Seminarvortrag: **Randomisiertes Runden**

Thomas Friedrich
27. Januar 1997

Seminar über Approximation
Universität zu Köln

1 Einleitung

1.1 Methodik

Wir wollen uns hier mit der Methode des randomisierten Rundens beschäftigen, einer effizienten Technik zur Lösung von schwierigen Problemen, vor allem bei der kombinatorischen Optimierung.
Meist formuliert man dazu das gegebene Optimierungsproblem als ein ganzzahliges lineares Programm. Dieses wird dann relaxiert (gelockert) auf ein nicht-ganzzahliges lineares Programm, für das polynomielle Algorithmen zur Lösung existieren, z. B. Innere-Punkt-Methoden. Durch Randomisierung wird dann aus der gewonnenen Lösung des LP's möglichst effektiv die Ganzzahligkeit wiederhergestellt.
Wir wollen das randomisierte Runden vor allem anwenden auf das integer multicommodity flow problem (ganzzahliges Mehrgüterfluß-Problem), außerdem auf Überdeckungs- und Packprobleme, sowie das ungerichtete Multicut-Problem. Dabei soll sich jeweils eine Analyse der Approximationsgüte anschließen.
Eine zweite Möglichkeit, randomisiertes Runden anzuwenden, nämlich mittels semidefiniter Programmierung, werden wir hier nicht betrachten.

1.2 Einführendes Beispiel

Zur kurzen Einführung soll uns hier das Problem der Gitterapproximation (lattice-approximation problem, LAP) dienen. Hierbei geht es darum, zu einer gegebenen $n \times n$-Matrix A, deren Einträge nur aus 0 und 1 bestehen,

und einem gegebenen Spaltenvektor p, dessen n Einträge aus dem Intervall [0,1] stammen, einen Vektor q zu finden, dessen Komponenten nur 0 oder 1 sind, und der p bzgl. A möglichst genau approximiert, also $\| A \cdot (p - q) \|_\infty$ minimiert.
Eine naive Lösungsmethode, die einem sofort einfällt, besteht darin, für alle Einträge $p_i \geq 0.5$ das jeweilige q_i auf 1 zu setzen, sonst auf 0. Hierbei kann es allerdings vorkommen, daß für den resultierenden Vektor q gilt, daß $\| A \cdot (p - q) \|_\infty$ in etwa so groß ist wie $\| A \cdot q \|_\infty$, die Approximation also nicht allzu gut ausfällt.
Besonders bei einem Spezialfall des Gitterapproximations-Problems, dem set-balancing problem, wird dieses Vorgehen ad absurdum geführt: hier besteht nämlich der gegebene Vektor p nur aus Einträgen von 0.5; sämtliche Komponenten des Vektors q würden also geradeso auf 1 gesetzt. Dies ergibt aber verständlicherweise eine nicht sehr brauchbare Lösung.
Setzt man hingegen die Komponente q_i mit der *Wahrscheinlichkeit* p_i auf 1 und sonst auf 0, ist zu erwarten, daß der resultierende Vektor q „nahe“ bei p liegt. So kann man zeigen, daß für eine gegebene Instanz $\{A, p\}$ des Gitterapproximations-Problems durch diesen Algorithmus eine Lösung q gefunden wird, für die $\| A \cdot (p - q) \|_\infty \leq \sqrt{4n \ln n}$ mit einer Wahrscheinlichkeit von mindestens $(1 - \frac{1}{n})$ gilt.
Wie man bereits sieht, spielt die Wahrscheinlichkeitstheorie beim randomisierten Runden eine große Rolle.

2 Stochastische Grundlagen

Definition:
Bezeichne mit $\Pr[A]$ die Wahrscheinlichkeit, daß Ereignis A eintritt.
Zwei Ereignisse A und B heißen *unabhängig*, falls gilt:

$$\Pr[A \wedge B] = \Pr[A] \cdot \Pr[B].$$

Eine *Zufallsvariable* X ist eine Funktion $X : \Omega \to I\!R$ von der Menge der Ereignisse in die reellen Zahlen.
Eine *Bernoullische Zufallsvariable* ist eine Zufallsvariable, die nur den Wert 0 oder 1 annimmt.
Der *Erwartungswert* $\mathrm{E}[X]$ einer Zufallsvariablen X wird definiert als:

$$\mathrm{E}[X] := \sum_{x \in \Omega} x \cdot p(x),$$

wobei $p(x) := \Pr[X = x]$, also die Wahrscheinlichkeit, daß $X = x$ ist.

Lemma 1: (Linearität des Erwartungwertes)
Für zwei Zufallsvariablen X, Y gilt:

$$\mathrm{E}[X+Y] = \mathrm{E}[X] + \mathrm{E}[Y].$$

Beweis:

$$\begin{aligned}
\mathrm{E}[X] + \mathrm{E}[Y] &= \sum_{x \in \Omega} x \mathrm{Pr}[X = x] + \sum_{x \in \Omega} x \mathrm{Pr}[Y = x] \\
&= \sum_{x \in \Omega} x (\mathrm{Pr}[X = x] + \mathrm{Pr}[Y = x]) \\
&= \mathrm{E}[X + Y].
\end{aligned}$$

q. e. d.

Bemerkung:
Induktiv läßt sich Lemma 1 erweitern auf jede endliche Anzahl von Zufallsvariablen $X_1, \ldots, X_n$, also:

$$\mathrm{E}\left[\sum_{i=1}^{n} X_i\right] = \sum_{i=1}^{n} \mathrm{E}[X_i]$$

und sogar auf eine abzählbare Menge von Zufallsvariablen $X_1, X_2, X_3, \ldots$:

$$\mathrm{E}\left[\sum_{i=1}^{\infty} X_i\right] = \sum_{i=1}^{\infty} \mathrm{E}[X_i].$$

Definition:
Das *k-te Moment* einer Zufallsvariablen X ist definiert als $\mathrm{E}[X^k]$.

Satz 2:
Die *Ungleichung für das k-te Moment* lautet:

$$\mathrm{Pr}[|X| \geq \lambda] \leq \frac{1}{\lambda^k} \cdot \mathrm{E}[|X|^k]$$

für eine Zufallsvariable X und $\lambda > 0$.

Beweis:

$$\begin{aligned}
\mathrm{E}[|X|^k] &= \sum_{x\in\Omega}(|x|^k p(x)) \\
&= \sum_{|x|<\lambda}(|x|^k p(x)) + \sum_{|x|\geq\lambda}(|x|^k p(x)) \\
&\geq \sum_{|x|\geq\lambda}(|x|^k p(x)) \\
&\geq \lambda^k \sum_{|x|\geq\lambda} p(x) \\
&= \lambda^k \Pr[|X|\geq\lambda],
\end{aligned}$$

also: $\Pr[|X|\geq\lambda] \leq \frac{1}{\lambda^k}\cdot\mathrm{E}[|X|^k]$.

q. e. d.

Bemerkungen:

(i) Für $k=1$ und $\lambda\cdot\mathrm{E}[X]$ statt λ ergibt sich:

$$\Pr[|X|\geq\lambda\cdot\mathrm{E}[X]] \leq \frac{1}{\lambda}.$$

Diese Ungleichung heißt *Markows Ungleichung.*

(ii) Für $k=2$ und $X-\mathrm{E}[X]$ statt X, sowie $\lambda\cdot\sigma$ für λ, wobei $\sigma := \sqrt{\mathrm{E}[(X-\mathrm{E}[X])^2]}$ die *Standardabweichung* ist, ergibt sich:

$$\begin{aligned}
\Pr[|X-\mathrm{E}[X]|\geq\lambda\cdot\sigma] &\leq \frac{1}{(\lambda\sigma)^2}\cdot\mathrm{E}[(X-\mathrm{E}[X])^2] \\
&= \frac{1}{\lambda^2\sigma^2}\cdot\sigma^2 = \frac{1}{\lambda^2}
\end{aligned}$$

Diese Ungleichung heißt *Tschebyschews Ungleichung.*

(iii) Der Beweis von Satz 2 bleibt gültig, wenn man $(\ldots)^k$ durch eine beliebige monoton steigende Funktion $g(x)$ auf $[0,\infty[$ ersetzt:

$$\Pr[|X|\geq\lambda] \leq \frac{1}{g(\lambda)}\cdot\mathrm{E}[g(|X|)],$$

also z. B. $g(x) := \exp(tx)$. Man erhält:

$$\Pr[|X|\geq\lambda] \leq \exp(-t\lambda)\cdot\mathrm{E}[\exp(tX)],$$

die *Tschernoff-Ungleichung*, wobei man t so wählen kann, daß die rechte Seite möglichst klein wird.

3 Multicommodity Flow I

Dieser Abschnitt soll den Schwerpunkt dieses Seminarvortrages darstellen.

3.1 Das Problem

Eine Variante des ganzzahligen Mehrgüterfluß-Problems, mit der wir uns nun beschäftigen wollen, läßt sich wie folgt formulieren: Gegeben sei ein gerichteter Graph $G = (V, E)$ mit Quellen $s_i \in V$ und Senken $t_i \in V$, $i = 1, \ldots, k$. Die Aufgabe besteht nun darin, für jedes i einen Pfad P_i von s_i nach t_i durch G zu finden, so daß die Maximalbelastung der Kanten im Graphen so gering wie möglich gehalten wird. Die Belastung einer Kante ist dabei definiert als die Anzahl der Pfade, die über diese Kante laufen.
Im allgemeinen ist dieses Problem NP-vollständig.

3.2 Formulierung als ganzzahliges lineares Optimierungsproblem

Bezeichne mit $x_i(v, w)$ (wobei $(v, w) \in E$), ob der Pfad P_i von s_i nach t_i über (v, w) läuft ($x_i(v, w) = 1$), oder nicht ($x_i(v, w) = 0$).
Dann sieht das zugehörige ganzzahlige lineare Optimierungsproblem wie folgt aus:

(1) minimiere W
unter Berücksichtigung der Restriktionen:

(2) $$\sum_{w \in V} x_i(v, w) - \sum_{w \in V} x_i(w, v) = \begin{cases} 1, & \text{falls } v = s_i \\ -1, & \text{falls } v = t_i \\ 0, & \text{sonst} \end{cases} \qquad i = 1, \ldots, k$$

(3) $\sum_{i=1}^{k} x_i(v, w) \leq W$ für alle $(v, w) \in E$

(4) $x_i(v, w) \in \{0, 1\}$ für alle $(v, w) \in E$ und $i = 1, \ldots, k$,

wobei (2) (evtl. nicht einfache) Pfade von s_i nach t_i erzwingt, da hier für jeden Knoten die Anzahl der Eingangskanten eines Pfades von der Zahl herausführender Kanten desselben Pfades abgezogen werden. Dies muß für den Startknoten 1 sein, für den Zielknoten -1 und für alle anderen 0. Ungleichung (3) hält die Belastung jeder Kante unter der Schwelle W, und (4) garantiert, daß die x_i nur die beiden zulässigen Werte 0 und 1 annehmen können.
(1) minimiert dann über die Maximalbelastung W einer Kante im Graphen.

3.3 Relaxierung auf LP

Wir können das ganzzahlige lineare Optimierungsproblem in ein lineares Programm (LP) überführen, indem wir die Bedingung (4): $x_i(v,w) \in \{0,1\}$ lockern auf $x_i(v,w) \geq 0$. Da nun aber der Raum der zulässigen Lösungen ungemein größer ist, werden die optimalen Lösungen W^* für das ganzzahlige Problem und W_{LP} für das relaxierte Problem im allgemeinen nicht übereinstimmen.
Jedoch läßt sich nun das LP mittels geeigneter Algorithmen in polynomieller Zeit lösen. Die erhaltene Optimallösung wird i. a. nicht ganzzahlig sein, was auch nicht zu erwarten war.

3.4 Flow Decomposition

Die gebrochene Lösung kann in Pfade zerlegt werden mit Hilfe von flow decomposition (Flußzerlegung), einer Standardtechnik bei Flüssen im Netzwerk:
Sei dazu $x \geq 0$ und

$$\sum_{w\in V} x(v,w) - \sum_{w\in V} x(w,v) = \begin{cases} a, & \text{falls } v = s_i \\ -a, & \text{falls } v = t_i \\ 0, & \text{sonst} \end{cases} \qquad i = 1,\dots,k$$

(Beachte die Ähnlichkeit zu (2) mit $a = 1$.)
Dann kann man für festes $i \in \{1,\dots,k\}$ Pfade $P_1,\dots,P_l$ von s_i nach t_i finden mit $\alpha_1,\dots,\alpha_l \in I\!R^+$, so daß

$$\sum_{j=1}^{l} \alpha_j = a,$$
$$\sum_{j:(v,w)\in P_j} \alpha_j \leq x(v,w).$$

Dies sei hier ohne Beweis angegeben.

3.5 Der Algorithmus nach Raghavan und Thompson

Der folgende Algorithmus, der zur Lösung des Problems randomisiertes Runden verwendet, geht auf Raghavan und Thompson zurück.

Algorithmus (RT):

1. Löse das relaxierte LP, erhalte dabei die Lösung W_{LP}.

2. Zerlege die gebrochene Lösung mittels flow decomposition in Pfade P_{ij}, $i = 1,\dots,k$, $j = 1,\dots,j_i$, wobei P_{ij} ein Pfad von s_i nach t_i

durch G bezeichnet. Man erhält die $\alpha_{ij} \in I\!R^+$ mit $\sum_{j=1}^{l_i} \alpha_{ij} = 1$, so daß gilt:

$$\sum_{i=1}^{k} \sum_{j:(v,w)\in P_{ij}} \alpha_{ij} \leq W_{LP}.$$

3. Randomisierungsschritt: Für jedes $i \in \{1,\ldots,k\}$ werfen wir einen j_i-seitigen „Würfel", bei dem mit Wahrscheinlichkeit α_{ij} die „Seite" j erscheint. Sei f das Ergebnis dieses Wurfes; wähle dann den Pfad P_{if} als *den* Pfad von s_i nach t_i.

3.6 Analyse

Die folgende Analyse des Algorithmus soll zeigen, daß wir mit hoher Wahrscheinlichkeit eine geringe Kantenbelastung im Graphen erhalten.

Lemma 3: (Tschernoff-Schranke)
Seien X_i, $i = 1,\ldots,k$, unabhängige Bernoulli-Zufallsvariablen mit Wahrscheinlichkeiten $\Pr[X_i = 1] = p_i$ (und damit $\Pr[X_i = 0] = (1 - p_i)$) für alle i. Dann gilt für alle $\alpha > 0$ und alle $t > 0$:

$$\begin{aligned}\Pr\left[\sum_{i=1}^{k} X_i \geq t\right] &\leq \exp(-\alpha t)\prod_{i=1}^{k}\mathrm{E}[\exp(\alpha X_i)]\\ &= \exp(-\alpha t)\prod_{i=1}^{k}(p_i\exp(\alpha) + (1 - p_i)).\end{aligned}$$

Beweis:

$$\begin{aligned}\Pr\left[\sum_{i=1}^{k} X_i \geq t\right] &\leq \exp(-\alpha t)\cdot \mathrm{E}\left[\exp\left(\alpha\sum_{i=1}^{k} X_i\right)\right]\\ &= \exp(-\alpha t)\cdot \mathrm{E}\left[\prod_{i=1}^{k}\exp(\alpha X_i)\right]\\ &= \exp(-\alpha t)\cdot\prod_{i=1}^{k}\mathrm{E}[\exp(\alpha X_i)].\end{aligned}$$

q. e. d.

Korollar 4:
Seien $X_i,\ i = 1, \ldots, k$, unabhängige Bernoulli-Zufallsvariablen, und sei $M := \mathrm{E}[\sum_{i=1}^k X_i] = \sum_{i=1}^k p_i$. Dann gilt für alle $\beta > 0$:

$$\begin{aligned}\Pr\left[\sum_{i=1}^k X_i \geq (1+\beta)\cdot M\right] &\leq (1+\beta)^{-(1+\beta)\cdot M} \prod_{i=1}^k \mathrm{E}[(1+\beta)^{X_i}] \\ &\leq \left(\frac{\exp(\beta)}{(1+\beta)^{(1+\beta)}}\right)^M.\end{aligned}$$

Die zweite Ungleichung folgt aus:

$$\mathrm{E}[(1+\beta)^{X_i}] = p_i(1+\beta) + (1-p_i) = 1 + \beta p_i \leq \exp(\beta p_i).$$

Theorem 5:
Sei $\varepsilon > 0$; ferner sei W^* die Optimallösung des obigen multicommodity flow-Problems mit $W^* = \Omega(\log n)$, wobei $n = |V|$. Dann erzeugt der Algorithmus (RT) mit Wahrscheinlichkeit $(1-\varepsilon)$ eine Lösung für die Maximalbelastung einer Kante im Graphen

$$W \leq W^* + c \cdot \sqrt{W^* \ln n},$$

wobei c und die Konstante aus der Ω-Notation von ε abhängen.

Beweis:
Sei $(v,w) \in E$ beliebig, aber fest. Die Kante (v,w) wird vom Produkt i mit der Wahrscheinlichkeit

$$p_i = \sum_{j:(v,w)\in P_{ij}} \alpha_{ij}$$

benutzt.
Sei nun X_i eine Bernoulli-Zufallsvariable, die angibt, ob (v,w) von P_i benutzt wird ($X_i = 1$), oder nicht ($X_i = 0$). Dann ist $W(v,w) := \sum_{i=1}^k X_i$ die Belastung der Kante (v,w). Damit ist

$$\mathrm{E}[W(v,w)] = \sum_{i=1}^k p_i = \sum_{i=1}^k \sum_{j:(v,w)\in P_{ij}} \alpha_{ij} \leq W_{LP} \leq W^*.$$

Mit Lemma 3 (Tschernoff-Schranke) und Korollar 4 folgt:

$$\begin{aligned}\Pr[W(v,w) \geq (1+\beta)W^*] &\leq \left(\frac{\exp(\beta)}{(1+\beta)^{(1+\beta)}}\right)^{W^*} \\ &= \exp((\beta - (1+\beta)\ln(1+\beta))W^*).\end{aligned}$$

Sei jetzt $\beta \leq 1$; dann gilt:

$$\frac{\exp(\beta)}{(1+\beta)^{(1+\beta)}} = \exp(\beta - (1+\beta)\ln(1+\beta)) \leq \exp(\frac{-\beta^2}{3}).$$

Somit ist für $\beta := \sqrt{\frac{3\ln\frac{n^2}{\varepsilon}}{W^*}}$:

$$\Pr[W(v,w) \geq (1+\beta)\cdot W^*] \leq \frac{\varepsilon}{n^2}.$$

(Die Annahme $\beta \leq 1$ trifft zu, wenn $W^* \geq 6\ln n - 3\ln\varepsilon$ ist.)
Mit obiger Wahl von β ist also

$$\begin{aligned}(1+\beta)\cdot W^* = W^* + \beta W^* &= W^* + \sqrt{\frac{3\ln\frac{n^2}{\varepsilon}}{W^*}}\cdot W^* \\ &= W^* + \sqrt{\frac{W^{*2}\cdot 3\ln\frac{n^2}{\varepsilon}}{W^*}} = W^* + \sqrt{3W^*\ln\frac{n^2}{\varepsilon}}.\end{aligned}$$

Jetzt gilt für die Wahrscheinlichkeit, daß die Maximalbelastung *größer* ist als $(1+\beta)\cdot W^*$:

$$\begin{aligned}\Pr[\max_{(v,w)\in E} W(v,w) > (1+\beta)\cdot W^*] &\leq \sum_{(v,w)\in E}\Pr[W(v,w) \geq (1+\beta)\cdot W^*] \\ &\leq |E|\cdot\frac{\varepsilon}{n^2} \leq \varepsilon,\end{aligned}$$

was wir zeigen wollten.

q. e. d.

3.7 Derandomisierung

Die Derandomisierung erfolgt mit Mitteln der bedingten Wahrscheinlichkeiten. Außerdem fließt noch ein „kleiner Trick“ mit ein.
Wir stellen den Wahrscheinlichkeitsraum als Entscheidungsbaum dar. Wir starten an der Wurzel, an der wir noch keine Entscheidung getroffen haben. Während wir uns nun den Baum hinabhangeln, legen wir nacheinander fest, welche Pfade von den einzelnen Gütern benutzt werden sollen. Dabei entscheiden wir zunächst für Produkt 1, dann für Produkt 2, usw. An jedem Knoten des Baumes stellen also die ausgehenden Kanten die möglichen Entscheidungen über die Wahl des Pfades für das nächste Produkt dar. Somit

hat die Wurzel j_1 Söhne, da wir ja j_1 verschiedene Pfade für Produkt 1 zur Auswahl haben. Jeder dieser Söhne hat wiederum j_2 Söhne, zur Wahl des Pfades für Produkt 2, usw.
Irgendwann erreichen wir die k-te Ebene (ganz unten). Hier befinden sich die Blätter des Baumes, die zusammen alle möglichen Entscheidungen für die k Produkte repräsentieren.
Seien wir beim Hinabsteigen des Baumes nach i bereits getroffenen Entscheidungen $l_1, \ldots, l_i$ für die Produkte $1, \ldots, i$ bei einem Knoten in Ebene i angekommen (wobei die Wurzel in Ebene 0 liegt). Wir definieren jetzt:

$$\begin{aligned} g(l_1, \ldots, l_i) := \Pr[\max_{(v,w)\in E} W(v,w) \geq (1+\beta) \cdot W^* | \\ l_1 \text{ für Produkt } 1, \ldots, l_i \text{ für Produkt } i]. \end{aligned}$$

Wir wollen nun einen Pfad für das $(i+1)$te Produkt festlegen. Es gilt:

$$g(l_1, \ldots, l_i) = \sum_{j=1}^{j_i} \alpha_{ij} \cdot g(l_1, \ldots, l_i, j) \geq \min_j g(l_1, \ldots, l_i, j).$$

Könnte man jetzt g effizient berechnen, könnten wir, ausgehend von $g(\emptyset)$ und bei jeweiliger Wahl des Minimums auf jeder Ebene des Entscheidungsbaumes, eine Folge $g(\emptyset) \geq g(l_1) \geq g(l_1, l_2) \geq \ldots \geq g(l_1, \ldots, l_k)$ konstruieren.
Unglücklicherweise geht dies aber nicht.
Hier kommt der „kleine Trick“ ins Spiel: Anstelle der exakten Berechnung von g fragen wir einen „pessimistischen Einschätzer“, wie groß die Wahrscheinlichkeit für einen Fehler ist.
Wegen Korollar 4 und aus der (RT)-Analyse wissen wir:

$$\Pr[\max_{(v,w)\in E} W(v,w) \geq (1+\beta)\cdot W^*] \leq (1+\beta)^{-(1+\beta)\cdot W^*} \sum_{(v,w)\in E} \mathrm{E}\left[\prod_{i=1}^{k} \exp(\beta X_i^{(v,w)})\right],$$

wobei $X_i^{(v,w)}$ von der Kante (v,w) abhängt.
Sei $h(l_i, \ldots, l_i)$ die Summe auf der rechten Seite dieser Ungleichung; insgesamt ergibt sich dann:

1. $h(l_1, \ldots, l_i)$ kann leicht berechnet werden
2. $g(l_1, \ldots, l_i) \leq h(l_1, \ldots, l_i)$
3. $h(l_1, \ldots, l_i) \geq \min_j h(l_1, \ldots, l_{i-1}, j)$.

Wenn wir also in der letzten Ungleichung in jeder Ebene das Minimum wählen, ergibt sich die Folge

$$1 > \varepsilon \geq h(\emptyset) \geq h(l_1) \geq h(l_1, l_2) \geq \ldots \geq h(l_1, \ldots, l_k).$$

Weil $g(l_1, \ldots, l_k) \in \{0, 1\}$ sein muß, kann sich durch die Wahl der Pfade in diesem deterministischen Algorithmus nur eine maximale Belastung von höchstens $(1 + \beta) \cdot W^*$ ergeben.

4 Multicommodity Flow II

Hier wollen wir jetzt das ganzzahlige Mehrgüterfluß-Problem in seiner klassischen Form betrachten.

4.1 Problem

Die Aufgabe ist in diesem Fall quasi umgekehrt: zu einem gerichteten Graphen $G = (V, E)$, ist zu jeder Kante $e \in E$ eine maximale Kapazität $c(e)$ gegeben. Außer den Quellen $s_i \in V$ und den Senken $t_i \in V$, $i = 1, \ldots, k$, besteht eine Instanz noch aus ganzzahligen Nachfragen d_i nach Produkt i. Gesucht sind ganzzahlige Flüsse $f_i(e)$ über die Kanten von G, so daß mindestens d_i Einheiten von Produkt i von s_i nach t_i fließen.
Bei der Lösung des Problems sind also folgende Restriktionen zu beachten:

(1) Sämtliche Flüsse $f_i(e)$ über jede Kante e sollen ganzzahlig sein:
$f_i(e) \in \{0, 1\}$ für alle $e \in E$, $i = 1, \ldots, k$.

(2) Für jedes Produkt sollen die in einem Knoten eingehenden und ausgehenden Flüsse gleich sein (außer natürlich Start- und Endknoten).

(3) Die Kapazität einer Kante darf nicht überschritten werden:
$\sum_{i=1}^{k} f_i(e) \leq c(e)$ für alle $e \in E$.

(4) Die Nachfragen d_i sollen erfüllt werden:
$\sum_{e \in A_i} f_i(e) \geq d_i$,
wobei A_i die Menge des Produktes i angibt, die s_i verläßt.

Es kann passieren, daß sich dieses Problem nicht lösen läßt, nämlich dann, wenn die Kapazitäten der Kanten nicht ausreichen, um die Nachfragen zu decken.

Wir beschäftigen uns hier mit einer Optimierungsvariante: Gegeben sei eine Instanz $(s_i, t_i, d_i),\ i = 1, \ldots, k$, dieses ganzzahligen Mehrgüterfluß-Problems. Wir wollen nun den Gesamtfluß

$$\sum_{i=1}^{k} \sum_{e \in A_i} f_i(e)$$

maximieren. Dabei beschränken wir uns auf den Fall, daß alle $d_i = 1$ sind. Dieselbe Technik ist aber mit geringen Änderungen auch auf den allgemeinen Fall $d_i \geq 1$ anwendbar.
Nun ergibt sich in Verbindung mit (1) - (4) ein ganzzahliges lineares Optimierungsproblem. Relaxiert man (1): $f_i(e) \in \{0, 1\}$ auf $f_i(e) \in [0, 1]$ für alle $e \in E,\ i = 1, \ldots, k$, könnte man das entstandene LP wieder effizient lösen. Stattdessen wollen wir ein leicht modifiziertes LP lösen; wir setzen dazu die Kapazität jeder Kante auf $(1 - \varepsilon) \cdot c(e)$, für eine positive Konstante $\varepsilon < \frac{\sqrt{5}-1}{2}$, sonst lassen wir alles gleich. Die gebrochene Lösung dieses linearen Programms bezeichnen wir mit $\hat{f}_i(e)$, der entstehende Gesamtfluß sei

$$\hat{F} = \sum_{i=1}^{k} \sum_{e \in A_i} f_i(e).$$

Ferner bezeichne F die Optimallösung des ganzzahligen Problems. Dann ist $\hat{F} \geq (1-\varepsilon)F$ und der Fluß über jede Kante bei der nicht-ganzzahligen Lösung ist höchstens das $(1 - \varepsilon)$-fache ihrer Kapazität.

4.2 Randomisierung

Wir wenden jetzt das randomisierte Runden wie folgt an: Für jedes Produkt i gehen wir zufällig von s_i nach t_i durch den Graphen, indem wir erst einmal eine Münze mit „Kopf" und „Zahl" werfen, wobei die Wahrscheinlichkeit, daß „Kopf" erscheint, gleich $\sum_{e \in A_i} \hat{f}_i(e) \leq 1$ sei. Wenn die Münze „Zahl" zeigt, setzen wir hier den Fluß für Produkt i auf 0. Zeigt die Münze aber „Kopf", gehen wir weiter durch den Graphen. Seien wir so bei Knoten v angekommen; wir wählen den weiteren Weg über Kante $a \in A(v)$ mit der Wahrscheinlichkeit $\frac{\hat{f}_(a)}{\sum_{e \in A(v)} \hat{f}_i(e)}$, wobei $A(v)$ die Menge der v verlassenden Kanten sei. Der Weg durch den Graphen für Produkt i endet mit dem Erreichen der Senke t_i, was nach spätestens $(n - 1)$ Schritten der Fall ist, da die Menge der Kanten, deren Durchfluß größer als 0 ist, o. B. d. A. als kreisfrei angenommen werden kann.

Lemma 6:
Seien $X_1, \ldots, X_n$ unabhängige Bernoulli-Zufallsvariablen mit $\Pr[X_i = 1] = p_i$ und $\Pr[X_i = 0] = (1 - p_i)$ für alle $i = 1, \ldots, n$. Sei ferner $Y = \sum_{i=1}^{n} X_i$, also $\mathrm{E}[Y] = \sum_{i=1}^{n} p_i$. Dann gilt für alle $\beta \in [0, 1]$:

$$\Pr[|Y| - E[Y]| \geq \beta \mathrm{E}[Y]] \leq 2 \exp(-0.38\beta^2 \mathrm{E}[Y]).$$

Diese Behauptung folgt aus Bemerkung (iii) zu Satz 2.

Theorem 7:
Sei eine Instanz des Mehrgüterfluß-Problems gegeben, bei der die Kapazität jeder Kante mindestens $5.2 \ln 4m$ beträgt, wobei $m := |E|$. Sei ferner eine positive Konstante $\varepsilon < \frac{\sqrt{5}-1}{2}$ gegeben. Dann liefert obiger Algorithmus eine ganzzahlige Lösung mit einem Gesamtfluß von mindestens $(1 - \varepsilon)^2 F$ mit einer Wahrscheinlichkeit von mindestens $(1 - \frac{1}{m} - 2\exp(-0.38\varepsilon^2 F))$, wobei F der Gesamtfluß in der ganzzahligen Optimallösung ist.

Beweis:
Der Algorithmus liefert immer nur eine ganzzahlige Lösung. Also kann nur einer der folgenden beiden Fälle eintreten:
1. Die Kapazität einer Kante ist überschritten.
Die Wahrscheinlichkeit, daß dieser Fall eintritt, beträgt höchstens $\frac{1}{m^2}$. Die Wahrscheinlichkeit über alle Kanten summiert ist $m \cdot \frac{1}{m^2} = \frac{1}{m}$.
2. Der Gesamtfluß ist kleiner als $(1 - \varepsilon)^2 F$.
Wir haben bereits festgestellt, daß der Gesamtfluß mindestens $(1 - \varepsilon)F$ beträgt. Die Anwendung von Lemma 6 mit $\beta = \varepsilon$ liefert die Schranke $2\exp(-0.38\varepsilon^2 F)$ für die Wahrscheinlichkeit, daß der Fluß kleiner als $(1-\varepsilon)^2 F$ ist.
Insgesamt ergibt sich also die Wahrscheinlichkeit $1 - (\frac{1}{m}) - (2\exp(-0.38\varepsilon^2 F)$, was zu zeigen war.

q. e. d.

Bemerkung:
Die Fehlerwahrscheinlichkeit läßt sich durch wiederholtes (voneinander unabhängiges) Anwenden des randomisierten Rundens vermindern. Bei den meisten Anwendungen ist nämlich das Finden der Optimallösung für das nicht-ganzzahlige LP der eigentliche Rechenaufwand. Der Schritt des randomisierten Rundens ist schnell durchführbar und kann deshalb auch leicht mehrmals wiederholt werden.

5 Covering & Packing

5.1 Allgemeine Problemstellung

Seien A eine $m \times n$-Matrix über $\mathbb{Z}_+$, $b \in \mathbb{Z}_+^m$, und seien $x, w \in \mathbb{Z}_+^n$. Dann heißt das Problem

$$\min\{w^T x | Ax \geq b\}$$

Überdeckungsproblem (covering problem) und das Problem

$$\max\{w^T x | Ax \leq b\}$$

Packproblem (packing problem).
Im Sinne der linearen Programmierung sind Überdeckungs- und Packprobleme also zueinander dual. Außerdem sind beide Problemsorten i. a. NP-vollständig.

5.2 Beispiel: Set Cover

In diesem Abschnitt wollen wir randomisiertes Runden auf einen Spezialfall der Packprobleme anwenden, nämlich auf das Problem der Mengenüberdeckung (set cover problem, SC). Sei dazu $V = \{v_1, \ldots, v_n\}$ eine Menge von Elementen, $S = \{S_1, \ldots, S_m\}$ eine Familie von Teilmengen von V. Es gibt jetzt zwei verschiedene Problemstellungen für das set cover problem.
Bei der ersten Formulierung wird jedem Element $v_i \in V$ ein Gewicht $w(v_i)$ zugeordnet. Jetzt soll eine Teilmenge V' von V mit minimalem Gewicht gefunden werden, so daß aus jeder Menge aus S mindestens ein Element in V' enthalten ist. Das Gewicht einer Menge ist dabei die Summe der Gewichte der in ihr enthaltenen Elemente.
Bei der zweiten Version wird jeder Menge $S_j \in S$ ein Gewicht $w(S_j)$ zugeordnet. Es sind nun Mengen $S_{k_1}, \ldots, S_{k_l} \in S$ mit minimalem Gesamtgewicht gesucht, deren Vereinigung ganz V überdeckt.
Es ist leicht einzusehen, daß diese beiden Probleme zueinander äquivalent sind. Wir beschäftigen uns daher nur mit dem ersten Fall, der auch als hitting set problem bekannt ist.
Wir formulieren zunächst wieder ein ganzzahliges Optimierungsproblem. Dabei bezeichne x_i, $i = 1, \ldots, n$, ob Element v_i zur Überdeckung benutzt wird ($x_i = 1$), oder nicht ($x_i = 0$).

Wir erhalten:

$$\begin{array}{lll} \text{Minimiere } \sum_{i=1}^{n} w_i \cdot x_i, & & \\ \text{so daß} & \sum_{i:v_i \in S_j} x_i \geq 1 & \text{für alle } S_j \in S \\ \text{und} & x_i \in \{0,1\} & i = 1, \ldots, n. \end{array}$$

Wir relaxieren dieses ganzzahlige Problem wieder in ein nicht-ganzzahliges LP, indem wir nicht mehr $x_i \in \{0,1\}$, sondern nur noch $x_i \in [0,1]$ fordern. Sei $\hat{x}_i$ der Wert der Variablen x_i nach der Lösung des relaxierten LP's.
Jetzt wollen wir ja wissen, welche der v_i wir für eine Überdeckung brauchen. Dafür wenden wir nun wieder randomisiertes Runden an, indem wir jede Variable x_i mit der Wahrscheinlichkeit $\hat{x}_i$ auf 1 setzen. Der Erwartungswert für das Gesamtgewicht des so entstandenen V' ist $\sum_{i=1}^{n} w_i \hat{x}_i$.
Wie groß ist jetzt die Wahrscheinlichkeit, daß wir eine Menge S_j auch wirklich überdeckt haben ? Dazu enthalte S_j die Elemente $v_1, \ldots, v_k$. Nach der ersten Bedingung gilt: $\sum_{i=1}^{k} \hat{x}_j \geq 1$. Somit ist die Wahrscheinlichkeit, daß die Menge S_j überdeckt ist:

$$1 - \prod_{i=1}^{k} (1 - \hat{x}_i) \geq 1 - \left(1 - \frac{1}{k}\right)^k \geq 1 - \frac{1}{\mathrm{e}},$$

wobei e die Eulersche Zahl 2.71828... ist. Die Wahrscheinlichkeit, S_j überdeckt zu haben, ist also mindestens so groß wie die Konstante $(1 - \frac{1}{\mathrm{e}}) \approx 0.632$.
Um die Wahrscheinlichkeit, alle Mengen aus S überdeckt zu haben, zu erhöhen, wenden wir das randomisierte Runden erneut auf diejenigen x_i an, die nicht auf 1 gesetzt wurden. Wenn wir so z. B. $t = O(\log m)$-mal das randomisierte Runden anwenden, ist die Wahrscheinlichkeit, daß eine Menge S_j nicht überdeckt wurde, höchstens $\frac{1}{2m}$. Somit ist nach den t Anwendungen des Algorithmus die Wahrscheinlichkeit, daß nicht alle Mengen aus S getroffen wurden, höchstens $\frac{1}{2}$ und der Erwartungswert des Gesamtgewichtes von V' höchstens $t \cdot \sum_{i=1}^{k} w_i \hat{x}_i,$.

Aus diesen Überlegungen ergibt sich insgesamt:

Theorem 8:
Zu jeder Instanz $\{V, S\}$ des set cover problem kann mit randomisiertem Runden eine $O(\log m)$-approximative Überdeckung mit einer Wahrscheinlichkeit von mindestens $\frac{1}{2}$ gefunden werden.

Bemerkung:
Die Schranke aus Theorem 8 ist noch nicht optimal, sie kann noch verbessert werden; jedoch kann man zeigen, daß sie höchstens $1+\ln\Delta$ sein kann, wobei Δ die maximale Anzahl von Mengen aus S ist, die mit einem einzigen Element aus V überdeckt werden können.

6 Undirected Multicut

6.1 Problemstellung

Eine Instanz des ungerichteten Multicut-Problems besteht aus einem ungerichteten Graphen $G = (V, E)$, einer auf den Kanten definierten Gewichtsfunktion $c : E \to I\!R^+$, sowie Knotenpaaren (s_i, t_i), $i = 1, \ldots, k$. Die Aufgabe lautet: Finde eine Menge von Kanten, so daß nach dem Entfernen dieser Kanten aus dem Graphen jede Quelle s_i von der zugehörigen Senke t_i getrennt ist, also nicht in derselben Zusammenhangskomponente liegt. Dabei soll die Summe über die Gewichte der entfernten Kanten minimal sein.
Auch dieses Problem ist NP-vollständig.

6.2 Zurückführung auf Set Cover

Wir können das Problem in kanonischer Weise in das set cover problem überführen. Dazu entsprechen die Kanten im Graphen den (gewichteten) Elementen aus dem SC-Problem, und jeder Pfad zwischen der Quelle s_i und der Senke t_i definiere eine Teilmenge aus S. Eine Überdeckung der Menge S durch Kanten aus G bedeutet, daß aus jedem Pfad von s_i nach t_i mindestens eine Kante entfernt wird. Die minimale Überdeckung beim set cover problem entspricht dann dem minimalen Schnitt durch G, so daß jeweils s_i und t_i getrennt werden.
Bei diesem Vorgehen wird allerdings nur ein sehr schlechter Approximationsfaktor erreicht, der in der Größenordnung von $\Omega(|V|)$ liegen kann.
Deshalb formulieren wir das ungerichtete Multicut-Problem auf eine andere Weise in das set cover problem um. Das folgende Verfahren stammt von Bertsimas und Vohra:
Seien in F alle diejenigen Teilmengen von V mit der Eigenschaft, daß für jedes $S \in F$ höchstens einer der beiden Knoten s_i und t_i, $i = 1, \ldots, k$, in S enthalten ist. Wir bezeichnen mit $\delta(S)$ die Menge der Kanten, die genau einen Endpunkt in S haben. Ferner sei $c(A)$ definiert als die Summe der Gewichte in Menge A. a_{iS} gebe an, ob sich die Quelle s_i in S befindet ($a_{iS} = 1$), oder

nicht ($a_{iS} = 0$). Analog sei b_{iS} für die Senken definiert. Die Variable $x(S)$ sei 1, wenn die Menge S für die Überdeckung benötigt wird, ansonsten 0.
Dies ergibt folgendes ganzzahlige lineare Programm:

$$
\begin{array}{lll}
\text{Minimiere } \frac{1}{2}\sum_{S\in F} c(\delta(S)) \cdot x(S), & & \\
\text{so daß} & \sum_{S\in F} a_{iS} \cdot x(S) \geq 1 & \text{für alle } s_i,\ i = 1,\dots,k \\
& \sum_{S\in F} b_{iS} \cdot x(S) \geq 1 & \text{für alle } t_i,\ i = 1,\dots,k \\
& x(S) \in \{0,1\} & \text{für alle } S \in F.
\end{array}
$$

Hier wird also versucht, sämtliche Quellen und Senken im Graphen durch Mengen aus F zu überdecken. Somit induziert eine Lösung dieses ganzzahligen LP's einen zulässigen Multicut in G, und umgekehrt.

Jetzt relaxieren wir wieder die letzte Bedingung $x(S) \in \{0,1\}$ auf $x(S) \in [0,1]$, und lösen das nicht-ganzzahlige LP, was ja in polynomieller Zeit möglich ist. An dieser Stelle wenden wir auf die so erhaltene Lösung den Randomisierungsschritt des set cover problem (vgl. Kapitel 5) an, um eine Lösung für den Multicut zu erhalten.
Weil die Anzahl der zu überdeckenden Elemente $2k$ ist, ergibt sich wie für das set cover problem ein Approximationsfaktor von $O(\log k)$. Somit erzielen wir mit randomisiertem Runden ein vergleichbar gutes Ergebnis wie der Algorithmus von Garg, Vazirani und Yannakakis, von dem wir schon vorher im Seminar gehört haben, und der deterministisch arbeitet.

Inhaltsverzeichnis

erstellt mit LaTeX2ε.