Universität Augsburg

Seminar: Approximationsalgorithmen und Spieltheorie

Dozent: Prof. Dr. Tobias HARKS

Lukas GRAF 12. Mai 2016 Sommersemester 2016

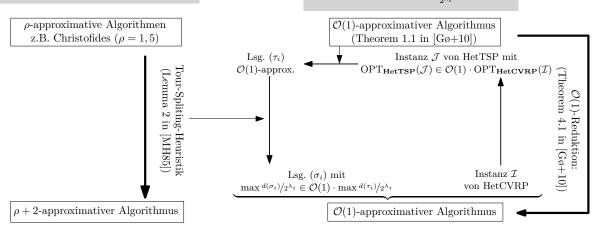
Heterogenes k-CVRP

Metrisches TSP:

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

Heterogenes k-TSP:

- Vollständiger Graph G = (V, E)
- Metr. Abstands funktion $d:E\to\mathbb{R}$
- Startpunkt $s \in V$
- k Fahrzeuge mit Geschw. $(2^{\lambda_i})_{i=1}^k$
- Lsgen: Touren (τ_i) , die bei s beginnen und gemeinsam ganz V abdecken
- **Ziel:** Minimiere max $\frac{d(\tau_i)}{2^{\lambda_i}}$



(Metrisches) CVRP:

- Vollständiger Graph G = (V, E)
- Metr. Abstands funktion $d:E\to\mathbb{R}$
- Startpunkt/Depot $s \in V$
- Kapazität Q (polynomiell in Eingabe)
- Bedarfe $(q_v)_{v \in V}, q_v \in \{0, ..., Q\}$
- Lsgen: Route (σ) , die bei s beginn alle Bedarfe erfüllen nie mehr als Q Elemente transportiert
- **Ziel:** Minimiere $d(\sigma)$

Heterogenes k-CVRP:

- Vollständiger Graph G = (V, E)
- Metr. Abstands funktion $d:E\to\mathbb{R}$
- Startpunkt/Depot $s \in V$
- \bullet einheitliche Kapazität Q
- Bedarfe $(q_v)_{v \in V}, q_v \in \{0, ..., Q\}$
- k Fahrzeuge mit Geschw. (2^{λ_i})
- Lsgen: Touren (σ_i) , die bei s beginnen, gemeinsam alle Bedarfe erfüllen, wobei kein Fahrzeug jemals mehr als Q Elemente transportiert
- **Ziel:** Minimiere max $\frac{d(\sigma_i)}{2^{\lambda_i}}$

[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.

 $[G\phi+10] \\ \mbox{Inge Li Gϕrtz u.a. ,$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.

Algorithmus für heterogenes k-TSP

Algorithm 1 Hettsp-Approx(G, d)

- 1: Rate M mit OPT $\leq M \leq 2 \cdot \text{OPT}$
- 2: $\mathcal{H} := (H_l)_{l>0} \leftarrow \text{Level-Prime } (G, d)$
- 3: $\mathcal{T} := (\mathcal{T}_l)_{l \geq 0} \leftarrow \text{Decomposition } (\mathcal{H})$
- 4: $(x_{Ti}) \leftarrow \bar{\text{FractionalAssignment}} (\mathcal{T})$
- 5: $(\tau_i) \leftarrow \text{RoundingAssignment } (x_{Ti})$
- 6: return (τ_i)

Algorithm 2 Level-Prim(G, d)

- 1: $V_0 := \{ v \in V \mid d(s, v) \le M \}$ 2: $V_l := \{ v \in V \mid 2^{l-1}M < d(s, v) \le 2^l M \}$
- 3: for $i \ge 0$ do $H_l \leftarrow \text{MST}$ auf $G[V_{\le l}]/\hat{V}_{< l}$
- 4: **return** $(H_l)_{l>0}$

Algorithm 3 Decomposition(\mathcal{H})

- 1: $S_0 := \{H_0\}$
- 2: $\mathcal{S}_l := \operatorname{Zerl}. \ \mathcal{H} \cap E_l$ in Bäume mit Wurzel in $V_{< l}$
- 3: $\mathcal{S}_l^{\geq} := \left\{ \tau \in \mathcal{S}_l \mid d(\tau) \geq 2^{l-3} M \right\}, \mathcal{S}_l^{\leq} := \mathcal{S}_l \setminus \mathcal{S}_l^{\geq}$
- 4: for $\tau \in \mathcal{S}_l^{<}$ do $h(\tau) := \tau' \in \mathcal{S}_{l-1}^{\geq}$ mit $\tau \cup \tau'$ zsh
- 5: for $\tau \in \mathcal{S}_l^{\geq}$ do
- $\mathcal{T}_l(\tau) \leftarrow \text{Partition von } \tau \cup h^{-1}(\tau) \text{ in Bäume der}$ Länge $\left[2^{l+1}M,2^{l+2}M\right]$ (und evtl. ein kürzerer)
- $\mathcal{T}'_l(\tau) \leftarrow \{T_r \cup \{ \text{ Kante zu } s \} \mid T_r \in \mathcal{T}_l(\tau) \}$
- 8: end for
- 9: $\mathcal{T}_l := \bigcup_{\tau \in \mathcal{S}_i^{\geq}} \mathcal{T}_l'(\tau)$
- 10: **return** $(\mathcal{T}_l)_{l>0}$

Algorithm 4 Fractional Assignment (\mathcal{T})

- 1: $L := \{T \in \mathcal{T}\}, \quad b(T) := \overline{d(T)}$
- 2: $R := \{i \mid 1 \le i \le k\}, \quad b(i) := \beta M 2^{\lambda_i}$
- 3: $F := \{ \{T, i\} \mid T \in \mathcal{T}_l, \lambda_i \ge l 1 \}$
- 4: $(x_{Ti}) \leftarrow L$ -überdeckendes b-Matching
- 5: return (x_{Ti})

Algorithm 5 ROUNDINGASSIGNMENT (x_{Ti})

- 1: $(x'_{Ti}) \leftarrow \text{RoundScheduling}(p_{Ti} := \frac{d(T)}{2^{\lambda_i}}, \tilde{x}_{Ti} :=$
- 2: $\tau_i \leftarrow \text{Tour durch die Bäume } T \text{ mit } x'_{Ti} = 1.$
- 3: return (τ_i)

Satz 1 (Theorem 1.1 in $[G\emptyset + 10]$).

Algorithmus 1 ist ein $\mathcal{O}(1)$ -approximativer Algorithmus für **HetTSP**.

Lemma 2 (Korollar 3.5 in $[G\emptyset+10]$). Ein von Algorithmus 2 gefundener Baum $(H_l)_{l>0}$ erfüllt:

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

Definition 3 (Definition 3.1 in $[G\emptyset + 10]$). 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: $\sum_{j>l} d(\mathcal{T}_j) \leq$ $\beta M \sum_{j>l} 2^j \mu_j$

d.h. die Fahrzeuge mit Geschw. $\geq 2^l$ können den Wald $\mathcal{T}_{>l}$ in $\mathcal{O}(\beta M)$ besuchen.

Lemma 4 (Lemma 3.11 in $[G\emptyset+10]$). Die von Algorithmus 3 bestimmte Zerlegung T = $(\mathcal{T}_i)_{i>0}$ ist (6,40)-zuweisbar.

Definition 5. (x_{Ti}) ist L-sättigendes b-Matching für $(L \cup R, F)$, wenn gilt:

$$\sum_{i} x_{Ti} = b(T), \quad \sum_{T} x_{Ti} \le b(i), \quad x_{Ti} \in \mathbb{N}$$

Lemma 6 (Seite 54f in [Coo+11]). Der in Algorithmus 4 definierte Hilfsgraph besitzt ein L-sättigendes b-Matching.

Lemma 7 (Theorem 1 in [LST90], Lemma 3.2 in $[G\emptyset+10]$).

Gegeben einen (α, β) -zuweisbaren Wald, liefern Algorithmus 4 und Algorithmus 5 eine $(4\alpha+2\beta)$ approximative Lösung für **HetTSP**.