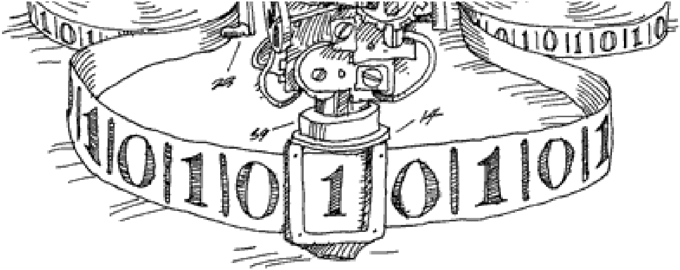
Karlsruher Institut für Technologie

Theoretische Grundlagen der Informatik

Tutorium 9

Institut für Theoretische Informatik



Motivation

- Manche Optimierungsprobleme sind $\mathcal{NP}$-schwer
  $\Rightarrow$ (wahrscheinlich) keine effiziente Lösung.
- Wir wollen trotzdem eine schnelle Lösung!
  (muss dafür nicht optimal sein)
  $\Rightarrow$ Approximation

Motivation

- Manche Optimierungsprobleme sind $\mathcal{NP}$-schwer
  $\Rightarrow$ (wahrscheinlich) keine effiziente Lösung.
- Wir wollen trotzdem eine schnelle Lösung!
  (muss dafür nicht optimal sein)
  $\Rightarrow$ Approximation

Motivation

- Manche Optimierungsprobleme sind $\mathcal{NP}$-schwer
  $\Rightarrow$ (wahrscheinlich) keine effiziente Lösung.
- Wir wollen trotzdem eine schnelle Lösung!
  (muss dafür nicht optimal sein)
  $\Rightarrow$ Approximation

Gütegarantien

- Vergleich der Worst-Case-Lösung mit der optimalen Lösung
- Π Optimierungsproblem, I Instanz von Π:
  OPT(I) ist der Wert der/einer optimalen Lösung.
- $\mathcal{A}(I)$ ist der Wert der (Approximations-)Lösung, die Algorithmus $\mathcal{A}$ für I liefert.

Absolute Gütegarantie (Differenzengarantie)

- $|\mathrm{OPT}(I) - \mathcal{A}(I)| \leq K$ für alle $I \in D_\Pi$
- Wünschenswerteste Gütegarantie
- Für $\mathcal{NP}$-vollständige Probleme gibt es oft
  keinen absoluten Approximationsalgorithmus.

Gütegarantien

- Vergleich der Worst-Case-Lösung mit der optimalen Lösung
- Π Optimierungsproblem, I Instanz von Π:
  OPT(I) ist der Wert der/einer optimalen Lösung.
- $\mathcal{A}(I)$ ist der Wert der (Approximations-)Lösung, die Algorithmus $\mathcal{A}$ für I liefert.

Absolute Gütegarantie (Differenzengarantie)

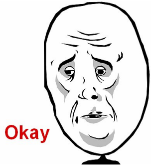
- $|\mathrm{OPT}(I) - \mathcal{A}(I)| \leq K$ für alle $I \in D_\Pi$
- Wünschenswerteste Gütegarantie
- Für $\mathcal{NP}$-vollständige Probleme gibt es oft
  keinen absoluten Approximationsalgorithmus.

Gütegarantien

- Vergleich der Worst-Case-Lösung mit der optimalen Lösung
- Π Optimierungsproblem, I Instanz von Π:
  OPT(I) ist der Wert der/einer optimalen Lösung.
- $\mathcal{A}(I)$ ist der Wert der (Approximations-)Lösung, die Algorithmus $\mathcal{A}$ für I liefert.

Absolute Gütegarantie (Differenzengarantie)

- $|\mathrm{OPT}(I) - \mathcal{A}(I)| \leq K$ für alle $I \in D_\Pi$
- Wünschenswerteste Gütegarantie
- Für $\mathcal{NP}$-vollständige Probleme gibt es oft keinen absoluten Approximationsalgorithmus.



Gütegarantien

- Vergleich der Worst-Case-Lösung mit der optimalen Lösung
- Π Optimierungsproblem, I Instanz von Π:
  OPT(I) ist der Wert der/einer optimalen Lösung.
- $\mathcal{A}(I)$ ist der Wert der (Approximations-)Lösung, die Algorithmus $\mathcal{A}$ für I liefert.

Relative Gütegarantie

- $$\mathcal{R}_{\mathcal{A}}(I) := \begin{cases} \frac{\mathcal{A}(I)}{\mathrm{OPT}(I)} & \text{falls } \Pi \text{ Minimierungsproblem} \\ \\ \frac{\mathrm{OPT}(I)}{\mathcal{A}(I)} & \text{falls } \Pi \text{ Maximierungsproblem} \end{cases}$$
- Ist $\mathcal{R}_{\mathcal{A}}(I) \leq 1 + \varepsilon$ für alle I, so heißt $\mathcal{A}$ Algorithmus mit relativer Gütegarantie, insbesondere ε-*approximativ*.

Beispiel: Greedy-Algorithmus $\mathcal{A}$ für KNAPSACK

- Sortiere die Gegenstände nach „Gewichtsdichten“ $p_i := \frac{c_i}{w_i}$
- Nimm so viele Gegenstände mit möglichst großer Gewichtsdichte, bis der Rucksack voll ist.
- Laufzeit in $\mathcal{O}(n \log n)$

Theorem
$\mathcal{A}$ ist ein 1-Approximationsalgorithmus für KNAPSACK.

Proof.
O.B.d.A. sei $w_1 \leq W$. Es gilt $\mathcal{A}(I) \geq c_1 \cdot \left\lfloor \frac{W}{w_1} \right\rfloor$ für alle I.

$$\mathrm{OPT}(I) \leq c_1 \cdot \frac{W}{w_1} \leq c_1 \cdot \left(\left\lfloor \frac{W}{w_1} \right\rfloor + 1 \right) \leq 2 \cdot c_1 \cdot \left\lfloor \frac{W}{w_1} \right\rfloor \leq 2 \cdot \mathcal{A}(I)$$

Also $\mathcal{R}_{\mathcal{A}}(I) \leq 2$. □

Beispiel: Greedy-Algorithmus $\mathcal{A}$ für KNAPSACK

- Sortiere die Gegenstände nach „Gewichtsdichten“ $p_i := \frac{c_i}{w_i}$
- Nimm so viele Gegenstände mit möglichst großer Gewichtsdichte, bis der Rucksack voll ist.
- Laufzeit in $\mathcal{O}(n \log n)$

Theorem

$\mathcal{A}$ ist ein 1-Approximationsalgorithmus für KNAPSACK.

Proof.

O.B.d.A. sei $w_1 \leq W$. Es gilt $\mathcal{A}(I) \geq c_1 \cdot \left\lfloor \frac{W}{w_1} \right\rfloor$ für alle I.

$$\mathrm{OPT}(I) \leq c_1 \cdot \frac{W}{w_1} \leq c_1 \cdot \left(\left\lfloor \frac{W}{w_1} \right\rfloor + 1 \right) \leq 2 \cdot c_1 \cdot \left\lfloor \frac{W}{w_1} \right\rfloor \leq 2 \cdot \mathcal{A}(I)$$

Also $\mathcal{R}_{\mathcal{A}}(I) \leq 2$. □

Beispiel: Greedy-Algorithmus $\mathcal{A}$ für KNAPSACK

- Sortiere die Gegenstände nach „Gewichtsdichten" $p_i := \frac{c_i}{w_i}$
- Nimm so viele Gegenstände mit möglichst großer Gewichtsdichte, bis der Rucksack voll ist.
- Laufzeit in $\mathcal{O}(n \log n)$

Theorem

$\mathcal{A}$ ist ein 1-Approximationsalgorithmus für KNAPSACK.

Proof.

O.B.d.A. sei $w_1 \leq W$. Es gilt $\mathcal{A}(I) \geq c_1 \cdot \left\lfloor \frac{W}{w_1} \right\rfloor$ für alle I.

$$\mathrm{OPT}(I) \leq c_1 \cdot \frac{W}{w_1} \leq c_1 \cdot \left(\left\lfloor \frac{W}{w_1} \right\rfloor + 1 \right) \leq 2 \cdot c_1 \cdot \left\lfloor \frac{W}{w_1} \right\rfloor \leq 2 \cdot \mathcal{A}(I)$$

Also $\mathcal{R}_{\mathcal{A}}(I) \leq 2$. □

Beispiel: Greedy-Algorithmus $\mathcal{A}$ für KNAPSACK

- Sortiere die Gegenstände nach „Gewichtsdichten“ $p_i := \frac{c_i}{w_i}$
- Nimm so viele Gegenstände mit möglichst großer Gewichtsdichte, bis der Rucksack voll ist.
- Laufzeit in $\mathcal{O}(n \log n)$

Theorem

$\mathcal{A}$ ist ein 1-Approximationsalgorithmus für KNAPSACK.

Proof.

O.B.d.A. sei $w_1 \leq W$. Es gilt $\mathcal{A}(I) \geq c_1 \cdot \left\lfloor \frac{W}{w_1} \right\rfloor$ für alle I.

$$\mathrm{OPT}(I) \leq c_1 \cdot \frac{W}{w_1} \leq c_1 \cdot \left(\left\lfloor \frac{W}{w_1} \right\rfloor + 1 \right) \leq 2 \cdot c_1 \cdot \left\lfloor \frac{W}{w_1} \right\rfloor \leq 2 \cdot \mathcal{A}(I)$$

Also $\mathcal{R}_{\mathcal{A}}(I) \leq 2$. □

Gütegarantien revisited

Karlsruher Institut für Technologie

- Oft interessiert uns nur der asymptotische Gütewert:

$$\mathcal{R}_{\mathcal{A}}^{\infty} := \inf \{ r \mid \exists N_0 > 0 \, \forall I \in D_\Pi, \mathrm{OPT}(I) \geq N_0 : \mathcal{R}_{\mathcal{A}} \leq r \}$$

- „Beste allgemeingültige Gütegarantie von $\mathcal{A}$“

Theorem

Falls $\mathcal{P} \neq \mathcal{NP}$, dann existiert kein relativer Approximationsalgorithmus $\mathcal{A}$ für COLOR mit $\mathcal{R}_{\mathcal{A}}^{\infty} \leq \frac{4}{3}$.

Gütegarantien revisited

- Oft interessiert uns nur der asymptotische Gütewert:

$$\mathcal{R}_{\mathcal{A}}^{\infty} := \inf\{r \mid \exists N_0 > 0\, \forall I \in D_{\Pi}, \mathrm{OPT}(I) \geq N_0 : \mathcal{R}_{\mathcal{A}} \leq r\}$$

- „Beste allgemeingültige Gütegarantie von $\mathcal{A}$“

Theorem

Falls $\mathcal{P} \neq \mathcal{NP}$, dann existiert kein relativer Approximationsalgorithmus $\mathcal{A}$ für COLOR mit $\mathcal{R}_{\mathcal{A}}^{\infty} \leq \frac{4}{3}$.

Approximationsschemata

- Ziel: **Beliebig genaue** Approximation an optimale Lösung.
- Natürlich gilt: Bessere Approximation $\Rightarrow$ höhere Laufzeit.

Polynomiales Approximationsschema (PAS) für ein Optimierungsproblem Π

Familie von Algorithmen $\{A_\varepsilon \mid \varepsilon > 0\}$, sodass A_ε ein ε-approximativer Algorithmus für Π ist.

Falls die Laufzeit von A_ε polynomial von $\frac{1}{\varepsilon}$ abhängt, heißt das Approximationsschema **vollpolynomial**.

Beispiel

Skript S. 85/86: Ein ε-approximativer Algorithmus für KNAPSACK mit Laufzeit $\mathcal{O}(n^3 \cdot \frac{1}{\varepsilon})$

$\Rightarrow \{A_\varepsilon \mid \varepsilon > 0\}$ ist ein FPAS für Knapsack.

Approximationsschemata

Karlsruher Institut für Technologie

- Ziel: **Beliebig genaue** Approximation an optimale Lösung.
- Natürlich gilt: Bessere Approximation $\Rightarrow$ höhere Laufzeit.

Polynomiales Approximationsschema (PAS) für ein Optimierungsproblem Π

Familie von Algorithmen $\{A_\varepsilon \mid \varepsilon > 0\}$, sodass A_ε ein ε-approximativer Algorithmus für Π ist.

Falls die Laufzeit von A_ε polynomial von $\frac{1}{\varepsilon}$ abhängt, heißt das Approximationsschema **vollpolynomial**.

Beispiel

Skript S. 85/86: Ein ε-approximativer Algorithmus für KNAPSACK mit Laufzeit $\mathcal{O}(n^3 \cdot \frac{1}{\varepsilon})$

$\Rightarrow \{A_\varepsilon \mid \varepsilon > 0\}$ ist ein FPAS für Knapsack.

Approximationsschemata

- Ziel: **Beliebig genaue** Approximation an optimale Lösung.
- Natürlich gilt: Bessere Approximation $\Rightarrow$ höhere Laufzeit.

Polynomiales Approximationsschema (PAS) für ein Optimierungsproblem Π

Familie von Algorithmen $\{A_\varepsilon \mid \varepsilon > 0\}$, sodass A_ε ein ε-approximativer Algorithmus für Π ist.

Falls die Laufzeit von A_ε polynomial von $\frac{1}{\varepsilon}$ abhängt, heißt das Approximationsschema **vollpolynomial**.

Beispiel

Skript S. 85/86: Ein ε-approximativer Algorithmus für KNAPSACK mit Laufzeit $\mathcal{O}(n^3 \cdot \frac{1}{\varepsilon})$

$\Rightarrow \{A_\varepsilon \mid \varepsilon > 0\}$ ist ein FPAS für Knapsack.

Aufgabe

Karlsruher Institut für Technologie

Gegeben ist ein ungerichteter Graph G. Ein *Matching* M in G ist eine Kantenmenge, sodass keine zwei Kanten in M zum gleichen Knoten inzident sind. Ein *inklusionsmaximales Matching* ist ein Matching, das keine echte Teilmenge von einem anderen Matching ist. Ein *maximales Matching* ist ein Matching von maximaler Kardinalität.

- Gib in dem Graphen an der Tafel ein inklusionsmaximales und ein maximales Matching an.

Aufgabe

Gegeben ist ein ungerichteter Graph G. Ein *Matching* M in G ist eine Kantenmenge, sodass keine zwei Kanten in M zum gleichen Knoten inzident sind. Ein *inklusionsmaximales Matching* ist ein Matching, das keine echte Teilmenge von einem anderen Matching ist. Ein *maximales Matching* ist ein Matching von maximaler Kardinalität.

- Gib in dem Graphen an der Tafel ein inklusionsmaximales und ein maximales Matching an.

Aufgabe

Gegeben ist ein ungerichteter Graph G. Ein *Matching* M in G ist eine Kantenmenge, sodass keine zwei Kanten in M zum gleichen Knoten inzident sind. Ein *inklusionsmaximales Matching* ist ein Matching, das keine echte Teilmenge von einem anderen Matching ist. Ein *maximales Matching* ist ein Matching von maximaler Kardinalität.

- Gib einen Greedy-Approximationsalgorithmus an, der ein inklusionsmaximales Matching berechnet.

Aufgabe

Gegeben ist ein ungerichteter Graph G. Ein *Matching* M in G ist eine Kantenmenge, sodass keine zwei Kanten in M zum gleichen Knoten inzident sind. Ein *inklusionsmaximales Matching* ist ein Matching, das keine echte Teilmenge von einem anderen Matching ist. Ein *maximales Matching* ist ein Matching von maximaler Kardinalität.

- Gib einen Greedy-Approximationsalgorithmus an, der ein inklusionsmaximales Matching berechnet.

  Idee: Erweitere M sukzessive um Kanten, die noch nicht zu einer Kante aus M adjazent sind.

Aufgabe

Gegeben ist ein ungerichteter Graph G. Ein *Matching* M in G ist eine Kantenmenge, sodass keine zwei Kanten in M zum gleichen Knoten inzident sind. Ein *inklusionsmaximales Matching* ist ein Matching, das keine echte Teilmenge von einem anderen Matching ist. Ein *maximales Matching* ist ein Matching von maximaler Kardinalität.

- Zeige, dass die Größe eines maximalen Matchings eine untere Schranke für die Größe jedes Vertex Covers ist.

Aufgabe

Gegeben ist ein ungerichteter Graph G. Ein *Matching* M in G ist eine Kantenmenge, sodass keine zwei Kanten in M zum gleichen Knoten inzident sind. Ein *inklusionsmaximales Matching* ist ein Matching, das keine echte Teilmenge von einem anderen Matching ist. Ein *maximales Matching* ist ein Matching von maximaler Kardinalität.

- Zeige, dass die Größe eines maximalen Matchings eine untere Schranke für die Größe jedes Vertex Covers ist.

  Idee: Für jede gematchte Kante muss ein inzidenter Knoten im Vertex Cover liegen, diese inzidenten Knoten sind nach Def. paarweise verschieden

Aufgabe

Karlsruher Institut für Technologie

- Betrachte ein inklusionsmaximales Matching M in $G = (V, E)$. Sei

  $$T := \{v \in V \mid v \text{ ist adjazent zu einer Kante in } M\}$$

  Was kann man über den Subgraph sagen, der von den Knoten $V \setminus T$ induziert wird?

Aufgabe

- Betrachte ein inklusionsmaximales Matching M in $G = (V, E)$. Sei

  $$T := \{v \in V \mid v \text{ ist adjazent zu einer Kante in } M\}$$

  Was kann man über den Subgraph sagen, der von den Knoten $V \setminus T$ induziert wird?
  Er ist kantenfrei!

Aufgabe

Karlsruher Institut für Technologie

- Betrachte ein inklusionsmaximales Matching M in $G = (V, E)$. Sei

  $$T := \{v \in V \mid v \text{ ist adjazent zu einer Kante in } M\}$$

  Was kann man über den Subgraph sagen, der von den Knoten $V \setminus T$ induziert wird?

  Er ist kantenfrei!
- Schließe aus dem letzten Teil, dass $2|M|$ die Größe eines Vertex Covers von G ist.

Aufgabe

- Zeige, dass die Größe eines maximalen Matchings eine untere Schranke für die Größe jedes Vertex Covers ist.
- Schließe aus dem letzten Teil, dass $2|M|$ die Größe eines Vertex Covers von G ist.
- Benutze die letzten Teile, um zu zeigen, dass der Greedy-Approximationsalgorithmus, der ein inklusionsmaximales Matching berechnet, eine 1-Approximation für ein maximales Matching ist.

Aufgabe

- Zeige, dass die Größe eines maximalen Matchings eine untere Schranke für die Größe jedes Vertex Covers ist.
- Schließe aus dem letzten Teil, dass $2|M|$ die Größe eines Vertex Covers von G ist.
- Benutze die letzten Teile, um zu zeigen, dass der Greedy-Approximationsalgorithmus, der ein inklusionsmaximales Matching berechnet, eine 1-Approximation für ein maximales Matching ist.

  Berechne mit unserem Approximationsalgorithmus ein inklusionsmaximales Matching M. Sei V das dazugehörige Vertex Cover und M' ein maximales Matching. Dann gilt

$$2\,|M| = |V| \geq |M'|$$

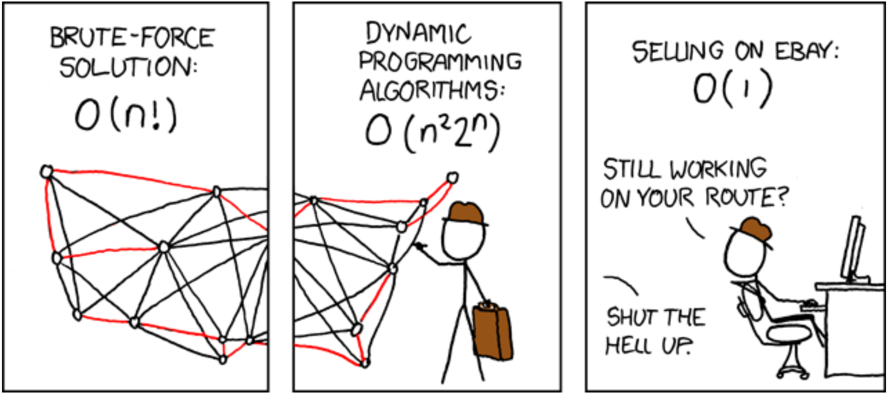
$$\Rightarrow R_{\mathcal{A}}(I) = \frac{\mathrm{OPT}(I)}{\mathcal{A}(I)} = \frac{|M'|}{|M|} \leq 2$$

Aufgabe

Sei $G = (V, E)$ ein ungerichteter Graph mit eindeutigen Kantengewichten $c(u, v)$ für jede Kante $\{u, v\}$ in E. Für jeden Knoten $v \in V$, sei $\max(v) = \arg\max_{(u,v)\in E}\{c(u, v)\}$ die gewichtsmaximale inzidente Kante von v. Sei $S_G = \{\max(v) \mid v \in V\}$ und sei T_G der gewichtsmaximale Spannbaum von G, das heißt der Spannbaum von maximalem Gesamtgewicht. Für jede Menge $E' \subseteq E$ definieren wir $c(E') = \sum_{(u,v)\in E'} c(u, v)$.

1. Finde ein Beispiel mit mindestens 4 Knoten, so dass $S_G = T_G$.
2. Finde ein Beispiel mit mindestens 4 Knoten, so dass $S_G \neq T_G$.
3. Beweise, dass $S_G \subseteq T_G$ für jeden Graph G gilt.
4. Beweise, dass $c(T_G) \leq 2c(S_G)$ für jeden Graphen G gilt.
5. Gib einen 1-Approximationsalgorithmus zur Berechnung eines maximalen Spannbaums an.

Bis zum nächsten Mal!



What's the complexity class of the best linear programming cutting-plane techniques? I couldn't find it anywhere. Man, the Garfield guy doesn't have these problems . . .

Lizenzen