Approximationsalgorithmen

Marco Ammon

14. Oktober 2020

Inhaltsverzeichnis

1 Kombinatorisches Optimierungsproblem Π

1.1 Definition

$$
\begin{aligned}
\mathcal{D} &= \text{Menge der Eingaben } I \\
\mathcal{S}(I \in \mathcal{D}) &= \text{Menge der zur Eingabe } I \text{ zulässigen Lösungen} \\
f : \mathcal{S}(I) \mapsto \mathbb{N}^{\neq 0} &= \text{Bewertungs-/Kosten-/Maßfunction} \\
\text{ziel} \in \{\min, \max\}
\end{aligned}
$$

1. Beschränkung auf natürliche Zahlen, weil Vergleich reeller Zahlen bislang nicht beweisbar schnell funktioniert.
2. Ausschluss der 0 für spätere Definitionen sinnvoll (lässt sich durch Modifikation von f in der Regel trivial erreichen)

Gesucht ist zu $I \in \mathcal{D}$ eine zulässige Lösung $\sigma_{\text{opt}} \in \mathcal{S}(I)$, sodass

$$\text{OPT}(I) = f(\sigma_{\text{opt}}) = \text{ziel}\{f(\sigma) \mid \sigma \in \mathcal{S}(I)\}$$

1.2 Beispiele

1.2.1 Das Rucksackproblem Rucksack

$$
\begin{aligned}
\mathcal{D} &= \{\langle W, \text{vol}, p, B\rangle \mid \{1, \ldots, n\}, \text{vol} : W \mapsto \mathbb{N}, p : W \mapsto \mathbb{N}, B \in \mathbb{N}, \forall w \in W : vol(w) \leq B\} \\
\mathcal{S}(\langle W, \text{vol}, p, B\rangle) &= \{A \subseteq W \mid \sum_{w \in A} \text{vol}(w) \leq B\} \\
f(A) &= \sum_{w \in A} p_w \\
\text{ziel} &= \max
\end{aligned}
$$

Die Restriktion der verfügbaren Waren auf solche, die in den leeren Rucksack passen, hilft gegen das künstliche Aufblähen der Eingabe.

1.2.2 Das Knotenfärbungsproblem Col

$$
\begin{aligned}
\mathcal{D} &= \{\langle G\rangle \mid G = (V, E) \, \text{ein ungerichteter Graph mit mindestens einer Kante}\} \\
\mathcal{S}(\langle G\rangle) &= \{c_{\text{V}} \mid c_{\text{V}} \, \text{ist eine Knotenfärbung von} \, G\} \\
f(c_{\text{V}}) &= |c_{\text{V}}(V)| \\
\text{ziel} &= \min
\end{aligned}
$$

Die Größe der kleinsten möglichen Knotenfärbung ist die chromatische Zahl $\chi(G)$.

1.2.3 Das Kantenfärbungsproblem EdgeCol

$$
\begin{aligned}
\mathcal{D} &= \{\langle G\rangle \mid G = (V, E) \, \text{ein ungerichteter Graph mit mindestens einer Kante}\} \\
\mathcal{S}(\langle G\rangle) &= \{c_{\text{E}} \mid c_{\text{E}} \, \text{ist eine Kantenfärbung von} \, G\} \\
f(c_{\text{E}}) &= |c_{\text{E}}(E)| \\
\text{ziel} &= \min
\end{aligned}
$$

Die Größe der kleinsten möglichen Kantenfärbung ist der chromatische Index $\chi'(G)$.

1.2.4 Das Traveling-Salesperson-Problem TSP

$$
\begin{aligned}
\mathcal{D} &= \{\langle K_n, c\rangle \mid K_n \,\text{vollständiger Graph auf } n \text{ Knoten}, c : E \mapsto \mathbb{N}, \} \\
\mathcal{S}(\langle K_n, c\rangle) &= \{C \mid C = (v_{i_1}, v_{i_2}, \ldots, v_{i_n}, v_{i_i}) \,\text{ist ein Hamiltonkreis}\} \\
f(c_{\mathrm{E}}) &= c(v_{i_n}, v_{i_1}) + \sum_{j=1}^{n-1} c(v_{i_j}, v_{i_{j+1}}) \\
\text{ziel} &= \min
\end{aligned}
$$

1.2.5 Das metrische Traveling-Salesperson-Problem ΔTSP

$$
\begin{aligned}
\mathcal{D} &= \{\langle K_n, c\rangle \mid K_n \,\text{vollständiger Graph auf } n \text{ Knoten}, c : E \mapsto \mathbb{N}, \\
&\quad \underbrace{\forall u, v, w \in V : c(u,v) \leq c(u,w) + c(w,v)}_{\text{Dreiecksungleichung}}\} \\
\mathcal{S}(\langle K_n, c\rangle) &= \{C \mid C = (v_{i_1}, v_{i_2}, \ldots, v_{i_n}, v_{i_i}) \,\text{ist ein Hamiltonkreis}\} \\
f(c_{\mathrm{E}}) &= c(v_{i_n}, v_{i_1}) + \sum_{j=1}^{n-1} c(v_{i_j}, v_{i_{j+1}}) \\
\text{ziel} &= \min
\end{aligned}
$$

1.2.6 Das Problem der Unabhängigen Knotenmengen IS

$$
\begin{aligned}
\mathcal{D} &= \{\langle G\rangle \mid G = (V, E) \,\text{ein Graph}\} \\
\mathcal{S}(\langle G\rangle) &= \{U \mid U \subseteq V, \forall u, v \in U : (u,v) \notin E\} \\
f(U) &= |U| \\
\text{ziel} &= \max
\end{aligned}
$$

2 $t(n)$-Zeit-Approximationsalgorithmus A

Für Eingabe $I \in \mathcal{D}$ berechnet A in Zeit $t(|I|)$ eine Ausgabe $\sigma_I^A \in \mathcal{S}(I)$. Es gilt die Schreibweise $A(I) = f(\sigma_I^A)$.

3 Konstante Gütegarantie

3.1 Definition

1. A hat bei Eingabe I absolute Güte von
$$\kappa_A(I) = |A(I) - \mathrm{OPT}(I)|$$

2. Die absolute Worst-Case-Güte von A abhängig von der Eingabelänge $n = |I|$ ist die Funktion
$$\kappa_A^{\mathrm{wc}}(n) = \max\{\kappa_A(I) \mid I \in \mathcal{D}, |I| <= n\}$$

3. A garantiert eine absolute Güte von $\kappa_A : \mathbb{N} \mapsto \mathbb{N}$, falls für alle $n \in \mathbb{N}$ gilt:
$$\kappa_A^{\mathrm{wc}}(n) \leq \kappa_A(n)$$

4. A hat eine absolute Abweichung von $\kappa'_A : \mathbb{N} \mapsto \mathbb{N}$, falls für unendlich viele n gilt

$$\kappa'_A(n) \leq \kappa_A^{wc}(n)$$

Eine unendlich große Menge $\mathcal{D}' \subseteq \mathcal{D}$ heißt $\kappa'_A(n)$-Zeugenmenge gegen A, wenn für alle $I \in \mathcal{D}'$ gilt:

$$\kappa_A(I) \geq \kappa'_A(|I|)$$

3.2 Unmöglichkeitsergebnis für das Rucksackproblem

Satz. *Falls $P \neq NP$, dann gibt es keine Konstante $n \in \mathbb{N}$, sodass es einen polynomiellen Approximationsalgorithmus A für das Rucksackproblem gibt mit*

$$|A(I) - \mathrm{OPT}(I)| \leq k$$

Widerspruchsbeweis. Unter der Annahme, dass A und k existieren, kann Rucksack in Polynomzeit exakt gelöst werden, was $P = NP$ zur Folge hat:

Konstruiere aus einer Instanz $I = \langle W, \mathrm{vol}, p, B\rangle$ eine neue Probleminstanz $I' = \langle W, \mathrm{vol}, p', B\rangle$ mit $p'(w) = (k+1) \cdot p(w)$. Eine zulässige Lösung σ für I ist auch eine zulässige Lösung für I'. Gleiches gilt aufgrund der Monotonie der Multiplikation auch für optimale Lösungen. Durch die Multiplikation aller Preise mit $k+1$ beträgt die „Lücke“ zwischen den optimalen und der ersten nicht-optimalen Lösung für I' mindestens $k+1$.

Da A eine absolute Güte von k garantiert und in Polynomzeit terminiert, kann es nur eine optimale Lösung für I', welche auf optimal für I ist, zurückgeben. Damit ist das NP-vollständige Rucksack in Polynomzeit exakt lösbar. □

Die hierbei verwendete Vorgehensweise einer Selbstreduktion sowie das „Aufblasen“ des Problems („Scaling“, „Gap Amplification“) lässt sich auch auf viele andere Probleme wie etwa SetCover anwenden. Folglich kann eine konstante Gütegarantie nur für vergleichsweise wenig Probleme erreicht werden.

4 Relative Gütegarantie

4.1 Definition

1. A hat bei Eingabe I eine relative Güte von

$$\rho_A(I) = \max\left\{\frac{A(I)}{\mathrm{OPT}(I)}, \frac{\mathrm{OPT}(I)}{A(i)}\right\} \geq 1$$

2. Die relative worst-case-Güte von A ist die Funktion

$$\rho_A^{\mathrm{wc}}(n) = \max\{\rho_A(I) \mid I \in \mathcal{D}, |i| \leq n\}$$

3. A garantiert eine relative Güte von $\rho_A : \mathbb{N} \mapsto \mathbb{N}$, falls für alle $n \in \mathbb{N}$ gilt

$$\rho_A^{\mathrm{wc}}(n) \leq \rho_A(n)$$

4. A macht für die Eingabe $I \in \mathcal{D}$ einen relativen Fehler von

$$\varepsilon_A(I) = \frac{|A(I) - \mathrm{OPT}(I)|}{\mathrm{OPT}(I)} = |\frac{A(I)}{\mathrm{OPT}(I)} - 1|$$

5. A garantiert einen relativen Fehler von $\varepsilon_A(n)$, falls für alle $\{I \mid I \in \mathcal{D}, |I| \leq n\}$ gilt

$$\varepsilon_A(I) \leq \varepsilon_A(n)$$

6. A hat eine relative Abweichung von $\rho'_A : \mathbb{N} \mapsto \mathbb{N}$, falls für unendlich viele n gilt

$$\rho_A^{\text{wc}}(n) \geq \rho'_A(n)$$

Eine unendlich große Menge $\mathcal{D}' \subseteq \mathcal{D}$ heißt $\rho'_A(n)$-Zeugenmenge gegen A, wenn für alle $I \in \mathcal{D}'$ gilt

$$\rho_A(I) \geq \rho'_A(|I|)$$

Es folgen daraus direkt, dass

1. bei einem Minimierungsproblem $1 + \varepsilon_A(n) = \rho_A(n)$ ist.
2. bei einem Maximierungsproblem $1 - \varepsilon_A(n) = \frac{1}{\rho_A(n)}$ ist.
3. für alle Probleme $\varepsilon_A(n) \leq \rho_A(n) - 1$ ist.

Weiter lassen sich damit obere bzw. untere Schranken der Optimallösung aus einer approximierten Lösung angeben. Es folgt, dass

1. bei einem Minimierungsproblem gilt

$$\frac{1}{\rho_A(|I|)} \cdot A(I) \leq \text{OPT}(I) \leq A(I) \leq \rho_A(|I|) \cdot \text{OPT}(I)$$

2. bei einem Maximierungsproblem gilt

$$\frac{1}{\rho_A(|I|)} \cdot \text{OPT}(I) \leq A(I) \leq \text{OPT}(I) \leq \rho_A(|I|) \cdot A(I)$$

3. bei beiden Problemtypen mit der Beziehung

$$|A(I) - \text{OPT}(I)| \leq \varepsilon_A(|I|) \cdot \text{OPT}(I)$$

gilt

$$(1 - \varepsilon_A(|I|)) \cdot \text{OPT}(I) \leq A(I) \leq (1 + \varepsilon_A(|I|)) \cdot \text{OPT}(I)$$

4.2 Unmöglichkeitsergebnis für das allgemeine (volle) TSP

Satz. *Wenn es einen polynomiellen Approximationsalgorithmus A mit konstanter relativer Gütegarantie r für das volle* TSP *gibt, dann gilt $P = NP$.*

Beweis durch Reduktion. Durch Benutzung von A mit beliebiger konstanter relativer Gütegarantie $r \in \mathbb{N}$ kann Hamilton auf das volle TSP reduziert werden. Es wird also Hamilton $\leq$ pTSP$[r]$ für alle r gezeigt:

Sei der Graph $G = (V, E)$ mit $n = |V|$, gegeben. Dazu wird nun passend eine Probleminstanz $I_G = \langle K_n, c \rangle$ für TSP erzeugt. c wird wie folgt konstruiert:

$$c(u, v) = \begin{cases} 1 & \text{falls } \{u, v\} \in E \text{ („kurze“ Kante)} \\ (r-1) \cdot n + 2 & \text{sonst („lange“ Kante)} \end{cases}$$

I_G kann in Polynomzeit aus G berechnet werden. Es gilt weiter:

- $G \in$ Hamilton $\Rightarrow$ kürzeste Rundreise in I_G hat die Länge n
- $G \notin$ Hamilton $\Rightarrow$ kürzeste Rundreise in I_G nimmt mindestens eine der langen Kanten und hat damit eine Länge von mindestens

$$(r-1) \cdot n + 2 + n - 1 = r \cdot n + 1 > r \cdot n$$

- I_G besitzt keine zulässige Lösung σ mit $n+1 \leq c(\sigma) \leq r \cdot n$.

Durch den folgenden Algorithmus kann also HAMILTON entschieden werden:

konstruiere I_G
approximiere mit A eine kürzeste Rundreise $A(I_G)$
if $A(I_G) > r \cdot |V|$ **then**
  return $G \notin$ HAMILTON
else
  return $G \in$ HAMILTON
end if

□

Der Ansatz der Konstruktion von Probleminstanzen anderer NP-schwerer Probleme und der anschließenden Verwendung eines Scaling-Arguments kann auch für weitere Probleme verwendet werden. Ebenfalls können damit bestimmte Bereiche für mögliche konstante relative Gütegarantien ausgeschlossen werden, etwa $\rho < \frac{3}{2}$ bei BINPACKING.

5 Approximationsschemata

Während die Ergebnisse von Approximationsalgorithmen mit absoluter oder relativer Gütegarantie nur durch eine Modifikation oder Wechsel des Algorithmus verbessert werden können, ist es manchmal gewünscht, im Gegenzug für eine verlängerte Laufzeit eine bessere Güte zu erreichen. Dafür sind sogenannte Approximationsschemata geeignet.

5.1 Definition

Sei Π ein Optimierungsproblem und A ein Approximationsalgorithmus für Π, der eine Probleminstanz I von Π und ein $0 < \varepsilon < 1$ bekommt.

1. A ist ein polynomielles Approximationsschema (PAS) für Π, wenn A zu jedem I und für jedes ε in Zeit $\mathcal{O}(\text{poly}(|I|))$ eine zulässige Lösung zu I mit relativem Fehler $\varepsilon_A(I,\varepsilon) \leq \varepsilon$ berechnet.

2. A ist ein streng polynomielles Approximationsschema (FPAS), wenn A ein PAS mit Laufzeit $\mathcal{O}(poly\left(|I|, \frac{1}{\varepsilon}\right))$ ist.

Satz (Umwandlung eines (F)PAS in einen exakten Algorithmus)**.** *Sei A ein (F)PAS und zu jeder Eingabe I $Z(I)$ eine obere Schranke. Sei $\varepsilon^* = \frac{1}{Z(I)+1}$, dann ist $A(I,\varepsilon^*) = \text{OPT}(I)$. Sofern A ein FPAS ist, liegt die Laufzeit in $\mathcal{O}(\text{poly}(|I|, Z(I)))$.*

Beweis. Starte A mit Eingabe I und ε^*. Es wird eine zulässige Lösung zu I gefunden, für ihren relativen Fehler gilt

$$\varepsilon_a(I,\varepsilon^*) = \frac{|\text{OPT}(I) - A(I,\varepsilon^*)|}{\text{OPT}(I)} \leq \varepsilon^*$$

Weil $\text{OPT}(I) \leq Z(I)$ beschränkt ist, folgt für die Abweichung

$$|\text{OPT}(I) - A(I,\varepsilon^*)| \leq \varepsilon^* \cdot \text{OPT}(I) = \frac{\text{OPT}(I)}{Z(I)+1} < 1$$

Da die Werte zulässiger Lösungen immer ganzzahlig sind, folgt $|\text{OPT}(I) - A(I,\varepsilon^*)| = 0$, damit also die Optimalität von $A(I,\varepsilon^*)$. □

6 GreedyIS für IS

$U = \emptyset, t = 0, V^{(0)} = V$
while $V^{(t)} \neq \emptyset$ **do**
  $G^{(t)} =$ der durch $V^{(t)}$ induzierte Graph
  $u_l =$ ein Knoten mit minimalem Grad in $G^{(t)}$
  $V^{(t+1)} = V^{(t)} - (\{u_l\} \cup \Gamma_{G^{(t)}}(u_l))$
  $U = U \cup \{u_l\}$
  $t = t + 1$
end while
return U

Satz. *Sei G ein knoten-k-färbbarer Graph, dann ist*

$$\textsc{GreedyIS}(G) \geq \left\lceil \log_k \left(\frac{|V|}{3} \right) \right\rceil$$

Beweis. Mit dem folgenden Hilfslemma kann eine Beziehung zwischen der Anzahl der notwendigen Farben und dem minimalen Grad des Graphs hergestellt werden.

Lemma. *Sei G ein knoten-k-färbbarer Graph, dann gilt:*

$$\exists u \in V : \deg_G(u) \leq \left\lfloor \left(1 - \frac{1}{k}\right) \cdot |V| \right\rfloor$$

Beweis. Da G mit k Farben gefärbt ist, gibt es k Mengen U_i an Knoten, die jeweils mit der gleichen Farbe i gefärbt sind. Es muss nach einem Durchschnitsargument eine Menge U_i mit $|U_i| \geq \left\lceil \frac{1}{k} \cdot |V| \right\rceil$ geben. Jeder der Knoten u in U_i kann maximal mit allen Knoten aus $V \setminus U_i$ verbunden sein. Es folgt also

$$\deg_G(u) \leq |V| - |U_i| \leq |V| - \left\lceil \frac{1}{k} \cdot |V| \right\rceil = \left\lfloor \left(1 - \frac{1}{k}\right) \cdot |V| \right\rfloor$$

□

Zur Vereinfachung gelte $n = |V|$ und $n_t = |V^{(t)}|$. Es kann $k \geq 2$ angenommen werden. Mit dem Hilfslemma ergibt sich für die Anzahl der Knoten folgende Rekursion:

$$\begin{aligned} n_0 &= n \\ n_{t+1} &\geq n_t - \left\lfloor \left(1 - \frac{1}{k}\right) \cdot n_t \right\rfloor - 1 \geq \frac{n_t}{k} - 1 \end{aligned}$$

Sie kann zur Ungleichung

$$n_t \geq \frac{n}{k^t} - \underbrace{\frac{k}{k-1} \cdot \left(1 - \frac{1}{k^t}\right)}_{\leq 2 \text{ für } k \geq 2} \geq \frac{n}{k^t} - 2$$

aufgelöst werden. Solange $n_t \geq 1$ gilt, wird ein neuer Knoten nach U gelegt. Durch Umformen obiger Ungleichung lässt sich dies für $t \geq \log_k \left(\frac{n}{3}\right)$ garantieren. Es folgt also $|U| \geq \left\lceil \log_k \left(\frac{n}{3}\right) \right\rceil$. □

7 Knotenfärbungsalgorithmen

7.1 GreedyCol

for all $u_i \in V$ **do**

$c_{\mathrm{V}}(u_i) = \infty$
end for
for all $i \in [1, \ldots, |V|]$ **do**
$c_{\mathrm{V}}(u_i) = \min\{\mathbb{N} \setminus \{c_{\mathrm{V}}(\Gamma(u_i))\}\}$
end for
return c_{V}

Satz. GreedyCol *berechnet in Zeit* $\mathcal{O}(|V| + |E|)$ *eine Knotenfärbung aus höchstens* $\Delta(G) + 1$ *Farben.*

Beweis. Da ein Knoten u maximal $\Delta(G)$ viele Nachbarn haben kann, muss in $[1, \ldots, \Delta(G) + 1]$ noch mindestens eine Farbe frei sein. □

Satz. GreedyCol *garantiert eine absolute Güte von*

$$\kappa_{\text{GreedyCol}}(G) = \text{GreedyCol}(G) - \text{OPT}(G) \leq \Delta(G) + 1 - 2 = \Delta(G) - 1$$

, weil die untere Schranke $\text{OPT}(G) \geq 2$ *für Graphen mit* $|V| \geq 2$ *gilt.*

Zeuge. $\Delta(G) - 1$*-Zeuge gegen* GreedyCol*: TODO (Abbildung 2.1)*

7.2 GreedyCol2

$t = 1, V^{(1)} = V$
while $V^{(t)} \neq \emptyset$ **do**
$G^{(t)} =$ der durch $V^{(t)}$ induzierte Graph
$U_t = \text{GreedyIS}(G^{(t)})$
färbe alle Knoten in U_t mit Farbe t
$V^{(t+1)} = V^{(t)} - U_t$
$t = t + 1$
end while
return berechnete Färbung

Satz. *Für einen knoten-k-färbbaren Graph* $G = (V, E)$ *mit* $n = |V|$ *gibt* GreedyCol2 *eine Färbung mit höchstens* $\frac{3n}{\log_k\left(\frac{n}{16}\right)}$*. Die relative Gütegarantie liegt als in* $\mathcal{O}\left(\frac{n}{\log n}\right)$*.*

Beweis. Zur Vereinfachung bezeichne $n_t = |V^{(t)}|$. Aus der Analyse von GreedyIS folgt $|U_t| \geq \log_k\left(\frac{n_t}{3}\right)$. Es ergibt sich die Rekursion

$$n_1 = n$$
$$n_{t+1} \leq n_t - \log_k\left(\frac{n_t}{3}\right)$$

Nun wird bestimmt, für welches t $n_t < 1$ eintritt, denn dann bricht der Algorithmus ab. Behelfsmäßig sei $n_t \geq \frac{n}{\log_k\left(\frac{n}{16}\right)}$. Mit der Beziehung $\frac{n}{\log_k n} \geq \frac{3}{4} \cdot \sqrt{n}$ ergibt sich

$$\log_k\left(\frac{n_t}{3}\right) \geq \log_k\left(\frac{n}{3 \cdot \log_k n}\right) \geq \log_k\left(\sqrt{\frac{n}{16}}\right) = \frac{1}{2} \cdot \log_k\left(\frac{n}{16}\right)$$

Solange $n_t \geq \frac{n}{\log_k\left(\frac{n}{16}\right)}$ also gilt, werden pro Runde mindestens $\frac{1}{2} \cdot \log_k\left(\frac{n}{16}\right)$ Knoten pro Runde gefärbt. Nach höchstens $t \leq \frac{2n}{\log_k\left(\frac{n}{16}\right)}$ gilt die Ungleichung nicht mehr. Färbt man jetzt alle verbliebenen Knoten mit jeweils einer eigenen Farbe, werden insgesamt maximal $\frac{n}{\log_k\left(\frac{n}{16}\right)} + t \leq \frac{3n}{\log_k\left(\frac{n}{16}\right)}$ vergeben.

Mit $k = \chi_G = \mathrm{OPT}(G)$ ergibt sich für die relative Gütegarantie:

$$\frac{\textsc{GreedyCol2}(\mathrm{G})}{\mathrm{OPT}(G)} \leq \frac{\frac{3n}{\log_k\left(\frac{n}{16}\right)}}{k} = \mathcal{O}\left(\frac{n}{\log n}\right)$$

□

8 Kantenfärbungsalgorithmen

TODO: Übung

9 Christofides' Algorithmus CH für das metrische TSP ΔTSP

Ein Matching M eines kantengewichteten Graphen G ist ein Teilgraph von G mit $\Delta(G) \leq 1$. Ist G ein vollständiger Graph mit $|V|$ gerade, dann gibt es perfekte Matchings. In einem perfekten Matching haben alle Knoten genau den Grad 1. Ein perfektes Matching mit kleinstmöglichem Gewicht wird als leichtestes Matching bezeichnet. Ein solches leichtestes Matching kann in $\mathcal{O}(n^{2.5} \cdot (\log n)^4)$ berechnet werden.

Als Multi-Graph wird ein Graph bezeichnet, der um mehrere Kanten zwischen den gleichen Knoten erweitert wurde.

Wird in einem Pfad jede Kante des (Multi-)Graph genau einmal besucht, so spricht man von einem Euler-Pfad. Bildet der Pfad einen Kreis, so nennt man ihn Euler-Kreis oder Euler-Tour. Haben alle Knoten von G geraden Grad, so existiert eine Euler-Tour in G. Diese lässt sich in $\mathcal{O}(|V| + |E|)$ berechnen.

Der Algorithmus von Christofides (CH) geht wie folgt vor:

1: berechne einen minimalen Spannbaum T_{CH} von $I = \langle K_n, c\rangle$
2: $S = \{v \in T_{\mathrm{CH}} \mid \deg_{T_{\mathrm{CH}}}(v) \text{ ungerade}\}$ ▷ $|S|$ ist gerade
3: berechne auf dem durch S induzierten Teilgraphen des K_n ein leichtestes Matching M_{CH}
4: berechne eine Euler-Tour $E = (u_1, u_2, \dots)$ auf $T_{\mathrm{CH}} \uplus M_{\mathrm{CH}}$ ▷ $T_{\mathrm{CH}} \uplus M_{\mathrm{CH}}$ kann Multi-Graph sein, alle Knoten haben geraden Grad
5: entferne Wiederholungen von Knoten in E, sodass man E' erhält
6: **return** E'

Satz. CH, *gestartet mit einer Eingabe auf* n *Knoten, garantiert eine relative Güte von* $\rho_{\mathrm{CH}} \leq \frac{3}{2} - \frac{1}{n}$ *in einer Laufzeit von* $\mathcal{O}(n^{2.5} \cdot (\log n)^4)$.

Beweis. Sei R^* eine optimale Rundreise für I, d.h. $c(R^*) = \mathrm{OPT}(I)$. Es gilt $\mathrm{CH}(I) = c(E') \leq (\frac{3}{2} - \frac{1}{n}) \cdot c(R^*)$ zu zeigen.

1. Da R^* aus n Kanten besteht, muss durch ein Durchschnittsargument mindestens eine Kante e mit $c(e) \geq \frac{c(R^*)}{n}$ existieren. Wird diese aus R^* entfernt, so enthält man einen Spannbaum des K_n. Da T_{CH} minimal ist, gilt

$$c(T_{\mathrm{CH}}) \leq c(R^*) - \frac{c(R^*)}{n} = \left(1 - \frac{1}{n}\right) \cdot c(R^*)$$

2. In beliebigen Bäumen ist die Anzahl der Knoten mit ungeradem Grad gerade.

3. Zur Vereinfachung werden die Knoten so umbenannt, dass $R^* = (u_1, u_2, \dots, u_n, u_1)$ ist. S kann dann als $S = \{u_{i_1}, \dots, u_{i_{|S|}}\}$ mit $i_1 < \cdots < i_{|S|}$ geschrieben werden.

   Aus S kann ein Kreis $H = (u_{i_1}, \dots, u_{i_{|S|}}, u_{i_1})$ gebildet werden. Durch die Dreiecksungleichung ($|H| \leq n$ und jede „Abkürzung“ ist maximal gleich lang wie der Weg in R^*) gilt $c(H) \leq c(R^*)$.

Es können zwei perfekte Matching M_1 und M_2 auf H berechnet werden, denn $|S|$ ist gerade. Weil M_{CH} minimal ist, folgt o.B.d.A. mit $c(M_1) \leq c(M_2)$ die Aussage

$$c(M_{\text{CH}}) \leq c(M_1) \leq \frac{1}{2} \cdot (c(M_1) + c(M_2)) = \frac{1}{2} \cdot c(H) \leq \frac{1}{2} \cdot c(R^*)$$

4. Da jeder Knoten in $T_{\text{CH}} \uplus M_{\text{CH}}$ geraden Grad hat, kann eine Euler-Tour E berechnet werden. Weil diese nur Kanten aus $T_{\text{CH}} \uplus M_{\text{CH}}$ benutzt, kann ihre Länge mit den vorherigen Ergebnissen wie folgt beschränkt werden:

$$c(E) = c(T_{\text{CH}} \uplus M_{\text{CH}}) \leq \left(1 - \frac{1}{n}\right) \cdot c(R^*) + \frac{1}{n} \cdot c(R^*) = \left(\frac{3}{2} - \frac{1}{n}\right) \cdot \text{OPT}(I)$$

5. Durch die Dreiecksungleichung kann E' nicht länger als E werden.

□

Zeuge. *TODO*

10 DynRucksack für Rucksack