Randomisiertes Runden

Frédéric Reinhardt

7. Februar 2007

1 Einleitung

1.1 Mathematische Programmierung

Viele theoretisch wie praktisch relevante Probleme lassen sich in natürlicher Weise als Maximierungs- bzw. Minimierungsproblem einer reellwertigen Funktion unter Nebenbedingungen modellieren. Schränkt man dabei die Klasse der zugelassenen Funktionen und Nebenbedingungen geeignet ein bekommt man unterschiedliche Klassen von sogenannten Programmierungsproblemen. Für die Klasse der linearen Programme (LP), bei denen sowohl die zu optimierende Funktion als auch die Nebenbedingungen linear sein müssen, existieren effiziente Lösungsalgorithmen. Ein Großteil der uns interessierenden Optimierungsprobleme sind jedoch NP-schwer und lassen sich unter der Annahme, dass $P \neq NP$ gilt, nicht effizient als Lineares Programm darstellen. Wohl aber gibt es für diese Probleme oftmals eine Formalisierung als sog. Ganzzahliges Programm (IP). Der einzige formale Unterschied zwischen IP's und LP's besteht darin, dass der Definitionsbereich der zu optimierenden Funktion bei einem IP auf einen diskreten Bereich, wie den ganzen Zahlen, eingeschränkt ist, wohingegen bei einem LP ein kontinuierlicher Definitionsbereich, wie etwa die rationalen Zahlen, vorliegt.

1.2 Relax & Round

Die effiziente Lösbarkeit von LP's auf der einen Seite und die leichte Darstellbarkeit vieler Optimierungsprobleme als IP's auf der anderen Seite legen folgenden Lösungsansatz nahe: Gegeben sei ein kombinatorisches Optimierungsproblem. Finde eine geeignete Darstellung des Problems als IP und erweitere den Definitionsbereich der zu optimierenden Funktion, so dass ein (effizient lösbares) LP entsteht. Das so entstandene LP bezeichnet man als Relaxation des IP. Man hegt nun die Hoffnung aus einer Lösung der Relaxation eine gute Approximation des ursprünglichen IP berechnen zu können. Dazu muß die aus der Lösung der Relaxation gewonnene Lösung, ein Vektor rationaler Zahlen, auf einen ganzzahligen Vektor, der die Nebenbedingungen des IP erfüllt, gerundet werden. Besonders gute Ergebnisse erzielt man dabei, wenn man die Rundungsprozedur randomisiert. Insbesondere macht die Randomisierung die Approximationsgüte des so gerundeten Ergebnisses einer mathematischen Analyse mittels etablierter probabilistischer Techniken zugänglich. Die Technik des randomisierten Rundens stellt also ein Entwurfsschema für randomisierte Approximationsalgorithmen dar, welches auf eine große Klasse von Optimierungsproblemen anwendbar ist.

2 0.75-Approximationsalgorithmen für MAX SAT

Im Folgenden werden mittels randomisiertem Runden 0.75-Approximationsalgorithmen für MAX SAT hergeleitet.

2.1 MAX SAT als IP und seine LP-Relaxation

Eine Instanz des MAX SAT-Problems besteht aus einer Menge $\mathcal{C}$ von aussagenlogischen Klauseln $C_j \in \mathcal{C}$ über einer Menge von Variablen y_i, $i = 1 \dots n$ und dazugehörigen nichtnegativen rationalen Gewichten w_j. Ein Algorithmus, der MAX SAT exakt löst, muß zu jeder Eingabeinstanz eine Belegung der Variablen ausgeben, so dass die Summe der Gewichte aller durch diese Belegung erfüllten Klauseln maximal wird. Man sieht leicht ein, dass das folgende IP eine äquivalente Beschreibung von MAX SAT darstellt. Dabei bezeichnen I_j^+ bzw. I_j^- jeweils die Indexmengen der in der Klausel C_j positiv bzw. negativ vorkommenden Literale.

$$\begin{array}{ll} \max & \sum\limits_{C_j \in \mathcal{C}} w_j z_j \\ \text{s.d.} & \sum\limits_{i \in I_j^+} y_i + \sum\limits_{i \in I_j^-} (1 - y_i) \geq z_j \qquad \forall C_j \in \mathcal{C} \\ & y_i \in \{0,1\} \qquad 1 \leq i \leq n \\ & z_j \in \{0,1\} \qquad \forall C_j \in \mathcal{C} \end{array}$$

Erweitert man den Definitionsbereich der Variablen von der Menge $\{0,1\}$ auf das ganze Intervall $[0,1]$, so erhält man eine LP-Relaxation des Problems, welche in Polynomialzeit lösbar ist. Bezeichne (y^*, z^*) einen optimalen Lösungsvektor der Relaxation und $Z_{LP}^* := \sum_{C_j} w_j z_j^*$ das an dieser Stelle angenommene Maximum, so wie Z_{IP}^* das Maximum des ursprünglichen IP's. Da jede Lösung des IP's insbesondere eine Lösung der Relaxation ist, gilt $Z_{IP}^* \leq Z_{LP}^*$.

2.2 Eine geeignete Rundungsprozedur

Die rationale Lösung y^* soll nun geeignet gerundet werden, um zu einer ganzzahligen Lösung zurückzukehren, die möglichst nah am Optimum liegt. Ein erster Ansatz könnte darin bestehen, den Vektor y^* als Wahrscheinlichkeitsverteilung aufzufassen und jede Variable y_i unabhängig von den anderen mit der Wahrscheinlichkeit $P(y_i = 1) = y_i^*$ auf 1 und mit Wahrscheinlichkeit $P(y_i = 0) = 1 - y_i^*$ auf 0 zu setzen. Diese Idee läßt sich jedoch sogleich verallgemeinert darstellen, indem man zunächst eine Funktion $f : [0,1] \to [0,1]$ auf den Rohvektor y^* ansetzt und dann jede Variable y_j unabhängig von den anderen mit der Wahrscheinlichkeit $f(y_j^*)$ auf 1 und $1 - f(y_j^*)$ auf 0 setzt. Eine Klausel C_j wird genau dann durch diese Zufallsbelegung erfüllt, wenn mindestens ein Literal aus C_j erfüllt ist, d.h.

$$P(C_j \text{ ist erfüllt}) = 1 - P(\forall i \in I_j^+ : y_i = 0 \wedge \forall i \in I_j^- : y_i = 1) = 1 - \prod_{i \in I_j^+} (1 - f(y_i^*) \prod_{i \in I_j^-} f(y_i^*).$$

Der Erwartungswert $E[W]$ des Gewichts W der durch die Zufallsbelegung y erfüllten Klauseln berechnet sich dann wie folgt:

$$E[W] = \sum_j w_j P(C_j \text{ ist erfüllt}) = \sum_j w_j (1 - \prod_{i \in I_j^+} (1 - f(y_i^*)) \prod_{i \in I_j^-} f(y_i^*)).$$

Es geht nun darum, die Funktion f geeignet zu wählen, so dass sich der Erwartungswert gut nach unten abschätzen läßt. Eine gute Wahl bilden, wie von Goemans und Williamson in [2] herausgestellt, Funktionen, die folgende Eigenschaft haben.

Definition 1 (3/4-Eigenschaft) $f : [0,1] \to [0,1]$ *hat die 3/4-Eigenschaft, falls*

$$1 - \prod_{i=1}^{l} (1 - f(y_i)) \prod_{i=l+1}^{k} f(y_i) \geq \frac{3}{4} min(1, \sum_{i=1}^{l} y_i + \sum_{i=l+1}^{k} (1 - y_i))$$

für alle $1 \leq l \leq k$ *und alle* $y_1, y_2, ..., y_k \in [0,1]$.

Eine Funktion f mit der 3/4-Eigenschaft ermöglicht nämlich folgende Abschätzung.

$$\sum_j w_j (1 - \prod_{i \in I_j^+} (1 - f(y_i^*)) \prod_{i \in I_j^-} f(y_i^*)) \geq \sum_j w_j \frac{3}{4} (\sum_{i \in I^+} y_i^* + \sum_{i \in I^-} (1 - y_i^*)) \geq \sum_j w_j \frac{3}{4} z_j^*$$

und somit insgesamt, wegen $\frac{3}{4} \sum_j w_j z_j^* = \frac{3}{4} Z_{LP}^* \geq \frac{3}{4} Z_{IP}^*$, auch

$$E[W] \geq \frac{3}{4} Z_{LP}^* \geq \frac{3}{4} Z_{IP}^*.$$

Jede in Polynomialzeit berechenbare Funktion mit der 3/4-Eigenschaft führt so also zu einem randomisierten Polynomialzeit-3/4-Approximationsalgorithmus für MAX SAT, wobei hier noch angemerkt sei, dass sich diese Algorithmen mittels der wohlbekannten Methode der bedingten Erwartungswerte leicht derandomisieren lassen. Goemans und Williamson geben eine Reihe von einfacheren für die 3/4-Eigenschaft hinreichenden Bedingungen an [2]. Wir wollen hier nur exemplarisch eine konkrete solche Funktion vorstellen, um klarzustellen, dass es solche Funktionen auch wirklich gibt. Setze

$$f(y) := \begin{cases} 4^{y-1} & y \geq 1/2 \\ 1 - 4^{-y} & y < 1/2 \end{cases}$$

Man verifiziert leicht die Ungleichung $1 - 4^{-y} \leq f(y) \leq 4^{y-1}$ für alle $y \in [0,1]$. Da außerdem die Funktion $g(y) := 1 - 4^{-y}$ wegen $g''(y) \leq 0$ im Intervall $[0,1]$ konkav ist, liegt sie dort über der Sekante $s(y) := \frac{3}{4}y$, die ihren Funktionsgraphen in den Punkten $g(0) = 0$ und $g(1) = \frac{3}{4}$ schneidet, d.h. $1 - 4^{-y} \geq \frac{3}{4}y$ für alle $y \in [0,1]$. Dies wird nachfolgend zum Nachweis der 3/4-Eigenschaft für f benutzt. Es gilt

$$\prod_{i=1}^{l}(1 - f(y_i)) \prod_{i=l+1}^{k} f(y_i) \leq \prod_{i=1}^{l} 4^{-y_i} \prod_{i=l+1}^{k} 4^{y_i - 1} = 4^{-(\sum_{i=1}^{l} y_i + \sum_{i=l+1}^{k}(1-y_i))}$$

und somit folgt

$$1 - \prod_{i=1}^{l}(1 - f(y_i)) \prod_{i=l+1}^{k} f(y_i) \geq 1 - 4^{-(\sum_{i=1}^{l} y_i + \sum_{i=l+1}^{k}(1-y_i))} \geq \frac{3}{4}\min(1, \sum_{i=1}^{l} y_i + \sum_{i=l+1}^{k}(1 - y_i)).$$

2.3 Bemerkungen

Der Approximationsfaktor von $\frac{3}{4}$ läßt sich mit keiner verfeinerteren Analyse, die Z_{LP}^* als obere Schranke für Z_{IP}^* benutzt, verbessern, denn bei folgender MAX SAT-Instanz wird die Schranke erreicht:

$$C_1 := y_1 \vee y_2, C_2 := \neg y_1 \vee y_2, C_3 := y_1 \vee \neg y_2, C_4 := \neg y_1 \vee \neg y_2, w_1 := w_2 := w_3 := w_4 := 1.$$

Das Maximum der LP-Relaxation wird hier mit dem Wert $Z_{LP}^* = 4$ erreicht, wenn $y_1 = y_2 = \frac{1}{2}$ und $z_1 = z_2 = z_3 = z_4 = 1$ gilt, aber es sind nur drei Klauseln gleichzeitig erfüllbar, womit $Z_{IP}^* = 3$ gilt, also $Z_{IP}^* = \frac{3}{4}Z_{LP}^*$. Eine andere interessante Beobachtung ist, dass die LP-Relaxation für jede MAX SAT-Instanz, die keine Klausel mit nur einem Literal enthält, immer die optimale Lösung $y_i^* = 1/2$ und $z_j = 1$ für alle j hat. In diesem Fall kann man sich also die Berechnung des LP sparen und einfach jede Variable mit der gleichen Wahrscheinlichkeit 1/2 auf 1 setzen. Mit analogen Abschätzungen des Erwartungswertes, wie sie oben vorgenommen wurden, kann man zeigen, dass so, für Instanzen bei denen jede Klausel mindestens $k \geq 2$ Literale enthält, eine Approximationsgüte von $(1 - 1/2^k)$ erreicht wird. Mittels der Random Hyperplane-Technik, welche im nächsten Abschnitt anhand des MAX CUT-Problems demonstriert wird, konnten Feige und Goemans [3] die Approximationsgüte für MAX 2SAT sogar auf 0.931 verbessern.

3 Die Random Hyperplane-Technik

In [1] erweiterten Goemanns & Williamson den Anwendungsbereich von randomisiertem Runden auf Semidefinite Programme bzw. Vektorprogramme, einer Verallgemeinerung von Linearen Programmen, und führten eine neue randomisierte Rundungstechnik ein. Die von diesen entwickelte Random Hyperplane-Technik konnte in mehreren Fällen zur Entwicklung von Approximationsalgorithmen mit eklatant besseren Approximationsgüten als den bis dato bekannten beitragen.

3.1 Vektorprogramme und Semidefinite Programmierung

3.1.1 Integrality gap

Gegeben sei ein Optimierungsproblem P und eine Relaxation R dieses Problems. Bezeichnen jeweils $OPT_P(I)$ bzw. $OPT_R(I)$ die Optima dieser Probleme auf der Instanz I, so wird durch den Ausdruck $\alpha := inf_I \frac{OPT_P(I)}{OPT_R(I)}$ die maximal mögliche Abweichung der Relaxation vom ursprünglichen Problem gemessen (integrality gap). Benutzt man bei der Analyse der Approximationsgüte eines Algorithmus $OPT_R(I)$ als eine obere Schranke für $OPT_P(I)$ so kann der Approximationsfaktor, den man damit erreicht, also nicht größer als α sein. Im letzten Abschnitt wurde so beispielsweise gezeigt, dass $\frac{3}{4}$ für die angegebene LP-Relaxation von MAX SAT bestmöglich ist. Will man bessere Approximationsalgorithmen finden, muß man also andere Arten von Relaxationen in Betracht ziehen als Lineare Programme. Eine allgemeinere Art mathematischer Programme, die sich dafür eignen, sind Semidefinite Programme.

3.1.2 Semidefinite Programme

Definition 2 (positiv semidefinite Matrix) *Eine Matrix $X \in \mathcal{R}^{n\times n}$ heißt positiv semidefinit (psd), falls*

$$\forall a \in \mathcal{R}^n : a^T X a \geq 0.$$

Definition 3 (Semidefinites Programm) *Ein semidefinites Programm (SDP) ist ein Optimierungsproblem der Form*

$$\begin{array}{ll} \max & \sum\limits_{1\leq i,j\leq n} c_{ij}x_{ij} \\ \text{s.d.} & \sum\limits_{1\leq i,j\leq n} a_{ijk}x_{ij} = b_k \quad k = 1,2,\ldots,m \\ & X = (x_{ij}) \in \mathcal{R}^{n\times n} \textit{ ist positiv semidefinit und symmetrisch} \end{array}$$

wobei $c_{ij}, a_{ijk}, b_k \in \mathcal{R} \quad \forall i,j,k.$

SDP's lassen sich mittels der Ellipsoid-Methode bis auf einen additiven Fehler ϵ in Zeit, die polynomiell in n und logarithmisch in $\frac{1}{\epsilon}$ wächst, lösen. Da sich der additive Fehler ϵ beliebig verkleinern läßt, wird er im Folgenden vernachlässigt.

3.1.3 Vektorprogramme

Eine weitere Verallgemeinerung Linearer Programme stellen die Vektorprogramme dar. Diese haben nahezu die gleiche Form wie SDP's mit dem Unterschied, dass es sich bei den Variablen um Vektoren des $\mathcal{R}^n$ handelt und die Multiplikation zweier Vektoren die Skalarmultiplikation im $\mathcal{R}^n$ bedeutet.

Definition 4 *Ein Vektorprogramm (VP) ist ein Optimierungsproblem der Form*

$$\begin{array}{ll} \max & \sum\limits_{1 \le i,j \le n} c_{ij}(v_i \cdot v_j) \\ \text{s.d.} & \sum\limits_{1 \le i,j \le n} a_{ijk}(v_i \cdot v_j) = b_k \quad k = 1, 2, \ldots, m \\ & v_i \in \mathcal{R}^n \quad \forall i \end{array}$$

wobei $c_{ij}, a_{ijk}, b_k \in \mathcal{R} \quad \forall i, j, k.$

Jedes VP läßt sich auch als SDP formulieren und umgekehrt, was eine Konsequenz des folgenden Lemmas ist.

Lemma 1 (Charakterisierung symmetrischer psd-Matrizen) *Sei* $X \in \mathcal{R}^{n \times n}$

1. *Die folgenden Aussagen sind äquivalent:*
   a) X *ist eine symmetrische psd-Matrix.*
   b) X *läßt sich in der Form* $X = V^T V$ *mit einer Matrix* $V \in \mathcal{R}^{n \times n}$ *darstellen*
2. *Es gibt einen Polynomialzeitalgorithmus, der die unter b) angegebene Zerlegung beliebig genau approximieren kann.*

Auf den Beweis dieses Lemmas verzichten wir an dieser Stelle. Aus diesem Lemma ergibt sich jedoch folgendes Korollar, welches für unsere Zwecke interessant ist.

Korollar 1 (Äquivalenz von VP's und SDP's) *Bezeichne* P_{VP} *ein VP und* P_{SDP} *das entsprechende SDP, welches aus* P_{VP} *entsteht, indem man die Ausdrücke* $v_i \cdot v_j$ *durch Variablen* x_{ij} *ersetzt und somit* P_{VP} *als SDP auffaßt. Zu jeder Lösung* X *von* P_{SDP} *läßt sich in Polynomialzeit eine Lösung* V *von* P_{VP} *, die den gleichen Wert hat, beliebig genau approximieren und umgekehrt.*

Beweis 1 *Aus einer Lösung* $X = (x_{ij})$ *von* P_{SDP} *bekommt man nach Lemma 1 eine Matrix* $V \in \mathcal{R}^{n \times n}$ *mit* $V^T V = X$*. Die Spalten von* V *bilden dann offensichtlich eine Lösung von* P_{VP} *mit dem gleichen Wert. Umgekehrt schreibe man die Lösung* $v_1, v_2, \ldots v_n$ *von* P_{VP} *als Spalten in eine Matrix* V *und berechne* $X := V^T V$*.*

Somit sind sowohl SDP's als auch VP's effizient lösbar, was sie zu geeigneten Kandidaten für Relaxationen macht.

3.2 Formulierung von MAX CUT als Vektorprogramm

Eine Instanz des MAX CUT-Problems besteht aus einer Liste $w_{ij}, 1 \le i < j \le n$ von nichtnegativen rationalen Kantengewichten eines ungerichteten Graphen mit n Ecken (nicht vorhandene Kanten bekommen das Kantengewicht 0). Ein Algorithmus, der das MAX CUT-Problem exakt löst, muß zu jeder Eingabeinstanz eine Teilmenge S der Eckenmenge ausgeben, so dass das Gewicht der zwischen S und $\overline{S}$ verlaufenden Kanten maximal ist. Das MAX CUT-Problem läßt sich auch wie folgt als (nichtlineares) IP darstellen:

$$\begin{array}{ll} \max & \frac{1}{2} \sum\limits_{1 \leq i < j \leq n} w_{ij}(1 - y_i y_j) \\ & y_i \in \{-1, 1\} \end{array}$$

Man sieht leicht ein, dass für jede Lösung y dieses IP $S := \{i : y_i = 1\}$ eine Lösung des MAX CUT-Problems ist und umgekehrt. Betrachten wir nun die VP-Relaxation des IP, die man erhält, wenn man den Definitionsbereich der Variablen y_i auf normierte Vektoren $v_i \in \mathcal{R}^n$, d.h. Vektoren, die auf der Einheitssphäre $S^{n-1} = \{v \in \mathcal{R}^n : ||v|| = 1\}$ liegen, erweitert.

$$\begin{array}{ll} \max & \frac{1}{2} \sum\limits_{1 \leq i < j \leq n} w_{ij}(1 - v_i \cdot v_j) \\ & v_i \cdot v_i = 1 \\ & v_i \in \mathcal{R}^n \end{array}$$

Zunächst macht man sich klar, dass dies tatsächlich eine Relaxation des Problems ist, dass es also zu jeder Lösung y_i des IP auch eine gleichwertige Lösung des VP gibt. Setzt man nämlich $v_i := (y_i, 0, \ldots, 0)$, so gilt $v_i \cdot v_j = y_i y_j$, womit die v_i eine Lösung des VP mit dem gleichen Wert darstellen. Das Optimum Z^*_{VP} stellt also, wie gewünscht, eine obere Schranke für Z^*_{IP} dar, d.h. es gilt $Z^*_{VP} \geq Z^*_{IP}$.

3.3 Runden mit zufälligen Hyperebenen

Der randomisierten Rundungstechnik, die im Folgenden beschrieben werden soll, liegt folgende heuristische Beobachtung zugrunde. Bezeichne $\Theta_{ij} := \angle(v_i, v_j)$ immer den Winkel zwischen v_i und v_j. Wegen $v_i \cdot v_j = ||v_i|| ||v_j|| \cos \Theta_{ij}$ und $||v_i|| = 1$, gilt für die in der zu maximierenden Funktion vorkommenden Summanden $1 - v_i \cdot v_j = 1 - \cos \Theta_{ij}$. Dieser Ausdruck wird also umso größer, je näher der Winkel Θ_{ij} bei π liegt, d.h. je weiter die Vektoren auseinanderklaffen. Es erscheint deshalb sinnvoll, die Wahrscheinlichkeit, zwei Ecken i, j in verschiedenen Partitionen unterzubringen, von der Größe dieses Winkels abhängig zu machen. Dies erreicht man, indem man, anschaulich gesprochen, einen zufälligen Schnitt durch die Einheitssphäre macht, d.h. zufällig eine durch den Ursprung verlaufende Hyperebene wählt, und i, j genau dann in verschiedenen Partitionen unterbringt, wenn die Hyperebene die beiden Vektoren v_i, v_j voneinander trennt. Denn je größer der Winkel zwischen den beiden Vektoren ist, desto größer ist intuitiv die Wahrscheinlichkeit, dass die Hyperebene zwischen diesen beiden Vektoren verläuft. Mathematisch entspricht die zufällige Wahl einer Hyperebene der zufälligen Wahl einer Normalen $r \in S^{n-1}$, die senkrecht auf der Hyperebene steht und sie somit eindeutig bestimmt. Auf welcher Seite der durch r definierten Hyperebene ein Vektor v_i liegt ist durch das Vorzeichen des Ausdrucks $r \cdot v_i$ bestimmt. Man rundet somit die Lösung v_i des VP auf eine Lösung y_i des IP, indem man

$$y_i := \begin{cases} +1 & \text{falls } r \cdot v_i \geq 0 \\ -1 & \text{falls } r \cdot v_i < 0 \end{cases}$$

setzt.

3.4 Analyse der Approximationsgüte

Ziel ist nun wieder eine Abschätzung des erwarteten Gewichts $E[W]$, der durch obige randomisierte Rundungsprozedur erzeugten Lösung y des IP herzuleiten. Dazu muß zunächst die Wahrscheinlichkeit $P(y_i \neq y_j)$ dafür, dass zwei Ecken i, j in unterschiedlichen Partitionen landen, bzw. dass die Vektoren v_i, v_j auf verschiedenen Seiten der Hyperebene liegen, berechnet werden. Das folgende Lemma ermöglicht es, sich dabei auf den anschaulicheren 2-dimensionalen Fall zu beschränken.

Lemma 2 (Gleichverteilung der Projektion) *Ist r eine auf S^{n-1} gleichverteilte Zufallsvariable, so ist die normierte Projektion r' von r auf einen 2-dimensionalen Untervektorraum von $\mathcal{R}^n$ ebenfalls gleichverteilt auf dem Einheitskreis des Untervektorraums.*

Der Beweis von Lemma 2 folgt anschaulich aus der Rotationssymmetrie der Gleichverteilung auf S^{n-1}. Bezeichne nun r' die Projektion von r auf die von v_i und v_j aufgespannte 2-dimensionale Ebene. Dann läßt sich r schreiben als $r = r' + r''$ mit einem Vektor r'', der senkrecht auf dieser Ebene steht. Mittels der Umformung $r \cdot v_k \geq 0 \Leftrightarrow (r' + r'') \cdot v_k = r' \cdot v_k \geq 0$ für $k = i, j$ ergibt sich nun zusammen mit Lemma 1, dass die Wahrscheinlichkeit dafür, dass r v_i und v_j trennt, gleich der Wahrscheinlichkeit ist, dass dies ein zufälliger Vektor r' aus der v_i-v_j-Ebene tut. Wie aus der Abbildung ersichtlich ist, hat der Bereich auf der Einheitskreislinie, auf dem r' liegen muß, damit die rote senkrecht zu r' verlaufende Strecke zwischen den beiden Vektoren verläuft, das Maß $2\Theta_{ij}$. Da der Kreisumfang 2π beträgt, gilt somit

$$P(\operatorname{sgn}(r' \cdot v_i) \neq \operatorname{sgn}(r' \cdot v_j)) = P(y_i \neq y_j) = \frac{\Theta_{ij}}{\pi}.$$

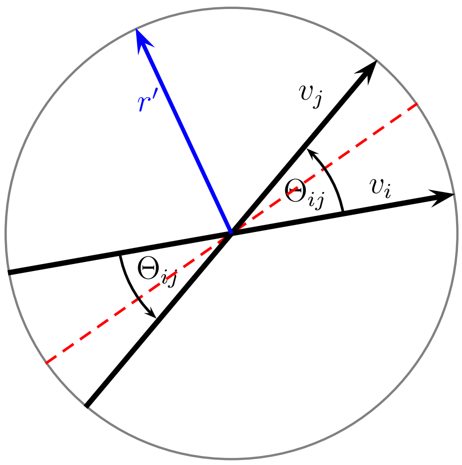

Abbildung 1: Projektion der Hyperebene in die v_i-v_j-Ebene

Lemma 3 *Es gilt die Abschätzung*

$$\min_{0 \leq \Theta \leq \pi} \frac{2}{\pi} \frac{\Theta}{1 - \cos\Theta} > 0.87856 =: \alpha.$$

Insbesondere folgt aus diesem Lemma, das Goemans und Williamson [1] numerisch verifiziert haben, $\frac{\Theta_{ij}}{\pi} > \alpha \frac{1-\cos\Theta_{ij}}{2}$. Damit erhält man schließlich die folgende untere Schranke des erwarteten Gewichts: Es gilt

$$E[W] = \sum_{1 \leq i < j \leq n} w_{ij} P(y_i \neq y_j) = \sum_{1 \leq i < j \leq n} w_{ij} \frac{\Theta_{ij}}{\pi} > \alpha \sum_{1 \leq i < j \leq n} w_{ij} \frac{1 - \cos\Theta_{ij}}{2} = \alpha Z^*_{VP}$$

also auch

$$E[W] > \alpha Z^*_{IP}.$$

Goemans und Williamson haben in ihrer Veröffentlichung [1] auch eine Möglichkeit aufgezeigt, den Algorithmus mittels der Methode der bedingten Erwartungswerte zu derandomisieren, womit sie also einen polynomialen deterministischen 0.87846 - Approximationsalgorithmus für das MAX CUT-Problem hergeleitet haben. Dies ist eine bemerkenswerte Verbesserung gegenüber dem bis dahin unübertroffenen 0.5-Approximationsalgorithmus.

4 Ausblick - Verfeinerte Techniken

4.1 Abhängiges randomisiertes Runden und schwer erfüllbare Nebenbedingungen

Bei den bisher vorgestellten Rundungstechniken wurden die einzelnen Komponenten des Lösungsvektors der Relaxation stochastisch unabhängig voneinander gerundet. Dies kann bei bestimmten schwer erfüllbaren Nebenbedingungen dazu führen, dass der gerundete Vektor nicht mehr in der Menge der zulässigen Lösungen liegt. Um dies zu berücksichtigen und somit die Technik des randomisierten Rundens auf weitere Probleme anwendbar zu machen, entwickelten Gandhi, Khuller, Parthasarathy und Srinivasan in [4] ein verfeinertes randomisiertes Rundungsschema. Wir stellen zunächst ein bestimmtes algorithmisches Problem zur Motivation vor.

4.1.1 Routing mit geringer Kantenbelastung

Eine Instanz dieses Problems besteht aus einem Graphen $G := (V, E)$ mit positiven Kantenkapazitäten $c_f > 0, \forall f \in E$, k Knotenpaaren (s_i, t_i) mit jeweils einem Fluß $\rho_i > 0$, einer Menge von Pfaden $\mathcal{P}_i$ zwischen dem Paar, sowie einer Zahl $1 \leq l_i \leq |\mathcal{P}_i|$. Die Aufgabe besteht darin, aus jedem Pfadsystem genau l_i Pfade auszuwählen, so dass die relative Kantenbelastung minimal wird. Dabei definiert man die relative Belastung einer einzelnen Kante durch das Verhältnis des die Kante belastenden Gesamtflusses zur Kantenkapazität c_f. Die relative Kantenbelastung ist dann als die maximale Belastung aller Kanten definiert. Dieses Problem läßt sich wie folgt als IP formalisieren, wobei mit $\mathcal{P}_{ij} \in \mathcal{P}_i$ jeweils der j-te Pfad aus dem i-ten Pfadsystem bezeichnet wird und die

Schreibweise $f \in \mathcal{P}_{ij}$ bedeutet, dass f eine Kante auf diesem Pfad ist.

$$\begin{array}{ll} \min & C^* \\ & \sum_{1 \leq j \leq |\mathcal{P}_i|} z_{ij} = l_i \quad \text{,i=1,...,k} \\ & \sum_{(i,j): f \in \mathcal{P}_{ij}} \rho_i z_{ij} \leq c_f C^* \quad , \forall f \in E \\ & z_{ij} \in \{0,1\} \end{array}$$

Die Äquivalenz zum oben geschilderten Problem wird deutlich, wenn man die Variablen z_{ij} folgendermaßen interpretiert:

$$z_{ij} = 1 \Leftrightarrow \text{ aus dem } i\text{-ten Pfadystem wird der } j\text{-te Pfad ausgewählt}$$

Eine natürliche LP-Relaxation des Problems erhält man, wenn man den Definitionsbereich der Variablen auf das Intervall $[0,1]$ ausdehnt. Die bisherigen Rundungsschemata lassen sich nun jedoch nicht ohne weiteres übernehmen, denn die erste Nebenbedingung, welche besagt, dass aus dem i-ten Pfadsystem jeweils genau l_i Pfade ausgewählt werden müssen, könnte durch unabhängiges Runden verletzt werden.

4.1.2 Ein abhängiges Rundungsschema

Das Rundungsschema von Gandhi, Khuller, Parthasarathy und Srinivasan ist ein randomisierter Polynomialzeitalgorithmus, der eine gegebene Liste von Zahlen $x_{ij} \in [0,1]$, welche als Kantengewichte eines bipartiten Graphen $(V = (A,B), E)$ interpretiert werden, auf Werte in $\{0,1\}$ rundet und dabei zusätzlich die nachfolgend aufgeführten 3 spezifischen Eigenschaften erfüllt.

1. **Randverteilung:** $\forall (i,j) \in E : P(X_{ij} = 1) = x_{ij}$
2. **Graderhaltung:** Für alle $i \in V$ gilt mit $d_i := \sum_{j:(i,j) \in E} x_{ij}$
$$\sum_{j:(i,j) \in E} X_{ij} \in \{\lfloor d_i \rfloor, \lceil d_i \rceil\}$$
3. **Negative Korrelation:** Für alle $i \in V$ und alle Teilmengen $S \subset E$ von Kanten, die mit i inzidieren:
$$\forall b \in \{0,1\} : P(\bigwedge_{f \in S} X_f = b) \leq \prod_{f \in S} P(X_f = b)$$

Insbesondere impliziert die zweite Eigenschaft, dass der (nicht notwendigerweise ganzzahlige) Knotengrad d_i erhalten bleibt, falls er schon vor dem Runden ganzzahlig war. Diese Eigenschaft macht das Rundungsschema auf das obige Problem anwendbar, da die erste Nebenbedingung hier durch das abhängige Runden nicht verletzt werden kann. Dies soll nur als kleiner Ausblick auf derzeitige Weiterentwicklungen auf dem Gebiet des randomisierten Rundens dienen. Für den detaillierten Rundungsalgorithmus, so wie Anwendungen zur Herleitung von Approximationsalgorithmen für Routing- und Schedulingproblemen der obigen Art, verweisen wir auf die ausführliche Veröffentlichung [4].

Literatur

[1] M.X. Goemans and D.P. Williamson, *Improved Approximation Algorithms for Maximum Cut and Satisfiability Problems Using Semidefinite Programming*, J. ACM **42** (1995), 1115–1145.

[2] M.X. Goemans and D.P. Williamson, *New 3/4-Approximation Algorithms for the Maximum Satisfiability Problem*, SIAM Journal on Discrete Mathematics **7** (1994), 656–666.

[3] Uriel Feige and Michel X. Goemans, *Approximating the Value of Two Prover Proof Systems, with Applications to MAX 2SAT and MAX DICUT*, Proceedings of theThirdIsrael Symposium on Theory of Computing and Systems (1995), 182–189.

[4] R. Gandhi, S. Khuller, S. Parthasarathy and A. Srinivasan *Dependent rounding and its applications to approximation algorithms*, Journal of the ACM **53** (2006), Issue 3, 324–360.