Einführung in die Optimierung
Mittelseminar

TECHNISCHE UNIVERSITÄT DARMSTADT

Fachbereich Mathematik
Prof. Dr. Alexander Martin
Dipl. Math. Andrea Peter

WS 2009/2010

Mittelseminaraufgabe: Der Netzwerk-Simplex-Algorithmus

Aus der Vorlesung ist der Simplex-Algorithmus bereits bekannt. Minimalkosten-Netzwerk-Fluss-Probleme kann man auch als Lineare Probleme formulieren und mit Hilfe des Simplex-Algorithmus lösen. Es gibt jedoch eine auf Minimalkosten-Netzwerk-Fluss-Probleme angepasste Version des Simplex-Algorithmus, der die Netzwerk-Struktur ausnutzt und solche Probleme wesentlich schneller löst. Ein solcher Algorithmus soll in diesem Mittelseminar hergeleitet werden.

1 Erweiterung des Simplex-Algorithmus

a) Erweitere den von dir implementierten Simplex-Algorithmus aus Übung 7 zu dem Simplex-Algorithmus mit oberen Schranken (Algorithmus 5.21).

2 Ein Minimalkosten-Netzwerk-Fluss-Problem

Wir betrachten einen Graphen (z.B. eine Straßenkarte oder ein System von Abwasserkanälen), bei dem die Kanten (z.B. Straßen oder Rohre) gewisse Kapazitäten haben. Das Nutzen einer Kante verursacht Kosten pro darüberfließende Einheit (z.B. pro Auto oder pro Liter Wasser). An den Knoten des Graphen (z.B. Kreuzung, Parkhaus oder Auffangbecken) werden die Ressourcen entweder bereitgestellt, angefordert oder einfach nur weitergeleitet. Wir wollen nun einen vorgebenen Fluss (z.B. eine bestimmte Menge an Autos oder an Abwasser) durch dieses Netzwerk führen, sodass alle Kapazitätsbeschränkungen und Knotenbedingungen eingehalten werden und minimale Kosten verursacht werden.

Formal lässt sich das Minimalkosten-Netzwerk-Fluss-Problem wie folgt beschreiben:
Gegeben sei ein gerichteter Graph $G = (N,A)$, definiert durch eine Menge N mit n Knoten und eine Menge A mit m gerichteten Kanten. Zu jeder Kante $(i, j) \in A$ sind die Kosten c_{ij} definiert, die die Kosten pro einer Einheit Fluss auf dieser Kante beschreiben. Die Flusskosten steigen hierbei linear mit der Menge des Flusses. Zu jeder Kante gibt es eine Kapazität u_{ij}, die die maximale Flussmenge auf dieser Kante beschreibt und eine untere Schranke l_{ij}, die den minimalen Fluss dieser Kante beschreibt (wir gehen in dieser Aufgabe davon aus, dass $l_{ij} = 0$ für alle $(i, j) \in A$). Zu jedem Knoten $i \in N$ repräsentiere $b(i)$ dessen Angebot bzw. Bedarf. Wenn $b(i) > 0$, so nennen wir i einen Angebotsknoten, ist $b(i) < 0$, nennen wir i einen Bedarfsknoten und ist $b(i) = 0$, so ist i ein Durchflussknoten. Die Summe der $b(i)$'s ergibt immer 0. Die Variable $x_{ij} \geq 0$ soll den Fluss auf einer Kante $(i, j) \in A$ beschreiben. Eine Funktion $x : A \to \mathbb{R}$ heißt Fluss, falls sie die Kapazitätsbedingungen ($l_{ij} \leq x_{ij} \leq u_{ij}$) und die Flusserhaltung (für jeden Knoten $i \in N$ ist die Summe der Flüsse der eingehenden Kanten plus dem Angebot/Bedarf dieses Knotens gleich der Summe der Flüsse der abgehenden Kanten) erfüllt. Ziel ist es, die minimalen Kosten zu berechnen, die entstehen, wenn ein Fluss durch das Netzwerk geleitet wird, der alle Knoten nach Wunsch bedient. Dabei müssen die Kapazitätsbeschränkungen auf jeder Kante eingehalten werden, sowie die Forderungen an jedem Knoten exakt eingehalten werden.

3 Ein Beispiel

Gegeben sei der Graph aus Abbildung 1.
Bei Knoten 1 befindet sich eine Fabrik für Weihnachtsplätzchen, bei Knoten 6 befindet sich der Weihnachtsmarkt, auf dem die Plätzchen verkauft werden sollen. Die Kanten zwischen den Knoten stellen Straßen dar, über die jeweils nur eine bestimmte Menge an Plätzchen mit LKWs transportiert werden darf. An den Knoten 2, 3, 4 und 5 sind Umladestationen, an denen die Plätzchen optimal (und kostenfrei) umsortiert werden können. An den Umladestationen werden weder Plätzchen hergestellt noch konsumiert. In den folgenden Tabelle findest du die jeweiligen Kanten- und Knotenbeschränkungen:

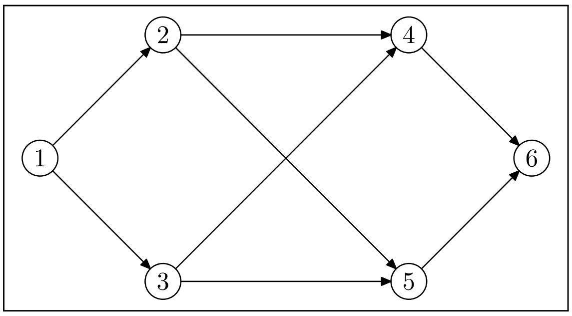

Abbildung 1: Das Plätzchenproblem

Knoten	1	2	3	4	5	6
$b(i)$ (in Tonnen)	20	0	0	0	0	-20

Kante	(1,2)	(1,3)	(2,4)	(2,5)	(3,4)	(3,5)	(4,6)	(5,6)
l_{ij} (in Tonnen)	0	0	0	0	0	0	0	0
u_{ij} (in Tonnen)	15	13	10	10	8	9	9	17
c_{ij} (in Euro)	100	150	150	130	80	80	100	120

Betrachten wir nun ein leicht modifiziertes Plätzchenproblem. Bei Knoten 5 wohnt eine fleißige Oma, die zusätzlich Plätzchen backt, die auch zum Weihnachtsmarkt transportiert werden sollen. An Knoten 2 wohnt ein hungriger Mitarbeiter der Transportfirma, der Plätzchen konsumiert. In der folgenden Tabelle findest du die aktualisierten Knotenbeschränkungen:

Knoten	1	2	3	4	5	6
$b(i)$ (in Tonnen)	20	-5	0	0	5	-20

b) Formuliere die beiden beschriebenen Probleme als LP (der Form 5.12). Formuliere auch das allgemeine Minimalkosten-Netzwerk-Fluss-Problem als LP.

4 Graphentheorie

Um die nächsten Aufgaben besser bearbeiten zu können, wiederholen (oder erarbeiten) wir ein wenig Basiswissen der Graphentheorie.

c) Gib die Definition eines aufspannenden Baumes eines Graphen an.

d) Finde in dem Graphen aus Abbildung 1 einen aufspannenden Baum.

e) Wie viele verschiedene aufspannende Bäume kann es in einem Graphen mit n Knoten und m Kanten höchstens geben? Gib eine grobe Abschätzung nach oben an.

f) Beschreibe einen Algorithmus deiner Wahl, um aufspannende Bäume zu finden. Gib seine Laufzeit und alle notwendigen graphentheoretischen Definitionen an.

g) Gib die Definition einer Knoten-Kanten-Inzidenzmatrix an.

h) Gib die Knoten-Kanten-Inzidenzmatrix für den Graphen aus Abbildung 1 an.

5 Simplex

i) Finde den optimalen Transportplan: Löse das in Aufgabenteil b) aufgestellte LP mit deinem Simplex-Algorithmus aus a). Interpretiere jeden Basiswechsel des Simplex-Algorithmus im Graphen.

6 Die harte Mathematik

j) Beweise den folgenden Satz:
Die Zeilen und Spalten einer Knoten-Kanten-Inzidenzmatrix eines aufspannenden Baumes können zu einer unteren Dreiecksmatrix umgeordnet werden.

k) Beweise den folgenden Satz:
Jeder aufspannende Baum eines Graphen G definiert eine Basis des Minimalkosten-Netzwerk-Fluss-Problems und umgekehrt definiert jede Basis des Minimalkosten-Netzwerk-Fluss-Problems einen aufspannenden Baum in G.

l) Benutze die in j) und k) bewiesenen Sätze, um die Schritte des Simplex-Algorithmus für das Minimalkosten-Netzwerk-Fluss-Problem zu vereinfachen. Erkläre hierzu für jeden Schritt, was sich ändert. Wie sehen die neuen reduzierten Kosten aus? Wie können die Basisvariablen passend umsortiert werden? Gib den angepassten Algorithmus an. Terminiert der Algorithmus (unter der Annahme, dass kein degenerierter Basiswechsel erfolgt)?

7 Spaß am Computer

m) Implementiere den von dir in l) gefundenen Netzwerk-Simplex-Algorithmus.

n) Berechne damit eine Lösung für das Problem aus a).

o) Vergleiche die Rechenzeit für Aufgabe i) und n).

8 Integrality

p) Gib die Definition für eine (total) unimodulare Matrix an.

q) Beweise den folgenden Satz:
Sei A eine Matrix mit ganzzahligen Einträgen und linear unabhängigen Zeilen. Dann sind die folgenden drei Bedingungen äquivalent:

i) A is unimodular
ii) Jede zulässige Lösung von $Ax = b$, $x \geq 0$, ist ganzzahlig für ganzzahlige Vektoren b.
iii) Jede Basismatrix A_B von A hat ein ganzzahliges Inverses A_B^{-1}.

r) Beweise den folgenden Satz:
Die Knoten-Kanten-Inzidenzmatrix eines gerichteten Graphen ist total unimodular.

s) Was folgerst du aus den in q) und r) gezeigten Sätzen?

9 Noch mehr Spaß am Computer

t) Auf der Veranstaltungshomepage sind unter `Mittelseminar` Graphenprobleme bereitgestellt. Eingabeformat und gewünschtes Ausgabeformat werden dort ebenfalls beschrieben. Löse das Minimalkosten-Netzwerk-Fluss-Problem für diese Graphen mit deinem Algorithmus (ohne Zuhilfenahme kommerzieller Tools!). Gib den Output deines Programmes mit ab.

Der Veranstalter hat noch weitere (unveröffentlichte) Testdatensätze. Derjenige, dessen Algorithmus diese Datensätze am schnellsten lösen kann bekommt einen Kasten Bier oder Ähnliches. Hierfür schickt euren Code bitte bis zum Abgabetermin an apeter@mathematik.tu-darmstadt.de

10 Formalia

Schreibe eine mindestens 5-, maximal 12-seitige, eigenständige Ausarbeitung in Latex und bearbeite darin die gestellten Aufgaben. Gebe diese bitte zusammen mit Listings deiner Matlab-Programme (oder C++) bis zum **21./22.01.2010** in der Übung oder in Büro S4|10 130 ab. Gebe auch evtl. verwendete Literatur an. Nicht genannte Quellen und Abschreiben können einen Erhalt des Seminarscheins verhindern.
Auf der Seite `https://moodle.tu-darmstadt.de/` ist ein Forum zur Mittelseminaraufgabe eingerichtet. Hier könnt ihr mit Kommilitonen über die Aufgaben diskutieren und die Veranstalter nach Tipps fragen. Vielleicht findet ihr ja den einen oder anderen hilfreichen Hinweis dort.

DISCRETE
OPTIMIZATION