Einführung in die Optimierung
10. Übungsblatt

Fachbereich Mathematik **WS 2013/14**
Prof. Dr. Marc Pfetsch **16./17.01.2014**
Dipl.-Math. Oec. Andreas Tillmann

Hinweis zur Programmieraufgabe R3: Falls der Zielfunktionsvektor c (des LPs (P): max $c^T x$ s.t. $Ax \leq b,\ x \geq 0$) nichtpositiv ist (also $c \leq 0$), lässt sich für jedes LP, dass aus (P) durch beibehalten lediglich einer (beliebigen) Ungleichung gewonnen wird, eine dual zulässige Basis bestimmen. (Man kann dann z.B. in der zugehörigen Standardform dieses "einzeiligen" LPs jeden Spaltenindex als (einelementige) Basis durchprobieren, bis eine solche Basis gefunden ist, die dual zulässig ist; mindestens die Schlupfbasis funktioniert garantiert!)

Das beschriebene Vorgehen garantiert im Fall $c \not\leq 0$ leider nicht, dass man so immer eine dual zulässige Basis zu einem "einzeiligen" LP findet. Für die Implementierung (Bearbeitung der Aufgabe) wäre es dennoch in Ordnung, den Ansatz umzusetzen – dann sollten jedoch alle möglichen Zeilen durchprobiert werden, solange nicht mit einer vorigen eine passende (einelementige) Basis ermittelt werden konnte. Im Falle, dass auch nach Überprüfung aller Zeilen (und jeweils aller Spalten) keine dual zulässige Basis gefunden wurde, kann mit der Mitteilung "Keine passende Startbasis gefunden" abgebrochen werden. (Es sei jedoch bemerkt, dass man hieraus im Allgemeinen *nicht* folgern kann, dass das primale LP (P) unbeschränkt ist, bzw. das duale unzulässig!)

Andere Ansätze zum Finden einer Startbasis für den inkrementellen Löser, oder Erweiterungen obiger Vorgehensweise, sind natürlich denkbar! (D.h., niemand muss seinen Code ändern, wenn bereits ein anderer begründbarer Ansatz umgesetzt wurde.)

Gruppenübung

Aufgabe G1 (Kodierungslänge)
Zeigen Sie:

(a) Für jedes $r \in \mathbb{Q}$ gilt: $|r| \leq 2^{\langle r \rangle - 1} - 1$.

(b) Für je zwei rationale Zahlen $r, s \in \mathbb{Q}$ gilt: $\langle rs \rangle \leq \langle r \rangle + \langle s \rangle$.

Aufgabe G2 (Kettenbrüche)
Diese Aufgabe, und auch die nächste, vertiefen die Beweisidee zur polynomialen Lösbarkeit des Problems der rationalen Approximation, welches in der Vorlesung im Zusammenhang mit der Reduktion von LPs auf Zulässigkeitsprobleme mittels Binärer Suche auftauchte (vgl. Abschnitt 6.2 im Skript).
Sei $a_0 \in \mathbb{Z}$ und seien $a_1, a_2, \ldots, a_k \in \mathbb{N}$ (also insbesondere positiv). Ein Ausdruck der Form

$$a_0 + \cfrac{1}{a_1 + \cfrac{1}{a_2 + \cfrac{1}{\ddots \; a_{k-1} + \cfrac{1}{a_k}}}}$$

heißt *(endlicher) Kettenbruch*; abkürzend schreiben wir hierfür $a_0 + 1\rfloor\overline{a_1} + \cdots + 1\rfloor\overline{a_k}$. Offensichtlich ergibt ein Kettenbruch $a_0 + 1\rfloor\overline{a_1} + \cdots + 1\rfloor\overline{a_k}$ eine rationale Zahl. Tatsächlich kann *jede* rationale Zahl $r \in \mathbb{Q}$ durch einen Kettenbruch dargestellt werden; die entsprechenden a_i (und die "Länge" k_r) findet folgender Algorithmus:

1. Setze $a_0 := \lfloor r \rfloor$ und $r_1 := 1/(r - a_0)$
2. Für $i = 1, 2, \ldots$
   i. Setze $a_i := \lfloor r_i \rfloor$

ii. Falls $a_i = r_i$, setze $k_r := i$ und breche ab, ansonsten setze $r_{i+1} := 1/(r_i - a_i)$

Im Folgenden bezeichne $\beta_i := a_0 + 1\rfloor\overline{a_1} + \cdots + 1\rfloor\overline{a_i}$ $(0 \le i \le k_r)$ die i-te Kettenbruch-Näherung von $r \in Q$; insbesondere gilt also $\beta_{k_r} = r$. Zudem seien p_i und q_i zusammen mit a_i im obigen Algorithmus derart berechnete Zahlen, dass

$$\begin{aligned} p_i &:= a_i\, p_{i-1} + p_{i-2} \quad (i = 0, 1, \ldots, k_r), \qquad \text{initialisiert mit } p_{-2} := 0,\ p_{-1} := 1; \\ q_i &:= a_i\, q_{i-1} + q_{i-2} \quad (i = 0, 1, \ldots, k_r), \qquad \text{initialisiert mit } q_{-2} := 1,\ q_{-1} := 0. \end{aligned}$$

Beweisen Sie (induktiv) folgende Aussagen: Für alle $0 \le i (\le k_r)$ gilt

(a) $\beta_i = \frac{p_i}{q_i}$,

(b) $p_{i+1}\, q_i - p_i\, q_{i+1} = (-1)^i$.

Bemerkung: Eine Konsequenz dieser Aussagen ist, dass p_i und q_i teilerfremd sind; insbesondere berechnet der oben angegebene Algorithmus also eine Darstellung $r = p/q$ mit $p = p_{k_r}$, $q = q_{k_r}$ teilerfremd. Zudem kann man zeigen, dass $k_r \in O(\log q)$, der Algorithmus also polynomiale Laufzeit besitzt.

Aufgabe G3 (Rationale Approximation mit Kettenbrüchen)
Es liegt nahe, zu betrachten, wie gut die Näherungswerte β_i (siehe vorige Aufgabe) die rationale Zahl r approximieren. Hierzu ist die Beobachtung hilfreich, dass diese Näherungen abwechselnd kleiner oder größer sind als r selbst, und (etwas vage ausgedrückt) dass die erreichbare Genauigkeit einer rationalen Approximation mit der erlaubten Größe des Kettenbruch-Nenners (q_i) zusammenhängt. Zudem sei $r \notin \mathbb{Z}$ (ansonsten gäbe es nichts zu tun, da ganze Zahlen für beliebige Nennergrößen am besten durch sich selbst "approximiert" werden).

(a) Sei $0 \le i < k_r$. Beweisen Sie, dass dann $\beta_i < r$, falls i eine gerade Zahl ist, und $\beta_i > r$, falls i ungerade ist.

(b) Sei $0 < \varepsilon < 1$ und sei j der größte Index so, dass $q_j \le 1/\varepsilon$. Zeigen Sie, dass gilt:

$$|r - \beta_j| < \frac{\varepsilon}{q_j}.$$

Bemerkung: Mit noch etwas mehr Arbeit kann man letztlich zeigen, dass für jede Zahl $r \in \mathbb{Q}$ und eine beliebige positive Zahl $N \in \mathbb{N}$ die beste rationale Approximation mit Nenner höchstens N in polynomialer Zeit (in $\langle N \rangle$ und $\langle r \rangle$) gefunden werden kann.

Hausübung

Aufgabe H1 (Reoptimierung)

(a) Ihr Vorgesetzter gibt Ihnen folgendes LP. Lösen Sie dieses per Hand mit Hilfe von Algorithmen aus der Vorlesung.

$$\begin{array}{rcl} \min\ 3x_1 + x_2 + 5x_3 & & \\ \text{s.t.}\ \ x_1 + x_2 + 2x_3 & \ge & 2 \\ 3x_1 - x_3 & \ge & 5 \\ x_1, x_2, x_3 & \ge & 0 \end{array}$$

(b) Danach stellt sich heraus, dass bei der Modellierung ein wichtiger Rohstoff vergessen wurde. Somit wird eine neue Variable benötigt. Diese geht in die erste Restriktion mit -1, in die zweite mit $+5$ ein, soll natürlich nichtnegativ sein und hat einen Zielfunktionskoeffizienten von $+2$ (alle Angaben beziehen sich auf die Formulierung aus Aufgabenteil (a)). Lösen Sie das so entstandene LP per Hand mit Hilfe der Lösung aus Aufgabenteil (a) und Algorithmen aus der Vorlesung.

(c) In der Zwischenzeit spricht Ihr Vorgesetzter mit dem Kunden. Aus diesen Verhandlungen ergibt sich, dass sich die rechte Seite des LPs aus Aufgabenteil (b) zu $(2, 1)^T$ ändert. Wie lautet die Optimallösung, wenn Sie diese Veränderung zusätzlich berücksichtigen?

(d) Nach Ihrer Reoptimierung wird festgestellt, dass der Faktor 1 mit dem die Variable x_1 in die erste Restriktion eingegangen ist, falsch kalkuliert wurde. Dieser wird nun auf 3 korrigiert. Lösen Sie das so entstandene LP per Hand mit Hilfe der Lösung aus Aufgabenteil (c) und Algorithmen aus der Vorlesung.

Hinweis: Wenden Sie die Methoden der Sensitivitätsanalyse in dieser Aufgabe an (Nicht jeden Aufgabenteil unabhängig von den anderen lösen!). Die Änderungen sind immer bezüglich des LPs in der Form aus Aufgabenteil (a) gegeben. Die Änderungen sind zeitlich nacheinanderfolgend, d.h. in Teil (d) sind auch Änderungen aus (b) und (c) zu berücksichtigen.

Aufgabe H2 (Kodierungslänge und Laufzeit)

(a) Gegeben sei folgendes Optimierungsproblem

$$\begin{array}{rrcl} \min & \frac{1}{2}x_1 + 2x_2 & & \\ \text{s.t.} & -x_1 - x_2 & \leq & -3 \\ & x_2 & \leq & -2 \\ & -x_1 + x_2 & \leq & 3 \\ & x_1 - x_2 & \leq & 5 \\ & x_1, x_2 & \geq & 0. \end{array}$$

Bestimmen Sie die Kodierungslänge dieses LPs. Interpretieren Sie dabei alle auftretenden Zahlen als rationale Zahlen!

(b) Betrachten Sie den Algorithmus $\mathscr{A}_1$. Gegeben seien folgende Eingabedaten

Algorithm $\mathscr{A}_1$: Arithmetisches Mittel

INPUT: $q_1, \dots, q_n \in \mathbb{Q}, n \in \mathbb{N}$

OUTPUT: $q = \frac{1}{n}\sum_{i=1}^{n} q_i$

$q \leftarrow 0$
for $i = 1, \dots, n$ **do**
  $q \leftarrow q + q_i$
end for
$q \leftarrow \frac{q}{n}$
return q

$$(q_1, \dots, q_6) = \left(\frac{1}{2}, \frac{1}{3}, \frac{1}{5}, \frac{1}{7}, \frac{1}{11}, \frac{1}{17}\right).$$

Bestimmen Sie für das durch die gegebene Eingabesequenz resultierende Problem Π explizit die Laufzeit $L_{\mathscr{A}_1}(\Pi)$.

Aufgabe H3 (Laufzeit)

Betrachten Sie Algorithmus $\mathscr{A}_2$, der eine moderne Version des Euklidischen Algorithmus zur Bestimmung des größten gemeinsamen Teilers zweier Zahlen $a, b \in \mathbb{N}$ darstellt. Hat dieser eine polynomiale Laufzeit? Bei der Laufzeitanalyse können Vergleiche, Zuweisungen und modulo-Rechnungen als elementare Rechenschritte angesehen werden.

Algorithm $\mathscr{A}_2$: Moderne Version des Euklidischen Algorithmus

INPUT: $a, b \in \mathbb{N}$

OUTPUT: $\mathrm{ggT}(a, b)$

while $b \neq 0$ **do**
  $h \leftarrow a \bmod b$
  $a \leftarrow b$
  $b \leftarrow h$
end while
return a