Universität Augsburg
Seminar: Approximationsalgorithmen und Spieltheorie
Dozent: Prof. Dr. Tobias Harks

Lukas Graf
2. Juni 2016
Sommersemester 2016

Heterogenes k-CVRP

Metrisches TSP:

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

Heterogenes k-TSP:

- Vollständiger Graph $G = (V, E)$
- Metr. Abstandsfunktion $d : E \to \mathbb{R}$
- 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 $\max \frac{d(\tau_i)}{2^{\lambda_i}}$

ρ-approximative Algorithmen
z.B. Christofides ($\rho = 1,5$)

Tour-Splitting-Heuristik
(Lemma 2 in [HK85])

$\rho + 2$-approximativer Algorithmus

$\mathcal{O}(1)$-approximativer Algorithmus
(Theorem 1.1 in [Gø+10])

Lsg. (τ_i)
$\mathcal{O}(1)$-approx.

Instanz $\mathcal{J}$ von HetTSP mit
$\mathrm{OPT}_{\mathbf{HetTSP}}(\mathcal{J}) \in \mathcal{O}(1) \cdot \mathrm{OPT}_{\mathbf{HetCVRP}}(\mathcal{I})$

$\mathcal{O}(1)$-Reduktion:
(Theorem 4.1 in [Gø+10])

Lsg. (σ_i) mit
$\max d(\sigma_i)/2^{\lambda_i} \in \mathcal{O}(1) \cdot \max d(\tau_i)/2^{\lambda_i}$

Instanz $\mathcal{I}$
von HetCVRP

$\mathcal{O}(1)$-approximativer Algorithmus

(Metrisches) CVRP:

- Vollständiger Graph $G = (V, E)$
- Metr. Abstandsfunktion $d : E \to \mathbb{R}$
- 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 beginnend alle Bedarfe erfüllt und nie mehr als Q Elemente transportiert
- **Ziel:** Minimiere $d(\sigma)$

Heterogenes k-CVRP:

- Vollständiger Graph $G = (V, E)$
- Metr. Abstandsfunktion $d : E \to \mathbb{R}$
- Startpunkt/Depot $s \in V$
- einheitliche Kapazität Q
- Bedarfe $(q_v)_{v \in V}, q_v \in \{0, ..., Q\}$
- k Fahrzeuge mit Geschw. $(2^{\lambda_i})_{i=1}^k$
- **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.

[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.

Algorithmus für heterogenes k-TSP

Algorithm 1 HetTSP-Approx(G, d)

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

Algorithm 2 Level-Prim(G, d)

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

Algorithm 3 Decomposition$(\mathcal{H})$

$\mathcal{S}_0 := \{H_0\}$
$\mathcal{S}_l :=$ Zerl. $\mathcal{H} \cap E_l$ in Bäume mit Wurzel in $V_{<l}$
$\mathcal{S}_l^{\geq} := \left\{\tau \in \mathcal{S}_l \mid d(\tau) \geq 2^{l-3}M\right\}, \mathcal{S}_l^{<} := \mathcal{S}_l \backslash \mathcal{S}_l^{\geq}$
for $\tau \in \mathcal{S}_l^{<}$ **do** $h(\tau) := \tau' \in S_{l-1}^{\geq}$ mit $\tau \cup \tau'$ zsh
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)
  $\mathcal{T}_l'(\tau) \leftarrow \{T_r \cup \{$ 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 $(\mathcal{T}_l)_{l \geq 0}$

Algorithm 4 FractionalAssignment$(\mathcal{T})$

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

Algorithm 5 RoundingAssignment(x_{Ti})

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

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

Lemma 2 (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.*
- $\forall l \geq -1 : \sum_{j > l} d(H_j) \leq 8M \cdot \sum_{j \geq l} 2^j \mu_j$

Definition 3 (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 4 (Lemma 3.11 in [Gø+10]).
Die von Algorithmus 3 bestimmte Zerlegung $\mathcal{T} = (\mathcal{T}_l)_{l \geq 0}$ ist $(6, 40)$-zuweisbar.

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

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

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 [Gø+10]).
*Gegeben einen (α, β)-zuweisbaren Wald, liefern Algorithmus 4 und Algorithmus 5 eine $(8\alpha + 4\beta)$-approximative Lösung für **HetTSP**.*