Anwendungen der Semidefiniten Programmierung

Minimum Bandwith Problem, Minimum Bisection

Johannes Ebke

11. September 2007

1. Minimum Bandwidth
   - Problem
   - Ziel des Algorithmus
   - Algorithmus
   - Analyse

2. Minimum Bisection
   - Problem
   - Ziel des Algorithmus
   - Algorithmus
   - Analyse

Minimum Bandwidth - Problem

- Die Ecken eines ungerichteten Graphen $G = (V, E)$ sollen auf die ersten $|V|$ natürlichen Zahlen gelegt werden
- Es soll dabei die maximal vorkommende Kantenlänge[1] minimiert werden.
- Verwendung: Hauptsächlich bei Matrixalgorithmen zur Konzentration der Einträge nahe der Diagonalen.

[1] Abstand zweier durch eine Kante verbundenen Ecken

Minimum Bandwidth - Problem

- Die Ecken eines ungerichteten Graphen $G = (V, E)$ sollen auf die ersten $|V|$ natürlichen Zahlen gelegt werden
- Es soll dabei die maximal vorkommende Kantenlänge[1] minimiert werden.
- Verwendung: Hauptsächlich bei Matrixalgorithmen zur Konzentration der Einträge nahe der Diagonalen.

[1] Abstand zweier durch eine Kante verbundenen Ecken

Minimum Bandwidth - Problem

- Die Ecken eines ungerichteten Graphen $G = (V, E)$ sollen auf die ersten $|V|$ natürlichen Zahlen gelegt werden
- Es soll dabei die maximal vorkommende Kantenlänge[1] minimiert werden.
- Verwendung: Hauptsächlich bei Matrixalgorithmen zur Konzentration der Einträge nahe der Diagonalen.

[1] Abstand zweier durch eine Kante verbundenen Ecken

Zielvorstellung

- Sei b^* die optimal erreichbare Bandbreite
- Idee: Konstruiere ein äquivalentes Problem, das wir SDP-Relaxieren können
- Ziel: Algorithmus, der eine Ordnung der Bandbreite $b = b^* \mathcal{O}(\sqrt{\frac{n}{b}} \log n)$ findet
- Bemerkung: Das Problem ist sehr schwer approximierbar, kein $(1,5 - \epsilon)$-Algorithmus möglich.

Zielvorstellung

- Sei b^* die optimal erreichbare Bandbreite
- Idee: Konstruiere ein äquivalentes Problem, das wir SDP-Relaxieren können
- Ziel: Algorithmus, der eine Ordnung der Bandbreite $b = b^*\mathcal{O}(\sqrt{\frac{n}{b}} \log n)$ findet
- Bemerkung: Das Problem ist sehr schwer approximierbar, kein $(1,5-\epsilon)$-Algorithmus möglich.

Zielvorstellung

- Sei b^* die optimal erreichbare Bandbreite
- Idee: Konstruiere ein äquivalentes Problem, das wir SDP-Relaxieren können
- Ziel: Algorithmus, der eine Ordnung der Bandbreite $b = b^* \mathcal{O}(\sqrt{\frac{n}{b}} \log n)$ findet
- Bemerkung: Das Problem ist sehr schwer approximierbar, kein $(1,5 - \epsilon)$-Algorithmus möglich.

Zielvorstellung

- Sei b^* die optimal erreichbare Bandbreite
- Idee: Konstruiere ein äquivalentes Problem, das wir SDP-Relaxieren können
- Ziel: Algorithmus, der eine Ordnung der Bandbreite $b = b^*\mathcal{O}(\sqrt{\frac{n}{b}}\log n)$ findet
- Bemerkung: Das Problem ist sehr schwer approximierbar, kein $(1,5-\epsilon)$-Algorithmus möglich.

Konstruktion eines Hilfsproblems

Idee: Ordne die Ecken auf einem Viertelkreis des Radius n in gleichem Abstand an und minimiere den maximalen euklidschen Abstand zweier verbundener Ecken.
(Zeichnung)
Minimiere die Bandbreite b unter $|v_i| = n$ und
$|v_i - v_j| \leq b \, \forall (i,j) \in E$

Relaxation des Hilfsproblems

Relaxiere Dimension von $|v_i|$, lasse Integralitätsbedingung fallen, minimiere b unter:

$$\begin{aligned} v_i v_j &\geq 0 & \forall i,j \in \{1,\dots,n\} \\ |v_i| &= n & \forall i \in \{1,\dots,n\} \\ |v_i - v_j| &\leq b & \forall (i,j) \in E \\ \frac{1}{|S|}\sum_{j\in S}(v_i - v_j)^2 &\geq \frac{1}{6}\left(\frac{|S|}{2}+1\right) & \forall S \subseteq \{1,\dots,n\}, \forall i \in \{1,\dots,n\} \end{aligned}$$

Die letzte Bedingung stellt dabei sicher, dass die Punkte weit genug ausgebreitet sind ("spreading constraint") Fällt etwas auf an diesen Bedingungen?

Das Abstandsorakel

$$\frac{1}{|S|}\sum_{j \in S}(v_i - v_j)^2 \geq \frac{1}{6}\left(\frac{|S|}{2}+1\right) \forall S \subseteq \{1,\ldots,n\}, \forall i \in \{1,\ldots,n\}$$

- Sortiere für jedes i die Klauseln $(v_i - v_j)$ nach Größe
- Für jedes $|S|$ überprüfe die größten $|S|$ Klauseln
- Da die rechte Seite nur von $|S|$ abhängt, ist dies bereits die größtmögliche Summe der Klauseln

Damit lässt sich diese Bedingung in Polynomzeit erfüllen.

Das Abstandsorakel

$$\frac{1}{|S|}\sum_{j\in S}(v_i - v_j)^2 \geq \frac{1}{6}\left(\frac{|S|}{2}+1\right) \forall S \subseteq \{1,\ldots,n\}, \forall i \in \{1,\ldots,n\}$$

- Sortiere für jedes i die Klauseln $(v_i - v_j)$ nach Größe
- Für jedes $|S|$ überprüfe die größten $|S|$ Klauseln
- Da die rechte Seite nur von $|S|$ abhängt, ist dies bereits die größtmögliche Summe der Klauseln

Damit lässt sich diese Bedingung in Polynomzeit erfüllen.

Das Abstandsorakel

$$\frac{1}{|S|}\sum_{j\in S}(v_i - v_j)^2 \geq \frac{1}{6}\left(\frac{|S|}{2}+1\right) \forall S \subseteq \{1,\ldots,n\}, \forall i \in \{1,\ldots,n\}$$

- Sortiere für jedes i die Klauseln $(v_i - v_j)$ nach Größe
- Für jedes $|S|$ überprüfe die größten $|S|$ Klauseln
- Da die rechte Seite nur von $|S|$ abhängt, ist dies bereits die größtmögliche Summe der Klauseln

Damit lässt sich diese Bedingung in Polynomzeit erfüllen.

Das Abstandsorakel

$$\frac{1}{|S|}\sum_{j\in S}(v_i - v_j)^2 \geq \frac{1}{6}\left(\frac{|S|}{2}+1\right) \forall S \subseteq \{1,\dots,n\}, \forall i \in \{1,\dots,n\}$$

- Sortiere für jedes i die Klauseln $(v_i - v_j)$ nach Größe
- Für jedes $|S|$ überprüfe die größten $|S|$ Klauseln
- Da die rechte Seite nur von $|S|$ abhängt, ist dies bereits die größtmögliche Summe der Klauseln

Damit lässt sich diese Bedingung in Polynomzeit erfüllen.

Das Abstandsorakel

$$\frac{1}{|S|}\sum_{j\in S}(v_i - v_j)^2 \geq \frac{1}{6}\left(\frac{|S|}{2}+1\right) \forall S \subseteq \{1,\dots,n\}, \forall i \in \{1,\dots,n\}$$

- Sortiere für jedes i die Klauseln $(v_i - v_j)$ nach Größe
- Für jedes $|S|$ überprüfe die größten $|S|$ Klauseln
- Da die rechte Seite nur von $|S|$ abhängt, ist dies bereits die größtmögliche Summe der Klauseln

Damit lässt sich diese Bedingung in Polynomzeit erfüllen.

Der Algorithmus

1) Löse die SDP-Relaxation für G, erhalte $v_1, \ldots, v_n$
2) Beschaffe einen zufälligen Einheitsvektor $\hat{e}$
3) Projiziere $v_1, \ldots, v_n$ auf der Linie durch $\hat{e}$
4) Wähle als Ordnung die Ordnung entlang dieser Linie

Dieser Algorithmus findet mit hoher Wahrscheinlichkeit eine Lösung der Bandbreite $b = b^* \mathcal{O}(\sqrt{\frac{n}{b}} \log n)$

Analyse - Vorgehensweise

- Schneide die Kugel des Radius n um den Ursprung in Streifen senkrecht zu $\hat{e}$ der Breite $\frac{8b\sqrt{\log n}}{\sqrt{n}}$
- Mit hoher Wahrscheinlichkeit kreuzt keine Kante in G mehr als 2 Streifen (Lemma 1)
- Ebenfalls mit hoher Wkt. fallen alle Punkte bei der Projektion auf $\hat{e}$ auf die mittleren n/b Streifen.
- Durch Čebyšev zeigen wir dass höchstwahrscheinlich in keinen Streifen mehr als $\mathcal{O}(\sqrt{nb}\log n)$ Punkte fallen. (Lemma 2)
- Damit ist die maximale Bandbreite bei dieser Ordnung $b\mathcal{O}(\sqrt{n/b}\log n)$

Analyse - Vorgehensweise

- Schneide die Kugel des Radius n um den Ursprung in Streifen senkrecht zu ê der Breite $\frac{8b\sqrt{\log n}}{\sqrt{n}}$
- Mit hoher Wahrscheinlichkeit kreuzt keine Kante in G mehr als 2 Streifen (Lemma 1)
- Ebenfalls mit hoher Wkt. fallen alle Punkte bei der Projektion auf ê auf die mittleren n/b Streifen.
- Durch Čebyšev zeigen wir dass höchstwahrscheinlich in keinen Streifen mehr als $\mathcal{O}(\sqrt{nb}\log n)$ Punkte fallen. (Lemma 2)
- Damit ist die maximale Bandbreite bei dieser Ordnung $b\mathcal{O}(\sqrt{n/b}\log n)$

Analyse - Vorgehensweise

- Schneide die Kugel des Radius n um den Ursprung in Streifen senkrecht zu $\hat{e}$ der Breite $\frac{8b\sqrt{\log n}}{\sqrt{n}}$
- Mit hoher Wahrscheinlichkeit kreuzt keine Kante in G mehr als 2 Streifen (Lemma 1)
- Ebenfalls mit hoher Wkt. fallen alle Punkte bei der Projektion auf $\hat{e}$ auf die mittleren n/b Streifen.
- Durch Čebyšev zeigen wir dass höchstwahrscheinlich in keinen Streifen mehr als $\mathcal{O}(\sqrt{nb}\log n)$ Punkte fallen. (Lemma 2)
- Damit ist die maximale Bandbreite bei dieser Ordnung $b\mathcal{O}(\sqrt{n/b}\log n)$

Analyse - Vorgehensweise

- Schneide die Kugel des Radius n um den Ursprung in Streifen senkrecht zu $\hat{e}$ der Breite $\frac{8b\sqrt{\log n}}{\sqrt{n}}$
- Mit hoher Wahrscheinlichkeit kreuzt keine Kante in G mehr als 2 Streifen (Lemma 1)
- Ebenfalls mit hoher Wkt. fallen alle Punkte bei der Projektion auf $\hat{e}$ auf die mittleren n/b Streifen.
- Durch Čebyšev zeigen wir dass höchstwahrscheinlich in keinen Streifen mehr als $\mathcal{O}(\sqrt{nb}\log n)$ Punkte fallen. (Lemma 2)
- Damit ist die maximale Bandbreite bei dieser Ordnung $b\mathcal{O}(\sqrt{n/b}\log n)$

Analyse - Vorgehensweise

- Schneide die Kugel des Radius n um den Ursprung in Streifen senkrecht zu $\hat{e}$ der Breite $\frac{8b\sqrt{\log n}}{\sqrt{n}}$
- Mit hoher Wahrscheinlichkeit kreuzt keine Kante in G mehr als 2 Streifen (Lemma 1)
- Ebenfalls mit hoher Wkt. fallen alle Punkte bei der Projektion auf $\hat{e}$ auf die mittleren n/b Streifen.
- Durch Čebyšev zeigen wir dass höchstwahrscheinlich in keinen Streifen mehr als $\mathcal{O}(\sqrt{nb}\log n)$ Punkte fallen. (Lemma 2)
- Damit ist die maximale Bandbreite bei dieser Ordnung $b\mathcal{O}(\sqrt{n/b}\log n)$

Keine Kante in G kreuzt mehr als zwei Streifen

Lemma

Für $v \in \mathbb{R}^n$ und einen zufälligen Einheitsvektor $\hat{e}$ gilt:

$$P\left(|v\hat{e}| \leq \frac{c}{\sqrt{n}}|v|\right) \geq 1 - e^{-c^2/4}$$

Verteilung der Ecken bei Zufallsprojektion

Lemma

Erfülle $v_1, \ldots, v_n$ die SDP-Bedingungen 1, 2 und 4. Sei X die Anzahl der Punkte die bei zufälliger Projektion in einem Intervall I der Länge W landen. Dann gilt:

$$E[X] = \mathcal{O}(W\sqrt{n}) \text{ und } E[X^2] = \mathcal{O}(W^2 n \log n)$$

Weiter gilt nach Čebyšev:

$$\frac{b}{4n} \geq P(X > \sqrt{4n/b}\sqrt{W^2 n \log n})$$

Nochmal der Algorithmus

1) Löse die SDP-Relaxation für G, erhalte $v_1, \ldots, v_n$
2) Beschaffe einen zufälligen Einheitsvektor $\hat{e}$
3) Projiziere $v_1, \ldots, v_n$ auf der Linie durch $\hat{e}$
4) Wähle als Ordnung die Ordnung entlang dieser Linie

Dieser Algorithmus findet mit hoher Wahrscheinlichkeit eine Lösung der Bandbreite $b = b^* \mathcal{O}(\sqrt{\frac{n}{b}} \log n)$

Das Minimum Bisection Problem

Definition

Eine Bisektion vom Graphen $G = (V, E)$ ist eine Partition von V in die Mengen S und $\bar{S}$ mit $|S| = |V|/2$. Die gesuchte minimale Bisektionsgrösse $b(G) = min_S|E(S, \bar{S})|$ ist die minimale Anzahl der Kanten von S nach $\bar{S}$.

Solche Bisektionen sind wichtig z.B. für Clustering, jedoch sehr schwer zu nähern - die "globale" Bedingung, dass S und $\bar{S}$ gleichmächtig sind ist hinderlich.

Modellierung des Problems durch Zufallsgraphen

Es ist extrem schwer, Algorithmen auf der Menge aller Graphen zu testen bzw. zu analysieren. Man schränkt sich daher auf ein Subset ein:

- Wähle $V, n, S \subset V$ mit $|V| = n$ und $|S| = n/2$
- Verbinde zwei Punkte mit Wahrscheinlichkeit p falls beide in S oder $\bar{S}$ liegen
- Verbinde zwei Punkte mit Wahrscheinlichkeit q falls einer in S und der andere in $\bar{S}$ liegt

Die am Anfang gewählte Menge S ist nun mit hoher Wahrscheinlichkeit die minimale Bisektion, falls gilt:

$$p - q \geq c\sqrt{p \log n / n}$$

Der nun entstehende Graph sei G_{Rand}

Modellierung des Problems durch Zufallsgraphen

Es ist extrem schwer, Algorithmen auf der Menge aller Graphen zu testen bzw. zu analysieren. Man schränkt sich daher auf ein Subset ein:

- Wähle $V, n, S \subset V$ mit $|V| = n$ und $|S| = n/2$
- Verbinde zwei Punkte mit Wahrscheinlichkeit p falls beide in S oder $\bar{S}$ liegen
- Verbinde zwei Punkte mit Wahrscheinlichkeit q falls einer in S und der andere in $\bar{S}$ liegt

Die am Anfang gewählte Menge S ist nun mit hoher Wahrscheinlichkeit die minimale Bisektion, falls gilt:

$$p - q \geq c\sqrt{p \log n / n}$$

Der nun entstehende Graph sei G_{Rand}

Modellierung des Problems durch Zufallsgraphen

Es ist extrem schwer, Algorithmen auf der Menge aller Graphen zu testen bzw. zu analysieren. Man schränkt sich daher auf ein Subset ein:

- Wähle $V, n, S \subset V$ mit $|V| = n$ und $|S| = n/2$
- Verbinde zwei Punkte mit Wahrscheinlichkeit p falls beide in S oder $\bar{S}$ liegen
- Verbinde zwei Punkte mit Wahrscheinlichkeit q falls einer in S und der andere in $\bar{S}$ liegt

Die am Anfang gewählte Menge S ist nun mit hoher Wahrscheinlichkeit die minimale Bisektion, falls gilt:

$$p - q \geq c\sqrt{p \log n / n}$$

Der nun entstehende Graph sei G_{Rand}

Modellierung des Problems durch Zufallsgraphen

Es ist extrem schwer, Algorithmen auf der Menge aller Graphen zu testen bzw. zu analysieren. Man schränkt sich daher auf ein Subset ein:

- Wähle $V, n, S \subset V$ mit $|V| = n$ und $|S| = n/2$
- Verbinde zwei Punkte mit Wahrscheinlichkeit p falls beide in S oder $\bar{S}$ liegen
- Verbinde zwei Punkte mit Wahrscheinlichkeit q falls einer in S und der andere in $\bar{S}$ liegt

Die am Anfang gewählte Menge S ist nun mit hoher Wahrscheinlichkeit die minimale Bisektion, falls gilt:

$$p - q \geq c\sqrt{p \log n / n}$$

Der nun entstehende Graph sei G_{Rand}

Modellierung des Problems durch Zufallsgraphen

Es ist extrem schwer, Algorithmen auf der Menge aller Graphen zu testen bzw. zu analysieren. Man schränkt sich daher auf ein Subset ein:

- Wähle $V, n, S \subset V$ mit $|V| = n$ und $|S| = n/2$
- Verbinde zwei Punkte mit Wahrscheinlichkeit p falls beide in S oder $\bar{S}$ liegen
- Verbinde zwei Punkte mit Wahrscheinlichkeit q falls einer in S und der andere in $\bar{S}$ liegt

Die am Anfang gewählte Menge S ist nun mit hoher Wahrscheinlichkeit die minimale Bisektion, falls gilt:

$$p - q \geq c\sqrt{p \log n/n}$$

Der nun entstehende Graph sei G_{Rand}

Eine neue Bedrohung: Der monotone Gegner

- Wähle $V, n, S \subset V$ mit $|V| = n$ und $|S| = n/2$
- Verbinde zwei Punkte mit Wahrscheinlichkeit p falls beide in S oder $\bar{S}$ liegen
- Verbinde zwei Punkte mit Wahrscheinlichkeit q falls einer in S und der andere in $\bar{S}$ liegt

Ein monotoner Gegner darf nun Kanten des ersten Typs hinzufügen und Kanten des zweiten Typs entfernen.
Damit kann er nicht die optimale Lösung ändern. Der nun entstehende Graph sei G

Suche: Lösung

- Gesucht ist nun eine zunächst beliebige Funktion $h(G)$ mit $h(G) = b(G)$ oft.
- Erinnerung: Die Bisektionsgrösse $b(S) = |E(S, \bar{S})|$ ist die Anzahl der Kanten von S nach $\bar{S}$.
- Ansatz: SDP-Relaxierung!
- Erhalten einen Algorithmus, der $h(G_{Rand}) = b(G_{Rand})$ mit Wahrscheinlichkeit $1 - 2n^{-5}$ erfüllt

Suche: Lösung

- Gesucht ist nun eine zunächst beliebige Funktion $h(G)$ mit $h(G) = b(G)$ oft.
- Erinnerung: Die Bisektionsgrösse $b(S) = |E(S, \bar{S})|$ ist die Anzahl der Kanten von S nach $\bar{S}$.
- Ansatz: SDP-Relaxierung!
- Erhalten einen Algorithmus, der $h(G_{Rand}) = b(G_{Rand})$ mit Wahrscheinlichkeit $1 - 2n^{-5}$ erfüllt

Suche: Lösung

- Gesucht ist nun eine zunächst beliebige Funktion $h(G)$ mit $h(G) = b(G)$ oft.
- Erinnerung: Die Bisektionsgrösse $b(S) = |E(S, \bar{S})|$ ist die Anzahl der Kanten von S nach $\bar{S}$.
- Ansatz: SDP-Relaxierung!
- Erhalten einen Algorithmus, der $h(G_{Rand}) = b(G_{Rand})$ mit Wahrscheinlichkeit $1 - 2n^{-5}$ erfüllt

Suche: Lösung

- Gesucht ist nun eine zunächst beliebige Funktion $h(G)$ mit $h(G) = b(G)$ oft.
- Erinnerung: Die Bisektionsgrösse $b(S) = |E(S, \bar{S})|$ ist die Anzahl der Kanten von S nach $\bar{S}$.
- Ansatz: SDP-Relaxierung!
- Erhalten einen Algorithmus, der $h(G_{Rand}) = b(G_{Rand})$ mit Wahrscheinlichkeit $1 - 2n^{-5}$ erfüllt

Vorarbeit vor dem Relaxen

Definition

h heißt **nach unten beschränkt**, falls. $h(G) \leq b(G)$
für jeden Graph G
h heißt **monoton beschränkt**, falls für G und G^+ mit $G^+ = G \oplus e$
gilt:

$$h(G) \leq h(G^+) \leq h(G) + 1$$

Die SDP-Relaxierung

Suche $X \in \mathbb{R}^n$ mit:

1) $\forall i, x_{ii} = 1$
2) $\sum_{ij} x_{ij} = 0$
3) X Symmetrisch und positiv definit
4) Definiere $h(G) = min_X h_X(G)$
5) mit $h_X(G) = \sum_{(i,j)\in E, i<j} \frac{1-x_{ij}}{2}$

"Robustheit" von h

Wir zeigen zunächst, dass die Funktion h einen monotonen Angreifer ignoriert.

Lemma

Die durch das SDP-Verfahren gegebene Funktion h hat eine obere Schranke und ist monoton beschränkt.

Güte von h

Lemma

Für genügend großes c ist die Funktion h "wahrscheinlich gut", i.e. ist mit Wahrscheinlichkeit $1 - 2n^{-5}$ gleich b über G_{Rand} falls gilt:

$$p - q \geq c\sqrt{p \log n / n}$$

Beweis.

Beweis: Benutze das Dual des SDP
Dies ist eine Maximierung, der Maximalwert muss dabei kleiner sein als der Minimalwert des ursprünglichen SDPs, aus diesem folgt $h(G_{Rand}) \geq b(G_{Rand})$. Zusammen mit der Robustheit ist die Gleichheit gezeigt. Die höchstwahrscheinlich richtige Lösung ist dabei auch eine des SDPs. $\square$

Rückblick

Minimale Bandbreite:

- Übertragung auf ein Hilfsproblem
- SDP Relaxierung
- Beweis der Güte eines randomisierten Algorithmus

Minimale Bisektion:

- Aufstellung eines Modells für zufällige Graphen
- SDP Relaxierung für eine Hilfsfunktion
- Betrachtung der Güte eines Algorithmus auf einer eingeschränkten Graphenmenge

Rückblick

Minimale Bandbreite:

- Übertragung auf ein Hilfsproblem
- SDP Relaxierung
- Beweis der Güte eines randomisierten Algorithmus

Minimale Bisektion:

- Aufstellung eines Modells für zufällige Graphen
- SDP Relaxierung für eine Hilfsfunktion
- Betrachtung der Güte eines Algorithmus auf einer eingeschränkten Graphenmenge

Rückblick

Minimale Bandbreite:

- Übertragung auf ein Hilfsproblem
- SDP Relaxierung
- Beweis der Güte eines randomisierten Algorithmus

Minimale Bisektion:

- Aufstellung eines Modells für zufällige Graphen
- SDP Relaxierung für eine Hilfsfunktion
- Betrachtung der Güte eines Algorithmus auf einer eingeschränkten Graphenmenge

Rückblick

Minimale Bandbreite:

- Übertragung auf ein Hilfsproblem
- SDP Relaxierung
- Beweis der Güte eines randomisierten Algorithmus

Minimale Bisektion:

- Aufstellung eines Modells für zufällige Graphen
- SDP Relaxierung für eine Hilfsfunktion
- Betrachtung der Güte eines Algorithmus auf einer eingeschränkten Graphenmenge

Rückblick

Minimale Bandbreite:

- Übertragung auf ein Hilfsproblem
- SDP Relaxierung
- Beweis der Güte eines randomisierten Algorithmus

Minimale Bisektion:

- Aufstellung eines Modells für zufällige Graphen
- SDP Relaxierung für eine Hilfsfunktion
- Betrachtung der Güte eines Algorithmus auf einer eingeschränkten Graphenmenge

Rückblick

Minimale Bandbreite:

- Übertragung auf ein Hilfsproblem
- SDP Relaxierung
- Beweis der Güte eines randomisierten Algorithmus

Minimale Bisektion:

- Aufstellung eines Modells für zufällige Graphen
- SDP Relaxierung für eine Hilfsfunktion
- Betrachtung der Güte eines Algorithmus auf einer eingeschränkten Graphenmenge

Literaturhinweise

[BKRV00] Avrim Blum, Goran Konjevod, R. Ravi, Santosh Vempala
Semi-definite relaxations for minimum bandwidth and other vertex-ordering problems
TCS 235, pp. 25¿42, 2000

[FK01] Uriel Feige, Joe Kilian
Heuristics for semirandom Graph Problems
JCSS 63, pp. 639¿671, 2001

[KARP02] Marek Karpinski
On Approximability of Minimum Bisection Problem
citeseer.ist.psu.edu/article/karpinski02approximability.htm