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 $d(\tau)$

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 $\max d(\tau_i)$

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 $\max \frac{d(\tau_i)}{\lambda_i}$

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 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 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 Algorithmus für HetTSP

Algorithm 1 HetTSP-Approx

procedure $\textsc{HetTSP}(G = (V, E),\ d : E \to \mathbb{R}_{\geq 0})$
  Rate M mit $\frac{M}{2} \leq \mathrm{OPT} \leq M$
  $\mathcal{H} := (H_i)_{i \geq 0} \leftarrow \textsc{Level-Prime}\ (G, d)$
    // $\mathcal{H}$ erfüllt: Wurzel-Blatt Pfade haben aufsteigende Knoten-Level
    //   und $\forall i : \sum_{j \geq i} d\,(H_j) \leq 8M \sum_{j \geq i-1} 2^j \mu_j$ (wenn M korrekt geraten)
  $\mathcal{T} := (\mathcal{T}_i)_{i \geq 0} \leftarrow \textsc{Decomposition}\ (\mathcal{H})$
    // $\mathcal{T}$ ist $(6, 40)$-zuweisbarer Wald
  $(x_{ij}) \leftarrow \textsc{FractionalAssignment}\ (\mathcal{T})$
  $(\tau_i) \leftarrow \textsc{RoundingAssignment}\ (x_{ij})$
    // $\mathcal{T}$ ist (α, β)-zuweisbar $\Rightarrow$ (τ_i) ist $(4\alpha + 2\beta)$-approx.
  return (τ_i)
end procedure

Satz 2.1 (Theorem 1.1 in [Gø+10]). *Algorithmus 1 ist ein $\mathcal{O}(1)$-approximativer Algorithmus für* HetTSP.

2.1 Level-Prime

Algorithm 2 Level-Prime

procedure $\textsc{Level-Prime}(G = (V, E),\ d : E \to \mathbb{R}_{\geq 0})$
  $V_0 := \{v \in V \mid d(s, v) \leq M\}, \quad V_i := \{v \in V \mid 2^{i-1}M < d(s, v) \leq 2^i M\}$
  for $i \geq 0$ **do** $H_i \leftarrow$ Minimaler Spannbaum auf $G\,[V_{\leq i}]\,/V_{<i}$ **end for**
  return $(H_i)_{i \geq 0}$
end procedure

Lemma 2.2 (Theorem 3.3 in [Gø+10]). *Ein von Algorithmus 2 gefundener Baum $(H_l)_{l \geq 0}$ erfüllt:*

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

Korollar 2.3 (Korollar 3.5 in [Gø+10]). *Ein von Algorithmus 2 gefundener Baum $(H_l)_{l \geq 0}$ erfüllt:*

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

2.2 Decomposition-Algorithmus

Algorithm 3 Decomposition

procedure DECOMPOSITION($(\mathcal{H})$)
  $\mathcal{S}_0 := \{H_0\}$, $\mathcal{S}_i :=$ Zerl. von $\mathcal{H} \cap E_i$ in Bäume mit genau einer Kante nach V_i

  return $(\mathcal{T}_i)_{i \geq 0}$
end procedure

Definition 2.4 (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 $k \geq 1$ gilt: $\sum_{l>k} d(\mathcal{T}_l) \leq \beta M \sum_{l \geq k} 2^l \mu_l$
  d.h. die Fahrzeuge mit Geschw. $\geq 2^k$ können den Wald $\mathcal{T}_{>k}$ in $\mathcal{O}(\beta M)$ besuchen.

Lemma 2.5 (Lemma 3.11 in [Gø+10])**.** *Die von Algorithmus 3 bestimmte Zerlegung $\mathcal{T} = (\mathcal{T}_i)_{i \geq 0}$ ist $(6, 40)$-zuweisbar.*

2.3 Assignment-Algorithmen

Algorithm 4 FractionalAssignment

procedure FRACTIONALASSIGNMENT($(\mathcal{T})$)

  return (x_{ij})
end procedure

Proposition 2.6 (Seite 54 (?) in [Coo+11])**.**

Algorithm 5 RoundingAssignment

procedure ROUNDINGASSIGNMENT((x_{ij}))

  return (τ_i)
end procedure

Proposition 2.7 (Theorem 1 in [LST90])**.**

Lemma 2.8 (Lemma 3.2 in [Gø+10])**.** *Gegeben einen (α, β)-zuweisbaren Wald, liefern Algorithmus 4 und Algorithmus 5 eine $(4\alpha + 2\beta)$-approximative Lösung für* HETTSP*.*

3 Algorithmus für HetCVRP

Satz 3.1 (Theorem 4.1 in [Gø+10])**.** *Es gibt eine $\mathcal{O}(1)$-approximationserhaltende Reduktion von* HetTSP *auf* HetCVRP*.*

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.

[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.

[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.