Journal of the Operations Research Society of Japan
Vol. 63, No. 2, April 2020, pp. 60–70

IMPROVING APPROXIMATION RATIOS FOR THE CLUSTERED TRAVELING SALESMAN PROBLEM

Masamune Kawasaki
Tokyo Institute of Technology

Kenjiro Takazawa
Hosei University

(Received March 8, 2019; Revised October 10, 2019)

Abstract The clustered traveling salesman problem (CTSP) is a generalization of the traveling salesman problem (TSP) in which the set of cities is divided into clusters and the salesman must consecutively visit the cities of each cluster. It is well known that TSP is NP-hard, and hence CTSP is NP-hard as well. Guttmann-Beck et al. (2000) designed approximation algorithms for several variants of CTSP by decomposing it into subproblems including the traveling salesman path problem (TSPP). In this paper, we improve approximation ratios by applying a recent improved approximation algorithm for TSPP by Zenklusen (2019).

Keywords: Combinatorial optimization, clustered traveling salesman problem, traveling salesman problem, approximation algorithms

1. Introduction

Let $G = (V, E)$ be a complete undirected graph with vertex set V and edge set E associated with edge weights $w(e)$ $(e \in E)$ satisfying the triangle inequality. Let V be divided into clusters $V_1, \ldots, V_k$. *The clustered traveling salesman problem* (CTSP) is a problem of computing a shortest Hamilton cycle which visits the vertices of each cluster consecutively. The traveling salesman problem (TSP) can be viewed as a special case of CTSP where there is only one cluster $V_1 = V$ or each V_i consists of only one vertex.

Guttmann-Beck et al. [6] presented approximation algorithms for the following variants of CTSP:

(1) The starting and ending vertices in each cluster are given.
(2) The two ending vertices in each cluster are given. We are free to choose any one as the starting vertex and the other one as the ending vertex.
(3) Only the starting vertex in each cluster is given.
(4) No specific starting and ending vertices in each cluster are given.

Since all the variants are generalizations of TSP, they are all NP-hard. Table 1 shows approximation ratios for these variants by Guttmann-Beck et al. [6]. Their algorithms use previously known approximation algorithms to the following three closely related problems: the traveling salesman path problem (TSPP), the stacker crane problem (SCP), and the rural postman problem (RPP), which will be described in Section 2. In particular, Hoogeveen's 5/3-approximation algorithm for TSPP plays a key role in their algorithms.

The aim of this paper is to improve the approximation ratios for CTSP by incorporating a recent approximation algorithm for TSPP with improved approximation ratio 1.5 by Zenklusen [11]. We show that this improvement for TSPP improves the approximation

Table 1: Approximation ratios by Guttmann-Beck et al. [6]

Variant	Ratio
(1) Start and end vertices are given	1.9091
(2) Two end vertices are given	1.8
(3) Only starting vertex is given	2.643
(4) End vertices are not specified	2.75

ratios for Variant (1), (2), and (4), while it does not directly improves that for Variant (3). Table 2 shows the approximation ratios obtained in this paper.

Table 2: Approximation ratios obtained in this paper

Variant	Ratio
(1) Start and end vertices are given	1.875
(2) Two end vertices are given	1.714
(3) Only starting vertex is given	2.643 (Unchanged)
(4) End vertices are not specified	2.67

Bao et al. [1] recently improved the approximation ratios for variant (3) and (4). The approximation ratio for variant (3) is 1.9 and that for variant (4) is 2.5. We remark that, while they use Hoogeveen's 5/3-approximation algorithm for TSPP, applying Zenklusen's algorithm does not directly result in the improvement of the approximation ratio for variant (3) and (4).

The rest of the paper is organized as follows. In Section 2 we review approximation algorithms for TSP, TSPP, SCP, and RPP. Section 3 describes the approximation algorithms for CTSP by Guttmann-Beck et al. [6]. In Section 4, we improve the approximation ratios for CTSP by incorporating Zenklusen's approximation algorithm for TSPP. Section 5 concludes this paper.

2. Preliminaries

In this section, after describing some definitions and notation, we review TSP, TSPP, SCP, and RPP together with previous approximation algorithms for those problems. For a graph $G = (V, E)$, we denote by $w(e)$ the weight of an edge $e \in E$. For a subset $E' \subseteq E$, we denote $w(E') = \sum_{e \in E'} w(e)$, the total weight of the edges in E'. Let OPT denote both an optimal solution of the problem under consideration and its total weight, and MST(G) denote both a minimum-weight spanning tree of G and its weight. In this paper, we always assume that the edge weights satisfy the triangle inequality for a complete graph G, i.e.,

$$w(a, b) + w(b, c) \geq w(a, c) \quad (a, b, c \in V).$$

2.1. The Traveling Salesman Problem

Let $G = (V, E)$ be a complete undirected graph with edge weights $w(e)$ $(e \in E)$. The traveling salesman problem (TSP) is a problem of computing a Hamitlon cycle of minimum weight. The algorithm by Christofides [2] attains the current best approximation ratio 1.5.

2.2. The Traveling Salesman Path Problem

Let $G = (V, E)$ be a complete undirected graph with edge weights $w(e)$ $(e \in E)$. *The traveling salesman path problem* (TSPP) is a problem of computing a Hamilton path of minimum weight. Hoogeveen [7] considered the following three variants of TSPP:

1. Both end vertices are given.
2. Only one of the end vertices is given.
3. No end vertices are given.

Hoogeveen designed approximation algorithms for these three variants of TSPP. The approximation ratio for Variant 1 is 5/3 and that for Variants 2 and 3 is 1.5. Guttmann-Beck et al. [6] modified the algorithm for Variant 1 so that it can be applied to CTSP, while maintaining the approximation ratio.

Theorem 2.1 ([6]). *For TSPP in $G = (V, E)$ with two end vertices $s, t \in V$ given, there exists a polynomial algorithm that finds Hamilton paths S_1 and S_2 between s and t satisfying the following inequations:*

$$w(S_1) \leq 2\mathrm{OPT} - w(s,t),$$
$$w(S_2) \leq \frac{3}{2}\mathrm{OPT} + \frac{1}{2}w(s,t).$$

It is straightforward to see that the length of the shorter of the paths S_1 and S_2 is at most 5OPT/3: it holds that $w(S_1) \leq 5\mathrm{OPT}/3$ if $w(s,t) \geq \mathrm{OPT}/3$, and $w(S_2) \leq 5\mathrm{OPT}/3$ otherwise.

Recently, Zenklusen [11] devised a new algorithm for Variant 1 of TSPP with an improved approximation ratio 1.5. By combining the algorithms by Guttmann-Beck et al. [6] and Zenklusen [11], we can straightforwardly obtain the following.

Lemma 2.1. *For TSPP in $G = (V, E)$ with two end vertices $s, t \in V$ given, there exists a polynomial algorithm that finds Hamilton paths S_1 and S_2 between s and t satisfying the following inequations:*

$$w(S_1) \leq 2\mathrm{OPT} - w(s,t),$$
$$w(S_2) \leq \frac{3}{2}\mathrm{OPT}.$$

2.3. The Stacker Crane Problem

Let $G = (V, E, D)$ be a mixed multigraph with undirected edge set E and directed set D. The undirected graph (V, E) is a complete graph with edge weights $w(e)$ $(e \in E)$ satisfying the triangle inequality. Each vertex is either the head s_i or the tail t_i of exactly one directed edge in D. A directed edge is often called an arc or a special arc. *The stacker crane problem* (SCP) is a problem of computing a Hamilton cycle of minimum weight that traverses each arc (s_i, t_i) in the specified direction from s_i to t_i. The arc (s_i, t_i) represents that an object at vertex s_i must be moved to vertex t_i using a vehicle called stacker crane.

Since TSP can be reduced SCP by replacing each vertex by an arc of zero-weight, SCP is a generalization of TSP and hence NP-hard. Frederickson et al. [5] designed a 1.8-approximation algorithm for SCP. It finds two different solutions and then chooses the better of the two solutions. We briefly review the basic ideas in the two algorithms and name the whole algorithm Algorithm SCP. Let $U = \sum_i w(s_i, t_i)$ and $A = \mathrm{OPT} - U$.

- Algorithm SmallArcs: Shrink each arc to obtain a vertex, execute Christofides' algorithm and then adjoin the shrunk arcs to obtain a solution for the original problem.
- Algorithm LargeArcs: Find a minimum-weight bipartite matching between the heads and tails and add the set of directed egdes to obtain a directed cycle cover. We shrink

the resulting cycles to obtain vertices, and find a minimum-weight spanning tree for the vertices. We add two copies of each edge, and orient the copies in opposite directions. The resulting graph is an Eulerian tour. Finally, we make it into a Hamilton cycle by shortcutting vertices visited more than once.

If U is small compared to OPT, then the problem is close to TSP and Algorithm SmallArcs finds a good solution, while Algorithm LargeArcs works well if U is large. The weight of the solutions can be bounded as follows.

Theorem 2.2 ([5]). *Algorithm SmallArcs finds a solution to SCP with weight at most* $3A/2 + 2U$*, and Algorithm LargeArcs finds a solution to SCP with weight at most* $3A + U$*.*

2.4. The Rural Postman Problem

Let $G = (V, E)$ be a complete undirected graph, and $E' \subseteq E$ be a specified subset of edges. *The rural postman problem* (RPP) is a problem of computing a Hamilton cycle of minimum weight that traverses all the edges in E'. RPP is NP-hard, and Frederickson [4] designed a 1.5-approximation algorithm for RPP. (See also Eiselt et al. [3] and Jansen [8].) It finds two different solutions in the following manner and then choose the better of the two solutions. We name the whole algorithm Algorithm RPP.

- Algorithm SmallEdges: This Algorithm is an adaptation of Algorithm SmallArcs, where D is replaced with E'. It becomes greatly simplified when applied to RPP and turns out to be a straightforward generalization of Christofides' algorithm for TSP.
- Algorithm LargeEdges: This algorithm is similar to Algorithm LargeArcs for SCP. The difference is that D is a set of undirected edges.

Similarly as Theorem 2.2, the following theorem is established.

Theorem 2.3 ([4]). *Algorithm SmallEdges finds a solution to RPP with weight at most* $3(A+U)/2$*, and Algorithm LargeEdges finds a solution to RPP with weight at most* $3A+U$*.*

3. Previous Approximation Algorithms for CTSP

In this section, we describe the approximation algorithms for CTSP by Guttmann-Beck et al. [6]. Recall that CTSP is a problem of computing a shortest Hamilton cycle which visits the vertices of each cluster consecutively.

3.1. Start and end vertices are given

In this subsection, we describe the approximation algorithm for Variant (1) of CTSP: the starting vertex s_i and ending vertex t_i are given for each cluster V_i ($i = 1, \ldots, k$). The algorithm by Guttmann-Beck et al. [6] is based on the following idea. We divide the problem into two parts. Firstly, we find a Hamilton path P_i between s_i and t_i in each cluster V_i. Next, we connect the paths $P_1, \ldots, P_k$ to obtain a Hamilton cycle in the following manner. We replace each cluster V_i by a special arc from s_i to t_i to obtain an instance of SCP. We then find a solution to the instance of SCP, and replace each arc (s_i, t_i) by P_i. The algorithm is summarized as follows.

Algorithm 1

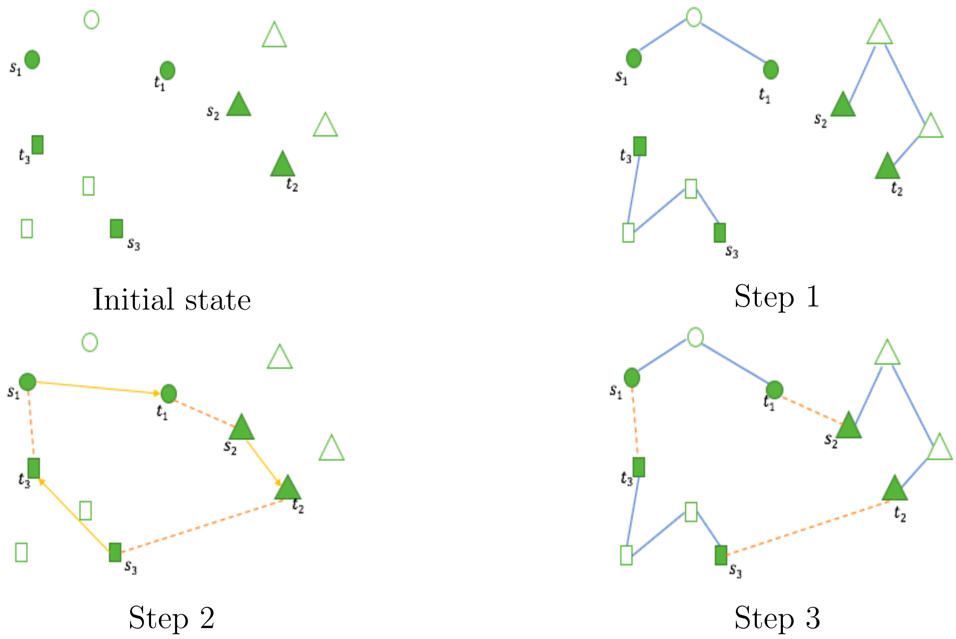
Step 1: For each cluster V_i ($i = 1, \ldots, k$), compute a Hamilton path P_i with starting and ending vertices s_i and t_i.

Step 2: Apply Algorithm SCP to the graph with special arcs $\{(s_i, t_i) \mid i = 1, \ldots, k\}$ to obtain Hamilton cycle T.

Step 3: In T, replace the special arc (s_i, t_i) by P_i for each $i = 1, \ldots, k$, and return the resulting Hamilton cycle.

Figure 1 illustrates an example. By using Lemma 2.1 and Theorem 2.2, Guttmann-Beck et al. [6] proved that Algorithm 1 yields 21/11-approximation.

Theorem 3.1 ([6]). *The approximation ratio of Algorithm 1 is* $21/11 \approx 1.9091$.



Initial state

Step 1

Step 2

Step 3

Figure 1: Illustration of Algorithm 1.

3.2. Two end vertices are given

In this subsection, we consider Variant (2) of CTSP: for each cluster V_i $(i = 1, \dots, k)$, we are given two specified vertices s_i^1 and s_i^2. We are free to choose any one of them as the starting vertex and the other vertex as the ending vertex. We modify Algorithm 1 by applying Algorithm RPP instead of Algorithm SCP, since each P_i can be oriented in any direction. The algorithm is summarized as follows.

Algorithm 2.

Step 1: For each cluster V_i $(i = 1, \dots, k)$, compute a Hamilton path P_i with ending vertices s_i^1 and s_i^2.

Step 2: Apply Algorithm RPP to the graph with the special edges $\{(s_i^1, s_i^2) \mid i = 1, \dots, k\}$ to obtain Hamilton cycle T.

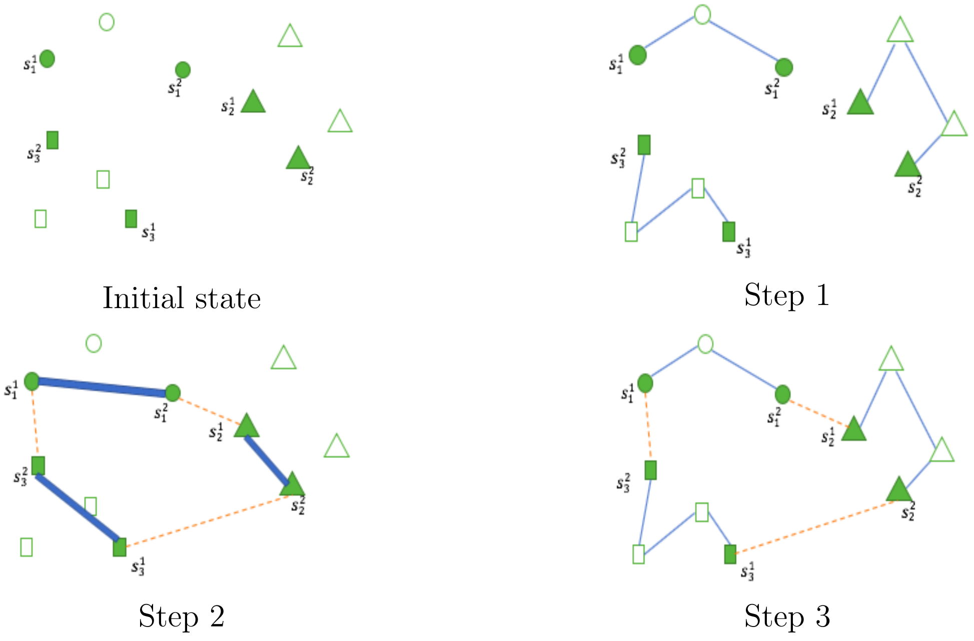
Step 3: In T, replace the special edge (s_i^1, s_i^2) by P_i for each $i = 1, \dots, k$, and return the resulting Hamilton cycle.

Figure 2 illustrates an example. By using Lemma 2.1 and Theorem 2.3, Guttmann-Beck et al. [6] proved that Algorithm 2 yields 1.8-approximation.

Theorem 3.2 ([6]). *The approximation ratio of Algorithm 2 is* 1.8.

3.3. End vertices are not specified

In this subsection, we consider Variant (4) of CTSP: we are free to choose the starting and ending vertices in the clusters. The approximation algorithm in [6] executes two different heuristic algorithms, and select the shorter of the obtained Hamilton cycles. The first algorithm is as follows. We apply a TSPP algorithm with unspecified ends to obtain a Hamilton path P_i in each cluster V_i $(i = 1, \dots, k)$. We then define the edges between the



Initial state

Step 1

Step 2

Step 3

Figure 2: Illustration of Algorithm 2.

ends of each P_i as special edges for an RPP instance, and compute an approximate RPP solution of this instance. We finally replace each special edge by the corresponding path P_i.

The second algorithm is as follows. In each cluster V_i $(i = 1, \dots, k)$, choose two vertices s_i and t_i that maximize $w(s_i, t_i)$ to be the end vertices in the cluster, and then apply Algorithm 2 (Section 3.2). The whole algorithm is summarized as follows.

Algorithm 3.

Step 1:

(a) Apply a TSPP algorithm with unspecified ending vertices to each V_i $(i = 1, \dots, k)$. Let P_i $(i = 1, \dots, k)$ be the resulting path on V_i, and denote its ending vertices by a_i and b_i.

(b) Apply Algorithm RPP with special edges (a_i, b_i) $(i = 1, \dots, k)$.

(c) Let T_h be a Hamilton cycle obtained by replacing the special edge (a_i, b_i) by P_i, for each $i = 1, \dots, k$.

Step 2: In each cluster V_i, find vertices s_i and t_i that maximize $w(s_i, t_i)$. Apply Algorithm 2 with ending vertices s_i and t_i, and let T_l be the obtained Hamilton cycle.

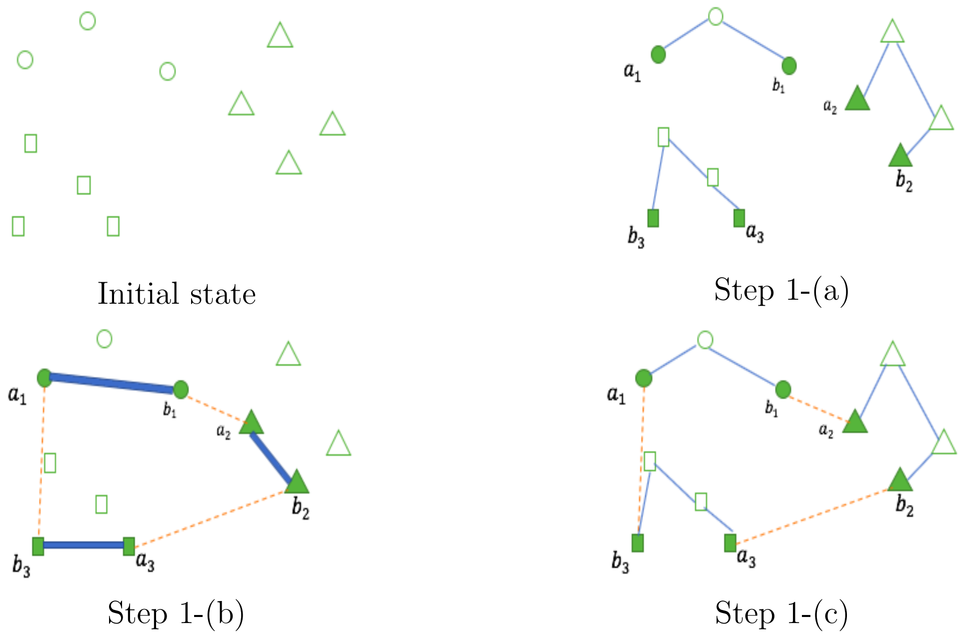
Step 3: Return the shorter of the Hamilton cycles T_h and T_l.

Figure 3 illustrates an example of the first algorithm. Guttmann-Beck et al. [6] proved that Algorithm 3 yields 2.75-approximation.

Theorem 3.3 ([6])**.** *The approximation ratio of Algorithm 3 is* 2.75.

4. Improving the Approximation Ratios for CTSP

In this section, we improve the approximation algorithm for CTSP [6] by incorporating the approximation algorithm for TSPP by Zenklusen [11]. Recall that Zenklusen [11] devised a new algorithm for the Variant 1 of TSPP with an improved approximation ratio 1.5, while the previous approximation algorithms for CTSP [6] apply 5/3-approximation algorithm for this variant of TSPP [7].



Initial state

Step 1-(a)

Step 1-(b)

Step 1-(c)

Figure 3: Illustration of Step 1 in Algorithm 3

4.1. Start and end vertices are given

In this subsection, we analyze the approximation ratio of the algorithm obtained by incorporating the algorithm for TSPP by Zenklusen [11] in Algorithm 1 (Section 3.1). We name the algorithm Algorithm A.

Theorem 4.1. *Let T_m be the Hamilton cycle returned by Algorithm A. Then,*

$$w(T_m) \leq \frac{15}{8}\mathrm{OPT}.$$

Proof. The algorithm consists of solving two subproblems of TSPP with given ending vertices and SCP. We introduce some notation to analyze the algorithm. Let W be the sum of the weights of the edges of OPT within each cluster V_i. Let A be the sum of the weight of edges of OPT that connect vertices of two different clusters. By definition, we have OPT $= A + W$. Let U be the total weight of arcs (s_i, t_i) for $i = 1, \ldots, k$. Let $w(P)$ be the total weights of P_i for $i = 1, \ldots, k$. It follows from Lemma 2.1 that

$$\begin{aligned} w(P) &\leq \min\left\{2W - U, \frac{3}{2}W\right\} \\ &\leq \frac{3}{4}(2W - U) + \frac{1}{4} \cdot \frac{3}{2}W \\ &= \frac{15}{8}W - \frac{3}{4}U. \end{aligned} \tag{1}$$

Note that the set consisting of edges of OPT connecting two different clusters and arcs (s_i, t_i) for $i = 1, \ldots, k$ is a solution to SCP of weight $A + U$. Thus, by Theorem 2.2, the two solutions S_1 and S_2 obtained by Algorithm SCP in Step 2 of Algorithm A satisfy $w(S_1) \leq 3A/2 + 2U$ and $w(S_2) \leq 3A + U$. Let T_s be the shorter of the two solutions. It then holds that

$$\begin{aligned} w(T_s) &\le \min\left\{\frac{3}{2}A + 2U, 3A + U\right\} \\ &\le \frac{3}{4}\left(\frac{3}{2}A + 2U\right) + \frac{1}{4}(3A + U) \\ &= \frac{15}{8}A + \frac{7}{4}U. \end{aligned} \tag{2}$$

In Step 3 of Algorithm A, the two solutions are combined by replacing arcs of weight U in the SCP solution by the TSPP solution. We obtain an upper bound on the weight of the solution T_m by combining (1) and (2):

$$\begin{aligned} w(T_m) &= w(P) - U + w(T_s) \\ &\le \left(\frac{15}{8}W - \frac{3}{4}U\right) - U + \left(\frac{15}{8}A + \frac{7}{4}U\right) \\ &= \frac{15}{8}(W + A) = \frac{15}{8}\mathrm{OPT}. \end{aligned}$$

□

Remark 1. If we use an algorithm for TSPP with approximation ratio α, it follows from the same analysis that the approximation ratio of Algorithm A becames $(12 - 3\alpha)/(7 - 2\alpha)$, provided $\alpha \le 3$.

4.2. Two end vertices are given

In this subsection, we analyze the approximation ratio of the algorithm obtained by incorporating the algorithm for TSPP by Zenklusen [11] in Algorithm 2 (Section 3.2). We name the algorithm Algorithm B.

Theorem 4.2. *Let T_m be the Hamilton cycle returned by Algorithm B. Then,*

$$w(T_m) \le \frac{12}{7}\mathrm{OPT}.$$

Proof. The proof is similar to that of Theorem 4.1. In Step 1 of Algorithm B, it follows from Lemma 2.1 that

$$\begin{aligned} w(P) &\le \min\left\{2W - U, \frac{3}{2}W\right\} \\ &\le \frac{3}{7}(2W - U) + \frac{4}{7}\cdot\frac{3}{2}W \\ &= \frac{12}{7}W - \frac{3}{7}U. \end{aligned} \tag{3}$$

Note that the set consisting of edges of OPT connecting two different clusters and special edges (s_i^1, s_i^2) for $i = 1, \ldots, k$ is a solution to RPP of weight $A + U$. By Theorem 2.3, the two solutions R_1 and R_2 obtained by Algorithm RPP in Step 2 of Algorithm B satisfy

$w(R_1) \leq 3(A+U)/2$ and $w(R_2) \leq 3A+U$. Let T_r be the shorter of the two solutions. It then holds that

$$\begin{aligned} w(T_r) &\leq \min\left\{\frac{3}{2}(A+U), 3A+U\right\} \\ &\leq \frac{6}{7}\left(\frac{3}{2}(A+U)\right) + \frac{1}{7}(3A+U) \\ &= \frac{12}{7}A + \frac{10}{7}U. \end{aligned} \tag{4}$$

In Step 3 of Algorithm B, the two solutions are combined by replacing edges of weight U in the RPP solution by the TSPP solution. We obtain an upper bound on the weight of the solution T_m by combining (3) and (4):

$$\begin{aligned} w(T_m) &= w(P) - U + w(T_r) \\ &\leq \left(\frac{12}{7}W - \frac{3}{7}U\right) - U + \left(\frac{12}{7}A + \frac{10}{7}U\right) \\ &= \frac{12}{7}(W+A) = \frac{12}{7}\mathrm{OPT}. \end{aligned}$$

□

Remark 2. If we use an algorithm for TSPP with approximation ratio α, it follows from the same analysis that the approximation ratio of Algorithm B becames $6/(5-\alpha)$, provided $\alpha \leq 3$.

4.3. End vertices are not specified

We analyze the approximation ratio of the algorithm obtained by incorporating the algorithm for TSPP by Zenklusen [11] in Algorithm 3 (Section 3.4). We name the algorithm Algorithm C.

Lemma 4.1 ([6])**.** *For the Hamilton cycle T_h computed in Step 1 of Algorithm C, it holds that*

$$w(T_h) \leq \frac{3}{2}\mathrm{OPT} + \frac{1}{2}W + 2U.$$

Lemma 4.2. *For the Hamilton cycle T_l computed in Step 2 of Algorithm C, it holds that*

$$w(T_l) \leq \begin{cases} \frac{3}{2}\mathrm{OPT} + 2W - 2U & \left(U > \frac{W}{2}\right), \\ \frac{3}{2}\mathrm{OPT} + \frac{3}{2}W - U & \left(U \leq \frac{W}{2}\right). \end{cases}$$

Proof. Here, we can just take the RPP solution computed by Algorithm SmallEdges to obtain the desired result. By Theorem 2.3, the RPP solution has weight $3(A+U)/2 \leq 3\mathrm{OPT}/2$. Then, we replace each special edge (s_i, t_i) by a path connecting s_i and t_i that includes all vertices in V_i. By Lemma 2.1, the weights of such paths is at most $2W-U$ or $3W/2$. Hence, we obtain

$$w(T_l) \leq \begin{cases} \frac{3}{2}\mathrm{OPT} - U + (2W-U) = \frac{3}{2}\mathrm{OPT} + 2W - 2U & \left(U > \frac{W}{2}\right), \\ \frac{3}{2}\mathrm{OPT} - U + \frac{3}{2}W = \frac{3}{2}\mathrm{OPT} + \frac{3}{2}W - U & \left(U \leq \frac{W}{2}\right). \end{cases}$$

□

Theorem 4.3. *Let T_m be the Hamilton cycle returned by Algorithm C. Then,*

$$w(T_m) \leq \frac{8}{3}\mathrm{OPT}.$$

Proof. If $U \leq W/3$, then, by Lemma 4.1 and the inequality $W \leq \mathrm{OPT}$, it holds that

$$\begin{aligned}
w(T_h) &\leq \frac{3}{2}\mathrm{OPT} + \frac{1}{2}W + 2U \\
&\leq \frac{3}{2}\mathrm{OPT} + \frac{1}{2}W + \frac{2}{3}W \\
&\leq \frac{3}{2}\mathrm{OPT} + \frac{7}{6}\mathrm{OPT} \\
&= \frac{8}{3}\mathrm{OPT}.
\end{aligned}$$

If $W/3 < U \leq W/2$, then, by Lemma 4.2 and the inequality $W \leq \mathrm{OPT}$, it holds that

$$\begin{aligned}
w(T_l) &\leq \frac{3}{2}\mathrm{OPT} + \frac{3}{2}W - U \\
&\leq \frac{3}{2}\mathrm{OPT} + \frac{3}{2}W - \frac{1}{3}W \\
&\leq \frac{3}{2}\mathrm{OPT} + \frac{7}{6}W \\
&\leq \frac{3}{2}\mathrm{OPT} + \frac{7}{6}\mathrm{OPT} \\
&= \frac{8}{3}\mathrm{OPT}.
\end{aligned}$$

If $U > W/2$, then, by Lemma 4.2 and the inequality $W \leq \mathrm{OPT}$, it holds that

$$\begin{aligned}
w(T_l) &\leq \frac{3}{2}\mathrm{OPT} + 2W - 2U \\
&\leq \frac{3}{2}\mathrm{OPT} + 2W - W \\
&\leq \frac{3}{2}\mathrm{OPT} + W \\
&\leq \frac{3}{2}\mathrm{OPT} + \mathrm{OPT} \\
&= \frac{5}{2}\mathrm{OPT}.
\end{aligned}$$

□

Remark 3. For this case, an approximation ratio obtained from an α-approximation algorithm for TSPP does not immediately follow from the same analysis.

5. Conclusions

In this paper, we have improved approximation ratios for Variants (1), (2), and (4) of CTSP [6] by applying a recent improved approximation algorithm for TSPP by Zenklusen [11], as shown in Tables 1 and 2. There is a similar problem, the subpath planning problem (SPP) and the subgroup planning problem (SGPP) [9]. Sumita et al. [10] considered approximation algorithms for SPP and SGPP. They proposed 1.5-approximation algorithm for SPP and 3-approximation algorithm for SGPP. Applying the approximation algorithm for Variant (4) of CTSP to SGPP, it might be possible to improve the approximation ratio for SGPP.

Acknowledgements

We appreciate the reviewers for helpful comments. The second author is partially supported by JSPS KAKENHI Grant Number JP16K16012, Japan.

References

[1] X. Bao, Z. Liu, W. Yu, and G. Li: A note on approximation algorithms of the clustered traveling salesman problem. *Information Processing Letters*, **127** (2017), 54–57.

[2] N. Christofides: Worst-case analysis of a new heuristic for the travelling salesman problem. *Report 388, Graduate School of Industrial Administration, Carnegie-Mellon University, Pittsburgh*, PA (1976).

[3] H. A. Eiselt, M. Gendreau, and G. Laporte: Arc routing problems, part II: The rural postman problem. *Operations Research*, **43** (1995), 399–414.

[4] G. N. Frederickson: Approximation algorithms for some postman problems. *Journal of the Association for Computing Machinery*, **26** (1979), 538–554.

[5] G. N. Frederickson, M. S. Hecht, and C. E. Kim: Approximation algorithms for some routing problems. *SIAM Journal on Computing*, **7** (1978), 178–193.

[6] N. Guttmann-Beck, R. Hassin, S. Khuller, and B. Raghavachari: Approximation algorithms with bounded performance guarantees for the clustered traveling salesman problem. *Algorithmica*, **28** (2000), 422–437.

[7] A. Hoogeveen: Analysis of Christofides' heuristic: Some paths are more difficult than cycles. *Operations Research Letters*, **10** (1991), 291–295.

[8] K. Jansen: An approximation algorithm for the general routing problem. *Information Processing Letters*, **41** (1992), 333–339.

[9] M. Safilian, S. M. Hashemi, S. Eghbali, and A. Safilian: An approximation algorithm for the subpath planning problem. *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence* (2016), 669–675.

[10] H. Sumita, Y. Yonebayashi, N. Kakimura, and K. Kawarabayashi: An improved approximation algorithm for the subpath planning problem. *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence* (2017), 4412–4418.

[11] R. Zenklusen: A 1.5-approximation for path TSP. *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms* (2019), 1539–1549.

Kenjiro Takazawa
Hosei University
3-7-2, Kajino-cho, Koganei-shi
Tokyo 184-8584, Japan
E-mail: takazawa@hosei.ac.jp