Übungsblatt 5 (mit Lösung)

Abgabe bis 26.05.2011 13:45 Uhr, Erdgeschoss Geb. 48

Vorlesung: Dr. Florentine Bunke, Jun.-Prof. Dr. Stefan Ruzika
Übungen: Michael Helmling, Florian Seipp

Dieses Übungsblatt sowie weitere Informationen zur Übung sind unter `http://optimierung.mathematik.uni-kl.de/teaching/ss11/prama.html` erhältlich.

Aufgabe 5.1 (4 Punkte)

Gegeben sei das folgende lineare Programm:

$$
\text{(LP)}\quad
\begin{array}{lrcrcrcrcl}
\min & 2x_1 & + & 3x_2 & + & 5x_3 & + & 6x_4 & & \\
\text{s.t.} & x_1 & + & 2x_2 & + & 3x_3 & + & x_4 & \geqslant & 2 \\
 & -x_1 & + & x_2 & - & x_3 & + & 3x_4 & \leqslant & -3 \\
 & & & & & & & x_1, x_2, x_3, x_4 & \geqslant & 0
\end{array}
$$

a) Stellen Sie das zu (LP) duale Problem (DLP) auf. Lösen Sie dieses mit einem Verfahren Ihrer Wahl (grafisch, Tableau, GLPK, Matlab). Benutzen Sie dann den Satz vom komplementären Schlupf, um eine primale Optimallösung für (LP) zu finden.

b) Lösen Sie (LP) mit dem dualen Simplex (mit den Schlupfvariablen als Startbasis).

Hinweis: Wenn Sie richtig rechnen, sollte in beiden Teilaufgaben das selbe Ergebnis herauskommen!

Lösung 5.1 (2+2 Punkte Punkte):

a) Zum Dualisieren kann man die 2. Ungleichung mit (-1) multiplizieren, dann erhält man als duale Variablen $\pi_1, \pi_2 \geqslant 0$ und (DLP) ist

$$
\begin{aligned}
\max\ & 2\pi_1 + 3\pi_2 \\
\text{s. t.}\ & \pi_1 + \pi_2 \leqslant 2 \\
& 2\pi_1 - \pi_2 \leqslant 3 \\
& 3\pi_1 + \pi_2 \leqslant 5 \\
& \pi_1 - 3\pi_2 \leqslant 6 \\
& \underline{\pi} \geqslant \underline{0}
\end{aligned}
$$

Die Optimallösung von (DLP) ist $\underline{\hat{\pi}} = (0,2)^T$ mit Zielfunktionswert 6. Also hat auch (LP) eine optimale Lösung $\underline{\hat{x}}$ mit ZFW 6. Diese bestimmen wir nun mit dem Satz vom komplementären Schlupf: $\underline{\hat{\pi}}$ erfüllt nur die erste Ungleichung von (DLP) mit Gleichheit, also kann nur $\hat{x}_1 \neq 0$ sein. Da umgekehrt $\hat{\pi}_2 \neq 0$ ist, muss die zweite Ungleichung von (LP) mit Gleichheit erfüllt sein, also gilt $\hat{x}_1 = 3$ und die Optimallösung ist $\underline{\hat{x}} = (3, 0, 0, 0)$.

b) Jetzt empfiehlt es sich, die erste Gleichung von (LP) mit (-1) zu multiplizieren. Dann ist das Starttableau

$$\begin{array}{rrrrrr|r} 2 & 3 & 5 & 6 & 0 & 0 & 0 \\ \hline -1 & -2 & -3 & -1 & 1 & 0 & -2 \\ -1 & 1 & -1 & 3 & 0 & 1 & -3 \end{array}$$

und nach einer Pivotoperation (mit dem Element $(t_{2,1})$) erhalten wir das optimale Tableau

$$\begin{array}{rrrrrr|r} 0 & 5 & 3 & 0 & 0 & 2 & -6 \\ \hline 0 & -3 & -2 & -4 & 1 & -1 & 5 \\ 1 & -1 & 1 & -3 & 0 & -1 & 3 \end{array}$$

Aufgabe 5.2 (4 Punkte)

Sei $A \in \mathbb{R}^{m \times n}$ und $\underline{b} \in \mathbb{R}^m$. Das *Lemma von Farkas* besagt, dass genau eines der beiden folgenden Systeme eine Lösung ($\underline{x} \in \mathbb{R}^n$ bzw. $\underline{y} \in \mathbb{R}^m$) besitzt:

$$(1)\text{: } A\underline{x} = \underline{b},\ \underline{x} \geqslant \underline{0} \qquad (2)\text{: } \underline{b}^T\underline{y} < 0,\ A^T\underline{y} \geqslant \underline{0}$$

a) Beweisen Sie das Lemma von Farkas, indem Sie ein geeignetes LP aufstellen und den starken Dualitätssatz anwenden.

b) Finden Sie eine geometrische Interpretation des Lemmas. Verwenden Sie dazu folgende Definition: Seien $\underline{a}^1, \ldots, \underline{a}^k \in \mathbb{R}^n$ gegeben, dann heißt

$$\operatorname{cone}(\underline{a}^1, \ldots, \underline{a}^k) := \left\{ \sum_{i=0}^{k} \lambda_i \underline{a}^i : \lambda_i \geqslant 0 \text{ für } i = 1, \ldots, n \right\}$$

der von $\underline{a}^1, \ldots, \underline{a}^k$ aufgespannte *Kegel*. Skizzieren Sie für $\underline{y} \in \mathbb{R}^2$ ein Beispiel.
Hinweis: Was bedeuten die beiden Aussagen für die Lage von $\underline{b}$ zu den *Spaltenvektoren* von A?

Lösung 5.2:

a) Betrachte das Problem

$$\begin{array}{lll} \text{(LP)} & \max & \underline{0}\,\underline{x} \\ & \text{s.t.} & A\underline{x} = \underline{b} \\ & & \underline{x} \geqslant \underline{0} \end{array}$$

mit dem dualen Problem

$$\begin{aligned} \text{(DLP)} \quad & \min \underline{b}^T \underline{y} \\ & \text{s. t.} \quad A^T \underline{y} \geqslant \underline{0} \end{aligned}$$

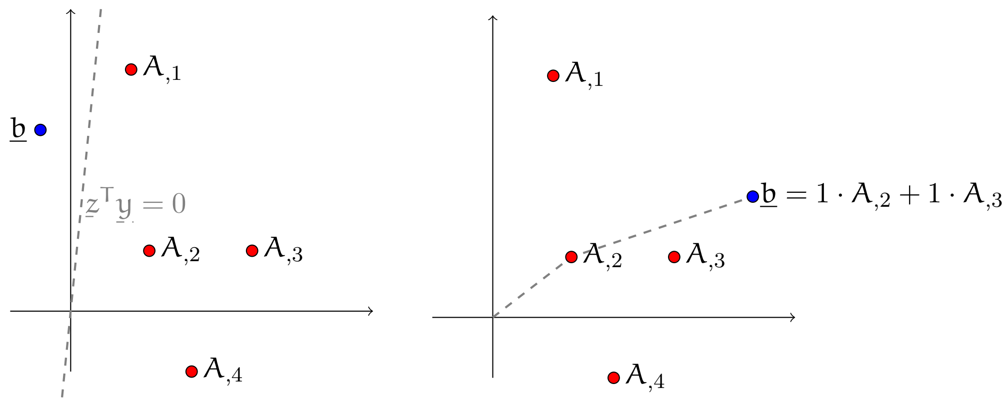
Angenommen (1) ist lösbar, dann ist (LP) zulässig und (da die Zielfunktion immer 0 ist) auch beschränkt, hat also eine Optimallösung mit ZFW 0. Nach dem Dualitätssatz ist dann auch (DLP) optimal lösbar mit ZFW 0, also existiert kein $\underline{y}$ das (2) erfüllt.

Sei umgekehrt (2) nicht lösbar. Damit ist $\underline{y} = \underline{0}$ eine Optimallösung für (DLP). Nach dem Dualitätssatz hat dann auch (LP) eine Optimallösung, d.h. (1) ist lösbar.

Wir haben also gezeigt: (1) lösbar $\Longleftrightarrow$ (2) nicht lösbar; mit anderen Worten: Genau eines der Systeme ist lösbar.

b) Eine Lösung $\underline{x}$ von (1) bedeutet, dass $\underline{b}$ im durch die Spalten von A aufgespannten Kegel liegt: $\underline{b} = \sum_{i=1}^{n} x_i A_{,i}$ mit $x_i \geqslant 0$ für $i = 1, \ldots, n$ ($\underline{x}$ entspricht also dem $\underline{\lambda}$ in der Definition des Kegels).

Wenn umgekehrt (2) lösbar ist, gibt es eine Hyperebene, definiert durch $H_{\underline{y}} := \{\underline{z} \in \mathbb{R}^m : \underline{z}^T \underline{y} = 0\}$, die $\underline{b}$ von den Spalten von A trennt: $\underline{b}$ liegt auf der einen Seite ($\underline{b}^T \underline{y} < 0$) und alle Spalten von A auf der anderen ($(A^T \underline{y})_j = A_{,j}^T \underline{y} \geqslant 0$). Ein Bild für $n = 4$, $m = 2$:



Aufgabe 5.3 (4 Punkte)

Auch wenn der Simplex-Algorithmus in den meisten Fällen sehr schnell ist und deshalb auch in der Praxis eingesetzt wird, ist seine Worst-Case-Laufzeit exponentiell. Auf diesem und dem nächsten Übungsblatt soll dies anhand des sogenannten *Klee-Minty-Würfels* gezeigt werden. Dabei handelt es sich um einen leicht verzerrten Hyperwürfel mit 2^n Eckpunkten, die im ungünstigsten Fall alle vom Simplex durchlaufen werden.

Es sei $0 < \varepsilon < \frac{1}{2}$ und $n \in \mathbb{N}$. Wir betrachten das LP

$$\max \{x_n \mid \varepsilon x_{i-1} \leqslant x_i \leqslant 1 - \varepsilon x_{i-1} \text{ für } i = 1, \ldots, n \text{ und } \underline{x} \geqslant \underline{0}\},$$

wobei $x_0 = 1$ festgesetzt (keine Variable) ist. Beachten Sie, dass der zulässige Bereich bei $\varepsilon = 0$ genau der Einheitswürfel wäre.

a) Bringen Sie das LP in Standardform, indem Sie für die Ungleichungen auf der linken Seite Schlupfvariablen r_i, für die auf der rechten Seite Schlupfvariablen s_i einführen.

b) Zeigen Sie, dass jede zulässige Basis alle x_i und für jedes $i \in \{1, \dots, n\}$ entweder r_i oder s_i enthält. Ist das Problem degeneriert?
Hinweis: Verwenden Sie, dass eine Variable, die nicht 0 sein kann, in jeder Basis enthalten sein muss, und dass die Basis so viele Elemente wie das LP Zeilen haben muss.

c) Die vorherige Teilaufgabe zeigt, dass in jeder Basislösung x jedes x_i entweder den minimalen $(r_i = 0)$ oder maximalen $(s_i = 0)$ Wert annimmt und dass die Basis dadurch bereits vollständig definiert ist – es existiert also eine 1-zu-1-Beziehung zwischen den Basislösungen und den Teilmengen $L \subseteq N := \{1, \dots, n\}$ durch die Festlegung $i \in L \Longleftrightarrow s_i = 0$. Wir erhalten dann für jedes solche L durch

$$J^L := \{x_1, \dots, x_n\} \cup \{r_i : i \in L\} \cup \{s_i : i \in N \setminus L\}$$

eine zulässige Basis. Sei x^L die zugehörige Basislösung. Zeigen Sie, dass für $L, L' \subseteq N$ mit $n \in L$ und $n \notin L'$ gilt: $x_n^{L'} < x_n^L$, d.h. dass jede Basis, die r_n enthält, einen echt größeren Zielfunktionswert hat als alle Basen, die r_n nicht enthalten. Zeigen Sie weiter, dass für $L' := L \setminus \{n\}$ gilt: $x_n^{L'} = 1 - x_n^L$.

Lösung 5.3:

a) Standardform:

$$\begin{aligned}
-\min\ & -x_n & \\
\text{s.t.}\ & x_1 + s_1 = 1 - \varepsilon & \\
& x_1 - r_1 = \varepsilon & \\
& \varepsilon x_{i-1} + x_i + s_i = 1 & i = 2, \dots, n \\
& -\varepsilon x_{i-1} + x_i - r_i = 0 & i = 2, \dots, n \\
& r_i, s_i \geqslant 0 & i = 1, \dots, n
\end{aligned}$$

b) Wir zeigen zunächst per Induktion, dass alle $x_i > 0$ sind. Induktionsanfang $(n = 1)$: $x_1 > \varepsilon x_0 = \varepsilon > 0$. Induktionsschluss $(n - 1 \to n)$: x_n erfüllt $\varepsilon x_{n-1} \leqslant x_n$, also $x_n > 0$ da $\varepsilon > 0$ und $x_{n-1} > 0$ nach Induktion.

Nun zeigen wir (ohne Induktion), dass für kein i sowohl $r_i = 0$ als auch $s_i = 0$ sein kann. Für $i = 1$: Falls $s_1 = r_1 = 0$ lassen sich die ersten zwei Gleichungen in der Standardform zusammenfassen zu $\varepsilon = 1 - \varepsilon$, was wegen $\varepsilon < \frac{1}{2}$ ein Widerspruch ist. Für $i > 1$ kann man unter der Annahme $s_i = r_i = 0$ analog die dritte und vierte Gleichung für dieses i zusammenfassen und erhält $1 - \varepsilon x_{i-1} = \varepsilon x_{i-1}$ bzw. $\frac{1}{2} = \varepsilon x_{i-1}$, was wegen $\varepsilon < \frac{1}{2}$

und $x_{i-1} < 1$ (was offensichtlich gilt) ein Widerspruch ist. Also sind alle x_i und jeweils mindestens eines von $\{r_i, s_i\}$ echt größer als 0. Da das LP $2n$ Nebenbedingungen hat, enthält jede Basis $2n$ Variablen, also können auch für kein i sowohl r_i als auch s_i in der Basis enthalten sein $\Leftrightarrow$ jede Basis enthält alle x_i und *genau* eine der Variablen $\{r_i, s_i\}$ für jedes $i = 1, \ldots, n$. Das Problem ist nicht degeneriert: wir haben gezeigt dass alle Basisvariablen > 0 sein müssen.

c) Sei $n \in L \Leftrightarrow r_n^L > 0$ und $s_n^L = 0 \Leftrightarrow x_n^L = 1 - \underbrace{\varepsilon x_{n-1}^L}_{<\frac{1}{2}} > \frac{1}{2}$. (1)

Sei $n \notin L' \Leftrightarrow r_n^{L'} = 0$ und $s_n^{L'} > 0 \Leftrightarrow x_n^{L'} = \underbrace{\varepsilon x_{n-1}^{L'}}_{<\frac{1}{2}} < \frac{1}{2}$. (2)

$\Leftrightarrow x_n^{L'} < x_n^L$.

Sei nun $L' = L \setminus \{n\}$. Dann ist

$$x_n^{L'} \overset{(2)}{=} \varepsilon x_{n-1}^{L'} = \varepsilon x_{n-1}^L \overset{(1)}{=} 1 - x_n^L,$$

wobei die mittlere Gleichung gilt da L und L' eingeschränkt auf $\{1, \ldots, n-1\}$ übereinstimmen.

Aufgabe 5.4 – Praktische Aufgabe (4 Punkte)

a) Implementieren Sie eine Matlab-/Octave-Funktion

```
function [A_Ph1, b_Ph1, c_Ph1, B_Ph1 ] = InitPhase1 (A, b, c)
```

die die Eingabedaten eines LP in Standardform so transformiert (also ggf. künstliche Variablen einfügt, die Zielfunktion ersetzt etc.), dass Phase 1 des Simplex direkt angewendet werden kann. Dabei ist `B_Ph1` $\in \mathbb{N}^m$ der Indexvektor der Startbasis für Phase 1.

b) Benutzen Sie Ihre bisherige Programme, um eine Funktion

```
function [EndTab, x_opt, z_opt] = TwoPhaseMethod(A,b,c, rule)
```

zu implementieren, die jedes beliebige LP in Standardform löst. Der Parameter `rule` soll sich so verhalten wie zuvor. Das Verhalten bei unbeschränkten LPs soll dem der Funktion `simplex` auf dem letzten Blatt entsprechen. Ist das LP unzulässig, soll `x_opt = z_opt = []` gelten.
Hinweis: Ändern Sie vorher Ihre Funktion `simplex` so ab, dass auch die optimale Basis `B_opt` ausgegeben wird.

Lösung 5.4:

a)

```
% ==== InitPhase1.m ===================================
function [A_Ph1,b_Ph1,c_Ph1,B_Ph1] = InitPhase1(A,b,c)

% Initialisierungen
[m n] = size(A);
A_Ph1 = A;
b_Ph1 = b;
B_Ph1 = zeros(1,m); % die Startbasis für Phase 1
c_Ph1 = zeros(1,n);
ev = []; % bereits gefundene oder konstruierte Einheitsvektoren

% Schritt 1 a), multipliziere alle Zeilen, fuer die b(i) < 0 ist, mit -1
for i=1:m
    if(b_Ph1(i) < 0)
        A_Ph1(i,:) = (-1)*A_Ph1(i,:);
        b_Ph1(i) = (-1)*b_Ph1(i);
    end
end

% Schritt 1 b), suche nach Variablen, die in nur einer Gleichung (mit
% positivem Koeffizienten) vorkommen
for j=1:n
    if (sum(A_Ph1(:,j) > 0) == 1) && (sum(A_Ph1(:,j) < 0) == 0)
        %Finde die Stelle an der ein Eintrag ungleich null ist und
        %speichere seinen Wert
        [wert i] = max(A_Ph1(:,j));

        if sum(ev == i) == 0 % noch kein i-ter Einheitsvektor vorhanden
            % Normiere den entsprechenden Eintrag und passe die zugehörige
            % Zeile von A_Ph1 und den Eintrag in b_Ph1 an
            b_Ph1(i) = b_Ph1(i) / wert;
            A_Ph1(i,:) = A_Ph1(i,:) ./ wert;
            % Die zu der gerade gefundenen Spalte gehörige Variable ist
            % Basisvariable => nimm j in B_Ph1 auf
            B_Ph1(i) = j;
            % Fuer Schritt 1c) wird sich hier noch gemerkt, welche
            % Einheitsvektoren schon erzeugt wurden.
            ev = [ev i];
        end
    end
end
% Schritt 1c), künstliche Variablen für die fehlenden Einheitsvektoren erzeugen
k = n+1;
for i = 1:m
    if sum(ev == i) == 0
        % Einheitsvektor ergänzen
        A_Ph1(:,k) = eye(m)(:,i);
        % jede kuenstliche Variable ist am Anfang Basisvariable
        B_Ph1(i) = k;
        c_Ph1(k) = 1;
        k = k+1;
    end
end
end % Ende der Funktion
```

b)

```
% ==== TwoPhaseMethod.m ===================================
function [EndTab x_opt z_opt] = TwoPhaseMethod(A,b,c, rule)

[m n] = size(A);
if (nargin < 4) % keine Regel angegeben -> Standardregel von Blatt3
    rule = 'blatt3';
end
% Umwandeln der Eingabedaten auf das für Phase 1 nötige Format
[A_Ph1,b_Ph1,c_Ph1,B_Ph1] = InitPhase1(A,b,c);

% Phase 1: Minimiere die Hilfszielfunktion (beachte: Funktion 'simplex'
% wurde so abgeändert, dass sie auch die optimale Basis zurückgibt)
[Tab_Ph1 x_Ph1 zfw_Ph1 B_Ph2] = simplex(A_Ph1,b_Ph1,c_Ph1,B_Ph1, rule);

% Überprüfen, ob der Wert der Hilfszielfunktion null ist
if zfw_Ph1 > 1e-10 % wg. numerischer Ungenauigkeit

    % Zielwert ungleich null, das Ausgangsproblem war unzulaessig
    EndTab = Tab_Ph1;
    x_opt = [];
    z_opt = [];
    disp('Das␣Problem␣ist␣unzulaessig!')

else
```

```
    % Zielwert ist null, überprüfe, ob sich noch künstliche Variablen
    % in der Basis befinden und pivotiere diese ggf. heraus,
    % um eine zulässige Startbasis für Phase 2 zu erhalten
    while sum(B_Ph2 > n) > 0

        % künstliche Basisvariable finden und Pivotzeile bestimmen
        [dummy z] = max(Tab_Ph1(2:m+1,max(B_Ph2)+1));
        % Pivotspalte wählen, so dass eine echte Variable in die Basis
        % eintritt
        [dummy s] = max(abs(Tab_Ph1(z+1,2:n+1)));
        % Pivotoperation durchführen
        [Tab_Ph1, B_Ph2] = pivot_operation(z,s,Tab_Ph1,B_Ph2);
    end

    %lese aus dem Endtableau von Phase 1 die nötigen Daten aus
    A_Ph2 = Tab_Ph1(2:m+1,2:n+1);
    b_Ph2 = Tab_Ph1(2:m+1,size(Tab_Ph1,2));

    % Phase 2: Simplex mit Original-Zielfunktion
    [EndTab x_opt z_opt] = simplex(A_Ph2,b_Ph2,c,B_Ph2, rule);
end
end
```