ulm university universität uulm

P R O B E K L A U S U R D E C K B L A T T

Name der Prüfung: **Optimierung/OR I**

Datum und Uhrzeit : 10.07.2013, 14:00 Uhr

Bearbeitungszeit: 90 min

Institut: Optimierung und Operations Research

Prüfer: Dr. Jens Maßberg

Vom Prüfungsteilnehmer auszufüllen:

Name: ____________________ Vorname: ____________________

Studiengang: ____________________ Abschluss: ____________________

Matrikelnummer: ______________

____

Datum, Unterschrift des Prüfungsteilnehmers

Hiermit erkläre ich, dass ich prüfungsfähig bin. Sollte ich nicht auf der Liste der angemeldeten Studierenden aufgeführt sein, dann nehme ich hiermit zur Kenntnis, dass diese Prüfung nicht gewertet werden wird.

Hinweise zur Prüfung:

- Für jede der 5 Aufgaben gibt es 10 Punkte.
- Gewertet werden die 4 besten Aufgaben.

Erlaubte Hilfsmittel:

- Schreibzeug (kein rotfarbiger Stift!)
- Lineal / Geodreieck

Viel Erfolg!

Bitte dieses Feld für den Barcode freilassen

Vom Prüfer auszufüllen:

Aufgabe 1	Aufgabe 2	Aufgabe 3	Aufgabe 4	Aufgabe 5	Summe - Min

Gesamtnote: ____________________

____

Unterschrift Prüfer

Dr. Jens Maßberg
Dr. Christian Löwenstein

08.07.2013

Institut für Optimierung und Operations Research

Probeklausur Optimierung I

Aufgabe 1. Gegeben sei das folgende lineare Optimierungsproblem (P)

$$\begin{array}{lrcl} \max & 2x_1 + x_2 & & \\ \text{wobei} & 3x_1 + 4x_2 & \leq & 12 \\ & -x_2 & \geq & -2 \\ & x_1 - x_2 & \leq & 3 \\ & x_1, x_2 & \in & \mathbb{R}_{\geq 0}. \end{array}$$

1. Skizzieren Sie den zulässigen Bereich von (P).
2. Bestimmen Sie A, b und c so, dass (P) dem Problem
$$(P') : \min\{c^T x \mid Ax = b, x \geq 0\}$$
entspricht.
3. Bestimmen Sie eine optimale Lösung x von (P') mit Hilfe des Simplex-Algorithmus. Geben Sie dazu die von ihnen gewählte Startecke des zulässigen Bereichs und das entsprechende Starttableau sowie alle Tableaus, die sich durch Anwendung des Simplex-Algorithmus ergeben, an.

Aufgabe 2. Gegeben seien das Problem

$$(P) : \min\{c^T x \mid Ax \geq b, x \geq 0\}$$

und das dazu duale Problem

$$(D) : \max\{b^T y \mid y^T A \leq c^T, y \geq 0\}$$

mit $A \in \mathbb{R}^{m \times n}$, $c \in \mathbb{R}^n$ und $b \in \mathbb{R}^m$.

P und D bezeichnen jeweils die zulässigen Mengen der beiden Probleme.

Zeigen Sie, dass genau eine der folgenden Aussagen gilt.

(i) P und D sind nicht leer und es gilt $\min\{c^T x \mid x \in P\} \geq \max\{b^T y \mid y \in D\}$.

(ii) $P = D = \emptyset$.

(iii) $P = \emptyset$ und $\sup\{b^T y \mid y \in D\} = \infty$.

(iv) $D = \emptyset$ und $\inf\{c^T x \mid x \in P\} = -\infty$.

(Hinweis: Sie können verwenden, dass genau eine der beiden folgenden Alternativen gilt.

(a) $\{x \in \mathbb{K}^n \mid Ax \leq b, x \geq 0\} \neq \emptyset$.

(b) $\{y \in \mathbb{K}^m \mid y^T A \geq 0, y \geq 0, y^T b < 0\} \neq \emptyset$.)

Bitte wenden.

Aufgabe 3. Beweisen Sie die folgende Aussage:

Für einen nicht leeren rationalen Polyeder $P = \{x \in \mathbb{R}^n \mid Ax \leq b\}$ sind folgende Aussagen äquivalent.

(1) $P = P_I$.

(2) Jede Seitenfläche von P enthält ganzzahlige Vektoren.

(3) Für jedes $c \in \mathbb{R}^n$, für welches $\max\{c^T x \mid x \in P\}$ endlich ist, wird das Maximum in einem ganzzahligen Vektor angenommen.

Sie dürfen verwenden, dass P_I ein Poyleder ist.

Aufgabe 4. Gegeben sei das folgende ganzzahlige lineare Optimierungsproblem (P)

$$\begin{array}{lrcl} \max & 2x_1 & & \\ \text{wobei} & 2x_1 + 5x_2 & \leq & 5 \\ & x_1, x_2 & \in & \mathbb{Z}. \end{array}$$

1. Lösen Sie das zu (P) gehörende relaxierte Problem mit dem Simplex-Algorithmus.
2. Berechnen Sie aus der Gleichung des letzten Tableaus einen Gomory-Schnitt.
3. Lösen Sie das zu (P) gehörende, um den Gomory-Schnitt erweiterte relaxierte Problem mit dem Simplex-Algorithmus. Stimmt die erhaltene Lösung mit der des ganzzahligen linearen Optimierungsproblems überein? Wieso?

Aufgabe 5. Wenden Sie den KRUSKAL ALGORITHMUS auf die Instanz (G, c) in Abbildung 1 an. Geben Sie die Folge der sortierten Kanten nach der ersten Zeile des Algorithmus an. Geben Sie die Menge $E(T)$

- direkt vor dem ersten Schleifendurchlauf von "**for** $i = 1$ **to** m **do**" und
- nach jedem Schleifendurchlauf von "**for** $i = 1$ **to** m **do**" an.

Was berechnet der Algorithmus und warum arbeitet er auf der gegebenen Instanz korrekt?

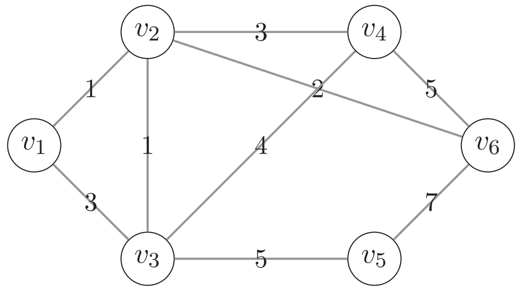

Abbildung 1: (G, c)

Algorithmus 1. KRUSKAL ALGORITHMUS

Input: *Ein zusammenhängender Graph* G *und eine Kostenfunktion* $c : E \to \mathbb{R}$.

```
begin
    Sortiere die Kanten von G so, dass c(e_1) ≤ c(e_2) ≤ ... ≤ c(e_m) gilt;
    T ← (V(G), ∅);
    for i = 1 to m do
        if T + e_i ist ein Wald then
            T ← (V(G), E(T) ∪ {e_i});
        end
    end
    return T;
end
```