

# An Improved Approximation Algorithm for the Capacitated Arc Routing Problem

Jingyang Zhao

University of Electronic Science and Technology of China  
jingyangzhao1020@gmail.com

Mingyu Xiao\*

University of Electronic Science and Technology of China  
myxiao@uestc.edu.cn

## Abstract

The Capacitated Arc Routing Problem (CARP), introduced by Golden and Wong in 1981, is an important arc routing problem in Operations Research, which generalizes the famous Capacitated Vehicle Routing Problem (CVRP). When every customer has a unit demand, the best known approximation ratio for CARP, given by Jansen in 1993, remains  $\frac{5}{2} - \frac{1.5}{k}$ , where  $k$  denotes the vehicle capacity. Based on recent progress in approximating CVRP, we improve this result by proposing a  $(\frac{5}{2} - \Theta(\frac{1}{\sqrt{k}}))$ -approximation algorithm, which to the best of our knowledge constitutes the first improvement over Jansen’s bound.

## 1 Introduction

The CAPACITATED ARC ROUTING PROBLEM (CARP), introduced by Golden and Wong (1981), is one of the most famous arc routing problems (van Bevern et al., 2014b). In this problem, we are given an undirected graph  $G = (V \cup \{v_0\}, E)$ , where each edge  $e \in E$  has an associated nonnegative cost  $c(e) \in \mathbb{R}_{\geq 0}$  and a nonnegative integer demand  $d(e) \in \mathbb{Z}_{\geq 0}$ . Edges with positive demand are referred to as *customers*. A fleet of identical vehicles, each with capacity  $k \in \mathbb{Z}_{\geq 1}$ , is initially located at a designated depot vertex  $v_0 \in V$ . The objective is to compute a set of routes (walks), each starting and ending at the depot, such that all customers are served, the total demand served by each vehicle does not exceed its capacity, and the total cost of the routes is minimized. It is typically assumed that each customer is served by exactly one vehicle (Jansen, 1993; Wøhlk, 2008b). When each edge’s demand is either 0 or 1, the problem is referred to as the *equal-demand CARP*; otherwise, it is known as the *general CARP*.

CARP generalizes many famous routing problems, e.g., the CHINESE POSTMAN PROBLEM (CPP) (Eiselt et al., 1995a), the RURAL POSTMAN PROBLEM (RPP) (Eiselt et al., 1995b), the

---

\*Corresponding author

TRAVELING SALESMAN PROBLEM (TSP) (Christofides, 2022), and so on. Moreover, if the demands are defined for the vertices instead of the edges, CARP reduces to the CAPACITATED VEHICLE ROUTING PROBLEM (CVRP) (Dantzig and Ramser, 1959), which is a representative problem in the area of node routing. Since CARP has various application in road networks, e.g., snow plowing (Perrier et al., 2007), waste collection (Fernández et al., 2016), newspaper delivery (Corberán and Laporte, 2015), and so on, it has been extensively studied in the areas of Operations Research and Computer Science (Corberán and Prins, 2010; Mourão and Pinto, 2017). A recent survey could be found in (Corberán et al., 2021).

We focus on approximation algorithms for CARP. For any minimization problem, an algorithm is called a  $\rho$ -approximation algorithm if it computes a solution with an objective value not exceeding  $\rho$  times the optimal value in polynomial time, where  $\rho \geq 1$  is called the *approximation ratio*.

## 1.1 Related work

It is well known that CARP can be solved in polynomial time when  $k = 1$  or  $k = 2$ . However, since CARP generalizes CVRP, it is APX-hard for any  $k \geq 3$  (Asano et al., 1996). Moreover, as general CARP also generalizes the BIN PACKING PROBLEM (Jansen, 1993), it is NP-hard to approximate within a factor better than  $\frac{3}{2}$ . When  $k = \infty$ , CARP reduces to the RPP, which in turn generalizes the TSP. Hence, both RPP and TSP remain APX-hard (Karpinski et al., 2015). However, if the subgraph induced by the edges with positive demand is connected, then RPP reduces to the CPP, which is solvable in polynomial time (Eiselt et al., 1995a).

CARP and its related problems has been extensively studied in both theory (Jansen, 1993; Wøhlk, 2008a; van Bevern et al., 2014a; Van Bevern et al., 2017; van Bevern et al., 2020) and practice (Brandão and Eglese, 2008; Santos et al., 2009; Mourão et al., 2009; Wøhlk and Laporte, 2018). For heuristic algorithms, a wide range of methods have been developed, including combinatorial methods (Wøhlk, 2008a), memetic search (Tang et al., 2009), tabu search (Brandão and Eglese, 2008), metaheuristic (Chen et al., 2016), simulated annealing (Babaei Tirkolaei et al., 2016), integer programming (Belenguer and Benavent, 2003), and so on. More results can be found in (Corberán and Prins, 2010; Mourão and Pinto, 2017; Corberán et al., 2021). For approximation algorithms, we first review the results on CVRP.

For the equal-demand CVRP, where each customer has unit demand, Haimovich and Kan (1985) proposed an  $(\alpha + 1 - \frac{\alpha}{k})$ -approximation algorithm under the assumption that the number of customers is divisible by  $k$ . Here,  $\alpha$  denotes the approximation ratio of the metric TSP, for which it is known that  $\alpha = \frac{3}{2}$  (Christofides, 2022) and was later slightly improved to  $\alpha = \frac{3}{2} - 10^{-36}$  (Karlin et al., 2021, 2023). Subsequently, Altinkemer and Gavish (1990) removed the divisibility assumption while maintaining the same approximation ratio of  $\alpha + 1 - \frac{\alpha}{k}$ . For the general CVRP, Altinkemer and Gavish (1987) proposed an  $(\alpha + 2 - \frac{2\alpha}{k})$ -approximation algorithm when  $k$  is even. Note that equal-demand (resp., general) CVRP is also referred to as the *unit-demand* (resp., *unsplittable*) CVRP. Another common variant is the *splittable* CVRP, where a customer may be served by multi-vehicles. However, as noted in the survey by Corberán et al. (2021), splittable service is less

common in arc routing problems.

Bompadre et al. (2006) improved these results by an additive term of  $\Omega(\frac{1}{k^3})$ . More recently, Blauth et al. (2023) achieved an improved approximation ratio of  $\alpha + 1 - \varepsilon$  for the equal-demand case and  $\alpha + 2 - 2\varepsilon$  for the general case, where  $\varepsilon > \frac{1}{3000}$  when  $\alpha = \frac{3}{2}$ . As a further improvement for the general case, Friggstad et al. (2022) achieved an approximation ratio of  $\alpha + 1 + \ln 2 - \varepsilon$  for some positive constant  $\varepsilon$ . In addition, Zhao and Xiao (2025a) studied the case where the vehicle capacity  $k$  is a fixed integer, and proposed a  $(\frac{5}{2} - \Theta(\frac{1}{\sqrt{k}}))$ -approximation algorithm for the equal-demand case, and a  $(\frac{5}{2} + \ln 2 - \Theta(\frac{1}{\sqrt{k}}))$ -approximation algorithm for the general case.

Approximation algorithms on multi-depot CVRP could be found in (Li and Simchi-Levi, 1990; Harks et al., 2013; Zhao and Xiao, 2025b).

For the equal-demand CARP, Jansen (1993) proposed a  $(\beta + 1 - \frac{\beta}{k})$ -approximation algorithm on metric graphs, where  $\beta = \frac{3}{2}$  is the approximation ratio of the RPP (Eiselt et al., 1995b). For the general CARP, Jansen (1993) gave a  $(\beta + \frac{k}{k_s} - \frac{\beta}{k_s})$ -approximation algorithm on metric graphs, where  $k_s = \lceil \frac{k}{2} \rceil$ . This yields an approximation ratio of  $\beta + 2 - \frac{2\beta}{k}$  when  $k$  is even, and  $\beta + 2 - \frac{2\beta}{k+1}$  when  $k$  is odd. Wøhlk (2008b) also obtained the same approximation ratios on metric graphs using a dynamic programming approach. Notably, there is no approximation gap between metric and non-metric graphs for CARP, since van Bevern et al. (2014a) showed that any  $\rho$ -approximation algorithm for CARP on metric graphs also yields a  $\rho$ -approximation algorithm on non-metric graphs.

It is worth noting that when  $k = O(1)$ , CVRP can be reduced to the MINIMUM WEIGHT  $k$ -SET COVER PROBLEM (Chvátal, 1979; Hassin and Levin, 2005; Gupta et al., 2023) in polynomial time  $n^{O(k)}$ , while preserving the approximation ratio (see (Zhao and Xiao, 2025a) for details). This reduction also applies to CARP. Therefore, based on the results of Gupta et al. (2023), both CVRP and CARP admit an approximation ratio of  $\min\{H_k - \frac{1}{8k}, H_k - \sum_{i=1}^k \frac{\log i}{8ki}\}$ , where  $H_k = \sum_{i=1}^k \frac{1}{i}$  denotes the  $k$ -th harmonic number.

Approximation algorithms on multi-depot CARP could be found in (Yu et al., 2023).

We observe that, in the early development of approximation algorithms, there was little to no gap between the approximation ratios for CVRP and CARP. However, as approximation algorithms for CVRP have advanced, the gap has grown significantly. In particular, the metric TSP now admits an approximation ratio of  $\alpha = \frac{3}{2} - 10^{-36}$  (Karlin et al., 2021, 2023), while the best known approximation ratio for the RPP remains  $\beta = \frac{3}{2}$  (Eiselt et al., 1995b). Moreover, the best known approximation ratios for CARP are still those given by Jansen (1993). This raises a natural question: can the techniques developed for CVRP be extended to CARP to improve upon the existing results?

## 1.2 Our results

In this paper, we focus on approximation algorithms for equal-demand CARP. According to the result of van Bevern et al. (2014a), it suffices to consider metric graphs. By extending the methods of analyzing the lower bounds of an optimal solution in (Zhao and Xiao, 2025a), we propose a  $(\frac{5}{2} - \Theta(\frac{1}{\sqrt{k}}))$ -approximation algorithm, which constitutes the first improvement over the previous

ratio of  $\frac{5}{2} - \frac{1.5}{k}$  by Jansen (1993). A summary of previous and our results for equal-demand CARP can be found in Table 1.

$k$	3	4	5	6	7	8	...
Previous Ratio	<b>1.792</b>	2.051	2.200	2.250	2.286	2.313	...
	(Gupta et al., 2023)	(Jansen, 1993)	(Jansen, 1993)	(Jansen, 1993)	(Jansen, 1993)	(Jansen, 1993)	
Our Ratio	1.889	<b>2.000</b>	<b>2.086</b>	<b>2.143</b>	<b>2.184</b>	<b>2.215</b>	...

Table 1: Previous and our approximation ratios for equal-demand CARP.

Similar to the previous algorithm, our algorithm first computes an RPP tour, which is a solution to RPP, and then obtain a solution to CARP by partitioning the RPP tour. The key difference is that we use two different RPP tours. The first is computed using the  $\frac{3}{2}$ -approximation algorithm for RPP (Eiselt et al., 1995b). The second is obtained in a different way: we begin by finding a minimum-cost even-degree multi-graph that includes all edges with positive demand, using a minimum-cost perfect matching algorithm (Schrijver, 2003); then, we connect the resulting components by greedily adding minimum-cost edges between them, similar to Kruskal’s algorithm (Schrijver, 2003); last, we form an Eulerian graph by doubling the added edges, and then obtain an RPP tour by shortcutting the Eulerian graph.

## 2 Preliminary

We consider CARP on metric graphs. The input graph, denoted by  $G = (V \cup \{v_0\}, E)$ , is an undirected complete graph, where  $v_0$  is the depot and the edge cost function  $c : E \rightarrow \mathbb{R}_{\geq 0}$  satisfies  $c(a, a) = 0$ ,  $c(a, b) = c(b, a)$ , and the triangle inequality  $c(a, h) \leq c(a, b) + c(b, h)$  for all  $a, b, h \in V$ . Let  $E^* \subseteq E$  denote the set of edges (customers) with positive demand. By making copies of vertices that are shared among multiple edges in  $E^*$ , we assume w.l.o.g. that the edges in  $E^*$  are vertex-disjoint. Moreover, by the triangle inequality, we can move any vertex in  $V$  that is not incident to an edge in  $E^*$ . Thus, we assume that each vertex in  $V$  is incident to exactly one edge in  $E^*$ , which implies that  $|E^*| = \frac{|V|}{2}$ . We let  $|V| = n$ , and for any positive integer  $t$ , we define  $[t] = \{1, 2, \dots, t\}$ . We assume that there is an unlimited number of vehicles at the depot, each with capacity  $k \in \mathbb{Z}_{\geq 1}$ .

For any subset  $E' \subseteq E^*$ , define  $\delta(E') = \sum_{(v_i, v_j) \in E'} \frac{c(v_0, v_i) + c(v_i, v_j) + c(v_0, v_j)}{2}$ . For any multi-graph subgraph  $S$  of  $G$ , let  $V(S)$  (resp.,  $E(S)$ ) denote the set of its vertices (resp., the (multi-)set of its edges). We define  $c(S) = \sum_{e \in E(S)} c(e)$ . Throughout the paper, we work with multi-edge sets, and the union of any two edge sets is taken with multiplicities.

A *walk*  $W$  in a graph, denoted by  $(v_1, v_2, \dots, v_l)$ , is a sequence of vertices where each consecutive pair  $(v_i, v_{i+1})$  is connected by an edge, and vertices may appear multiple times. We use  $E(W)$  (resp.,  $V(W)$ ) to denote the multi-set of edges  $\{(v_1, v_2), \dots, (v_{l-1}, v_l)\}$  (resp., the set of vertices  $\{v_1, \dots, v_l\}$ ) and define the cost of the walk as  $c(W) = \sum_{e \in E(W)} c(e)$ . A *closed walk* is a walk in which the first and last vertices are the same, while a *cycle* is a closed walk in which no other vertex is repeated. Given a closed walk  $W$ , we can obtain a new walk  $W'$  by skipping some vertices or edges along  $W$ ; this process is called *shortcutting*. By the triangle inequality, shortcutting does not increase the

cost, i.e.,  $c(W') \leq c(W)$ . A *tour* or *route* is a closed walk that starts and ends at the depot  $v_0$ , and an *RPP tour* is a tour that traverses every edge in  $E^*$  at least once. By the triangle inequality, we may assume that any RPP tour consists of exactly  $|V|+1$  distinct edges, i.e., it forms a *Hamiltonian cycle* in  $G$ . Let  $H^*$  denote an optimal RPP tour in  $G$ , and then it is a minimum-cost Hamiltonian cycle in  $G$  such that  $E^* \subseteq E(H^*)$ .

A solution to the CARP is a set of tours, where each tour corresponds to the route of a single vehicle and serves a set of edges (customers) with total demand at most  $k$  in  $E^*$ . By the triangle inequality, we may assume that each vehicle route forms a cycle and visits only the vertices of its served edges along with the depot. Then, each route  $T$  can be represented as  $(v_0, v_1, v_2, \dots, v_{2l-1}, v_{2l}, v_0)$ , where it serves the set of customers  $\{(v_{2i-1}, v_{2i}) \mid i \in [l]\}$ , and  $\sum_{i \in [l]} d(v_{2i-1}, v_{2i}) \leq k$ . Hence, we define the set of customers served by route  $T$  as  $E_T^* = E^* \cap E(T)$  and also let  $V_T = \{v_0, v_1, \dots, v_{2l}\}$ .

CARP is formally defined as follows.

**Definition 1** (CARP). *Given an undirected complete graph  $G = (V \cup \{v_0\}, E)$ , a set of vertex-disjoint edges  $E^* \subseteq E$ , where each edge  $e \in E^*$  has demand  $d(e) \in \mathbb{Z}_{\geq 1}$ , and a vehicle capacity  $k \in \mathbb{Z}_{\geq 1}$ , the objective is to find a set of tours  $\mathcal{T}$  in  $G$  such that*

- (1) *each tour  $T \in \mathcal{T}$  servers all customers in  $E_T^* = E^* \cap E(T)$  with total demand at most  $k$ , i.e.,  $\sum_{e \in E_T^*} d(e) \leq k$ ;*
- (2) *all tours together serve all customers, i.e.,  $\bigcup_{T \in \mathcal{T}} E_T^* = E^*$ ;*
- (3) *each customer is served by exactly one tour in  $\mathcal{T}$ , i.e.,  $E_T^* \cap E_{T'}^* = \emptyset$  for all  $T, T' \in \mathcal{T}$ ;*
- (4) *the total cost, i.e.,  $\sum_{T \in \mathcal{T}} c(T)$ , is minimized.*

By our assumptions, we also have

- (5) *the tours in  $\mathcal{T}^*$  are pairwise edge-disjoint, and any two tours share only the common vertex  $v_0$ , i.e.,  $E(T) \cap E(T') = \emptyset$  and  $V_T \cap V_{T'} = \{v_0\}$  for all  $T, T' \in \mathcal{T}$ .*

We consider unite-demand CARP, where we have  $d(e) = 1$  for all  $e \in E^*$ .

### 3 The Algorithms

We first review the previous approximation algorithm given by Jansen (1993).

#### 3.1 The previous tour partition algorithm

For equal-demand CVRP, a well-known algorithm is the *Iterated Tour Partitioning* (ITP) algorithm (Haimovich and Kan, 1985; Altinkemer and Gavish, 1990). ITP first computes an approximate Hamiltonian cycle and then obtains a solution to equal-demand CVRP by appropriately partitioning the cycle into fragments (paths), each of which is transformed into a tour.

For equal-demand CARP, Jansen (1993) extended the ITP framework and proposed a similar tour partitioning algorithm, referred to as JITP. In the first step of JITP, instead of computing an arbitrary Hamiltonian cycle, the algorithm constructs an RPP tour that traverses all edges in  $E^*$ . Then, analogous to ITP, JITP partitions the RPP tour into a set of fragments, and each fragment is transformed into a tour by connecting its two endpoints to the depot. JITP may explore multiple partitioning strategies and select the one that yields the minimum-cost CARP solution.

Specifically, let  $m = \frac{n}{2}$ , and suppose an RPP tour in  $G$  is given by  $(v_0, v_1, v_2, \dots, v_{2m-1}, v_{2m}, v_0)$ , where  $(v_{2i-1}, v_{2i}) \in E^*$  for each  $i \in [m]$ . For convince, let  $x_i = (v_{2i-1}, v_{2i})$  for each  $i \in [m]$ , and use  $(x_i, \dots, x_{i+i'})$  to represent the tour  $(v_0, v_{2i-1}, v_{2i}, \dots, v_{2i+2i'-1}, v_{2i+2i'}, v_0)$ , which is obtained by connecting the endpoints of the fragment  $(v_{2i-1}, v_{2i}, \dots, v_{2i+2i'-1}, v_{2i+2i'})$  to the depot. Then, JITP returns the best solution among  $k$  potential ones.

In the  $i$ -th potential solution, denoted by  $\mathcal{T}_i$ , we have  $\mathcal{T}_i = \{T_i^1, \dots, T_i^N\}$ , where  $N = \lceil \frac{n-i}{k} \rceil + 1$ ,  $T_i^1 = (x_1, \dots, x_i)$ ,  $T_i^j = (x_{i+(j-2)k+1}, \dots, x_{i+(j-1)k})$  for each  $2 \leq j < N$ , and  $T_i^N = (x_{i+(N-2)k+1}, \dots, x_n)$ .

Therefore, in the solution of JITP, except possibly for the first and the last tours, each tour serves exactly  $k$  customers. We remark that Wøhlk (2008b) proved that the optimal partitioning can be computed in polynomial time using a dynamic programming approach.

We have the following result.

**Lemma 1** ((Jansen, 1993; Wøhlk, 2008a; van Bevern et al., 2014a)). *For equal-demand CARP, given an RPP tour  $H$  in  $G$  with  $E^* \subseteq E(H)$ , there is a polynomial-time algorithm that computes a solution with cost at most  $\frac{2}{k}\delta(E^*) + \frac{k-1}{k}c(H)$ .*

### 3.2 Our algorithm

In our algorithm, we use JITP as a subroutine. We first construct two RPP tours, and then use the one with smaller cost to call the JITP algorithm. The first RPP tour, denoted by  $H_1$ , is obtained directly by using the  $\frac{3}{2}$ -approximation algorithm (Eiselt et al., 1995b). The second one, denoted by  $H_2$ , is obtained in the following way.

First, we find a minimum-cost perfect matching  $M$  in the graph  $G[V]$  (Schrijver, 2003). Then, in the multi-graph  $G' = (V \cup \{v_0\}, E(M) \cup E^*)$ , every vertex has an even degree. Moreover, the graph consists of a set of components. Then, similar to Kruskal's algorithm (Schrijver, 2003), we connect these components by greedily adding minimum-cost edges between them. Let  $F$  be the set of added edges. Then, adding all edges in  $F$  to  $G'$  results in a connected graph. Then, the multi-graph  $G'' = (V \cup \{v_0\}, E(M) \cup E^* \cup F \cup F)$  forms an *Eulerian graph*, i.e., a connected graph and every vertex in it has an even degree. Thus, a tour  $T$  such that  $E(T) = E(G'')$ , also known as an *Eulerian tour*, can be obtained in polynomial time (Schrijver, 2003). Then, by further shortcutting the tour  $W$ , an RPP tour is obtained.

We remark that the above algorithm uses the idea of the double-tree algorithm for TSP (Williamson and Shmoys, 2011).

An illustration of our algorithm for equal-demand CARP can be found in Algorithm 1.

---

**Algorithm 1** The approximation algorithm for equal-demand CARP

---

**Input:** An instance  $G = (V, E)$  of equal-demand CARP.

**Output:** A solution to equal-demand CARP.

- 1: Obtain an RPP tour  $H_1$  by using the  $\frac{3}{2}$ -approximation algorithm (Eiselt et al., 1995b).
  - 2: Find a minimum-cost perfect matching  $M$  in  $G[V]$  (Schrijver, 2003).
  - 3: Connect the components in  $G' = (V \cup \{v_0\}, E(M) \cup E^*)$  by greedily adding minimum-cost edges between them, and let  $F$  be the set of added edges.
  - 4: Obtain an RPP tour  $H_2$  by finding an Eulerian tour  $T$  in  $G'' = (V \cup \{v_0\}, E(M) \cup E^* \cup F \cup F)$  and then shortcutting  $T$ .
  - 5: Obtain a solution  $\mathcal{T}$  by using the RPP tour  $\arg \min\{c(H_1), c(H_2)\}$  to call the JITP algorithm.
  - 6: **return**  $\mathcal{T}$ .
- 

## 4 Performance Analysis

In this section, we use  $\mathcal{T}^*$  to denote an optimal solution to equal-demand CARP and define  $\text{OPT} = c(\mathcal{T}^*)$ . In the following, we first recall two lower bounds on  $\text{OPT}$  in (Jansen, 1993); then, we analyze the upper bounds on the cost of the used RPP tours in our algorithm and propose new lower bounds on  $\text{OPT}$ ; last, we analyze the approximation ratio of our algorithm by using these lower bounds on  $\text{OPT}$ .

Recall that  $H^*$  denotes an optimal RPP tour, and  $\delta(E') = \sum_{(v_i, v_j) \in E'} \frac{c(v_0, v_i) + c(v_i, v_j) + c(v_0, v_j)}{2}$  for any  $E' \subseteq E^*$ . There are two known lower bounds on  $\text{OPT}$ .

**Lemma 2** ((Jansen, 1993)). *It holds that  $\frac{2}{k}\delta(E^*) \leq \text{OPT}$  and  $c(H^*) \leq \text{OPT}$ .*

Next, we analyze the used RPP tours in our algorithm.

Let  $E'$  denote a minimum-cost edge set such that  $E' \cup E^*$  forms a spanning tree in  $G$ . Then, we define  $\text{MST} = c(E') + c(E^*)$ .

The first RPP tour  $H_1$  is obtained by using the  $\frac{3}{2}$ -approximation algorithm (Eiselt et al., 1995b). We have the following result.

**Lemma 3.** *It holds that  $c(H_1) \leq \text{MST} + \frac{1}{2}\text{OPT}$ .*

*Proof.* It is well known that  $c(H_1) \leq \text{MST} + \frac{1}{2}c(H^*)$  (Eiselt et al., 1995b). Since  $c(H^*) \leq \text{OPT}$  by Lemma 2, we obtain the desired result.  $\square$

**Lemma 4.** *It holds that  $c(H_2) \leq \text{OPT} + 2\text{MST} - 2c(E^*)$ .*

*Proof.* Recall that  $H_2$  is obtained by shortcutting the graph  $G'' = (V \cup \{v_0\}, E(M) \cup E^* \cup F \cup F)$ . By the triangle inequality, we have

$$c(H_2) \leq c(E^*) + c(M) + 2c(F). \quad (1)$$

By line 2, the graph  $G' = (V \cup \{v_0\}, E(M) \cup E^*)$  is the minimum-cost Eulerian graph containing all edges in  $E^*$ . By shortcutting the depot  $v_0$  from the optimal solution  $\mathcal{T}^*$ , we obtain an Eulerian

graph containing all edges in  $E^*$  with cost at most  $\text{OPT}$ . Thus, we have

$$c(E^*) + c(M) \leq \text{OPT}. \quad (2)$$

By line 3 and the proof of Kruskal's algorithm (Schrijver, 2003),  $F$  is the minimum-cost set of edges such that adding all edges in  $F$  to the graph  $G' = (V \cup \{v_0\}, E(M) \cup E^*)$  yields a connected graph. Recall that  $E'$  is a minimum-cost edge set such that  $E' \cup E^*$  forms a spanning tree in  $G$ . Thus, adding all edges in  $E'$  to  $G'$  yields a connected graph, and then we have

$$c(F) \leq c(E') = \text{MST} - c(E^*). \quad (3)$$

Therefore, by (1), (2), and (3), we have  $c(H_2) \leq \text{OPT} + 2\text{MST} - 2c(E^*)$ .  $\square$

For any tour  $T = (v_0, v_1, v_2, \dots, v_{2t-1}, v_{2t}, v_0)$  in the optimal solution  $\mathcal{T}^*$ , we have  $t \leq k$ . Recall that  $E_T^* = \{(v_{2i-1}, v_{2i}) \mid i \in [t]\}$  and  $V_T = \{v_0, v_1, \dots, v_{2t}\}$ . We also define  $\text{MST}_T$  be the cost of a minimum-cost spanning tree in the graph  $G[V_T]$  containing all edges in  $E_T^*$ .

We have the following property.

**Lemma 5.** *It holds that  $\delta(E^*) = \sum_{T \in \mathcal{T}^*} \delta(E_T^*)$  and  $\text{MST} \leq \sum_{T \in \mathcal{T}^*} \text{MST}_T$ .*

*Proof.* Since  $\mathcal{T}^*$  is a solution to equal-demand CARP, we have  $E^* = \bigcup_{T \in \mathcal{T}^*} E_T^*$  and  $E_T^* \cap E_{T'}^* = \emptyset$  for all  $T, T' \in \mathcal{T}$  by (2) and (3) in Definition 1. Thus, we have  $\delta(E^*) = \sum_{T \in \mathcal{T}^*} \delta(E_T^*)$ .

Moreover, by (5) in Definition 1, we have  $V = \bigcup_{T \in \mathcal{T}^*} V_T$  and  $V_T \cap V_{T'} = \{v_0\}$  for all  $T, T' \in \mathcal{T}^*$ . Since  $\text{MST}_T$  measures the cost of a spanning tree in  $G[V_T]$ , then  $\sum_{T \in \mathcal{T}^*} \text{MST}_T$  measures the cost of a spanning tree in  $G$ . By definition, any spanning tree in  $G$  containing all edges in  $E_T^*$  has cost at least  $\text{MST}$ . Thus, we have  $\text{MST} \leq \sum_{T \in \mathcal{T}^*} \text{MST}_T$ .  $\square$

By extending the techniques in (Zhao and Xiao, 2025a), we obtain the following key result.

**Lemma 6.** *For any tour  $T \in \mathcal{T}^*$ , there exist parameters  $\{\alpha_1, \alpha_2, \dots, \alpha_m\}$  such that*

$$(1) \quad \delta(E_T^*) \leq \left(\frac{k}{2} + \alpha - \sum_{i=1}^m i\alpha_i\right) c(T);$$

$$(2) \quad \text{MST}_T \leq \left(1 - \max_{1 \leq i \leq m} \frac{1}{2}\alpha_i\right) c(T);$$

$$(3) \quad c(E_T^*) = (1 - \alpha)c(T);$$

$$(4) \quad \sum_{i=1}^m \alpha_i = \alpha \leq 1, \text{ where } m = \infty \text{ and } 0 \leq \alpha_i \leq \alpha \text{ for each } 1 \leq i \leq m.$$

*Proof.* To avoid distraction from our main discussions, we delay the proof to Section 5.  $\square$

Next, we are ready to analyze the approximation ratio of our algorithm.

**Lemma 7.** *The approximation ratio of our algorithm is at most*

$$\max_{\substack{l \in \mathbb{Z}_{\geq 1} \\ 1 \geq \alpha \geq 0}} \min \{\tau(\alpha, l), \eta(\alpha, l)\},$$



where  $\tau(\alpha, l) = \frac{5k-3}{2k} - \frac{(2l^2-2l+k-1)\alpha}{2kl}$  and  $\eta(\alpha, l) = \frac{2k-1}{k} - \frac{(l^2+l-2kl+k-1)\alpha}{kl}$ .

*Proof.* Recall that our algorithm uses the RPP tour  $\arg \min\{c(H_1), c(H_2)\}$  to call the JITP algorithm. Then, by Lemma 1, the obtained solution, denoted by  $\mathcal{T}$ , satisfies that

$$\begin{aligned} c(\mathcal{T}) &\leq \frac{2}{k}\delta(E^*) + \frac{k-1}{k} \min\{c(H_1), c(H_2)\} \\ &\leq \frac{2}{k}\delta(E^*) + \frac{k-1}{k} \min\left\{\text{MST} + \frac{1}{2}\text{OPT}, \text{OPT} + 2\text{MST} - 2c(E^*)\right\} \\ &\leq \sum_{T \in \mathcal{T}^*} \left( \frac{2}{k}\delta(E_T^*) + \frac{k-1}{k} \min\left\{\text{MST}_T + \frac{1}{2}c(T), c(T) + 2\text{MST}_T - 2c(E_T^*)\right\} \right), \end{aligned} \quad (4)$$

where the second inequality follows from Lemmas 3 and 4, and the last inequality from Lemma 5 and the fact that  $\text{OPT} = \sum_{T \in \mathcal{T}^*} c(T)$ .

By Lemma 6, we have

$$\begin{aligned} &\frac{2}{k}\delta(E_T^*) + \frac{k-1}{k} \left( \text{MST}_T + \frac{1}{2}c(T) \right) \\ &\leq \frac{2}{k} \left( \frac{k}{2} + \alpha - \sum_{i=1}^m i\alpha_i \right) c(T) + \frac{k-1}{k} \left( 1 - \max_{1 \leq i \leq m} \frac{1}{2}\alpha_i + \frac{1}{2} \right) c(T) \\ &= \frac{5k-3}{2k}c(T) + \frac{2}{k} \left( \alpha - \sum_{i=1}^m i\alpha_i \right) c(T) - \left( \max_{1 \leq i \leq m} \frac{k-1}{2k}\alpha_i \right) c(T), \end{aligned} \quad (5)$$

and

$$\begin{aligned} &\frac{2}{k}\delta(E_T^*) + \frac{k-1}{k} (c(T) + 2\text{MST}_T - 2c(E_T^*)) \\ &\leq \frac{2}{k} \left( \frac{k}{2} + \alpha - \sum_{i=1}^m i\alpha_i \right) c(T) + \frac{k-1}{k} \left( 1 + 2 - \max_{1 \leq i \leq m} \alpha_i - 2(1-\alpha) \right) c(T) \\ &= \frac{2k-1}{k}c(T) + \frac{2}{k} \left( \alpha - \sum_{i=1}^m i\alpha_i \right) c(T) + \frac{k-1}{k} \left( 2\alpha - \max_{1 \leq i \leq m} \alpha_i \right) c(T) \\ &= \frac{2k-1}{k}c(T) + \left( 2\alpha - \frac{2}{k} \sum_{i=1}^m i\alpha_i \right) c(T) - \left( \max_{1 \leq i \leq m} \frac{k-1}{k}\alpha_i \right) c(T). \end{aligned} \quad (6)$$

Let

$$f(\vec{\alpha}) = \frac{5k-3}{2k} + \frac{2}{k} \left( \alpha - \sum_{i=1}^m i\alpha_i \right) - \left( \max_{1 \leq i \leq m} \frac{k-1}{2k}\alpha_i \right), \quad (7)$$

and

$$g(\vec{\alpha}) = \frac{2k-1}{k} + \left( 2\alpha - \frac{2}{k} \sum_{i=1}^m i\alpha_i \right) - \left( \max_{1 \leq i \leq m} \frac{k-1}{k}\alpha_i \right), \quad (8)$$

where  $\vec{\alpha} = (\alpha_1, \alpha_2, \dots, \alpha_m)$ .

By (4), (5), (6), and Lemma 6, the approximation ratio of our algorithm is at most

$$\max_{\substack{\alpha_1 + \alpha_2 + \dots + \alpha_m = \alpha \\ \alpha_1, \alpha_2, \dots, \alpha_m \geq 0 \\ 1 \geq \alpha \geq 0}} \min \{f(\vec{\alpha}), g(\vec{\alpha})\}. \quad (9)$$

Under the three conditions (or constraints) in (9), both  $f(\vec{\alpha})$  and  $g(\vec{\alpha})$  achieve their maximum values only when  $\alpha_1 \geq \alpha_2 \geq \dots \geq \alpha_m$ . As shown in (Zhao and Xiao, 2025a), the reason is that if there exists  $\alpha_p < \alpha_q$  for some  $p < q$ , then exchanging their values yields a larger solution: the value  $\max_{1 \leq i \leq m} \alpha_i$  remain unchanged, while the coefficients of  $\alpha_p$  and  $\alpha_q$  satisfy  $0 > \frac{-2p}{k} > \frac{-2q}{k}$ .

Therefore, the approximation ratio of our algorithm is at most

$$\max_{\substack{\alpha_1 + \alpha_2 + \dots + \alpha_m = \alpha \\ \alpha_1 \geq \alpha_2 \geq \dots \geq \alpha_m \geq 0 \\ 1 \geq \alpha \geq 0}} \min \{f(\vec{\alpha}), g(\vec{\alpha})\}. \quad (10)$$

Fixing the value of  $\alpha_1$ , and under the three conditions in (10), maximizing  $f(\vec{\alpha})$  (or  $g(\vec{\alpha})$ ) is equivalent to minimizing  $\sum_{i=1}^m i\alpha_i$ . It is easy to see that each  $\alpha_i$  should be set as large as possible, in order of increasing index. Specifically, when  $\alpha - \sum_{j=1}^{i-1} \alpha_j \geq 0$ , set  $\alpha_i = \min\{\alpha - \sum_{j=1}^{i-1} \alpha_j, \alpha_1\}$ . Then, we have  $\alpha_1 = \alpha_2 = \dots = \alpha_{l-1} \geq \alpha_l = \alpha - (l-1)\alpha_1 \geq 0$  and  $\alpha_{l+1} = \alpha_{l+2} = \dots = \alpha_m = 0$ , where  $l = \lceil \frac{\alpha}{\alpha_1} \rceil$ . Therefore, we have  $\sum_{i=1}^m i\alpha_i = \sum_{i=1}^{l-1} i\alpha_i + l\alpha_l = \frac{(l-1)l\alpha_1}{2} + l(\alpha - (l-1)\alpha_1) = l\alpha - \frac{(l-1)l\alpha_1}{2}$ .

Let

$$z(\alpha_1) = -l\alpha + \frac{(l-1)l\alpha_1}{2} - C_1 \cdot \alpha_1,$$

where  $C_1 > 0$  denotes a constant.

When  $\alpha_1 \in [\frac{\alpha}{t+1}, \frac{\alpha}{t})$  with  $t \in \mathbb{Z}_{\geq 1}$ , the function  $z(\alpha_1)$  is a continuous linear function. Moreover, since  $l = \lceil \frac{\alpha}{\alpha_1} \rceil = t+1$ , we have

$$\lim_{\alpha_1 \rightarrow (\frac{\alpha}{t})^-} z(\alpha_1) = -(t+1)\alpha + \frac{t(t+1) \cdot \frac{\alpha}{t}}{2} - C_1 \cdot \frac{\alpha}{t} = -\frac{(t+1)\alpha}{2} - C_1 \cdot \frac{\alpha}{t}. \quad (11)$$

When  $\alpha_1 = \frac{\alpha}{t}$ , since  $l = \lceil \frac{\alpha}{\alpha_1} \rceil = t$ , we have

$$z(\frac{\alpha}{t}) = -t\alpha + \frac{(t-1)t\alpha}{2} - C_1 \cdot \frac{\alpha}{t} = -\frac{(t+1)\alpha}{2} - C_1 \cdot \frac{\alpha}{t}. \quad (12)$$

Therefore, by (11) and (12), the function  $z(\alpha_1)$  is a continuous function when  $\alpha_1 \in (0, \alpha]$ . Recall that  $z(\alpha_1)$  is a continuous linear function when  $\alpha_1 \in [\frac{\alpha}{t+1}, \frac{\alpha}{t})$  with  $t \in \mathbb{Z}_{\geq 1}$ . So, it is easy to observe that there exists at least one value of  $\alpha_1$  maximizing  $z(\alpha_1)$  such that  $\frac{\alpha}{\alpha_1}$  is an integer.

Consequently, there exists at least one value of  $\alpha_1$  maximizing  $f(\vec{\alpha})$  (or  $g(\vec{\alpha})$ ) such that  $\frac{\alpha}{\alpha_1}$  is an integer.

Under  $\frac{\alpha}{\alpha_1} = l \in \mathbb{Z}_{\geq 1}$  and  $\alpha_1 = \alpha_2 = \dots = \alpha_l = \frac{\alpha}{l} = \frac{\alpha}{t}$ , as shown before, we have  $\sum_{i=1}^m i\alpha_i =$

$l\alpha - \frac{(l-1)l\alpha_1}{2} = l\alpha - \frac{(l-1)\alpha}{2} = \frac{(l+1)\alpha}{2}$ . Then, by (7) and (8), we have

$$\begin{aligned} f(\vec{\alpha}) &= \frac{5k-3}{2k} + \frac{2}{k} \left( \alpha - \sum_{i=1}^m i\alpha_i \right) - \left( \max_{1 \leq i \leq m} \frac{k-1}{2k} \alpha_i \right) \\ &= \frac{5k-3}{2k} + \frac{2}{k} \left( \alpha - \frac{(l+1)\alpha}{2} \right) - \frac{k-1}{2k} \cdot \frac{\alpha}{l} \\ &= \frac{5k-3}{2k} - \frac{(2l^2 - 2l + k - 1)\alpha}{2kl}, \end{aligned}$$

and

$$\begin{aligned} g(\vec{\alpha}) &= \frac{2k-1}{k} + \left( 2\alpha - \frac{2}{k} \sum_{i=1}^m i\alpha_i \right) - \left( \max_{1 \leq i \leq m} \frac{k-1}{k} \alpha_i \right) \\ &= \frac{2k-1}{k} + \left( 2\alpha - \frac{2}{k} \cdot \frac{(l+1)\alpha}{2} \right) - \frac{k-1}{k} \cdot \frac{\alpha}{l} \\ &= \frac{2k-1}{k} - \frac{(k-1)\alpha + l(l+1)\alpha - 2kl\alpha}{kl} \\ &= \frac{2k-1}{k} - \frac{(l^2 + l - 2kl + k - 1)\alpha}{kl}. \end{aligned}$$

Let

$$\tau(\alpha, l) = \frac{5k-3}{2k} - \frac{(2l^2 - 2l + k - 1)\alpha}{2kl} \quad \text{and} \quad \eta(\alpha, l) = \frac{2k-1}{k} - \frac{(l^2 + l - 2kl + k - 1)\alpha}{kl}.$$

Then, by (10), the approximation ratio of our algorithm is at most

$$\max_{\substack{\alpha_1 + \alpha_2 + \dots + \alpha_m = \alpha \\ \alpha_1 \geq \alpha_2 \geq \dots \geq \alpha_m \geq 0 \\ 1 \geq \alpha \geq 0}} \min \{f(\vec{\alpha}), g(\vec{\alpha})\} = \max_{\substack{l \in \mathbb{Z}_{\geq 1} \\ 1 \geq \alpha \geq 0}} \min \{\tau(\alpha, l), \eta(\alpha, l)\}, \quad (13)$$

as desired.  $\square$

**Theorem 1.** *For unit-demand CARP, there is a polynomial-time algorithm with an approximation ratio of  $\frac{5}{2} - \frac{2l^2 + 10l + k - 4}{2k(4l-1)}$ , where  $l = \lceil \frac{\sqrt{8k-7}-1}{4} \rceil$ .*

*Proof.* It is clear that our algorithm takes  $O(n^3)$  time.

Next, we compute the approximation ratio given by Lemma 7.

Recall that  $\tau(\alpha, l) = \frac{5k-3}{2k} - \frac{(2l^2 - 2l + k - 1)\alpha}{2kl}$  and  $\eta(\alpha, l) = \frac{2k-1}{k} - \frac{(l^2 + l - 2kl + k - 1)\alpha}{kl}$ . Therefore, when  $\alpha \leq \frac{l}{4l-1}$ , we have  $\eta(\alpha, l) \leq \tau(\alpha, l)$ ; otherwise, we have  $\eta(\alpha, l) \geq \tau(\alpha, l)$ .

**Case 1:**  $0 \leq \alpha \leq \frac{l}{4l-1}$ . In this case, by Lemma 7, the approximation ratio of our algorithm is at most  $\max_{l \in \mathbb{Z}_{\geq 1}, \frac{l}{4l-1} \geq \alpha \geq 0} \eta(\alpha, l)$ . Since  $\eta(\alpha, l)$  is a linear function w.r.t.  $\alpha$ , we have

$$\max_{\substack{l \in \mathbb{Z}_{\geq 1} \\ \frac{l}{4l-1} \geq \alpha \geq 0}} \eta(\alpha, l) = \max_{l \in \mathbb{Z}_{\geq 1}} \max \left\{ \eta(0, l), \eta\left(\frac{l}{4l-1}, l\right) \right\}. \quad (14)$$

We have  $\eta(0, l) = \frac{2k-1}{k}$ . Next, we consider  $\max_{l \in \mathbb{Z}_{\geq 1}} \eta(\frac{l}{4l-1}, l)$ .

First, we have

$$\eta(\frac{l}{4l-1}, l) = \frac{2k-1}{k} - \frac{l^2 + l - 2kl + k - 1}{k(4l-1)}. \quad (15)$$

Then, we obtain

$$\frac{d}{dl} \eta(\frac{l}{4l-1}, l) = \frac{-k(2l+1-2k)(4l-1) + 4k(l^2 + l - 2kl + k - 1)}{k^2(4l-1)^2} = \frac{-4l^2 + 2l + 2k - 3}{k(4l-1)^2}. \quad (16)$$

Therefore, by (16), when  $k \geq 3$  and  $l \geq 1$ ,  $\eta(\frac{l}{4l-1}, l)$  is a concave function w.r.t.  $l$ .

Consequently, we have

$$\max_{l \in \mathbb{Z}_{\geq 1}} \eta(\frac{l}{4l-1}, l) = \eta(\frac{\tilde{l}}{4\tilde{l}-1}, \tilde{l}), \quad (17)$$

where

$$\tilde{l} = \arg \min_{l \in \mathbb{Z}_{\geq 1}} \left\{ \eta(\frac{l}{4l-1}, l) - \eta(\frac{(l+1)}{4(l+1)-1}, (l+1)) \geq 0 \right\}. \quad (18)$$

Note that

$$\begin{aligned} & \eta(\frac{l}{4l-1}, l) - \eta(\frac{(l+1)}{4(l+1)-1}, (l+1)) \\ &= \left( \frac{(l+1)^2 + (l+1) - 2k(l+1) + k - 1}{k(4l+3)} \right) - \left( \frac{l^2 + l - 2kl + k - 1}{k(4l-1)} \right) \\ &= \frac{((l+1)^2 + (l+1) - 2k(l+1) + k - 1)(4l-1) - (l^2 + l - 2kl + k - 1)(4l+3)}{k(4l-1)(4l+3)} \\ &= \frac{2(2l^2 + l - k + 1)}{k(4l-1)(4l+3)}. \end{aligned} \quad (19)$$

By (19), when  $\eta(\frac{l}{4l-1}, l) - \eta(\frac{(l+1)}{4(l+1)-1}, (l+1)) = 0$  and  $l \geq 1$ , we have  $l = \frac{\sqrt{8k-7}-1}{4}$ . Then, by (18), we have  $\tilde{l} = \lceil \frac{\sqrt{8k-7}-1}{4} \rceil$ . Then, by (15) and (17), we have

$$\max_{l \in \mathbb{Z}_{\geq 1}} \eta(\frac{l}{4l-1}, l) = \frac{2k-1}{k} - \frac{\tilde{l}^2 + \tilde{l} - 2k\tilde{l} + k - 1}{k(4\tilde{l}-1)}, \quad \text{where } \tilde{l} = \left\lceil \frac{\sqrt{8k-7}-1}{4} \right\rceil. \quad (20)$$

Recall that  $\max_{l \in \mathbb{Z}_{\geq 1}} \eta(0, l) = \frac{2k-1}{k}$ . It can be verified that  $\frac{2k-1}{k} < \frac{2k-1}{k} - \frac{\tilde{l}^2 + \tilde{l} - 2k\tilde{l} + k - 1}{k(4\tilde{l}-1)}$  for all  $k \geq 3$ . Therefore, in the case where  $0 \leq \alpha \leq \frac{l}{4l-1}$ , by (14), the approximation ratio of our algorithm is at most  $\frac{2k-1}{k} - \frac{\tilde{l}^2 + \tilde{l} - 2k\tilde{l} + k - 1}{k(4\tilde{l}-1)}$ , where  $\tilde{l} = \lceil \frac{\sqrt{8k-7}-1}{4} \rceil$ .

**Case 2:**  $1 \geq \alpha \geq \frac{l}{4l-1}$ . In this case, by Lemma 7, the approximation ratio of our algorithm is at most  $\max_{l \in \mathbb{Z}_{\geq 1}, 1 \geq \alpha \geq \frac{l}{4l-1}} \tau(\alpha, l)$ . Since  $\tau(\alpha, l)$  is a linear function w.r.t.  $\alpha$ , we have

$$\max_{\substack{l \in \mathbb{Z}_{\geq 1} \\ 1 \geq \alpha \geq \frac{l}{4l-1}}} \tau(\alpha, l) = \max_{l \in \mathbb{Z}_{\geq 1}} \max \left\{ \tau(\frac{l}{4l-1}, l), \tau(1, l) \right\}. \quad (21)$$

Since  $\tau(\frac{l}{4l-1}, l) = \eta(\frac{l}{4l-1}, l)$ , by the previous analysis, we have

$$\max_{l \in \mathbb{Z}_{\geq 1}} \tau(\frac{l}{4l-1}, l) = \frac{2k-1}{k} - \frac{\tilde{l}^2 + \tilde{l} - 2k\tilde{l} + k - 1}{k(4\tilde{l}-1)}, \quad \text{where } \tilde{l} = \left\lceil \frac{\sqrt{8k-7}-1}{4} \right\rceil. \quad (22)$$

Next, we consider  $\max_{l \in \mathbb{Z}_{\geq 1}} \tau(1, l)$ .

First, we have

$$\tau(1, l) = \frac{5k-3}{2k} - \frac{2l^2 - 2l + k - 1}{2kl}. \quad (23)$$

Then, we obtain

$$\frac{d}{dl} \tau(1, l) = \frac{-2kl(4l-2) + 2k(2l^2 - 2l + k - 1)}{4k^2l^2} = \frac{-2l^2 + k - 1}{2k^2l^2}. \quad (24)$$

Therefore, by (24), when  $k \geq 3$  and  $l \geq 1$ ,  $\tau(1, l)$  is a concave function w.r.t.  $l$ .

Consequently, we have

$$\max_{l \in \mathbb{Z}_{\geq 1}} \tau(1, l) = \tau(1, \hat{l}), \quad (25)$$

where

$$\hat{l} = \arg \min_{l \in \mathbb{Z}_{\geq 1}} \{\tau(1, l) - \tau(1, (l+1)) \geq 0\}. \quad (26)$$

Note that

$$\begin{aligned} & \tau(1, l) - \tau(1, (l+1)) \\ &= \left( \frac{2(l+1)^2 - 2(l+1) + k - 1}{2k(l+1)} \right) - \left( \frac{2l^2 - 2l + k - 1}{2kl} \right) \\ &= \frac{(2(l+1)^2 - 2(l+1) + k - 1)l - (2l^2 - 2l + k - 1)(l+1)}{2kl(l+1)} \\ &= \frac{(2l^2 + 2l + k - 1)l - (2l^2 - 2l + k - 1)(l+1)}{2kl(l+1)} \\ &= \frac{(2l^3 + 2l^2 + kl - l) - (2l^3 + kl - 3l + k - 1)}{2kl(l+1)} \\ &= \frac{2l^2 + 2l - k + 1}{2kl(l+1)}. \end{aligned} \quad (27)$$

By (27), when  $\tau(1, l) - \tau(1, (l+1)) = 0$  and  $l \geq 1$ , we have  $l = \frac{\sqrt{2k-1}-1}{2}$ . Then, by (26), we have  $\hat{l} = \lceil \frac{\sqrt{2k-1}-1}{2} \rceil$ . Then, by (23) and (25), we have

$$\max_{l \in \mathbb{Z}_{\geq 1}} \tau(1, l) = \frac{5k-3}{2k} - \frac{2\hat{l}^2 - 2\hat{l} + k - 1}{2k\hat{l}}, \quad \text{where } \hat{l} = \left\lceil \frac{\sqrt{2k-1}-1}{2} \right\rceil. \quad (28)$$

Recall that  $\max_{l \in \mathbb{Z}_{\geq 1}} \tau(\frac{l}{4l-1}, l) = \frac{2k-1}{k} - \frac{\tilde{l}^2 + \tilde{l} - 2k\tilde{l} + k - 1}{k(4\tilde{l}-1)}$ , where  $\tilde{l} = \lceil \frac{\sqrt{8k-7}-1}{4} \rceil$ . It can be verified that  $\frac{5k-3}{2k} - \frac{2\hat{l}^2 - 2\hat{l} + k - 1}{2k\hat{l}} < \frac{2k-1}{k} - \frac{\tilde{l}^2 + \tilde{l} - 2k\tilde{l} + k - 1}{k(4\tilde{l}-1)}$  for all  $k \geq 3$ . Thus, in the case where  $1 \geq \alpha \geq \frac{l}{4l-1}$ , by

(21), the approximation ratio of our algorithm is at most  $\frac{2k-1}{k} - \frac{\tilde{l}^2 + \tilde{l} - 2k\tilde{l} + k - 1}{k(4\tilde{l}-1)}$ , where  $\tilde{l} = \lceil \frac{\sqrt{8k-7}-1}{4} \rceil$ .

In conclusion, the approximation ratio of our algorithm is at most

$$\frac{2k-1}{k} - \frac{\tilde{l}^2 + \tilde{l} - 2k\tilde{l} + k - 1}{k(4\tilde{l}-1)} = \frac{5}{2} - \frac{2\tilde{l}^2 + 10\tilde{l} + k - 4}{2k(4\tilde{l}-1)} = \frac{5}{2} - \Theta\left(\frac{1}{\sqrt{k}}\right),$$

where  $\tilde{l} = \lceil \frac{\sqrt{8k-7}-1}{4} \rceil$ . □

The approximation ratio in Theorem 1 is derived using two RPP tours  $H_1$  and  $H_2$  in  $G$ . Interestingly, it can be shown that the approximation ratio can be achieved using only the RPP tour  $H_1$  in  $G$  (see Section 6).

**Remark 1.** *CVRP can be viewed as a special case of CARP by replacing each customer with a zero-cost edge. Thus, CARP with  $\alpha = 1$  capture the CVRP. In fact, in the proof of Theorem 1, the ratio*

$$\frac{5k-3}{2k} - \frac{2\hat{l}^2 - 2\hat{l} + k - 1}{2k\hat{l}}, \quad \text{where } \hat{l} = \left\lceil \frac{\sqrt{2k-1}-1}{2} \right\rceil,$$

*obtained in the case  $\alpha = 1$ , exactly matches the approximation ratio for CVRP given in (Zhao and Xiao, 2025a).*

## 5 Proof of Lemma 6

**Lemma 6.** *For any tour  $T \in \mathcal{T}^*$ , there exist parameters  $\{\alpha_1, \alpha_2, \dots, \alpha_m\}$  such that*

- (1)  $\delta(E_T^*) \leq (\frac{k}{2} + \alpha - \sum_{i=1}^m i\alpha_i) c(T)$ ;
- (2)  $MST_T \leq (1 - \max_{1 \leq i \leq m} \frac{1}{2}\alpha_i) c(T)$ ;
- (3)  $c(E_T^*) = (1 - \alpha)c(T)$ ;
- (4)  $\sum_{i=1}^m \alpha_i = \alpha \leq 1$ , where  $m = \infty$  and  $0 \leq \alpha_i \leq \alpha$  for each  $1 \leq i \leq m$ .

*Proof.* We assume w.l.o.g. that  $c(T) > 0$ ; otherwise, the lemma holds trivially.

Let  $T = (v_0, v_1, v_2, \dots, v_{2l-1}, v_{2l}, v_0)$  and  $t = \lceil \frac{l+1}{2} \rceil$ . Then, we have  $l \leq k$ , and  $t \leq m$ . Note that we also have  $E_T^* = \{(v_{2i-1}, v_{2i}) \mid i \in [l]\}$  and  $|E_T^*| = l$ .

To construct the parameters  $\{\alpha_1, \alpha_2, \dots, \alpha_m\}$ , we set  $\alpha_i = 0$  for all  $t < i \leq m$ . Then, it remains to show how to set  $\alpha_i$  for each  $i \in [t]$ . We consider two cases based on the parity of  $l$ , i.e.,  $|E_T^*|$ .

**Case 1:  $|E_T^*|$  is odd.** In this case, we have  $t = \lceil \frac{l+1}{2} \rceil = \frac{l+1}{2}$ . We let

$$\alpha_i = \frac{c(v_{2i-2}, v_{2i-1}) + c(v_{2l+2-2i}, v_{(2l+3-2i) \bmod (2l+1)})}{c(T)}, \quad \forall i \in [t],$$

and

$$\beta_i = \frac{c(v_{2i-1}, v_{2i}) + c(v_{2l+1-2i}, v_{2l+2-2i})}{c(T)}, \quad \forall i \in [t-1].$$

Also, we let  $\beta_t = \frac{c(v_{2t-1}, v_{2t})}{c(T)}$ . Then, we have  $\alpha_i, \beta_i \geq 0$  for all  $i \in [t]$ .

Note that  $c(E(T) \setminus E_T^*) = \sum_{i=1}^t \alpha_i c(T)$  and  $c(E_T^*) = \sum_{i=1}^t \beta_i c(T)$ . Then,  $\sum_{i=1}^t (\alpha_i + \beta_i) = 1$ . Since  $\alpha_i = 0$  for any  $t < i \leq m$ , we have  $\alpha = \sum_{i=1}^m \alpha_i = \sum_{i=1}^t \alpha_i = 1 - \sum_{i=1}^t \beta_i \leq 1$ .

Therefore, the properties (3) and (4) in Lemma 6 are both satisfied.

Next, we prove the properties (1) and (2) in Lemma 6.

**Claim 1.** When  $|E_T^*|$  is odd, it holds that  $MST_T \leq (1 - \max_{1 \leq i \leq m} \frac{1}{2} \alpha_i) c(T)$ .

*Claim Proof.* Recall that  $MST_T$  measures the cost of a minimum-cost spanning tree in  $G[V_T]$  containing all edges in  $E_T^*$ . Deleting an arbitrary edge in  $E(T) \setminus E_T^*$  from  $T$  forms a spanning tree in  $G[V_T]$  containing all edges in  $E_T^*$ . Then, we have

$$\begin{aligned} MST_T &\leq c(T) - \max_{e \in E(T) \setminus E_T^*} c(e) \\ &= c(T) - \max_{1 \leq i \leq 2t} c(v_{2i-2}, v_{(2i-1) \bmod (2l+1)}) \\ &\leq c(T) - \max_{1 \leq i \leq t} \frac{1}{2} [c(v_{2i-2}, v_{2i-1}) + c(v_{2l+2-2i}, v_{(2l+3-2i) \bmod (2l+1)})] \\ &= \left(1 - \max_{1 \leq i \leq t} \frac{1}{2} \alpha_i\right) c(T) \\ &= \left(1 - \max_{1 \leq i \leq m} \frac{1}{2} \alpha_i\right) c(T), \end{aligned}$$

where the first equality follows from  $E(T) \setminus E_T^* = \{(v_{2i-2}, v_{(2i-1) \bmod (2l+1)}) \mid i \in [2t]\}$ , the second inequality from  $E(T) \setminus E_T^* = \{(v_{2i-2}, v_{2i-1}), (v_{2i-2}, v_{(2i-1) \bmod (2l+1)}) \mid i \in [t]\}$ , and the last inequality from the definition of  $\alpha_i$ .  $\square$

**Claim 2.** When  $|E_T^*|$  is odd, it holds that  $\delta(E_T^*) \leq (\frac{k}{2} + \alpha - \sum_{i=1}^m i \alpha_i) c(T)$ .

*Claim Proof.* Recall that  $\delta(E_T^*) = \sum_{(x,y) \in E_T^*} \frac{c(v_0, x) + c(x, y) + c(v_0, y)}{2}$ . Since  $E_T^* = \{(v_{2i-1}, v_{2i}) \mid i \in [2t-1]\}$ , we have

$$2\delta(E_T^*) = c(E_T^*) + \sum_{i=1}^{2t-1} [c(v_0, v_{2i-1}) + c(v_0, v_{2i})]. \quad (29)$$

Since  $E_T^* = \{(v_{2i-1}, v_{2i}), (v_{2l+1-2i}, v_{2l+2-2i}) \mid i \in [t-1]\} \cup \{(v_{2t-1}, v_{2t})\}$ , we have

$$\begin{aligned} &\sum_{i=1}^{2t-1} [c(v_0, v_{2i-1}) + c(v_0, v_{2i})] \\ &= c(v_0, v_{2t-1}) + c(v_0, v_{2t}) + \sum_{i=1}^{t-1} [c(v_0, v_{2i-1}) + c(v_0, v_{2i}) + c(v_0, v_{2l+1-2i}) + c(v_0, v_{2l+2-2i})] \quad (30) \\ &\leq c(v_0, v_{2t-1}) + c(v_0, v_{2t}) + \sum_{i=1}^{t-1} [2c(v_0, v_{2i-1}) + 2c(v_0, v_{2l+2-2i}) + \beta_i c(T)], \end{aligned}$$

where the inequality follows from  $c(v_0, v_{2i}) \leq c(v_0, v_{2i-1}) + c(v_{2i-1}, v_{2i})$  and  $c(v_0, v_{2l+1-2i}) \leq c(v_0, v_{2l+2-2i}) + c(v_{2l+1-2i}, v_{2l+2-2i})$  by the triangle inequality and  $\beta_i c(T) = \frac{c(v_{2i-1}, v_{2i}) + c(v_{2l+1-2i}, v_{2l+2-2i})}{c(T)}$  when  $i \in [t-1]$ .

For any  $i \in [t]$ , by the triangle inequality, we have

$$c(v_0, v_{2i-1}) \leq \sum_{j=1}^{2i-1} c(v_{j-1}, v_j) \quad \text{and} \quad c(v_0, v_{2l+2-2i}) \leq \sum_{j=1}^{2i-1} c(v_{2l+1-j}, v_{(2l+2-j) \bmod (2l+1)}). \quad (31)$$

Then, for any  $i \in [t]$ , we have

$$\begin{aligned} c(v_0, v_{2i-1}) + c(v_0, v_{2l+2-2i}) &\leq \sum_{j=1}^{2i-1} [c(v_{j-1}, v_j) + c(v_{2l+1-j}, v_{(2l+2-j) \bmod (2l+1)})] \\ &= \left( \sum_{j=1}^i \alpha_j + \sum_{j=1}^{i-1} \beta_j \right) c(T) \\ &= \left( \sum_{j=1}^i (\alpha_j + \beta_j) - \beta_i \right) c(T), \end{aligned} \quad (32)$$

where the first inequality follows from (31), and the first equality from the definitions of  $\alpha_i$  and  $\beta_i$ .

Then, we have

$$\begin{aligned} &c(v_0, v_{2t-1}) + c(v_0, v_{2t}) + \sum_{i=1}^{t-1} [2c(v_0, v_{2i-1}) + 2c(v_0, v_{2l+2-2i}) + \beta_i c(T)] \\ &\leq \left( \sum_{j=1}^t (\alpha_j + \beta_j) - \beta_t \right) c(T) + \sum_{i=1}^{t-1} \left( \sum_{j=1}^i 2(\alpha_j + \beta_j) - \beta_i \right) c(T) \\ &= \left( \sum_{j=1}^t \alpha_j + \sum_{i=1}^{t-1} \sum_{j=1}^i 2(\alpha_j + \beta_j) \right) c(T) \\ &= \left( \sum_{j=1}^t \alpha_j + \sum_{j=1}^t 2(t-j)(\alpha_j + \beta_j) \right) c(T) \\ &= \left( \sum_{j=1}^t \alpha_j + 2t - \sum_{j=1}^t 2j(\alpha_j + \beta_j) \right) c(T), \end{aligned} \quad (33)$$

where the inequality follows from (32) and the last equality from  $\sum_{j=1}^t (\alpha_j + \beta_j) = 1$ .



Therefore, by (29), (30), and (33), we obtain that

$$\begin{aligned}
2\delta(E_T^*) &= c(E_T^*) + \sum_{i=1}^{2t-1} (c(v_0, v_i) + c(v_0, v_{2l+1-i})) \\
&\leq \left( \sum_{j=1}^t (\alpha_j + \beta_j) + 2t - \sum_{j=1}^t 2j(\alpha_j + \beta_j) \right) c(T) \\
&= \left( l + 2 - \sum_{j=1}^t 2j(\alpha_j + \beta_j) \right) c(T) \\
&\leq \left( k + 2 - \sum_{j=1}^t 2j\alpha_j - \sum_{j=1}^t 2j\beta_j \right) c(T) \\
&= \left( k + \sum_{j=1}^t 2\alpha_j - \sum_{j=1}^t 2j\alpha_j \right) c(T) \\
&= \left( k + 2\alpha - \sum_{j=1}^t 2j\alpha_j \right) c(T).
\end{aligned} \tag{34}$$

where the first equality follows from (29), the second equality follows from  $\sum_{j=1}^t (\alpha_j + \beta_j) = 1$  and  $t = \frac{l+1}{2}$ , the last equality follows from  $\alpha = \sum_{j=1}^t \alpha_j$ , the first inequality follows from (30) and (33) and  $c(E_T^*) = \sum_{j=1}^t \beta_j c(T)$ , and the second inequality follows from  $l \leq k$  and  $\beta_j \geq 0$  for all  $j \in [t]$ .

Recall that  $\alpha_j = 0$  for  $t < j \leq m$ . Then, by (34), we obtain  $\delta(E_T^*) \leq (\frac{k}{2} + \alpha - \sum_{i=1}^m i\alpha_i) c(T)$ .  $\square$

By Claims 1 and 2, the properties (1) and (2) in Lemma 6 are both satisfied when  $|E_T^*|$  is odd. Therefore, Lemma 6 holds when  $|E_T^*|$  is odd.

Next, we consider the case that  $|E_T^*|$  is even.

**Case 2:  $|E_T^*|$  is even.** In this case, we have  $t = \lceil \frac{l+1}{2} \rceil = \frac{l+2}{2}$ . We let

$$\alpha_i = \frac{c(v_{2i-2}, v_{2i-1}) + c(v_{2l+2-2i}, v_{(2l+3-2i) \bmod (2l+1)})}{c(T)}, \quad \forall i \in [t-1],$$

and

$$\beta_i = \frac{c(v_{2i-1}, v_{2i}) + c(v_{2l+1-2i}, v_{2l+2-2i})}{c(T)}, \quad \forall i \in [t-1].$$

Also, we let  $\alpha_t = \frac{c(v_{2t-2}, v_{2t-1})}{c(T)}$  and  $\beta_t = 0$ . Then, we have  $\alpha_i, \beta_i \geq 0$  for all  $i \in [t]$ .

Similarly,  $c(E(T) \setminus E_T^*) = \sum_{i=1}^t \alpha_i c(T)$  and  $c(E_T^*) = \sum_{i=1}^t \beta_i c(T)$ . Then,  $\sum_{i=1}^t (\alpha_i + \beta_i) = 1$ . Since  $\alpha_i = 0$  for any  $t < i \leq m$ , we have  $\alpha = \sum_{i=1}^m \alpha_i = \sum_{i=1}^t \alpha_i = 1 - \sum_{i=1}^t \beta_i \leq 1$ .

Therefore, the properties (3) and (4) in Lemma 6 are both satisfied.

Next, we prove the properties (1) and (2) in Lemma 6.

**Claim 3.** When  $|E_T^*|$  is even, it holds that  $MST_T \leq (1 - \max_{1 \leq i \leq m} \frac{1}{2} \alpha_i) c(T)$ .

*Claim Proof.* Recall that  $\text{MST}_T$  measures the cost of a minimum-cost spanning tree in  $G[V_T]$  containing all edges in  $E_T^*$ . Deleting an arbitrary edge in  $E(T) \setminus E_T^*$  from  $T$  forms a spanning tree in  $G[V_T]$  containing all edges in  $E_T^*$ . Then, we have

$$\begin{aligned}
\text{MST}_T &\leq c(T) - \max_{e \in E(T) \setminus E_T^*} c(e) \\
&= c(T) - \max_{1 \leq i \leq 2t-1} c(v_{2i-2}, v_{(2i-1) \bmod (2l+1)}) \\
&\leq c(T) - \max_{1 \leq i \leq t-1} \max \left\{ \frac{1}{2} (c(v_{2i-2}, v_{2i-1}) + c(v_{2l+2-2i}, v_{(2l+3-2i) \bmod (2l+1)})), c(v_{2t-2}, v_{2t-1}) \right\} \\
&= \left( 1 - \max_{1 \leq i \leq t-1} \max \left\{ \frac{1}{2} \alpha_i, \alpha_t \right\} \right) c(T) \\
&\leq \left( 1 - \max_{1 \leq i \leq m} \frac{1}{2} \alpha_i \right) c(T),
\end{aligned}$$

where the first equality follows from  $E(T) \setminus E_T^* = \{(v_{2i-2}, v_{(2i-1) \bmod (2l+1)}) \mid i \in [2t-1]\}$ , the second inequality from  $E(T) \setminus E_T^* = \{(v_{2i-2}, v_{2i-1}), (v_{2i-2}, v_{(2i-1) \bmod (2l+1)}) \mid i \in [t-1]\} \cup \{(v_{2t-2}, v_{2t-1})\}$ , and the last inequality from the definition of  $\alpha_i$ .  $\square$

**Claim 4.** When  $|E_T^*|$  is even, it holds that  $\delta(E_T^*) \leq (\frac{k}{2} + \alpha - \sum_{i=1}^m i\alpha_i) c(T)$ .

*Claim Proof.* Since  $E_T^* = \{(v_{2i-1}, v_{2i}) \mid i \in [2t-2]\}$ , we have

$$2\delta(E_T^*) = c(E_T^*) + \sum_{i=1}^{2t-2} [c(v_0, v_{2i-1}) + c(v_0, v_{2i})]. \quad (35)$$

Since  $E_T^* = \{(v_{2i-1}, v_{2i}), c(v_{2l+1-2i}, v_{2l+2-2i}) \mid i \in [t-1]\}$ , we have

$$\begin{aligned}
\sum_{i=1}^{2t-2} [c(v_0, v_{2i-1}) + c(v_0, v_{2i})] &= \sum_{i=1}^{t-1} [c(v_0, v_{2i-1}) + c(v_0, v_{2i}) + c(v_0, v_{2l+1-2i}) + c(v_0, v_{2l+2-2i})] \\
&\leq \sum_{i=1}^{t-1} [2c(v_0, v_{2i-1}) + 2c(v_0, v_{2l+2-2i}) + \beta_i c(T)],
\end{aligned} \quad (36)$$

where the inequality follows from  $c(v_0, v_{2i}) \leq c(v_0, v_{2i-1}) + c(v_{2i-1}, v_{2i})$  and  $c(v_0, v_{2l+1-2i}) \leq c(v_0, v_{2l+2-2i}) + c(v_{2l+1-2i}, v_{2l+2-2i})$  by the triangle inequality and  $\beta_i c(T) = \frac{c(v_{2i-1}, v_{2i}) + c(v_{2l+1-2i}, v_{2l+2-2i})}{c(T)}$  when  $i \in [t-1]$ .

For any  $i \in [t-1]$ , by the triangle inequality, we have

$$c(v_0, v_{2i-1}) \leq \sum_{j=1}^{2i-1} c(v_{j-1}, v_j) \quad \text{and} \quad c(v_0, v_{2l+2-2i}) \leq \sum_{j=1}^{2i-1} c(v_{2l+1-j}, v_{(2l+2-j) \bmod (2l+1)}). \quad (37)$$

Then, for any  $i \in [t-1]$ , we have

$$\begin{aligned}
c(v_0, v_{2i-1}) + c(v_0, v_{2l+2-2i}) &\leq \sum_{j=1}^{2i-1} [c(v_{j-1}, v_j) + c(v_{2l+1-j}, v_{(2l+2-j) \bmod (2l+1)})] \\
&= \left( \sum_{j=1}^i \alpha_j + \sum_{j=1}^{i-1} \beta_j \right) c(T) \\
&= \left( \sum_{j=1}^i (\alpha_j + \beta_j) - \beta_i \right) c(T),
\end{aligned} \tag{38}$$

where the first inequality follows from (37), and the first equality from the definitions of  $\alpha_i$  and  $\beta_i$ .

Then, we have

$$\begin{aligned}
&\sum_{i=1}^{t-1} [2c(v_0, v_{2i-1}) + 2c(v_0, v_{2l+2-2i}) + \beta_i c(T)] \\
&\leq \sum_{i=1}^{t-1} \left( \sum_{j=1}^i 2(\alpha_j + \beta_j) - \beta_i \right) c(T) \\
&= \left( \sum_{i=1}^{t-1} \sum_{j=1}^i 2(\alpha_j + \beta_j) - \sum_{j=1}^{t-1} \beta_j \right) c(T) \\
&= \left( \sum_{j=1}^t 2(t-j)(\alpha_j + \beta_j) - \sum_{j=1}^{t-1} \beta_j \right) c(T) \\
&= \left( 2t - \sum_{j=1}^t 2j(\alpha_j + \beta_j) - \sum_{j=1}^t \beta_j \right) c(T),
\end{aligned} \tag{39}$$

where the inequality follows from (38) and the last equality from  $\sum_{j=1}^t (\alpha_j + \beta_j) = 1$  and  $\beta_t = 0$ .

Therefore, by (35), (36), and (39), we obtain that

$$\begin{aligned}
2\delta(E_T^*) &= c(E_T^*) + \sum_{i=1}^{2t-2} [c(v_0, v_i) + c(v_0, v_{2l+1-i})] \\
&\leq \left( 2t - \sum_{j=1}^t 2j(\alpha_j + \beta_j) \right) c(T) \\
&= \left( l + 2 - \sum_{j=1}^t 2j(\alpha_j + \beta_j) \right) c(T) \\
&\leq \left( k + 2 - \sum_{j=1}^t 2j\alpha_j - \sum_{j=1}^t 2\beta_j \right) c(T) \\
&= \left( k + \sum_{j=1}^t 2\alpha_j - \sum_{j=1}^t 2j\alpha_j \right) c(T) \\
&= \left( k + 2\alpha - \sum_{j=1}^t 2j\alpha_j \right) c(T).
\end{aligned} \tag{40}$$

where the first equality follows from (35), the second equality follows from  $\sum_{j=1}^t (\alpha_j + \beta_j) = 1$  and  $t = \frac{l+2}{2}$ , the last equality follows from  $\alpha = \sum_{j=1}^t \alpha_j$ , the first inequality follows from (36) and (39) and  $c(E_T^*) = \sum_{j=1}^t \beta_j c(T)$ , and the second inequality follows from  $l \leq k$  and  $\beta_j \geq 0$  for all  $j \in [t]$ .

Recall that  $\alpha_j = 0$  for  $t < j \leq m$ . Then, by (40), we have  $\delta(E_T^*) \leq (\frac{k}{2} + \alpha - \sum_{i=1}^m i\alpha_i) c(T)$ .  $\square$

By Claims 3 and 4, the properties (1) and (2) in Lemma 6 are both satisfied when  $|E_T^*|$  is even. Therefore, Lemma 6 also holds when  $|E_T^*|$  is odd.  $\square$

## 6 A Note on the Approximation Ratio

In this section, we show that the approximation ratio in Theorem 1 can be achieved by using only the RPP tour  $H_1$  in  $G$ .

By Lemma 4, it suffices to prove the following lemma.

**Lemma 9.** *It holds that  $c(H_1) \leq OPT + 2MST - 2c(E^*)$ .*

*Proof.* We first recall the details of the  $\frac{3}{2}$ -approximation algorithm (Eiselt et al., 1995b).

The algorithm begins by computing a minimum-cost spanning tree  $T^*$  in  $G$  such that  $E^* \subseteq E(T^*)$ . Then, it computes a minimum-cost perfect matching  $M^*$  in  $G[Odd(V(T^*))]$ , where  $Odd(V(T^*))$  denotes the set of odd-degree vertices in  $T^*$ . Clearly, the graph  $G^* = (V \cup \{v_0\}, E(T^*) \cup E(M^*))$  forms an Eulerian graph. Then, it computes an Eulerian tour in  $G^*$ , which then is transformed into the RPP tour  $H_1$  by shortcutting. Therefore, we have

$$c(H_1) \leq c(E(T^*)) + c(M^*).$$

By definition, we have  $c(E(T^*)) \leq \text{MST}$ . Now, we prove that  $c(M^*) \leq \text{OPT} + \text{MST} - 2c(E^*)$ . Let  $\bar{E}^* := E(T^*) \setminus E^*$  and  $M$  be a minimum-cost perfect matching in  $G[V]$ .

It is clear that  $M \cup E^*$  forms a graph where each vertex has an even degree. Moreover, since  $\bar{E}^* \cup E^* = E(T^*)$ , we know that  $M \cup E^* \cup \bar{E}^*$  forms a connected graph. Furthermore,  $M \cup E^* \cup \bar{E}^* \cup \bar{E}^*$  forms an Eulerian graph. Therefore, the set  $M \cup \bar{E}^*$  augment the tree  $T^*$  into an Eulerian graph.

Since the cost function  $c$  is a metric function, we known that  $M^*$  augments  $T^*$  into an Eulerian graph using minimum-cost. Therefore, we have

$$c(M^*) \leq c(M) + c(\bar{E}^*) = c(M) + c(E(T^*)) - c(E^*).$$

Recall that  $c(E(T^*)) \leq \text{MST}$  by definition and  $c(E^*) + c(M) \leq \text{OPT}$  by (2). Thus, we obtain

$$c(M^*) \leq \text{OPT} + \text{MST} - 2c(E^*),$$

as desired. □

## 7 Conclusion

In this paper, by extending the techniques in approximating CVRP (Zhao and Xiao, 2025a), we propose a  $(\frac{5}{2} - \Theta(\frac{1}{\sqrt{k}}))$ -approximation algorithm for equal-demand CARP. To our knowledge, this is the first improvement over the classic result of  $\frac{5}{2} - \frac{1.5}{k}$  (Jansen, 1993). In the future, it would be interesting to investigate whether the methods in (Blauth et al., 2023) can be adapted to equal-demand CARP to break the  $\frac{5}{2}$ -approximation barrier.

## References

- Altinkemer, K. and Gavish, B. (1987). Heuristics for unequal weight delivery problems with a fixed error guarantee. *Oper. Res. Lett.*, 6(4):149–158.
- Altinkemer, K. and Gavish, B. (1990). Technical note—heuristics for delivery problems with constant error guarantees. *Transportation Science*, 24(4):294–297.
- Asano, T., Katoh, N., Tamaki, H., and Tokuyama, T. (1996). Covering points in the plane by k-tours: a polynomial approximation scheme for fixed k. *IBM Tokyo Research Laboratory Research Report RT0162*.
- Babae Tirkolaee, E., Alinaghian, M., Bakhshi Sasi, M., and Seyyed Esfahani, M. (2016). Solving a robust capacitated arc routing problem using a hybrid simulated annealing algorithm: a waste collection application. *Journal of Industrial Engineering and Management Studies*, 3(1):61–76.
- Belenguer, J. M. and Benavent, E. (2003). A cutting plane algorithm for the capacitated arc routing problem. *Computers & Operations Research*, 30(5):705–728.

- Blauth, J., Traub, V., and Vygen, J. (2023). Improving the approximation ratio for capacitated vehicle routing. *Math. Program.*, 197(2):451–497.
- Bompadre, A., Dror, M., and Orlin, J. B. (2006). Improved bounds for vehicle routing solutions. *Discret. Optim.*, 3(4):299–316.
- Brandão, J. and Eglese, R. (2008). A deterministic tabu search algorithm for the capacitated arc routing problem. *Computers & Operations Research*, 35(4):1112–1126.
- Chen, Y., Hao, J.-K., and Glover, F. (2016). A hybrid metaheuristic approach for the capacitated arc routing problem. *European Journal of Operational Research*, 253(1):25–39.
- Christofides, N. (2022). Worst-case analysis of a new heuristic for the travelling salesman problem. *Oper. Res. Forum*, 3(1).
- Chvátal, V. (1979). A greedy heuristic for the set-covering problem. *Math. Oper. Res.*, 4(3):233–235.
- Corberán, A., Eglese, R., Hasle, G., Plana, I., and Sanchis, J. M. (2021). Arc routing problems: A review of the past, present, and future. *Networks*, 77(1):88–115.
- Corberán, Á. and Laporte, G. (2015). *Arc routing: problems, methods, and applications*. SIAM.
- Corberán, A. and Prins, C. (2010). Recent results on arc routing problems: An annotated bibliography. *Networks*, 56(1):50–69.
- Dantzig, G. B. and Ramser, J. H. (1959). The truck dispatching problem. *Manag. Sci.*, 6(1):80–91.
- Eiselt, H. A., Gendreau, M., and Laporte, G. (1995a). Arc routing problems, part I: The chinese postman problem. *Oper. Res.*, 43(2):231–242.
- Eiselt, H. A., Gendreau, M., and Laporte, G. (1995b). Arc routing problems, part II: the rural postman problem. *Oper. Res.*, 43(3):399–414.
- Fernández, E., Fontana, D., and Speranza, M. G. (2016). On the collaboration uncapacitated arc routing problem. *Computers & Operations Research*, 67:120–131.
- Friggstad, Z., Mousavi, R., Rahgoshay, M., and Salavatipour, M. R. (2022). Improved approximations for capacitated vehicle routing with unsplittable client demands. In Aardal, K. I. and Sanità, L., editors, *Integer Programming and Combinatorial Optimization - 23rd International Conference, IPCO 2022, Eindhoven, The Netherlands, June 27-29, 2022, Proceedings*, volume 13265 of *Lecture Notes in Computer Science*, pages 251–261. Springer.
- Golden, B. L. and Wong, R. T. (1981). Capacitated arc routing problems. *Networks*, 11(3):305–315.
- Gupta, A., Lee, E., and Li, J. (2023). A local search-based approach for set covering. In Kavitha, T. and Mehlhorn, K., editors, *2023 Symposium on Simplicity in Algorithms, SOSA 2023, Florence, Italy, January 23-25, 2023*, pages 1–11. SIAM, SIAM.

- Haimovich, M. and Kan, A. H. G. R. (1985). Bounds and heuristics for capacitated routing problems. *Math. Oper. Res.*, 10(4):527–542.
- Harks, T., König, F. G., and Matuschke, J. (2013). Approximation algorithms for capacitated location routing. *Transportation Science*, 47(1):3–22.
- Hassin, R. and Levin, A. (2005). A better-than-greedy approximation algorithm for the minimum set cover problem. *SIAM J. Comput.*, 35(1):189–200.
- Jansen, K. (1993). Bounds for the general capacitated routing problem. *Networks*, 23(3):165–173.
- Karlin, A. R., Klein, N., and Gharan, S. O. (2021). A (slightly) improved approximation algorithm for metric TSP. In Khuller, S. and Williams, V. V., editors, *STOC '21: 53rd Annual ACM SIGACT Symposium on Theory of Computing, Virtual Event, Italy, June 21-25, 2021*, pages 32–45. ACM.
- Karlin, A. R., Klein, N., and Gharan, S. O. (2023). A deterministic better-than-3/2 approximation algorithm for metric TSP. In Pia, A. D. and Kaibel, V., editors, *Integer Programming and Combinatorial Optimization - 24th International Conference, IPCO 2023, Madison, WI, USA, June 21-23, 2023, Proceedings*, volume 13904 of *Lecture Notes in Computer Science*, pages 261–274. Springer.
- Karpinski, M., Lampis, M., and Schmied, R. (2015). New inapproximability bounds for tsp. *Journal of Computer and System Sciences*, 81(8):1665–1677.
- Li, C. and Simchi-Levi, D. (1990). Worst-case analysis of heuristics for multidepot capacitated vehicle routing problems. *INFORMS J. Comput.*, 2(1):64–73.
- Mourão, M. C., Nunes, A. C., and Prins, C. (2009). Heuristic methods for the sectoring arc routing problem. *European Journal of Operational Research*, 196(3):856–868.
- Mourão, M. C. and Pinto, L. S. (2017). An updated annotated bibliography on arc routing problems. *Networks*, 70(3):144–194.
- Perrier, N., Langevin, A., and Campbell, J. F. (2007). A survey of models and algorithms for winter road maintenance. part iv: Vehicle routing and fleet sizing for plowing and snow disposal. *Computers & Operations Research*, 34(1):258–294.
- Santos, L., Coutinho-Rodrigues, J., and Current, J. R. (2009). An improved heuristic for the capacitated arc routing problem. *Computers & Operations Research*, 36(9):2632–2637.
- Schrijver, A. (2003). *Combinatorial optimization: polyhedra and efficiency*, volume 24. Springer.
- Tang, K., Mei, Y., and Yao, X. (2009). Memetic algorithm with extended neighborhood search for capacitated arc routing problems. *IEEE Transactions on Evolutionary Computation*, 13(5):1151–1166.

- van Bevern, R., Fluschnik, T., and Tsidulko, O. Y. (2020). On approximate data reduction for the rural postman problem: Theory and experiments. *Networks*, 76(4):485–508.
- van Bevern, R., Hartung, S., Nichterlein, A., and Sorge, M. (2014a). Constant-factor approximations for capacitated arc routing without triangle inequality. *Oper. Res. Lett.*, 42(4):290–292.
- Van Bevern, R., Komusiewicz, C., and Sorge, M. (2017). A parameterized approximation algorithm for the mixed and windy capacitated arc routing problem: Theory and experiments. *Networks*, 70(3):262–278.
- van Bevern, R., Niedermeier, R., Sorge, M., and Weller, M. (2014b). The complexity of arc routing problems. In *Arc routing: problems, methods, and applications*, pages 19–52. SIAM.
- Williamson, D. P. and Shmoys, D. B. (2011). *The design of approximation algorithms*. Cambridge university press.
- Wøhlk, S. (2008a). An approximation algorithm for the capacitated arc routing problem. *Open Operational Research Journal*, 2:8–12.
- Wøhlk, S. (2008b). A decade of capacitated arc routing. In *The vehicle routing problem: latest advances and new challenges*, pages 29–48. Springer.
- Wøhlk, S. and Laporte, G. (2018). A fast heuristic for large-scale capacitated arc routing problems. *Journal of the Operational Research Society*, 69(12):1877–1887.
- Yu, W., Liao, Y., and Yang, Y. (2023). Exact and approximation algorithms for the multi-depot capacitated arc routing problems. *Tsinghua Science and Technology*, 28(5):916–928.
- Zhao, J. and Xiao, M. (2025a). Improved approximation algorithms for capacitated vehicle routing with fixed capacity. In *50th International Symposium on Mathematical Foundations of Computer Science, MFCS 2025, August 25-29, 2025, Warsaw, Poland*, volume 345 of *LIPIcs*, pages 93:1–93:19. Schloss Dagstuhl - Leibniz-Zentrum für Informatik.
- Zhao, J. and Xiao, M. (2025b). Multidepot capacitated vehicle routing with improved approximation guarantees. *Theor. Comput. Sci.*, 1043:115265.