Approximationsalgorithmen

Facility Location
K-Median

Cheng, Wei

12. Juli

Facility Location

Definition

Gegeben:

- mögliche Standorte $F = \{F_1 , F_2 \dots F_m\}$
- Städte $C = \{C_1 , C_2 \dots C_n\}$
- Eröffnungskosten f_i für F_i
- Verbindungskosten c_{ij} zwischen F_i und C_j
  - *Dreiecksungleichung*

Definition

Gesucht:

- Eröffne Teilmenge $I \subseteq F$ von Standorten
- Funktion $\phi : C \to I$
  - verbinde jede Stadt mit offenem Standort.
- Gesamtkosten minimieren
  - Eröffnungskosten der Standorte
  - Verbindungskosten zwischen Stadt und Standort

Primales/ Duales LP

Primal:

Minimiere:

$$\sum_{i \in F, j \in C} c_{ij} x_{ij} + \sum_{i \in F} f_i y_i$$

Nebenbedienungen:

$$\sum_{i \in F} x_{ij} \geq 1 \quad , \quad j \in C$$

$$y_i - x_{ij} \geq 0 \quad , \quad i \in F, j \in C$$

$$x_{ij} \geq 0 \quad , \quad i \in F, j \in C$$

$$y_i \geq 0 \quad , \quad i \in F$$

Dual:

Maximiere:

$$\sum_{j \in C} \alpha_j$$

Nebenbedienungen:

$$\alpha_j - \beta_{ij} \leq c_{ij} \quad , \quad i \in F, j \in C$$

$$\sum_{j \in C} \beta_{ij} \leq f_i \quad , \quad i \in F$$

$$\alpha_j \geq 0 \quad , \quad j \in C$$

$$\beta_{ij} \geq 0 \quad , \quad i \in F, j \in C$$

Verwandte Arbeiten

- Hochbaum [1982] O(log n)
- Shmoys, Tardos and Aardal [1998] 3,16
- Jain and Vazirani [2001] 3
- Mahdian, Ye and Zhang [2004] 1,52

- Guha and Khuller [1998]
  untere Schranke 1,463

Primal-Dual basierter Algorithmus

- Phase 1
  - Öffnen mehrere Standorte, und verbinden alle Städte mit Standorten. (Eine Stadt kann mit mehr einem Standort verbinden.)
- Phase 2
  - Schließen unnötige Standorte, und verbinden alle Städte mit nur einem Standort.

Algorithmus (Phase 1)

- Am Anfang
  - Zeit 0
  - Alle Stadte sind *unverbunden*
  - Alle Variablen = 0

Algorithmus (Phase 1)

- Erhöhe α_j der *unverbundenen* Städte *j* um 1
- Wenn $\alpha_j = c_{ij}$ von Kante, ist Kante (i,j) *bezahlt* .
  => Duale Variable β_{ij} wird erhöht, laut $\alpha_j - \beta_{ij} \leq c_{ij}$.
- β_{ij} bezahlt Eröffnungskosten f_i für F_i
  - Kanten (i, j) mit $\beta_{ij} > 0$ heißen *besonders*.
- Wenn $\sum_j \beta_{ij} = f_i$, ist Standort i *bezahlt*.
  - Standort i ist *temporär geöffnet*.

Algorithmus (Phase 1)

- Stadt mit *bezahlter* Kante zu eröffnetem Standort F_i ist *verbunden*.
- Standort i ist *Verbindungszeuge* dieser Städte.
- α_j *verbundener* Städte werden nicht mehr erhöht.
- Sobald *unverbundene* Stadt j *bezahlte* Kante zu eröffnetem Standort i hat:
  - Stadt j ebenfalls *verbunden*.
  - Standort i ist V*erbindungszeuge* von j.
  - Achtung: Hier ist $\beta_{ij} = 0$, und Kante (i, j) *unbesonders*.

Algorithmus (Phase 1)

- Phase 1 endet, wenn alle Städte *verbunden*.
- Eine Stadt zu mehreren Standorten verbunden.
- Zu viele Standorte geöffnet.

Primal-Dual basierter Algorithmus

- Phase 1
  - Öffnen mehrere Standorte, und verbinden alle Städte mit Standorten. (Eine Stadt kann mit mehr einem Standort verbinden.)
- Phase 2
  - Schließen unnötige Standorte, und verbinden alle Städte mit nur einem Standort.

Algorithmus (Phase 2)

- F_t – *temporär geöffnete* Standorte
- T – durch *besondere Kanten* induzierter Teilgraph
- T^2 – $\forall (u,v) \in T \Rightarrow (u,v) \in T^2$

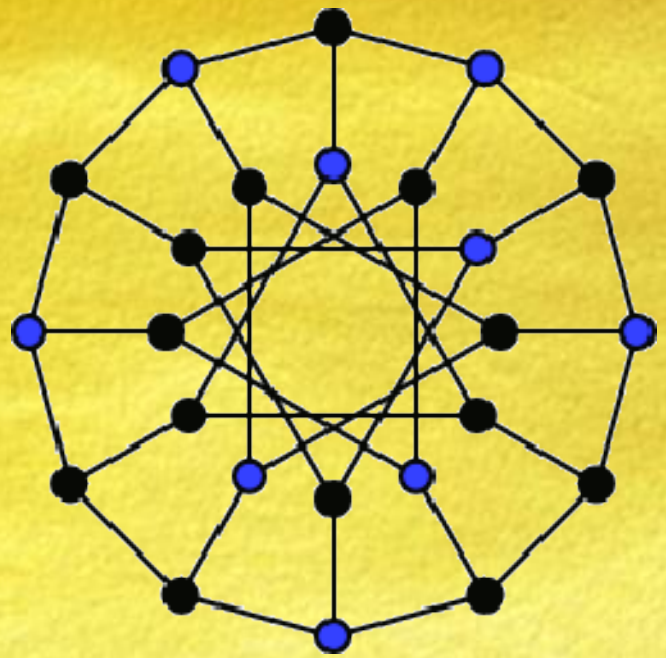
  $\forall (u,v) \in T \wedge (v,w) \in T \Rightarrow (u,w) \in T^2$
- H – von F_t induzierter Graph in T^2

Algorithmus (Phase 2)

- Wähle *I* als *Maximal Independent Set* in *H*
- Maximum Independent Set (NP-schwer)
- Maximal Independent Set:

  einfacher Greedy-Algorithmus

- Alle Standorte in *I* werden geöffnet.



Algorithmus (Phase 2)

- $\mathbf{F}_j = \{i \in F_t \mid \beta_{ij} > 0\}$, für jede Stadt j
- *I* independent set => höchstens ein Standort in $\mathbf{F}_j$ *geöffnet*.
- Fall 1: Ein Standort $i \in \mathbf{F}_j$ ist *geöffnet*.
  - $\Phi(j) = i$, und Stadt j ist *direkt verbunden*.
- Fall 2: Alle Standorte in $\mathbf{F}_j$ sind *ungeöffnet*.

Algorithmus (Phase 2)

- Fall 2: Alle Standorte in F_j sind *ungeöffnet*.
  Betrachte bezahlte Kante (i',j), $\beta_{i'j} = 0$
  i' ist V*erbindungszeuge* von j
  - Fall 2.1: $i' \in I$
    - Stadt j ist *direkt verbunden*. $\Phi(j) = i'$
    - Achtung: Bis jetzt ist die Lösung Optimal.
  - Fall 2.2: $i' \notin I$
    - i eröffneter Nachbar von *i'* in Graph H.
    - $\Phi(j) = i$, und Stadt j ist *indirekt verbunden*.

Algorithmus (Phase 2)

- (I, Φ) ist Lösung.
- $x_{ij} = 1 \Leftrightarrow \Phi(j) = i$
- $y_i = 1 \Leftrightarrow i \in I$

Analyse

- Definition: $\alpha_j = \alpha_j^f + \alpha_j^e$
- j *indirekt verbunden*
  - $\alpha_j^f = 0 \qquad \alpha_j^e = \alpha_j$
- j *direkt verbunden*
  - $\alpha_j^f = \beta_{ij} \qquad \alpha_j^e = c_{ij}$ für $\Phi(j) = i$

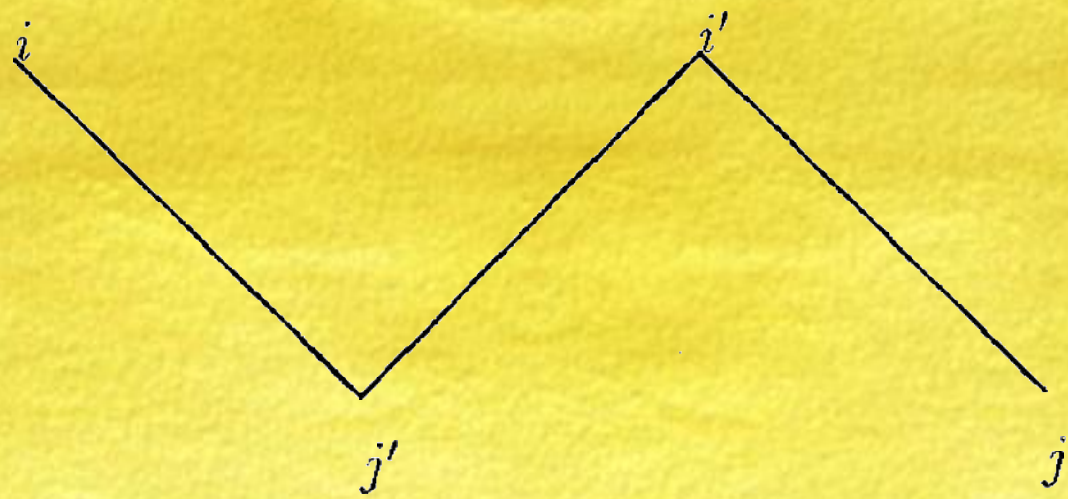
Analyse

- Lemma 1: $\sum_{j:\Phi(j)=i} \alpha_j^f = f_i$
- Folgerung 2: $\sum_{j \in C} \alpha_j^f = \sum_{i \in I} f_i$
- Lemma 3: Für eine *indirekt verbunden* Stadt j:

$$c_{ij} \leq 3\alpha_j^e \qquad i = \Phi(j)$$

Analyse

- i' V*erbindungszeuge* von Stadt j
- j verbindet *indirekt zu* i $=>$ $(i, i') \in H$
- (i, j') und (i', j') sind beide *besondere* Kanten.



Analyse

- Satz 4: $\sum_{i \in F, j \in C} c_{ij} x_{ij} + 3 \sum_{i \in F} f_i y_i \leq 3 \sum_{j \in C} \alpha_j$

- Beweis:
  - $c_{ij} \leq 3\alpha_j^e$ (Lemma 3) => $\sum_{i \in F, j \in C} c_{ij} x_{ij} \leq 3 \sum_{j \in C} \alpha_j^e$
  - $\sum_{j \in C} \alpha_j^f = \sum_{i \in I} f_i$

Zusammenfassung

- Implementierung mit Priority Queue
- Jede Kante wird max. 2-mal betrachtet.
  - *bezahlt*
  - *verbunden*
- Der Algorithmus berechnet in O(m log m) Zeit eine 3-Approximation für Facility Location.


K-Median

Definition

Gegeben:

- Eröffnungskosten $f_i = 0$ für F_i
- max. K Standorte eröffnen

Gesucht:

- Gesamtkosten minimieren

Primales/ Duales LP

Primal:

Minimieren:

$$\sum_{i \in F, j \in C} c_{ij} x_{ij}$$

Nebenbedienungen:

$$\sum_{i \in F} x_{ij} \geq 1 \quad , \quad j \in C$$

$$y_i - x_{ij} \geq 0 \quad , \quad i \in F, j \in C$$

$$\sum_{i \in F} -y_i \geq -k \quad ,$$

$$x_{ij} \geq 0 \quad , \quad i \in F, j \in C$$

$$y_i \geq 0 \quad , \quad i \in F$$

Dual:

Maximieren:

$$\sum_{j \in C} \alpha_j - zk$$

Nebenbedienungen:

$$\alpha_j - \beta_{ij} \leq c_{ij} \quad , \quad i \in F, j \in C$$

$$\sum_{j \in C} \beta_{ij} \leq z \quad , \quad i \in F$$

$$\alpha_j \geq 0 \quad , \quad j \in C$$

$$\beta_{ij} \geq 0 \quad , \quad i \in F, j \in C$$

$$z \geq 0 \quad ,$$

Verwandte Arbeiten

- Bartal [1998] O(log n log log n)
- Charikar [1999] 6+2/3
- Jain and Vazirani [2001] 6
- Arya [2002] 3+2/p

Laufzeit O(n^p)

Grundidee

- z: Eröffnungskosten eines Standorts
- Suchen geeignetes z, benutzen anschließend Algorithmus für Facility Location.
- Wir suchen $\sum_{i \in F} y_i = k$

Grundidee

- z = 0 => alle Standorte werden geöffnet.
- z sehr groß => nur ein Standort wird geöffnet.
- $z \in [0, nc_{\max}]$
- Finde z mit binärer Suche.

Grundidee

- z_1 und z_2 $\quad z_1 - z_2 \leq c_{\min} /(12n_c^2)$
- Finden entspricht $k_1<k$ und $k_2>k$
  - (x^s,y^s) $\quad \sum_{i \in F} y_i^s = k_1$
  - (x^l,y^l) $\quad \sum_{i \in F} y_i^l = k_2$

- $(x,y) = a\,(x^s,y^s) + b\,(x^l,y^l)$

  $k = a\ k_1 + b\ k_2$

  $a = (k_2-k)/(k_2-k_1)$ $\quad b = (k-k_1)/(k_2-k_1)$

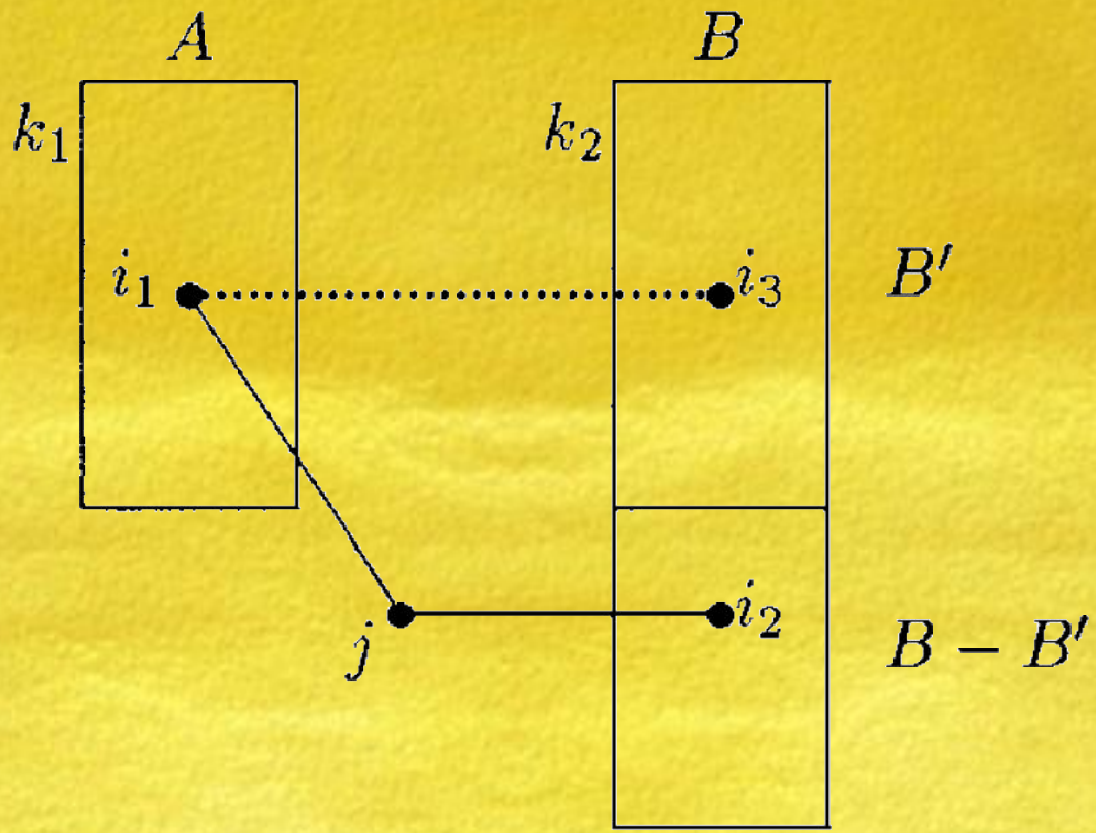
Lemma 5

- Die Kosten von (x,y) sind höchstens $(3+1/n_c)$ mal so groß wie die einer optimalen (nicht notwendig ganzzahligen) Lösung des K-Median Problems .

- Beweisskizze:

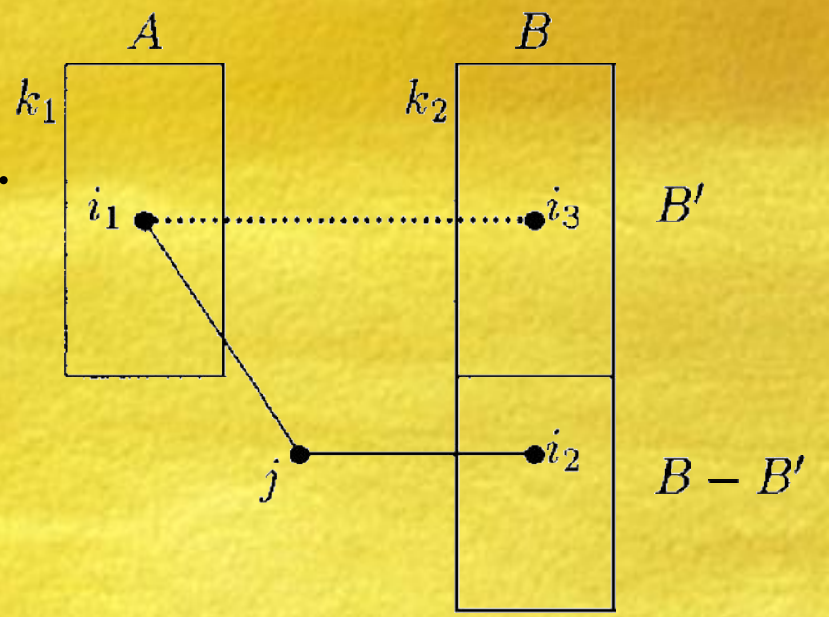
$$\sum_{i \in F,\, j \in C} c_{ij} x_{ij} \leq \left(3 + \frac{1}{n_c}\right)\left(\sum_{j \in C} \alpha_j - z_1 k\right)$$

Randomisiertes Runden



Randomisiertes Runden

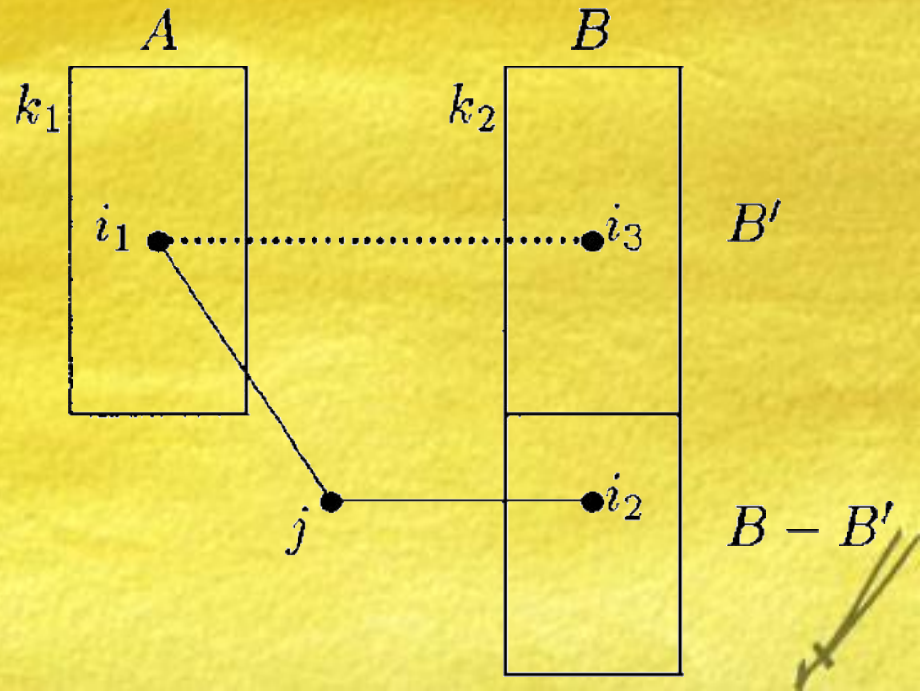
- Mit Wahrscheinlichkeit a werden alle Standorte in A geöffnet . Mit b = 1-a, alle Standorte in B'.
- Außerdem k-k_1 Standorte in B-B' werden zufällig geöffnet.
- I ist Menge der geöffneten Standorte, | I | = k.



Randomisiertes Runden

- Definiere $\Phi : C \to I$:

  für Stadt j, $i_1 \in A \quad i_2 \in B$
- Fall 1: $i_2 \in B'$ => verbinde zu vorhandenem Standort
- Fall 2: $i_2 \in B - B'$
  - $i_3 \in B'$ ist nächster Standort von i_1
  - Verbinde zu i_1 oder i_2 oder i_3
- $\text{cost}(j) = ac_{i_1 j} + bc_{i_2 j}$



Randomisiertes Runden

- Lemma 6: erwartete Kosten von Stadt j

$$E[c_{\Phi(j)j}] \le (1+\max(a,b))\text{cost}(j)$$

- Fall 1: $E[c_{\Phi(j)j}] = \text{cost}(j) = ac_{i_1 j} + bc_{i_2 j}$
- Fall 2:
  - i_2 geöffnet : wahrscheinlichkeit b
  - i_2 nicht geöffnet und i_1 geöffnet : $(1-b)a = a^2$
  - beide nicht öffnet: $(1-b)(1-a) = ab$

- $E[c_{\Phi(j)j}] \le bc_{i_2 j} + a^2 c_{i_1 j} + abc_{i_3 j}$

Approximation

- $E\left[\sum_{i\in F, j\in C} c_{ij} x_{ij}^{k}\right] \le (1+\max(a,b))\left(\sum_{i\in F, j\in C} c_{ij} x_{ij}\right)$

- $a \le 1-1/n_c$ ($k_1 = k-1$ und $k_2 = n_c$)
- $b \le 1-1/k$ ($k_1 = 1$ und $k_2 = k+1$)
- $(1+\max(a,b)) \le 2-1/n_c$

- Approximationsgarantie ist $(2-1/n_c)(3+1/n_c) < 6$

Laufzeit

- Binäre Suche in $[0, nc_{\max}]$ bis $z_1 - z_2 \leq c_{\min}/(12n_c^2)$
- $O(\log(c_{\max}/c_{\min}) + \log n)$ Proben
- Gesamtlaufzeit $O(m \log m(\log(c_{\max}/c_{\min}) + \log n))$

Zusammenfassung

- Der Algorithmus berechnet in O(m log m) Zeit eine 3-Approximation für Facility Location.

- Unser Algorithmus berechnet in $O(m \log m(\log(c_{\max} / c_{\min}) + \log n))$ Zeit eine 6-Approximation für K-Median.


Vielen Dank!