$$

  d.h. x wird zunächst auf 0 gesetzt und die Gleichheitsbedingung mithilfe der neuen Variablen z_i erfüllt.

- Indem die Summe der z_i minimiert wird (Minimum bei 0), muss sich wieder x um die Gleichheitsbedingung kümmern.

- Im Minimum haben wir eine Basislösung für das eigentliche Problem.

Pivoting

- $Ax = b$ muss weiter eingehalten werden, d.h. beim Erhöhen von x_q müssen einige der $x_i, i \in \mathcal{B}$ geeignet reduziert werden.

- Vor und nach der Änderung muss gelten:

$$Ax = Bx_B = Bx'_B + \boxed{A_q} x'_q$$

$q$te Spalte von A

- Durch Umformung erhalten wir also die notwendige Änderung von x_B beim Erhöhen von x_q :

$$x'_B = x_B - B^{-1} A_q x'_q \qquad d := B^{-1} A_q$$

- Mit dieser Information finden wir das kritische Element p

$$p = \operatorname{argmin}_{i \in \mathcal{B}} \frac{x_i}{d_i}$$

sowie $x'_q = \dfrac{x_p}{d_p}$

Update der Matrix

- Die aufwendigste Operation ist das Invertieren der Matrix B in den Operationen

$$B^\top \lambda = c_B \qquad\qquad Bd = A_q$$

- Nach dem Update der Basis möchten wir B nicht erneut invertieren.

- Besser: Faktorisierung von B in einer untere und obere Dreiecksmatrix:

$$B = LU$$

- L und U können nach dem Basiswechsel effizient aktualisiert werden (siehe Nocedal-Wright pp. 372)

Degenerierte Probleme

- Wir gingen bisher immer davon aus, dass $x_i > 0, \ \forall i \in \mathcal{B}$

- Es kann aber auch sein, dass $x_i = 0, \ i \in \mathcal{B}$

- Eine Basis mit einem solchen Element heißt **degeneriert**. Das Gesamtproblem heißt degeneriert, wenn es eine degenerierte Basis gibt.

- <u>Problem:</u> Beim Austausch eines degenerierten Basiselements wird die Zielfunktion möglicherweise nicht reduziert, nämlich dann wenn beim Pivoting $x_i = 0$ und $d_i \neq 0$.

- Der Basiswechsel sorgt dafür, dass das Verfahren dennoch terminiert, denn mit der anderen Basis gilt eventuell $x_i = 0$ und $d_i = 0$.

- Es muss jedoch sichergestellt werden, dass das Verfahren nicht zweimal die gleiche degenerierte Basis wählt (Zyklus).

Komplexität

- Das Simplex-Verfahren ist in fast allen Fällen sehr effizient (ca. $2m$ Iterationen)
- Die Worst-Case-Komplexität ist jedoch exponentiell in der Anzahl der Dimensionen n
- Dies gilt auch für alle anderen Varianten des Algorithmus, die bisher bekannt sind.
- **Innere-Punkt-Methoden** liefern bessere Garantien und sind bei sehr großen Problemen üblicherweise effizienter.
- Bei kleinen und mittelgroßen linearen Programmen ist der Simplex-Algorithmus im Normalfall die beste Wahl.

Zusammenfassung

- Lineare Programme lassen sich effizient mit dem Simplex-Algorithmus optimieren.
- Mindestens eines der globalen Optima ist eine Ecke des zulässigen Polyeders.
- Das Simplex-Verfahren wandert die Ecken des Polyeders ab, bis es die optimale Ecke gefunden hat.
- Die primalen und dualen Variablen werden dabei gleichzeitig optimiert.
- Das Verfahren unterscheidet zwischen der aktiven und inaktiven Menge der Ungleichheitsbedingungen.
- Ein Simplexschritt tauscht jeweils zwei Elemente zwischen diesen beiden Mengen aus.

Literatur

- J. Nocedal, S. J. Wright: Numerical Optimization, Springer, 2006.
  Kapitel 13 behandelt den Simplex-Algorithmus im Detail.

Übungsaufgabe

1. Ein Server hat einen Plattendurchsatz von 100MB/s, eine Netzwerkkapazität von 1000MBit/s und einen Hauptspeicher mit 8000MB. Er soll Jobs dreier verschiedener Arten verarbeiten. Jobs vom Typ 1 haben die höchste Priorität von 20 und benötigen 8MB/s Plattendurchsatz, 50MBit/s vom Netzwerk und 50MB Speicher (wir unterscheiden hier nicht zwischen mittlerer und Peakauslastung). Von Typ 1 stehen 10 in der Queue. Die 5 Jobs vom Typ 2 haben Priorität 8 und benötigen 5MB/s, 2MBit/s und 800MB. Außerdem gibt es noch 50 Jobs vom Typ 3 mit Priorität 2, welche 2MB/s Platte und 40MB Speicher benötigen. Alle Jobs dauern gleich lange. Wir möchten die Jobs so auswählen, dass die Summe der Prioritäten maximal wird. Formulieren Sie die Aufgabe als lineares Programm.

2. Bringen Sie die Formulierung in die Standardform.

3. Lösen Sie das Problem mit Scipy `linprog` unter Verwendung des Simplex-Algorithmus.