Supplementary Materials for “Approximation Algorithms for the Min-Max Cycle Cover Problem with Neighborhoods”

Lijia Deng, Wenzheng Xu, Weifa Liang, Jian Peng, Yingjie Zhou, Lei Duan, and Sajal K. Das

1 ANALYSIS OF THE APPROXIMATION ALGORITHM FOR THE ROOTLESS MIN-MAX CYCLE COVER PROBLEM WITH NEIGHBORHOODS

Lemma 1. The POIs served by each $C_i^*$ of the optimal $K$ tours must be contained in a single connected component $CC_i$, where $1 \leq t \leq T$.

Proof: We prove by contradiction. Assume that the POIs served by $C_i^*$ are contained in multiple connected components. For simplicity, assume that a POI $u$ served by tour $C_i^*$ is contained in a connected component $CC_{i'}$, where $t \neq t'$, and $1 \leq t, t' \leq T$. On one hand, since both disks $D(u)$ and $D(v)$ are visited by tour $C_i^*$, assume that they are visited at locations $p_u^*$ and $p_{v}^*$, respectively. Then, the traveling time $l(p_u^*, p_v^*)$ between $p_u^*$ and $p_v^*$ is no more than half the total sum of the traveling time $w(C_i^*)$ of tour $C_i^*$, i.e., $l(p_u^*, p_v^*) \leq \frac{w(C_i^*)}{2}$. Also, notice that the minimum traveling time $l(D(u), D(v))$ between disks $D(u)$ and $D(v)$ is no more than $l(p_u^*, p_v^*)$. We thus have

$$l(D(u), D(v)) \leq l(p_u^*, p_v^*) \leq \frac{w(C_i^*)}{2} \leq \frac{OPT}{2}$$

On the other hand, as POIs $u$ and $v$ are contained in two different connected components, their minimum traveling time $l(D(u), D(v))$ must be strictly longer than $\frac{B}{2}$, by following the construction of graph $G$. This however contradicts Eq.(1). Then, the assumption that the POIs served by $C_i^*$ are contained in multiple connected components is incorrect. That is, the POIs served by each tour $C_i^*$ must be contained in a single connected component $CC_i$. The lemma then follows.

2 ANALYSIS OF THE APPROXIMATION ALGORITHM FOR THE SINGLE-ROOTED MIN-MAX CYCLE COVER PROBLEM WITH NEIGHBORHOODS

Lemma 4. Algorithm approAlgOneRoot can find a solution with no more than $K$ subtours and the total weight of each subtour $C_i$ does not exceed $\frac{1}{K}(w(C') - e_{\text{max}}) + e_{\text{max}}$, where $e_{\text{max}} = \max_{p \in P}\{h(v_i) + 2l(p_i, s)\}$, $h(v_i)$ is the service time of POI $v_i$, $p_i$ is the location where $v_i$ is served by a vehicle in the neighborhood of $v_i$, $l(p_i, s)$ is the traveling time between $p_i$ and the root $s$, and $1 \leq i \leq K$.

Proof: Recall that, Algorithm approAlgOneRoot constructs $K$ subtours $C_1, C_2, \ldots, C_K$ from tour $C'$, and the total weight of each subtour $C_i$ ($1 \leq i \leq K$) is upper bounded as follows.

For the first subtour $C_1$, the total weight from depot $s$ to $p_i$ is no more than $\frac{1}{K}(w(C') - e_{\text{max}}) + \frac{e_{\text{max}}}{2}$, by following the proposed algorithm, where $p_i$ is the last location of subtour $C_1$. Since the edge weight $w'(p_i, s)$ between $p_i$ and depot $s$ does not exceed $\frac{e_{\text{max}}}{2}$, the total weight of subtour $C_1$ is no greater than $\frac{1}{K}(w(C') - e_{\text{max}}) + e_{\text{max}}$, where $e_{\text{max}} = \max_{p \in P}\{h(v_i) + 2l(p_i, s)\}$.

For each subtour $C_i$ with $2 \leq i \leq K - 1$, the total weight of the $i$-th split path from locations $p_{i-1}$ to $p_i$ is no greater than $\frac{1}{K}(w'(C') - e_{\text{max}}) + \frac{e_{\text{max}}}{2}$. Following Algorithm approAlgOneRoot. It can also be seen that both the edge weights $w'(s, p_{i-1})$ between $s$ and $p_{i-1}$, and $w'(p_i, s)$ between $p_i$ and $s$ do not exceed $\frac{e_{\text{max}}}{2}$.

Finally, the total weight of subtour $C_K$ is

$$w'(C_K) \leq \frac{1}{K}(w'(C') - e_{\text{max}}) + \frac{e_{\text{max}}}{2} + e_{\text{max}}$$

$$= \frac{1}{K}(w'(C') - e_{\text{max}}) + \frac{e_{\text{max}}}{2} + e_{\text{max}}$$

$$= \frac{1}{K}(w'(C') - e_{\text{max}}) + e_{\text{max}}.$$
Therefore, the total consumed time \( w(C_i) \) of each sub-tour \( C_i \) is no greater than \( \frac{1}{K}(w(C') - c_{\text{max}}) + c_{\text{max}} \) with \( 1 \leq i \leq K \). Also, recall that \( w(C') = w(C') \), we have \( w(C_i) \leq \frac{1}{K}(w(C') - c_{\text{max}}) + c_{\text{max}} \) with \( 1 \leq i \leq K \).

3 ANALYSIS OF THE APPROXIMATION ALGORITHM FOR THE MULTI-ROOTED MIN-MAX CYCLE COVER PROBLEM WITH NEIGHBORHOODS

Lemma 6. Assume that a tour node \( u_i \) is not matched to any depot in the maximum matching \( M \) of graph \( G = (U \cup S, E) \). Construct a maximal \( u_i \)-rooted tree \( T_A = (U_A \cup S_A, E_A) \) in \( G \), so that each path from node \( u_i \) to any leaf node in tree \( T_A \) is an alternating path. We have that

(i) Each leaf node in tree \( T_A \) is matched in \( M \).
(ii) Each leaf in \( T_A \) is a tour node but not a depot.
(iii) There is at least one depot in set \( S_A \), i.e., \( |S_A| \geq 1 \).
(iv) The number of tour nodes is one more than the number of depots in tree \( T_A \), i.e., \( |U_A| = |S_A| + 1 \).
(v) Assume that the tour nodes in \( U_A \) represent \( n'_U \) tours \( C_1, C_2, \ldots, C_n' \), where \( n'_U = |U_A| \). For each disk \( D(v) \) visited by one of the \( n'_U \) tours, assume that \( D(v) \) is visited by an optimal tour \( C_k \) that contains depot \( s_k \), where \( D(v) \) is the disk that centers at POI \( v \). Then, depot \( s_k \) must be contained in set \( S_A \) if \( B \geq \text{OPT}_m \).

Proof: We only prove the first three claims here. The proofs for claims (iv) and (v) are in the main body of the paper.

We first show claim (i) that each leaf node in tree \( T_A \) is matched in \( M \) by contradiction. Assume that there is a leaf that is not matched in \( M \). Consider the alternating path from \( u_i \) to the leaf. Since both \( u_i \) and the leaf node are not matched, we can construct another matching \( M' \) from the path and matching \( M \), such that the number of matched edges in \( M' \) is one more than the number in \( M \) (i.e., \( |M'| = |M| + 1 \)), by following the Lemma 1 in [11]. This however contradicts that \( M \) is the maximum matching in \( G \). Thus, each leaf must be matched in \( M \).

We then prove claim (ii) that each leaf in \( T_A \) is a tour node. Assume that there is a leaf that is a depot. Consider the path from \( u_i \) to the leaf in \( T_A \). Since the path is an alternating path and the first node \( u_i \) in the path represents an unmatched tour \( C_j \), we know that the odd order (i.e., 1st, 3rd, 5th, \ldots) nodes in the path are tour nodes, while the even order (i.e., 2nd, 4th, 6th, \ldots) nodes are depots. Also, assume that a depot \( s_k \) is the \((2j)\)th node in the path, where \( j \geq 1 \). Then, depot \( s_k \) must be matched to the \((2j + 1)\)th tour node in the path, if the path has at least \( 2j + 1 \) nodes. Since the last node in the path is a depot, it is not matched in \( M \). This contradicts claim (i) that each leaf node is matched. Then, we know that each leaf is a tour node.

We prove claim (iii) that there is at least one depot in set \( S_A \), i.e., \( |S_A| \geq 1 \). Consider any disk \( D(v) \) visited by the tours that are represented by the nodes in \( U_A \). Following the proposed algorithm, any guess \( B \) of \( \text{OPT}_m \) is no less than twice the traveling time

\[
\min_{k=1}^{K} \{ l(D(v), s_k) \} \text{ between disk } D(v) \text{ and its nearest depot in } S, \text{ i.e., } B \geq 2 \min_{k=1}^{K} \{ l(D(v), s_k) \}. \]

Assume that \( s_{k'} = \arg \min_{k=1}^{K} \{ l(D(v), s_k) \} \). Also, assume that POI \( v \) is served at a location \( p \in D(v) \) in a tour \( C \). Then, the traveling time \( l(p, s_{k'}) \) between location \( p \) and depot \( s_{k'} \) is no larger than \( l(D(v), s_{k'}) + 2r \), as \( p \) is in \( D(v) \). Then, we have

\[
l(C, s_{k'}) \leq l(p, s_{k'}) \text{, as location } p \text{ is in tour } C \\
\leq l(D(v), s_{k'}) + 2r \\
\leq \frac{B}{2} + 2r.
\]

(3)

Following the construction of graph \( G \), we know that there is an edge between the tour node \( u \) (representing tour \( C \)) and depot \( s_{k'} \). Therefore, there is at least one depot in \( S_A \). The lemma then follows.