Näherungsalgorithmen

(Approximationsalgorithmen)

WiSe 2008/09 in Trier

Henning Fernau
Universität Trier
fernau@uni-trier.de

Näherungsalgorithmen Gesamtübersicht

- Organisatorisches
- Einführung / Motivation
- Grundtechniken für Näherungsalgorithmen
- Approximationsklassen (Approximationstheorie)

Grundtechniken

- Greedy-Verfahren
- Partitionsprobleme
- Lokale Suche
- Lineares Programmieren
- Dynamisches Programmieren

Lokale Suche ist ein beliebte Heuristik mit folgendem Schema:

1. Initialisiere y mit (irgendeiner) zulässigen Lösung

2. Solange es in der „Nachbarschaft“ von y eine Lösung z gibt, die besser als y ist, setze $y := z$.

3. Liefere (lokales Optimum) y zurück.

Notwendig zum Algorithmenentwurf:

a) ein Verfahren, um den in Schritt 1 erforderlichen Startwert (d.h., um irgendeine zulässige Lösung) zu finden sowie

b) einen geeigneten *„Nachbarschaftsbegriff“* auf dem Raum der zulässigen Lösungen.

Lokale Suche als Näherungsalgorithmusstrategie:

(i) Schritt 1 (siehe a)) geht polynomiell.

(ii) Die gemäß b) definierte Nachbarschaft lässt sich in Polynomzeit auf bessere Lösungen hin untersuchen (Schritt 2).

(iii) Der Gesamtalgorithmus findet nach polynomiell vielen Schritten ein lokales Optimum.

Größtmöglicher (Kanten-)Schnitt (Maximum Cut MC)

I : Graph $G = (V, E)$

$S : \{\{V_1, V_2\} \mid V_1, V_2 \subseteq V, V_1 \cap V_2 = \emptyset, V_1 \cup V_2 = V\}$

$m : |C(V_2, V_2)|$, mit

$$C(V_1, V_2) = \{e \in E \mid e \cap V_1 \neq \emptyset \wedge e \cap V_2 \neq \emptyset\}$$

opt : max

Lokale Suche für MC:

ad a): Nimm $V_1 = \emptyset, V_2 = V$ (ist also trivial)

ad b) $\mathcal{N}(\{V_1, V_2\}) = \{\{U_1, U_2\} \in S \mid \exists v \in V :$
$U_1 \triangle V_1 = U_2 \triangle V_2 = \{v\}\}$

(Hamming-Nachbarschaft)

$\triangle$ bezeichnet die symmetrische Differenz von Mengen, d.h.:

$A \triangle B = A \setminus B \cup B \setminus A.$

Satz: Ist $G = (V, E)$ eine Instanz von MC und ist $\{V_1, V_2\}$ ein lokales Optimum bzgl. der obigen Nachbarschaft $\mathcal{N}(\{V_1, V_2\})$ mit dem Wert $m_{\mathcal{N}}(G)$, dann gilt:

$$m^*(G)/m_{\mathcal{N}}(G) \leq 2$$

Beweis: Ist $m := |E|$, so gilt sicher $m^*(G) \leq m$. Wir zeigen jetzt:

$$m_{\mathcal{N}}(G) \geq \frac{m}{2}.$$

Ist $m_i := |E(G(V_i))|$, $i = 1, 2$, so gilt:

$$m = m_1 + m_2 + m_{\mathcal{N}}(G) \qquad (*)$$

Ist $E_{i,u} = \{v \in V_i \mid \{u, v\} \in E\}$, $i = 1, 2$, und ist $m_{i,u} := |E_{i,u}|$, so gilt wegen der lokalen Optimalität von $\{V_1, V_2\}$:

$$m_{1,u} \leq m_{2,u}, \quad \text{falls } u \in V_1 \text{ (sonst } \{V_1 \setminus \{u\}, V_2 \cup \{u\}\} \text{ „besser“)}$$

$$m_{1,u} \geq m_{2,u}, \quad \text{falls } u \in V_2 \text{ (sonst } \{V_1 \cup \{u\}, V_2 \setminus \{u\}\} \text{ „besser“)}$$

Daraus folgt:

$$\sum_{u \in V_1} (m_{1,u} - m_{2,u}) = 2[!] \cdot m_1 - m_{\mathcal{N}}(G) \leq 0$$

(die Kanten zwischen V_1-Knoten werden doppelt gezählt) sowie

$$\sum_{u \in V_2} (m_{2,u} - m_{1,u}) = 2 \quad \cdot m_2 - m_{\mathcal{N}}(G) \leq 0;$$

zusammen

$$\rightsquigarrow m_1 + m_2 - m_{\mathcal{N}}(G) \leq 0$$

$$\overset{(*)}{\rightsquigarrow} m \leq 2m_{\mathcal{N}}(G) \qquad \square$$

Lineares Programmieren

Eine LP-Aufgabe ist gegeben durch:
n **reellwertige** Variablen $\vec{x} = (x_1, \ldots, x_n)$,
eine lineare Zielfunktion, die es zu minimieren oder zu maximieren gilt
unter der Einschränkung, dass die Lösung m gegebenen linearen Ungleichungen genügt.
Wie wir noch zeigen werden, lässt sich jedes lineare Programm wie folgt formulieren:

$$\begin{array}{l} \text{Ggb: } \vec{c} \in \mathbb{R}^n, \vec{b} \in \mathbb{R}^m, A \in \mathbb{R}^{m \times n} \\ \text{minimiere } \vec{c}^{\,T} \cdot \vec{x} \\ \text{unter den Bedingungen } A\vec{x} \geq \vec{b}, \\ \qquad\qquad\qquad\qquad \vec{x} \geq \vec{0}. \end{array} \qquad (*)$$

Bekannte **Lösungsverfahren** für LP:

1. *Simplexverfahren* von Dantzig
   (exponentiell (!) aber schnell),

2. *Ellipsoidmethode* von Khachian
   (polynomiell aber oft langsam).

Wir nehmen diese Algorithmen als “Black Box” gegeben an.
Es gibt gute Softwarepakete für LP-Aufgaben, z.B. CPLEX.

Äquivalente Formulierungen I

Ggb: $\vec{c} \in \mathbb{R}^n, \vec{b} \in \mathbb{R}^m, \vec{b}' \in \mathbb{R}^l, A \in \mathbb{R}^{m \times n}, A' \in \mathbb{R}^{l \times n}, r \leq n$
minimiere $\vec{c}^T \cdot \vec{x}$
unter den Bedingungen

$$A\vec{x} \geq \vec{b},$$
$$A'\vec{x} = \vec{b}'$$
$$x_1 \geq 0, \ldots, x_r \geq 0$$

Überführung in die Form $(*)$:

- Jede Gleichheitsbedingung $\vec{a}' \cdot \vec{x} = b_i'$ kann durch zwei lineare Ungleichungen ausgedrückt werden: $\vec{a}' \cdot \vec{x} \geq b_i'$ und $-\vec{a}' \cdot \vec{x} \geq -b_i'$.

- Jede Variable x_j ohne Vorzeichenbeschränkung wird durch zwei Variablen x_j^+, x_j^- ersetzt mit $x_j^+ \geq 0, x_j^- \geq 0$; jedes Vorkommen von x_j im ursprünglichen Linearen Programm wird durch $(x_j^+ - x_j^-)$ ersetzt.

Äquivalente Formulierungen II

Ggb: $\vec{c} \in \mathbb{R}^n, \vec{b} \in \mathbb{R}^m, A \in \mathbb{R}^{m \times n}$
minimiere $\vec{c}^T \cdot \vec{x}$
unter $A\vec{x} = \vec{b}$,
und $\vec{x} \geq 0$

Dazu ist jede Ungleichung $\vec{a} \cdot \vec{x} \geq b_i$ durch zwei Bedingungen $\vec{a} \cdot \vec{x} - s_i = b_i$ und $s_i \geq 0$ zu ersetzen; die neu eingeführte Variable s_i heißt auch *Schlupfvariable*.

Äquivalente Formulierungen III

Durch Übergang von $\vec{c}$ auf $(-\vec{c})$, von $\vec{b}$ auf $(-\vec{b})$ und von A auf $(-A)$ lässt sich auch die folgende Maximierungsaufgabe leicht in die Form $(*)$ bringen:

Ggb: $\vec{c} \in \mathbb{R}^n, \vec{b} \in \mathbb{R}^m, A \in \mathbb{R}^{m \times n}$

maximiere $\vec{c}^{\,T} \cdot \vec{x}$
unter $A \cdot \vec{x} \leq \vec{b}$,
$\vec{x} \geq 0$

$\rightsquigarrow$ minimiere $(-\vec{c})^T \cdot \vec{x}$
unter $(-A) \cdot \vec{x} \geq -\vec{b}$,
$\vec{x} \geq 0$

Duale Programme

Primal	*Dual*
Bedingung i: $\sum_{j=1}^{n} a_{ij}x_j \geq b_i$	Variable y_i: $y_i \geq 0$
Variable x_j: $x_j \geq 0$	Bedingung $\sum_{i=1}^{m} a_{ij}y_i \leq c_j$
minimiere $\vec{c}^{\,T}\vec{x}$	maximiere $\vec{b}^{\,T}\vec{y}$

Dabei gehen wir von einem primalen Programm der Form $(*)$ aus mit
$A = (a_{ij} \mid 1 \leq i \leq m, 1 \leq j \leq n)$,
$\vec{x} = (x_j \mid 1 \leq j \leq n)$,
$\vec{b} = (b_j \mid 1 \leq j \leq n)$ und
$\vec{c} = (c_i \mid 1 \leq i \leq m)$.

Der **Dualitätssatz** besagt:

$$\min\left\{\vec{c}^{\,T} \cdot \vec{x} \mid \vec{x} \geq 0, A\vec{x} \geq \vec{b}\right\} = \max\left\{\vec{b}^{\,T} \cdot \vec{y} \mid \vec{y} \geq 0, \vec{y}^{\,T}A \leq \vec{c}\right\}.$$

Ganzzahliges Programmieren und Relaxation

Aus $(*)$ (oder einer der besprochenen Umformulierungen) entsteht ein ganzzahlige lineares Programm, indem zusätzlich gefordert wird, dass zulässige Lösungen stets nur ganzzahlig sind oder gar nur bestimmte ganze Zahlen als Lösung akzeptiert werden.

Während LP (wie gesagt) in Polynomzeit lösbar ist,
ist ILP (integer LP) im Allgemeinen ein NP-hartes Problem.

Es ist für uns jedoch **von Interesse**, da

- viele Optimierungsaufgaben sich als ILP formulieren lassen und

- die Vernachlässigung (*Relaxation*) der Ganzzahligkeitsbedingungen zu einem LP führt, dessen Lösung eine Schranke für das ursprüngliche ILP bietet; die weiter gehende Hoffnung ist, dass man durch einfache Operationen wie Runden aus der LP-Lösung eine möglichst gute ILP-Lösung generieren kann.

Das **gewichtete Knotenüberdeckungsproblem** lässt sich wie folgt als ILP ausdrücken:

$$\begin{array}{ll} \text{minimiere} & \sum_{v_i \in V} c_i x_i \\ \text{unter} & x_i + x_j \geq 1 \text{ für } \{v_i, v_j\} \in E \\ & x_i \in \{0, 1\} \text{ für } v_i \in V \end{array}$$

Ausgegangen wird dabei von einem Graphen $G = (V, E)$, $V = \{v_1, \ldots, v_n\}$ mit Knotenkosten $c_i (c_i \geq 0)$ für v_i. Die Variable x_i „entspricht“ dem Knoten v_i und die Bedingung $x_i + x_j \geq 1$ „entspricht“ der Kante $\{v_i, v_j\}$.

Daher ist eine Knotenmenge $C \subseteq V$, gegeben durch $C = \{v_i \mid x_i = 1\}$, genau dann eine Knotenüberdeckung, wenn für alle Kanten $\{v_i, v_j\}$ die Ungleichung $x_i + x_j \geq 1$ gilt.

Da wir das Knotenüberdeckungsproblem als NP-hart kennen, folgt aus dieser Modellierung auch sofort die NP-Härte von ILP.

Eine mögliche Relaxation ist die Folgende:

minimiere	$\sum_{v_i \in V} c_i x_i$
unter	$x_i + x_j \geq 1$ für $\{v_i, v_j\} \in E$
und	$x_i \geq 0,\ i = 1, \ldots, n$

$\rightsquigarrow$ RoundVC$(G = (V, E), \vec{c})$

1. Bestimme die ILP-Formulierung des VC-Problems.

2. Bestimme die zugehörige LP-Formulierung durch Relaxation.

3. Es sei $\vec{x}^*_G = (x^*_1, \ldots, x^*_n)$ eine optimale Lösung für die Relaxation.

4. Liefere $\{v_i \mid x^*_i \geq 0,5\}$ zurück.

Satz 1: RoundVC findet eine zulässige Lösung eines gewichteten Knotenüberdeckungsproblems vom Wert $m_{LP}(G, \vec{c})$ mit

$$m_{LP}(G, \vec{c}) / m^*(G, \vec{c}) \leq 2.$$

Beweis: Die von RoundVC gelieferte Lösung $\{v_i \mid x_i^* \geq 0,5\}$ ist eine Knotenüberdeckung. Wäre nämlich $e = \{v_{i_1} v_{i_2}\}$ eine nicht abgedeckte Kante, so wäre $x_{i_1}^* + x_{i_2}^* < 1$, im Widerspruch zur Annahme, $\vec{x}_G^*$ sei Lösung der Relaxations-LP-Aufgabe.
Umgekehrt ist jede Lösung der ILP-Aufgabe auch eine Lösung der LP-Aufgabe, sodass insbesondere für die optimalen Lösungen gilt:

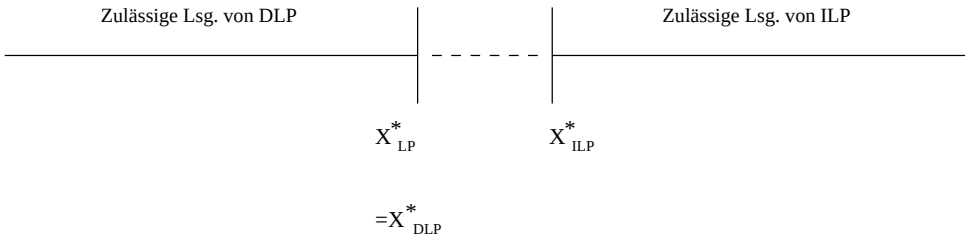
$$m^*(G, \vec{c}) \geq m_{LP}^*(G, \vec{c}) = \sum_{v_i \in V} c_i x_i^*.$$

Weiter ist:

$$
\begin{aligned}
m_{LP}(G, \vec{c}\,) &= \sum_{x_i^* \geq 0{,}5\ 1 \leq i \leq n} c_i \\
&\leq 2 \cdot \sum_{1 \leq i \leq n} c_i x_i^* \\
&= 2 \cdot \vec{c}^{\,T} \cdot \vec{x}_G^* \\
&= 2 \cdot m_{LP}^*(G, \vec{c}\,) \\
&\leq 2 \cdot m^*(G, \vec{c}\,) \qquad \square
\end{aligned}
$$

Primal-Duale Algorithmen für ILP

Betrachte ein (Minimierungs-)ILP, die zugehörige Relaxation LP sowie dessen (Maximierungs-)Dual-DLP mit optimalen Werten $x^*_{\mathrm{ILP}}, x^*_{\mathrm{LP}}$ und x^*_{DLP}. Aus dem Dualitätssatz ergibt sich für eine Verteilung der Werte von Lösungen auf der Zahlengeraden:



Tatsächlich wird bei den Primal-Dual-Näherungsalgorithmen für ILPs vermieden, explizit die optimale Schranke $x^*_{\mathrm{LP}} = x^*_{\mathrm{DLP}}$ zu berechnen; es werden statt dessen sukzessive bessere zulässige Lösungen des DLPs berechnet.

Ein Primal-Dual-Algorithmus für VC PDVC $(G, \vec{c})$

1. Bestimme die ILP-Formulierung des VC-Problems.

2. Bestimme die zugehörige LP-Formulierung durch Relaxation.

3. Bestimme die zugehörige DLP-Formulierung.

Bemerkung: DLP_{VC} lautet:

maximiere $\sum_{\{v_i,v_j\}\in E} y_{\{i,j\}}$

unter $\sum_{j:\{v_i,v_j\}\in E} y_{\{i,j\}} \leq c_i$ für jedes $v_i \in V$

$y_{\{i,j\}} \geq 0$ für alle $\{v_i, v_j\} \in E$

4. Setze jede duale Variable $y_{\{i,j\}}$ auf Null.
   [Dies ist eine zulässige Lösung des DLP.]

5. $V' := \emptyset$; [V' enthält die Knoten der Überdeckungsapprox.]

6. Solange V' keine Knotenüberdeckung ist, tue:

   6a. Wähle Kante $e = \{v_i, v_j\}$ mit $e \cap V' = \emptyset$.

   6b. $\varepsilon := \min \left\{ c_i - \sum_{k:\{v_i,v_k\}\in E} y_{\{i,k\}},\ c_j - \sum_{k:\{v_j,v_k\}\in E} y_{\{j,k\}} \right\}$

   6c. $y_{\{i,j\}} := y_{\{i,j\}} + \varepsilon$

   6d. Wenn $\sum_{k=\{v_i,v_k\}\in E} y_{i,k} = c_i$
   dann $V' := V' \cup \{v_i\}$
   sonst $V' := V' \cup \{v_j\}$

Satz 2: PDVC findet eine zulässige Lösung des gewichteten Knotenüberdeckungsproblems vom Wert $m_{PD}(G, \vec{c})$ mit

$$\frac{m_{PD}(G, \vec{c})}{m^*(G, \vec{c})} \leq 2.$$

Beweis: Die Schleife in Schritt 6 wird erst verlassen, wenn in V' eine zulässige Lösung „aufgesammelt“ wurde.
Schritt 6d gewährleistet für die von PDVC gelieferte Überdeckung V':

$$\forall v_i \in V' : \sum_{j:\{v_i,v_j\}\in E} y_{\{i,j\}} = c_i$$

Also ist:

$$\begin{aligned}
m_{PD}(G, \vec{c}) &= \sum_{v_i \in V'} c_i \\
&= \sum_{v_i \in V'} \sum_{j:\{v_i,v_j\} \in E} y_{\{i,j\}} \\
&\leq \sum_{v_i \in V} \sum_{j:\{v_i,v_j\} \in E} y_{\{i,j\}} \\
&= 2 \cdot \sum_{\{v_i,v_j\} \in E} y_{\{i,j\}} \\
&\leq 2 \cdot m^*(G, \vec{c}),
\end{aligned}$$

denn $\sum_{\{v_i,v_j\} \in E} y_{\{i,j\}}$ ist der Wert einer zulässigen Lösung des DLP. □

Bemerkung: Der soeben vorgestellte Algorithmus PDVC ist tatsächlich der schon früher behandelte von Bar-Yehuda und Even. Um dies einzusehen, setze man anfangs in PDVC $w(v_i) := c_i$ als Gewichtsreduzierung und als Schritt „6e“ füge man $w(v_i) := w(v_i) - \varepsilon$ und $w(v_j) := w(v_j) - \varepsilon$ ein. Dann gilt offenbar stets:

$$w(v_i) = c_i - \sum_{k:\{v_i,v_k\}\in E} y_{\{i,k\}}.$$

In termini des zweiten Kapitels ist ε daher der Nicht-Null-Wert der Gewichtsreduktionsfunktion $\delta_{\{v_i,v_j\}}$.

Dieser Zusammenhang lässt sich noch allgemeiner darstellen, siehe auch:
R. Bar-Yehuda, D. Rawitz: On the equivalence between the primal-dual scheme and the local-ratio technique. Seiten 24–35 IN: M. Goemanns u.a.: Approximation, Randomization and Combinatorical Optimization, LNCS 2129, Springer, 2001.

Primal-Dual Graphentheoretisch

Primale Welt	Duale Welt
ungew. Knotenüberdeckungsproblem	Maximum Matching
$\Downarrow$ ILP-Formulierung	$\Uparrow$ Graphentheoretische Interpretation
minimiere $\sum_{v_i \in V} x_i$ unter $x_i + x_j \geq 1$ für $\{v_i, v_j\} \in E$ $x_i \in \{0,1\}$ für $v_i \in V$	maximiere $\sum_{\{v_i,v_j\} \in E} y_{\{i,j\}}$ unter $\sum_{j:\{v_i,v_j\} \in E} y_{\{i,j\}} \leq 1$ für $v_i \in V$ $y_{\{i,j\}} \in \{0,1\}$ für $\{v_i, v_j\} \in E$
$\Downarrow$ Relaxation	$\Uparrow$ ganzzahlige Spezialisierung
minimiere $\sum_{v_i \in V} x_i$ unter $x_i + x_j \geq 1$ für $\{v_i, v_j\} \in E$ $x_i \geq 0$ für $v_i \in V$	maximiere $\sum_{\{v_i,v_j\} \in E} y_{\{i,j\}}$ unter $\sum_{j:\{v_i,v_j\} \in E} y_{\{i,j\}} \leq 1$ für $v_i \in V$ $y_{\{i,j\}} \geq 0$ für $\{v_i, v_j\} \in E$

$\Rightarrow$

Dualisierungssatz liefert **Folgerung**: Die Größe eines Maximum Matching ist stets eine untere Schranke für die Größe einer ungewichteten Knotenüberdeckung.
Achtung: Maximum Matching kann man in Polynomzeit berechnen.

Primal-Dual Graphentheoretisch

Die Überlegungen aus der vorigen Folie liefern auch noch einen weiteren Algorithmus für das (ungewichtete) Knotenüberdeckungsproblem:

```
Berechne ein Maximum Matching.
Gib die im Matching enthaltene Knotenmenge aus.
```

Satz 3: Maximum Matching induziert eine 2-Approximation für das ungewichtete Knotenüberdeckungsproblem.

Beweis: Faktor 2 ergibt sich aus dem Dualitätssatz.

Gäbe es eine Kante, die auf diese Weise nicht abgedeckt wäre, so heißt das, dass sie mit keiner Kante aus dem Matching einen Knoten gemeinsam hat; also könnte man sie zum Matching hinzunehmen, um ein größeres zu bekommen, im Widerspruch zu dessen Maximalität.