Universität Augsburg

Institut für Mathematik

Seminar„ausarbeitung“

zu einem Vortrag im Seminar Spieltheorie und Approximationsalgorithmen im SS 2016
zum Thema

Capacitated Vehicle Routing with Non-Uniform Speeds

Zusammengestellt:
Lukas Graf

Betreut von:
M. Sc. Manuel Surek,
Prof. Dr. Tobias Harks

1 Problemübersicht

Metrisches TSP:

- Vollständiger Graph $G = (V, E)$
- Metr. Abstandsfunktion $d : E \to \mathbb{R}$
- **Lsgen:** Tour τ durch ganz V
- **Ziel:** Minimiere $d(\tau)$

CVRP:

- Vollständiger Graph $G = (V, E)$
- Metr. Abstandsfunktion $d : E \to \mathbb{R}$
- Startpunkt/Depot $s \in V$
- Bedarfe $(q_v)_{v \in V}$ und Kapazität Q
- **Lsgen:** Tour (τ), die bei s beginn alle Bedarfe erfüllen nie mehr als Q Elemente transportiert
- **Ziel:** Minimiere $\max d(\tau_i)$

Homogenes TSP:

- Vollständiger Graph $G = (V, E)$
- Metr. Abstandsfunktion $d : E \to \mathbb{R}$
- Startpunkt $s \in V$
- k Fahrzeuge
- **Lsgen:** Touren (τ_i), die bei s beginnen und gemeinsam ganz V abdecken
- **Ziel:** Minimiere $\max d(\tau_i)$

Homogenes CVRP:

- Vollständiger Graph $G = (V, E)$
- Metr. Abstandsfunktion $d : E \to \mathbb{R}$
- Startpunkt/Depot $s \in V$
- Bedarfe $(q_v)_{v \in V}$
- k Fahrzeuge mit einheitlicher Kapazität Q
- **Lsgen:** Touren (τ_i), die bei s beginnen, gemeinsam alle Bedarfe erfüllen, wobei kein Fahrzeug jemals mehr als Q Elemente transportiert
- **Ziel:** Minimiere $\max d(\tau_i)$

Heterogenes TSP:

- Vollständiger Graph $G = (V, E)$
- Metr. Abstandsfunktion $d : E \to \mathbb{R}$
- Startpunkt $s \in V$
- k Fahrzeuge mit Geschw. $(\lambda_i)_{i=1}^k$
- **Lsgen:** Touren (τ_i), die bei s beginnen und gemeinsam ganz V abdecken
- **Ziel:** Minimiere $\max \frac{d(\tau_i)}{\lambda_i}$

Heterogenes CVRP:

- Vollständiger Graph $G = (V, E)$
- Metr. Abstandsfunktion $d : E \to \mathbb{R}$
- Startpunkt/Depot $s \in V$
- Bedarfe $(q_v)_{v \in V}$
- k Fahrzeuge mit Geschw. $\{\lambda_i\}$ und einheitlicher Kapazität Q
- **Lsgen:** Touren (τ_i), die bei s beginnen, gemeinsam alle Bedarfe erfüllen, wobei kein Fahrzeug jemals mehr als Q Elemente transportiert
- **Ziel:** Minimiere $\max \frac{d(\tau_i)}{\lambda_i}$

2 Algorithmen für HomTSPund CVRP

Vorbemerkungen

Verwendet folgende Konvention: Eine Funktion $f : M \to \mathbb{R}$ induziert Funktionen

$$\tilde{f} : \mathcal{P}(M) \to \mathbb{R} : N \mapsto \sum_{m \in N} f(m)$$

und

$$\hat{f}_n : \prod_{i=1}^{n} M \to \mathbb{R} : (m_i) \mapsto \sum_{i=1}^{n} f(m_i).$$

die wir unter Überladung der Notation ebenfalls mit f bezeichnen werden.

2.1 Algorithmus für HomTSP

Algorithm 1 HomTSP-Approx

1: **procedure** $\textsc{HomTSP}(G = (V, E),\ d : E \to \mathbb{R}_{\geq 0}, k)$
2: $\quad \tau \leftarrow$ TSP-$\textsc{Approx}$ (G, d)
3: $\quad (\pi_i)_{i=1}^{k} \leftarrow$ Teile τ durch Entfernen von Kanten in k Teilstrecken mit Länge $\leq \frac{d(\tau)}{k}$
4: $\quad (\tau_i)_{i=1}^{k} \leftarrow$ Verbinde die zwei Endpunkte von π_i mit s.
5: $\quad$ **return** (τ_i)
6: **end procedure**

Lemma 2.1 (Theorem 8 in [FHK76][1]). *Unter Verwendung eines ρ-approximativen Algorithmus (z.B. $\rho = \frac{3}{2}$ durch Christofides) für **TSP** ist* $\textsc{HomTSP}$ *$(\rho+1)$-approximativ.*

Beweis. Schritt 3 ist möglich, denn jeder Pfad π_i entspricht Kanten der Gesamtlänge $> \frac{d(\tau)}{k}$ in τ: Zunächst die (möglicherweise 0 vielen) Kanten in pi_i der Gesamtlänge $\leq \frac{d(\tau)}{k}$ und dann der entfernten nächsten Kante, die die Gesamtlänge über diese Schwelle gebracht hätte. Nach dem Erstellen der ersten $(k-1)$ Pfade sind daher höchstens noch Kanten der Länge

$$d(\tau) - \sum_{i=1}^{k-1} d(\pi_i) < d(\tau) - (k-1) \cdot \frac{d(\tau)}{k} = \frac{d(\tau)}{k}$$

übrig, die daher von π_k abgedeckt werden können.

Da jeder der Endpunkte u und v von π_i auch in einer optimalen Lösung besucht werden muss, gilt sicher $d(s,u), d(s,v) \leq \frac{1}{2}\mathrm{OPT}_{\mathbf{HomTSP}}$. Außerdem ist die Vereinigung aller Touren τ_i^* einer optimalen Lösung von **HomTSP** eine zulässige Lösung von **TSP** und damit $\mathrm{OPT}_{\mathbf{TSP}} \leq \sum_{i=1}^{k} d(\tau_i^*) \leq k \cdot \max_{i=1}^{k} d(\tau_i^*) = k \cdot \mathrm{OPT}_{\mathbf{HomTSP}}$.

Zusammen ergibt dies für jede der Touren τ_i (und damit insbesondere für die längste):

$$\begin{aligned} d(t_i) &\leq \frac{d(\tau)}{k} + 1 \cdot \mathrm{OPT}_{\mathbf{HomTSP}} \leq \frac{\rho \cdot \mathrm{OPT}_{\mathbf{TSP}}}{k} + \mathrm{OPT}_{\mathbf{HomTSP}} \\ &\leq \frac{\rho \cdot k \cdot \mathrm{OPT}_{\mathbf{HomTSP}}}{k} + \mathrm{OPT}_{\mathbf{HomTSP}} = (\rho+1)\mathrm{OPT}_{\mathbf{HomTSP}}) \end{aligned}$$

□

[1] Tatsächlich wird in [FHK76] mit einem etwas komplexeren Algorithmus sogar eine Approximationsgüte von $\rho + 1 - \frac{1}{k}$ erreicht.

2.2 Algorithmus für CVRP

Algorithm 2 CVRP-Approx

procedure CVRP($G = (V, E)$, $d : E \to \mathbb{R}_{\geq 0}, Q, (q_v)_{v \in V}$)
  $G' \leftarrow$ vollst. Graph auf $V' := \left\{ v^{(j)} \mid v \in V, j = 1, \ldots, q_v \right\}$
  $d'(v^{(i)}, u^{(j)}) := d(v, u)$
  $\tau \leftarrow$ TSP-Approx (G', d')
  for $v \in V'$ **do**
    $\left(\pi_r^{(v)}\right) \leftarrow$ Teile τ durch Kantenlöschen in Teilstrecken mit max. Q Knoten, wobei die erste Strecke bei v beginnt.
    $\left(\sigma_r^{(v)}\right) \leftarrow$ Verbinde die zwei Endpunkte von $\pi_r^{(v)}$ mit s.
  end for
  $(\sigma_r) \leftarrow \left(\sigma_r^{(v)}\right)$ mit $\sum_r d'(\sigma_r^{(v)})$ minimal.
  return (σ_r)
end procedure

Proposition 2.2 (Lemma 1 in [HK85]). *Es gilt:*

$$\mathrm{OPT}_{\boldsymbol{CVRP}} \geq \max \left\{ \mathrm{OPT}_{\boldsymbol{TSP}}, \frac{2}{Q} \sum_{v' \in V'} d'(s, v') \right\}$$

Beweis. Die erste Abschätzung ist klar (eine **CVRP**-Route ist insbesondere auch eine **TSP**-Tour).

Für die zweite Abschätzung betrachte eine optimale Route (σ_r^*), wobei $S_r \subseteq V'$ die in der r-ten Tour besuchten Knoten seien. Dann gilt offenbar:

$$d\left(\sigma_i^*\right) \geq 2 \cdot \max \left\{ d(s, v') \mid v' \in S_r \right\} \geq 2 \frac{\sum_{v' \in V'} d(s, v')}{|S_r|} \geq \frac{2}{Q} \sum_{v' \in S_r} d(s, v')$$

Da jeder Knoten aus V' in wenigstens einem S_r enthalten sein muss, folgt damit:

$$\mathrm{OPT}_{\mathbf{CVRP}} = \sum_r d\left(\sigma_i^*\right) \geq \frac{2}{Q} \sum_r \sum_{v' \in S_r} d(s, v') \geq \frac{2}{Q} \sum_{v' \in V'} d(s, v')$$

□

Lemma 2.3 (Lemma 2 in [HK85][2]). *Unter Verwendung eines ρ-approximativen Algorithmus für **TSP** ist* CVRP-Approx *$(\rho + 2)$-approximativ.*

[2]Wie über eine genauere Abschätzung in [HK85] gezeigt wird, erreicht der Algorithmus sogar eine Approximationsgüte von $\rho + \lceil \frac{n'}{Q} \rceil \frac{Q - \rho}{n'}$, wobei $n' := \sum_v q_v$

Beweis. Sei $n' := \sum_v q_v = |V'|$ und $m := \lceil \frac{n'}{Q} \rceil$. Dann gilt für die Summe aller möglichen Routen:

$$\sum_{v,r} d'(\sigma_r^{(v)}) \leq 2m \sum_{v' \in V'} d'(s, v') + n' \cdot d(\tau)$$

Denn jeder Knoten ist genau einmal Anfangs- und einmal Endpunkt für jede der m Teilstrecken (erster Summand) und jede Route entsteht durch Weglassen von Kanten aus der Tour τ (zweiter Summand).

Die beste dieser Routen ist sicher mindestens so gut wie der Durchschnitt aus allen möglichen Routen, also:

$$\sum_r d'(\sigma_r) = 2\frac{m}{n'} \sum_{v' \in V'} d'(s, v') + d(\tau) \leq 2\frac{\frac{n'}{Q} + 1}{n'} \sum_{v' \in V'} d'(s, v') + d(\tau) \leq 2 \cdot \frac{2}{Q} \sum_{v' \in V'} d'(s, v') + \rho \cdot \mathrm{OPT}_{\mathbf{TSP}}$$

Zusammen mit Proposition 2.2 ergibt dies:

$$\sum_r d'(\sigma_r) \leq 2 \cdot \mathrm{OPT}_{\mathbf{CVRP}} + \rho \cdot \mathrm{OPT}_{\mathbf{CVRP}} \leq (2 + \rho) \cdot \mathrm{OPT}_{\mathbf{CVRP}}$$

□

3 Algorithmus für HetTSP

Algorithm 3 HetTSP-Approx

1: **procedure** HETTSP($G = (V, E)$, $d : E \to \mathbb{R}_{\geq 0}$)
2: Rate M mit $\mathrm{OPT} \leq M \leq 2 \cdot \mathrm{OPT}$
3: $\mathcal{H} := (H_l)_{l \geq 0} \leftarrow$ LEVELPRIM (G, d)
// $\mathcal{H}$ erfüllt: Wurzel-Blatt Pfade haben aufsteigende Knoten-Level
// und $\forall l : \sum_{j>l} d(H_j) \leq 8M \sum_{j \geq l} 2^j \mu_j$ (wenn M korrekt geraten)
4: $\mathcal{T} := (\mathcal{T}_l)_{l \geq 0} \leftarrow$ DECOMPOSITION $(\mathcal{H})$
// $\mathcal{T}$ ist $(6, 40)$-zuweisbarer Wald
5: $(x_{Ti}) \leftarrow$ FRACTIONALASSIGNMENT $(\mathcal{T})$
6: $(\tau_i) \leftarrow$ ROUNDINGASSIGNMENT (x_{Ti})
// $\mathcal{T}$ ist (α, β)-zuweisbar $\Rightarrow$ (τ_i) ist $(4\alpha + 2\beta)$-approx.
7: **return** (τ_i)
8: **end procedure**

Satz 3.1 (Theorem 1.1 in [Gø+10]). *Algorithmus 3 ist ein $\mathcal{O}(1)$-approximativer Algorithmus für **HetTSP**.*

Beweis. Folgende Kapitel... □

3.1 LevelPrim

Algorithm 4 LevelPrim

procedure LEVELPRIM($G = (V, E)$, $d : E \to \mathbb{R}_{\geq 0}$)
  $V_0 := \{v \in V \mid d(s, v) \leq M\}$, $V_l := \left\{v \in V \mid 2^{l-1}M < d(s, v) \leq 2^l M\right\}$
  for $i \geq 0$ **do** $H_l \leftarrow$ Minimaler Spannbaum auf $G\left[V_{\leq l}\right]/V_{<l}$ **end for**
  return $(H_l)_{l \geq 0}$
end procedure

Lemma 3.2 (Theorem 3.3 in [Gø+10]). *Ein von Algorithmus 4 gefundener Baum* $(H_l)_{l \geq 0}$ *erfüllt:*

- *Die Knoten-Level entlang jedes Wurzel-Blatt-Pfades sind monoton wachsend.*
- $\forall l \geq 0 : \sum_{j \geq l} d(H_j) \leq 8 \cdot \mathrm{MST}(G/V_{<l})$

Korollar 3.3 (Korollar 3.5 in [Gø+10]). *Ein von Algorithmus 4 gefundener Baum* $(H_l)_{l \geq 0}$ *erfüllt:*

- *Die Knoten-Level entlang jedes Wurzel-Blatt-Pfades sind monoton wachsend.*
- $\forall l \geq -1 : \sum_{j > l} d(H_j) \leq 8M \cdot \sum_{j \geq l} 2^j \mu_j$

3.2 Zerlegungsalgorithmus

Algorithm 5 Decomposition

procedure DECOMPOSITION($(\mathcal{H})$)
  $\mathcal{S}_0 := \{H_0\}$, $\mathcal{S}_l :=$ Zerl. von $\mathcal{H} \cap E_l$ in Bäume mit genau einer Kante nach $V_{<l}$
  $\mathcal{S}_l^{\geq} := \left\{\tau \in \mathcal{S}_l \mid d(\tau) \geq 2^{l-3}M\right\}, \mathcal{S}_l^{<} := \mathcal{S}_l \setminus \mathcal{S}_l^{\geq}$
  for $\tau \in \mathcal{S}_l^{<}$ **do** $h(\tau) := \tau' \in S_{l-1}^{\geq}$ mit $\tau \cup \tau'$ zsh. (ex. eind.) **end for**
  for $\tau \in \mathcal{S}_l^{\geq}$ **do**
    $\mathcal{T}_l(\tau) \leftarrow$ Partition von $\tau \cup h^{-1}(\tau)$ in Bäume der Länge $\left[2^{l+1}M, 2^{l+2}M\right]$
      (und evtl. ein kürzerer)
    $\mathcal{T}_l'(\tau) \leftarrow \{T_r \cup \{$kürzeste Kante zu $s\} \mid T_r \in \mathcal{T}_l(\tau)\}$
  end for
  $\mathcal{T}_l := \bigcup_{\tau \in \mathcal{S}_l^{\geq}} \mathcal{T}_l'(\tau)$
  return $(\mathcal{T}_l)_{l \geq 0}$
end procedure

Definition 3.4 (Definition 3.1 in [Gø+10]). Ein Wald $\mathcal{T} = \bigcup_{l \geq 0} \mathcal{T}_l$ aus Bäumen mit Wurzel s heißt (α, β)-zuweisbar, wenn gilt:

- Für alle $T \in \mathcal{T}_l$ gilt: $d(T) \leq \alpha 2^l M$
  d.h. ein Baum aus $\mathcal{T}_l$ kann mit Geschw. 2^l in $\mathcal{O}(\alpha M)$ besucht werden.
- Für alle $l \geq -1$ gilt: $\sum_{j>l} d(\mathcal{T}_j) \leq \beta M \sum_{j \geq l} 2^j \mu_j$
  d.h. die Fahrzeuge mit Geschw. $\geq 2^l$ können den Wald $\mathcal{T}_{>l}$ in $\mathcal{O}(\beta M)$ besuchen.

Lemma 3.5 (Lemma 3.11 in [Gø+10])**.** *Die von Algorithmus 5 bestimmte Zerlegung $\mathcal{T} = (\mathcal{T}_i)_{i \geq 0}$ ist $(6, 40)$-zuweisbar.*

3.3 Assignment-Algorithmen

Algorithm 6 FractionalAssignment

```
procedure FRACTIONALASSIGNMENT((𝒯))
    L := {T ∈ 𝒯},   b(T) := d(T),   R := {i | 1 ≤ i ≤ k},   b(i) := βM2^{λ_i}
    F := {{T, i} | T ∈ 𝒯_l, λ_i ≥ l − 1}
    Bestimme L-überdeckendes b-Matching (x_{Ti}).
    return (x_{Ti})
end procedure
```

Definition 3.6. (x_{Ti}) ist *L-sättigendes b-Matching*, wenn gilt:

$$\sum_i x_{Ti} = b(T), \quad \sum_T x_{Ti} \leq b(i), \quad x_{Ti} \in \mathbb{N}$$

Proposition 3.7 (Seite 54f in [Coo+11])**.** *Ein L-sättigendes b-Matching existiert genau dann, wenn für alle $K \subseteq L : b(K) \leq b(N(K))$, wobei $N(K) := \{i \in R \mid \exists T \in K : \{T, i\} \in F\}$.*

Lemma 3.8. *Der in Algorithmus 6 definierte Hilfsgraph besitzt ein L-sättigendes b-Matching.*

Beweis. Sei $K \subseteq L$ und $l := \min\{j \mid K \cap \mathcal{T}_j \neq \emptyset\} - 1$, dann gilt:

$$b(K) = \sum_{T \in K} d(T) \leq \sum_{T \in \mathcal{T}_{>l}} d(T) \overset{(*)}{\leq} \beta M \sum_{j \geq l} 2^j \mu_j \overset{(\#)}{=} \beta M \sum_{i \in N(K)} 2^{\lambda_i} = b(N(K))$$

Dabei ist $(*)$ gerade die zweite Bedingung für (α, β)-Zuweisbarkeit und $(\#)$ folgt aus der Definition der Kantenmenge F (K enthält einen Baum mit Level $l + 1$. Dieser ist bereits mit allen Fahrzeugen von Geschwindigkeit 2^l und mehr verbunden). □

Algorithm 7 RoundingAssignment

1: **procedure** RoundingAssignment((x_{Ti}))
2: $(x'_{Ti}) \leftarrow$ RoundScheduling($p_{Ti} := \frac{d(T)}{2^{\lambda_i}}, \tilde{x}_{Ti} := \frac{x_{Ti}}{d(T)}$)
3: $\tau_i \leftarrow$ Tour durch die Bäume T mit $x'_{Ti} = 1$.
4: **return** (τ_i)
5: **end procedure**

Proposition 3.9 (Theorem 1 in [LST90]). *Gegeben folgendes Scheduling-Problem: Job T benötigt auf Maschine i Zeit p_{Ti}. $J_i(t) := \{T \mid p_{Ti} \leq t\}$ und $M_T(t) := \{i \mid p_{Ti} \leq t\}$. Eine rationale Lösung $\tilde{x}_{Ti}$ mit*

$$\sum_{i \in M_T(t)} \tilde{x}_{Ti} = 1, \quad \sum_{T \in J_i(t)} p_{Ti}\tilde{x}_{Ti} \leq \beta M, \quad \tilde{x}_{Ti} \geq 0$$

kann in polynomieller Zeit zu einer ganzen Lösung x'_{Ti} gerundet werden mit:

$$\sum_{i \in M_T(t)} x'_{Ti} = 1, \quad \sum_{T \in J_i(t)} p_{Ti}x'_{Ti} \leq \beta M + t, \quad x'_{Ti} \in \{0,1\}$$

Lemma 3.10 (Lemma 3.2 in [Gø+10]). *Gegeben einen (α, β)-zuweisbaren Wald, liefern Algorithmus 6 und Algorithmus 7 eine $(4\alpha + 2\beta)$-approximative Lösung für **HetTSP**.*

Beweis. $\tilde{x}_{Ti}$ erfüllt:

$$\sum_{i \in M_T(2\alpha M)} \tilde{x}_{Ti} = 1, \quad \sum_{T \in J_i(2\alpha M)} p_{Ti}\tilde{x}_{Ti} \leq \beta M, \quad \tilde{x}_{Ti} \geq 0$$

Somit gibt Algorithmus 7 eine Zuweisung mit

$$\sum_{T \in J_i(t)} p_{Ti}\tilde{x}_{Ti} \leq \beta M + 2\alpha M$$

Durch Verdoppeln der Kanten in den Bäumen, erhalten wir daraus Touren mit $d(\tau_i) \leq 2 \cdot (\beta M + 2\alpha M)$. □

4 Algorithmus für HetCVRP

Algorithm 8 Reduktion

1: **procedure** Reduktion($G = (V, E), d : E \to \mathbb{R}_{\geq 0}, s \in V, Q, (\lambda_i), (q_v)$)
2: $V' := \left\{v^{(j)} \mid v \in V, j = 1, \ldots, q_v\right\}$, $E' := \left\{\{v^{(j)}, v\} \mid v \in V, j = 1, \ldots, q_v\right\}$
3: $\tilde{G} := (V \cup V', E \cup E')$, $\tilde{d}(v^{(j)}, v) := \frac{d(s,v)}{Q}$, sonst wie d
4: **return** $\left(\tilde{G}, \tilde{d}, s, Q, (\mu_i)\right)$
5: **end procedure**

Lemma 4.1. *Für jede Instanz $\mathcal{I}$ von **HetCVRP** ist $\mathcal{J} := \textsc{Reduktion}(\mathcal{I})$ eine Instanz von **HetTSP** mit:*

$$\mathrm{OPT}_{\textbf{\textit{HetTSP}}}(\mathcal{J}) \in \mathcal{O}(1) \cdot \mathrm{OPT}_{\textbf{\textit{HetCVRP}}}(\mathcal{I})$$

Beweis. Sei (σ_i) eine optimale Lösung von $\mathcal{I}$. Definiere $c_i(v) :=$ #Einheiten von i an v geliefert. Dann gilt also:

$$\sum_i c_i(v) = q_v, \text{ für alle } v \in V$$

Folglich ist es möglich $U_i' \subseteq V'$ zu wählen, sodass gilt

$$\bigcup_i U_i' = V' \text{ und } \left|\left\{v^{(j)} \in U_i'\right\}\right| = c_i(v)$$

Definiere weiter $U_i := \left\{v \in V \mid v^{(j)} \in U_i'\right\}$.

Seien nun τ_i minimale **TSP**-Touren durch $U_i' \cup U_i \cup \{s\}$. Zusammen sind diese dann insbesondere eine zulässige Lösung für die **HetTSP**-Instanz $\mathcal{J}$ und es gilt:

$$\begin{aligned} d(\tau_i) &\leq 2 \cdot \mathrm{MST}(U_i' \cup \{s\}) = 2 \cdot \left(\mathrm{MST}(U_i \cup \{s\}) + \sum_{v' \in U_i'} \tilde{d}(v, v')\right) \\ &= 2 \cdot \left(\mathrm{MST}(U_i \cup \{s\}) + \sum_{v \in U_i} c_i(v) \frac{d(s,v)}{Q}\right) \leq 2 \cdot (d(\sigma_i) + d(\sigma_i)) = 4 \cdot d(\sigma_i) \end{aligned}$$

Die erste Abschätzung gilt, da man eine zulässige Tour erhalten kann, indem man einen minimalen Spannbaum nimmt, alle Kanten verdoppelt und dann eine Eulertour bestimmt. Bei der zweiten Abschätzung ist:

- $\mathrm{MST}(U_i \cup \{s\}) \leq d(\sigma_i)$, denn σ_i besucht alle Knoten in U_i, enthält also insbesondere einen Spannbaum von $U_i \cup \{s\}$.
- $\sum_{v \in U_i} c_i(v) \frac{d(s,v)}{Q} \leq d(\sigma_i)$, denn σ_i ist eine zulässige Lösung für das **CVRP**-Problem auf U_i mit Bedarfen $c_i(v)$. Damit folgt die Ungleichung aus Proposition 2.2.

Insgesamt folgt daher:

$$\mathrm{OPT}_{\textbf{HetTSP}}(\mathcal{J}) \leq \max d(\tau_i) \leq 4 \cdot \max d(\sigma_i) = 4 \cdot \mathrm{OPT}_{\textbf{HetCVRP}}(\mathcal{I})$$

$\square$

Algorithm 9 Deduktion

```
1: procedure DEDUKTION(𝓘, 𝓙, (τ_i))
2:     for i ← 1, ..., k do
3:         c_i(v) := |{v^(j) | v^(j) ∈ τ_i}|
4:         σ_i ← CVRP-APPROX(G, d, Q, (c_i(v))_{v∈V})
5:     end for
6:     return (σ_i)
7: end procedure
```

Lemma 4.2. *Für jede Lösung (τ_i) von $\mathcal{J}$ ist (σ_i) eine Lösung von $\mathcal{I}$ mit:*

$$\max \frac{d(\sigma_i)}{\lambda_i} \in \mathcal{O}(1) \cdot \max \frac{d(\tau_i)}{\lambda_i}$$

Beweis. Nach Lemma 2.3 liefert Algorithmus 2 eine Lösung mit

$$d(\sigma_i) \leq \rho \cdot \mathrm{OPT}_{\mathbf{TSP}}(U_i', d) + \frac{2}{Q} \sum_{v' \in U_i'} d'(s, v')$$

Ferner ist

$$\mathrm{OPT}_{\mathbf{TSP}}(U_i', d) \leq d(\tau_i) \text{ und } \frac{2}{Q} \sum_{v' \in U_i'} d'(s, v') \leq d(\tau_i)$$

Also zusammen

$$d(\sigma_i) \leq (\rho + 1) \cdot d(\tau_i)$$

□

Satz 4.3 (Theorem 4.1 in [Gø+10])**.** *Es gibt eine $\mathcal{O}(1)$-approximationserhaltende Reduktion von **HetTSP** auf **HetCVRP**.*

Beweis. Verwende Algorithmus 8 um die Eingabe zu einer Instanz von **HetTSP** zu machen. Finde eine $\mathcal{O}(1)$-approximative Lösung für diese mit Hilfe von Algorithmus 3. Schließlich erhalte aus dieser mit Algorithmus 9 eine Lösung der **HetCVRP**-Instanz.

Für diese Lösung gilt:

$$\max \frac{d(\sigma_i)}{\lambda_i} \in \mathcal{O}(1) \cdot \max \frac{d(\tau_i)}{\lambda_i} \subseteq \mathcal{O}(1) \cdot \mathrm{OPT}_{\mathbf{HetTSP}}(\mathcal{J}) \subseteq \mathcal{O}(1) \cdot \mathrm{OPT}_{\mathbf{HetCVRP}}(\mathcal{I})$$

□

Literatur

[Coo+11] W.J. Cook u. a. *Combinatorial Optimization*. Wiley Series in Discrete Mathematics and Optimization. Wiley, 2011. ISBN: 9781118031391. URL: https://books.google.de/books?id=tarLTNwM3gEC.

[FHK76] G. N. Frederickson, M. S. Hecht und C. E. Kim. „Approximation algorithms for some routing problems“. In: *Foundations of Computer Science, 1976., 17th Annual Symposium on*. 1976, S. 216–227. DOI: 10.1109/SFCS.1976.6.

[Gø+10] Inge Li Gørtz u. a. „Capacitated Vehicle Routing with Non-Uniform Speeds“. In: *CoRR* abs/1012.1850 (2010). URL: http://arxiv.org/abs/1012.1850.

[HK85] M. Haimovich und A. H. G. Rinnooy Kan. „Bounds and Heuristics for Capacitated Routing Problems“. In: *Mathematics of Operations Research* 10.4 (1985), S. 527–542. ISSN: 0364765X, 15265471. URL: http://www.jstor.org/stable/3689422.

[LST90] Jan Karel Lenstra, David B. Shmoys und Éva Tardos. „Approximation algorithms for scheduling unrelated parallel machines“. In: *Mathematical Programming* 46.1 (1990), S. 259–271. ISSN: 1436-4646. DOI: 10.1007/BF01585745. URL: http://dx.doi.org/10.1007/BF01585745.