TSINGHUA SCIENCE AND TECHNOLOGY
ISSN 1007-0214 09/14 pp916–928
DOI: 10.26599/TST.2022.9010052
Volume 28, Number 5, October 2023

Exact and Approximation Algorithms for the Multi-Depot Capacitated Arc Routing Problems

Wei Yu*, Yujie Liao, and Yichen Yang

Abstract: In this work, we investigate a generalization of the classical capacitated arc routing problem, called the Multi-depot Capacitated Arc Routing Problem (MCARP). We give exact and approximation algorithms for different variants of the MCARP. First, we obtain the first constant-ratio approximation algorithms for the MCARP and its nonfixed destination version. Second, for the multi-depot rural postman problem, i.e., a special case of the MCARP where the vehicles have infinite capacity, we develop a $(2-\frac{1}{2k+1})$-approximation algorithm (k denotes the number of depots). Third, we show the polynomial solvability of the equal-demand MCARP on a line and devise a 2-approximation algorithm for the multi-depot capacitated vehicle routing problem on a line. Lastly, we conduct extensive numerical experiments on the algorithms for the multi-depot rural postman problem to show their effectiveness.

Key words: approximation algorithm; multi-depot; vehicle routing problem; arc routing problem; rural postman problem

1 Introduction

The Capacitated Arc Routing Problem (CARP) is defined as follows. Let $G=(V,E)$ be an undirected (multi)graph, where V is the vertex set and E is the edge set. There is a nonnegative cost function $c: E \to \mathbf{R}^+$ and a nonnegative integer demand function $d: E \to \mathbf{Z}^+$. Initially, a fleet of identical vehicles with capacity Q is located at a special vertex $o \in V$, known as the depots. The objective is to determine a set of routes (or closed walks), which start from and end at the depot, for the vehicles to serve the edges with positive demands so that each vehicle serves a total demand of at most Q and the total cost of the routes is as small as possible. In the CARP, if the demands are defined for the vertices instead of the edges, we obtain the Capacitated Vehicle Routing Problem (CVRP).

Golden and Wong[1] showed that the CVRP is actually a special case of the CARP, since one can split each vertex in the CVRP into two vertices joined by a zero-cost edge whose demand equals the original vertex demand. The CARP has found applications in many practical problems, which include electric power line inspection[2], school bus routing[3], garbage collection[4] and distribution service[5].

The Multi-depot Capacitated Arc Routing Problem (MCARP) is a natural generalization of the CARP in which multiple depots are available and the routes of the vehicles may start from and end at any depots. Similarly, we can define the Multi-depot Capacitated Vehicle Routing Problem (MCVRP). The investigation of the MCARP/MCVRP has been motivated not only by their theoretical interest, but also by their emerging applications in various practical domains. For the CARP/CVRP defined on a large service area, the total cost of the routes may be very expensive due to the single-depot constraint. In this setting, a potential solution is to introduce multiple depots to meet the

• Wei Yu and Yujie Liao are with School of Mathematics, East China University of Science and Technology, Shanghai 200237, China. E-mail: yuwei@ecust.edu.cn; y30190191@mail.ecust.edu.cn.
• Yichen Yang is with Sabre Lab Research Team, Sabre Inc., Southlake, TX 76092, USA. E-mail: Yichen.Yang@sabre.com.
∗ To whom correspondence should be addressed.
Manuscript received: 2022-06-23; revised: 2022-10-03; accepted: 2022-11-10

service requirements[6]. These depots usually represent warehouses, supply points, relay boxes, and dumping places. For example, online shopping companies often distribute their service or goods at multiple depots (i.e., distribution centers) to improve customer satisfaction in urban areas[7]. Typical applications of the MCARP/MCVRP include mail delivery[8], police patrolling[9], and explosive waste recycling[10].

The CARP and the CVRP are clearly NP-hard, because they are extensions of the classical rural postman problem and the metric traveling salesman problem, respectively, where the vehicles have infinite capacity. This implies that the more general MCARP/MCVRP is also NP-hard. As a result, the current research on the MCARP/MCVRP has focused on branch-and-cut approach (e.g., see Refs. [11, 12]) and meta-heuristics (e.g., see Refs. [7, 8, 13]). Whereas, in this paper we mainly deal with approximation algorithms for the MCARP/MCVRP. Currently, few approximability results on the multi-depot extension of the CARP/CVRP are available in the literature. Especially, as far as we know there exists no approximation algorithm for the MCARP.

Haimovich and Kan[14] first studied approximation algorithms for the equal-demand CVRP, i.e., a restricted case of the CVRP where each vertex has unit demand. They proposed the well-known iterated tour partition heuristic $ITP(\alpha)$, where α is the best available approximation ratio for the metric TSP (the results in Refs. [15, 16] imply that $\alpha \leqslant 3/2$), and showed that $ITP(\alpha)$ has an approximation ratio of $1 + (1 - 1/Q)\alpha$ on condition that the number $n = |V|$ of vertices is divided by the capacity Q. Later, Haimovich et al.[17] and Altinkemer and Gavish[18] achieved the same ratio without the condition that n is divided by $Q^{\dagger}$. For the general CVRP where there is no restriction on the demands of the vertices, Altinkemer and Gavish[19] devised an approximation algorithm, known as $UITP(\alpha)$, with ratio $2 + (1 - 2/Q)\alpha$ by generalizing $ITP(\alpha)$. Haimovich et al.[17] simplified the proof of the approximation ratio of $UITP(\alpha)$. $ITP(\alpha)$ and $UITP(\alpha)$ had been the best available approximation algorithms for the CVRP for over 35 years. Recently, Blauth et al.[20] succeeded in developing an improved approximation algorithm for the CVRP with ratio $\alpha + 2(1 - \epsilon)$, where $\epsilon > 0$ is a small constant greater than $1/3000$. They also improved the ratio for the equal-demand CVRP to $\alpha + 1 - \epsilon$ for some small constant $\epsilon > 1/3000$. Friggstad et al.[21] then further improved the approximation ratio for the CVRP to $\alpha + \ln 2 + 1 \approx \alpha + 1.694$.

For the results on the CVRP defined on special graphs, Labbé et al.[22] obtained an approximation algorithm for the CVRP on trees with ratio 2, and Wu and Lu[23] proposed a 5/3-approximation algorithm for the CVRP on a line. Note that even the CVRP on a half-line (i.e., the depot vertex is one of the two end vertices of the line graph) is NP-hard[24]. Actually, the CVRP on a half-line has an inapproximability lower bound of 3/2[23].

Concerning the CARP with triangle inequality, Jansen[25] extended the above heuristics $ITP(\alpha)$ and $UITP(\alpha)$ to derive approximation algorithms whose ratios are $1 + (1 - 1/Q)\alpha_0$ for the equal-demand case and $2 + (1 - 2/Q)\alpha_0$ for the general case. Here α_0 denotes the best-known approximation ratio for the rural postman problem (we have $\alpha_0 \leqslant 3/2$ due to Refs. [2, 26]). Wøhlk[27] developed an alternative approximation algorithm with the same ratio for the CARP with triangle inequality. Interestingly, van Bevern et al.[26] showed that any γ-approximation algorithm for the CARP with triangle inequality can be transformed into a γ-approximation algorithm for the general CARP (without triangle inequality).

Li and Simchi-Levi[28] presented approximation algorithms for the multi-depot CVRP, whose ratios are $1 + (2 - 1/Q)\alpha$ for the equal-demand case and $2+(2-2/Q)\alpha$ for the general case. They also introduced the nonfixed destination MCVRP, which is a variant of the MCVRP allowing the vehicles to start from one depot but terminate at a different depot, and designed a $(1+(1-1/Q)\alpha)$-approximation algorithm for the equal-demand case and a $(2 + (1 - 2/Q)\alpha)$-approximation algorithm for the general case.

In this paper, we address exact and approximation algorithms for the MCARP and obtain the following results. First, we give the first constant-factor approximation algorithms for the MCARP as well as the nonfixed destination variant. Second, for a special case of the MCARP where the vehicle capacity is unbounded, called the Multi-depot Rural Postman Problem (MRPP), we propose a $(2 - \frac{1}{2k+1})$-approximation algorithm with k representing the number of depots. Third, we study the MCARP/MCVRP defined on a line graph. Although the line graph represents a simple topology, it does appear in some applications, including highways and rivers. Moreover, an algorithm designed for a problem on a line

† In fact, the versions of $ITP(\alpha)$ in Refs. [17, 18] slightly differ from that in Ref. [14]. However, we still call them $ITP(\alpha)$.

can be used as a subroutine to solve the same problem defined on a relatively complicated network (e.g., a tree), as shown by Bhattacharya and Hu[29]. For the equal-demand MCARP on a line, we give a polynomial time exact algorithm, and for the MCVRP on a line we devise a 2-approximation algorithm. Finally, we conduct extensive numerical experiments on the algorithms for the MRPP to show their effectiveness.

The remainder of this paper is organized as follows. We start with the formal description of the problems and some notations in Section 2. We analyze approximation algorithms for the nonfixed destination MCARP in Section 3. In Section 4, we deal with the (fixed destination) MCARP. We give the approximation algorithms for the MRPP in Section 5. In Section 6, we present exact and approximation algorithms for the MCARP/MCVRP defined on a line graph. Lastly, we conduct numerical experiments on the algorithms for the MRPP in Section 7.

2 Preliminary

The problems studied in this paper are defined formally as follows:

Definition 1 In the MCARP, we are given an undirected graph $G = (V, E)$ with vertex set V and edge set E. There is a nonnegative cost (or length) function $c : E \to \mathbf{R}^+$ and a nonnegative integer demand function $d : E \to \mathbf{Z}^+$. Let $D \subseteq V$ be the depot set. Initially, a fleet of identical vehicles with capacity Q is located at each depot in D. The objective is to determine a set of routes (or closed walks), each of which starts from and ends at the same depot, for the vehicles to serve the edges with positive demands so that each vehicle serves a total demand of at most Q and the total cost of the routes is minimized.

Definition 2 The Nonfixed Destination MCARP is a variant of the MCARP where the route of each vehicle is allowed to start from some depot and end at another depot.

Definition 3 The MCVRP is a variant of the MCARP where the demand function is defined on the vertex set V instead of the edge set and all the vertices have positive demands.

Throughout this paper, algorithms on different versions of the MCARP/MCVRP are analyzed. For the MCARP, the optimal value is given by Z^*. The optimal value of the nonfixed destination MCARP is denoted by Z_n^*. The objective value of the solution generated by some algorithm A is denoted with Z^A.

Suppose that $G = (V, E)$ is an underlying graph, we use $c(e) \geqslant 0$ to represent the cost (or length) of edge $e \in E$. An edge e with end vertices $v, w \in V$ is also denoted by $\{v, w\}$ and its cost is also represented by $c(v, w)$. We use a nonnegative integer $d(v)$ ($d(e)$) to indicate the demand of vertex v (edge e). An edge e with $d(e) > 0$ is referred to as a required edge. R denotes the set of all required edges. Q represents the vehicle capacity. $P(v, w)$ is the shortest path between v and w and its length is represented by $c_s(v, w)$. If H is a subgraph of G, we use $V(H)$ and $E(H)$ to represent the vertex set and edge (multi)set of H, respectively. We define the cost of H as $c(H) = \sum_{e \in E(H)} c(e)$. The total cost of the required edges in H is denoted by $c_R(H)$. As a result, the total cost of the non-required edges in H is $c(H) - c_R(H)$.

3 Nonfixed Destination MCARP

In this section, we generalize the algorithm for the nonfixed destination MCVRP in Ref. [28] to obtain an approximation algorithm $NMCARP(\gamma)$ for the nonfixed destination MCARP, where γ is the best-known approximation ratio for the CARP. Our algorithm has a very simple description by applying the algorithm for the CARP (without triangle inequality) in Ref. [26].

Assume that $G = (V, E)$ is the original graph for the nonfixed destination MCARP with depot set $D \subseteq V$. The algorithm $NMCARP(\gamma)$ is a two-stage procedure that invokes a γ-approximation algorithm for the CARP. In the first stage, the algorithm contracts the depot set D of G into a single depot d to derive a new graph G', and invokes any γ-approximation algorithm for the CARP defined on G' to obtain a solution consisting of some routes starting from and terminating at d. In the second stage, the algorithm uncontracts d back to the original set D of depots to generate a feasible solution of the original nonfixed destination MCARP. The procedure is described formally in Algorithm 1.

In Algorithm 1, note that in Step 2, the routes $C_1', C_2', \ldots, C_t'$ are generated by an algorithm for the CARP defined on G' and hence satisfy the capacity constraint. The definition of the demands of G' and the construction of $P_1, P_2, \ldots, P_t$ in Step 3 imply that $P_1, P_2, \ldots, P_t$ also satisfy the capacity constraint.

Next we analyze the approximation ratio of Algorithm $NMCARP(\gamma)$.

Lemma 1 $Z^{NMCARP(\gamma)} \leqslant \gamma Z_n^*$.

Algorithm 1 *NMCARP*(γ)

Step 1: Construct a new graph $G' = (V', E')$ from $G = (V, E)$ with $V' = \{d\} \cup (V \setminus D)$, where d represents the single depot, and each edge $\{v, w\} \in E$ corresponds to an edge $\{v', w'\} \in E'$ with the same cost and demand, such that

$$\begin{cases} v' = v, w' = w, & \text{if } v, w \in V \setminus D; \\ v' = v, w' = d, & \text{if } v \in V \setminus D, w \in D; \\ v' = d, w' = w, & \text{if } v \in D, w \in V \setminus D; \\ v' = w' = d, & \text{if } v, w \in D. \end{cases}$$

Note that the last case implies that $\{v', w'\}$ is a self-loop in G'.

Step 2: Run any γ-approximation algorithm for the CARP defined on G' to produce a solution composed of t routes $C_1', C_2', \ldots, C_t'$ starting from and terminating at the depot d. Without loss of generality, we suppose that each C_i' includes d exactly twice[†].

Step 3: For each $i = 1, 2, \ldots, t$, we replace each edge $\{v', w'\}$ of C_i' with the original edge $\{v, w\}$ in G corresponding to $\{v', w'\}$ to produce a route P_i in G whose both end vertices are depots in D (although they may be different).

Step 4: Output the routes $P_1, P_2, \ldots, P_t$.

Note:[†] Otherwise, one can break C_i' into several routes including d exactly twice.

Proof Suppose that $Z^*(G')$ is the optimal value of the CARP defined on G' in Step 2 of Algorithm 1. Any feasible solution, say $\mathcal{P}$, to the nonfixed destination MCARP can be transformed into a feasible solution, say $\mathcal{C}$, to the CARP defined on G' by contracting the depots in D into d. By the definition of G', the cost of $\mathcal{C}$ is no more than that of $\mathcal{P}$, which implies that $Z^*(G') \leqslant Z_n^*$. Also due to the definition of G', the total cost of the routes $C_1', C_2', \ldots, C_t'$ cannot exceed $\gamma Z^*(G')$. Note that in Step 3 the total cost of the routes $P_1, P_2, \ldots, P_t$ is equal to the total cost of the routes $C_1', C_2', \ldots, C_t'$. Consequently, we have

$$Z^{NMCARP(\gamma)} = \sum_{i=1}^{t} c(P_i) \leqslant \gamma Z^*(G') \leqslant \gamma Z_n^* .$$

■

The results in Refs. [2, 25, 26] indicate that for the CARP there is a $(2 + (1 - 2/Q)\alpha_0)$-approximation algorithm, which we call *UITP*(α_0), and for the equal-demand case there exists a $(1 + (1 - 1/Q)\alpha_0)$-approximation algorithm, which we refer to as *ITP*(α_0). As before, α_0 denotes the best available approximation ratio for the rural postman problem. The algorithm *UITP*(α_0) works as follows. First it generates an α_0-approximate rural postman tour (a rural postman tour is a closed walk traversing all the required edges at least once) and then partitions this tour properly into edge disjoint sub-walks. Eventually, for each sub-walk the algorithm connects the depot with each of the two end vertices of this sub-walk by the shortest path between them.

By applying Lemma 1 we have the following result.

Theorem 1 For the nonfixed destination MCARP, there exists a $(2 + (1 - 2/Q)\alpha_0)$-approximation algorithm. If the demands are equal, there is a $(1 + (1 - 1/Q)\alpha_0)$-approximation algorithm.

Remark 1 It can be seen that our algorithm has a very simple description, thanks to the adoption of the γ-approximation algorithm for the CARP without triangle inequality. More exactly, when obtaining the graph G' we simply contract the depot set without changing the costs and demands of the edges. In contrast, after contracting the depot set, the *UITP*$_n(\alpha)$ heuristic for the nonfixed destination CVRP in Ref. [28] needs to modify the edge costs by determining the shortest paths between all pairs of vertices in G' and adds some dummy edges. This is because the algorithm *UITP*$_n(\alpha)$ uses the *UITP*(α) heuristic for the CVRP with triangle inequality, but the edge costs in G' may violate the triangle inequality.

4 Fixed Destination MCARP

We now deal with the (fixed destination) MCARP where all the routes must start from and end at the same depot.

Our algorithm *MUITP*(α_0) (see Algorithm 2) for the MCARP is obtained by modifying the algorithm *NMCARP*(γ) as follows. First, we substitute the above-mentioned algorithm *UITP*(α_0) for the γ-approximation algorithm in Step 2 in Algorithm 2. Then we adjust the solution constructed in Step 4 in Algorithm 2 to produce a feasible solution for the MCARP.

In Algorithm 2, one can see that in Step 2 the routes $C_1', C_2', \ldots, C_t'$ are produced by the algorithm *UITP*(α_0) for the CARP defined on G' and hence satisfy the capacity constraint. The definition of the demands of G' in Step 1 and the construction of $C_1, C_2', \ldots, C_t$ in Step 4 imply that $C_1, C_2', \ldots, C_t$ also satisfy the capacity constraint.

By definition, we have

$$c(P_i) = c_s(d_1^{(i)}, v_1^{(i)}) + \sum_{h=1}^{r_i-1} c(v_h^{(i)}, v_{h+1}^{(i)}) + c_s(v_{r_i}^{(i)}, d_2^{(i)}).$$

To analyze the approximation ratio of the algorithm *MUITP*(α_0), we use L^* to denote the cost of the optimal rural postman tour with respect to G' in Step 2 in

Algorithm 2 ***MUITP(α_0)***

Step 1: Do the same as in Step 1 of Algorithm 1 to obtain a new graph $G' = (V', E')$ from $G = (V, E)$ with $V' = \{d\} \cup (V \setminus D)$, where d represents the single depot.

Step 2: Run the algorithm $UITP(\alpha_0)$ for the CARP defined on G' to produce a solution with t routes $C'_1, C'_2, \ldots, C'_t$ starting from and terminating at the depot d. As before we assume without loss of generality, each C'_i includes d exactly twice.

Step 3: Do the same as in Step 3 of Algorithm 1 to produce routes $P_1, P_2, \ldots, P_t$ in G whose both end vertices are depots in D.

Step 4: For each $i = 1, 2, \ldots, t$, according to algorithm *UITP*(α_0), P_i takes the following form: $d_1^{(i)}, P(d_1^{(i)}, v_1^{(i)}), v_1^{(i)}, v_2^{(i)}, \ldots, v_{r_i}^{(i)}, P(v_{r_i}^{(i)}, d_2^{(i)}), d_2^{(i)}$, where $d_1^{(i)}, d_2^{(i)} \in D$ and $P(d_1^{(i)}, v_1^{(i)})$ $(P(v_{r_i}^{(i)}, d_2^{(i)}))$ is the shortest path between $d_1^{(i)}$ $(v_{r_i}^{(i)})$ and $v_1^{(i)}$ $(d_2^{(i)})$, and $v_h^{(i)} \in V \setminus D$ $(h = 1, 2, \ldots, r_i)$. We modify P_i to generate C_i in the following way: if $d_1^{(i)}$ and $d_2^{(i)}$ are identical, then P_i is already feasible and we simply define $C_i = P_i$; otherwise C_i is redefined as

$$\begin{cases} d_1^{(i)}, P(d_1^{(i)}, v_1^{(i)}), v_1^{(i)}, v_2^{(i)}, \ldots, v_{r_i}^{(i)}, P(v_{r_i}^{(i)}, d_1^{(i)}), d_1^{(i)}, \\ \quad \text{if } c_s(d_1^{(i)}, v_1^{(i)}) + c_s(v_{r_i}^{(i)}, d_1^{(i)}) \leqslant \\ \qquad c_s(d_2^{(i)}, v_1^{(i)}) + c_s(v_{r_i}^{(i)}, d_2^{(i)}); \\ d_2^{(i)}, P(d_2^{(i)}, v_1^{(i)}), v_1^{(i)}, v_2^{(i)}, \ldots, v_{r_i}^{(i)}, P(v_{r_i}^{(i)}, d_2^{(i)}), d_2^{(i)}, \\ \quad \text{if } c_s(d_1^{(i)}, v_1^{(i)}) + c_s(v_{r_i}^{(i)}, d_1^{(i)}) > \\ \qquad c_s(d_2^{(i)}, v_1^{(i)}) + c_s(v_{r_i}^{(i)}, d_2^{(i)}). \end{cases}$$

Step 5: Output the routes $C_1, C_2, \ldots, C_t$.

Algorithm 2. That is, L^* indicates the length of the shortest closed walk in G' containing d and all required edges. Let $L(\alpha_0)$ be the cost of an α_0-approximate rural postman tour used by *UITP*(α_0). Obviously, $L(\alpha_0) \leqslant \alpha_0 L^*$. Furthermore, due to *UITP*(α_0) it holds that

$$\sum_{i=1}^{t} \sum_{h=1}^{r_i - 1} c(v_h^{(i)}, v_{h+1}^{(i)}) \leqslant L(\alpha_0).$$

This is because $Q_i = v_1^{(i)}, v_2^{(i)}, \ldots, v_{r_i}^{(i)}$ $(i = 1, 2, \ldots, t)$ is a consecutive segment along the α_0-approximate rural postman tour used by *UITP*(α_0) and all Q_i's are edge disjoint.

Now we can prove the following result.

Lemma 2 $Z^{MUITP(\alpha_0)} \leqslant (2 + (2 - 2/Q)\alpha_0)\, Z^*$.

Proof Similarly to the analysis of the algorithm *ITP*$_f(\alpha)$ for the MCVRP in Ref. [28], one can deduce that $c(C_i) \leqslant c(P_i) + \sum_{h=1}^{r_i-1} c(v_h^{(i)}, v_{h+1}^{(i)})$. Thus we have

$$Z^{MUITP(\alpha_0)} = \sum_{i=1}^{t} c(C_i) \leqslant$$

$$\sum_{i=1}^{t} c(P_i) + \sum_{i=1}^{t} \sum_{h=1}^{r_i-1} c(v_h^{(i)}, v_{h+1}^{(i)}) \leqslant$$

$$\left(2 + \left(1 - \frac{2}{Q}\right)\alpha_0\right) Z_n^* + L(\alpha_0).$$

Since $Z_n^* \leqslant Z^*$ and $L(\alpha_0) \leqslant \alpha_0 L^* \leqslant \alpha_0 Z^*$, the proof is completed. ■

By replacing algorithm *UITP*(α_0) with *ITP*(α_0) in Algorithm 2, we can derive an approximation algorithm for the equal-demand MCARP whose approximation ratio is at most $1 + (2 - 1/Q)\alpha_0$. In a nutshell, we have shown the following result for the MCARP.

Theorem 2 There is a $(2 + (2 - 2/Q)\alpha_0)$-approximation algorithm for the MCARP. In particular, for the equal-demand problem there exists a $(1 + (2 - 1/Q)\alpha_0)$-approximation algorithm.

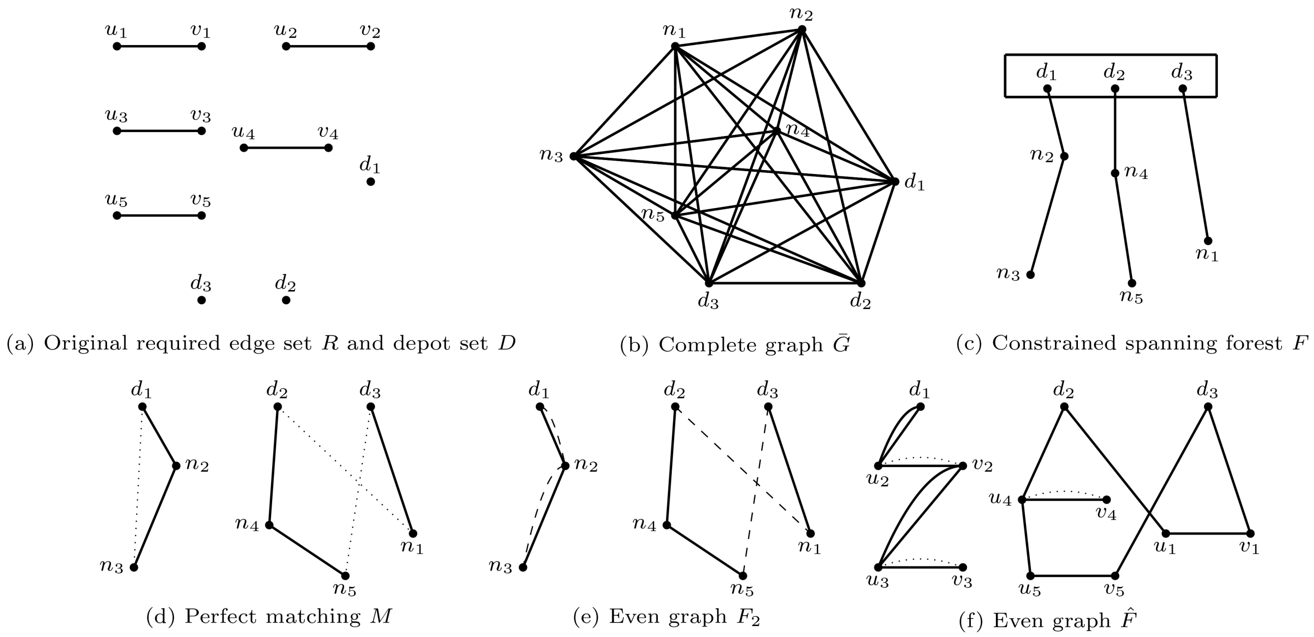
5 Multi-Depot Rural Postman Problem

In this section, we discuss the Multi-depot Rural Postman Problem (MRPP), i.e., a special case of the MCARP with infinite vehicle capacity. Let $k = |D|$ be the number of depots. In essence, the MRPP is to determine at most k closed walks covering all the required edges, such that each closed walk starts from and terminates at a distinct depot and the total cost of the walks is minimized.

In the rest of this section, we suppose that the input graph $G = (V, E)$ has the following two properties: (1) $V = V(R) \cup D$. That is, each vertex is either an end vertex of some required edge or a depot; (2) $V(R) \cap D = \varnothing$ and any two required edges do not contain common end vertices. This is without loss of generality because the second property holds by splitting properly some vertices and then the first property can be guaranteed by considering an equivalent instance defined on the reduced graph $\tilde{G} = (V(\tilde{G}), E(\tilde{G}))$ of G. Here the vertex set of $\tilde{G}$ is $V(\tilde{G}) = V(R) \cup D$ and the edge set $E(\tilde{G})$ of $\tilde{G}$ consists of R and $E' = V(\tilde{G}) \times V(\tilde{G})$, where the length of any edge in R is the same as in G and the length of each edge $e = \{u, v\} \in E'$ equals the length of the shortest path between u and v in G (see Refs. [2], [30], and [31] for more details). Furthermore, we also assume that $k \geqslant 2$. Because the MRRP with $k = 1$ is exactly the rural postman problem, which admits an approximation algorithm with ratio 3/2 as mentioned in the introduction.

Next we describe two algorithms for the MRPP. In the end, we will choose the better solution produced by these two algorithms. The first one is given below and an example demonstrating the steps of this algorithm can be found in Fig. 1.

It is not hard to see that Algorithm 3 runs in



(a) Original required edge set R and depot set D (b) Complete graph $\bar{G}$ (c) Constrained spanning forest F

(d) Perfect matching M (e) Even graph F_2 (f) Even graph $\hat{F}$

Fig. 1 Example for Algorithm *MRPP*1. The dotted lines indicate the perfect matching *M* in (d), the dashed lines denote the shortest paths corresponding to the matching edges in *M* in (e), and the dotted lines represent the copies of the odd edges in (f).

polynomial time and outputs a feasible solution to the MRPP.

Algorithm 3 *MRPP*1

Step 1: Suppose that

$$R = \{\{u_1, v_1\}, \{u_2, v_2\}, \ldots, \{u_{|R|}, v_{|R|}\}\}.$$

In G, we contract each required edge $\{u_i, v_i\} \in R$ into a vertex n_i to obtain a graph G_1 with vertex set $N \cup D$ where $N = \{n_i \mid i = 1, 2, \ldots, |R|\}$. Then we derive a complete graph $\bar{G}$ on $N \cup D$ by setting the cost $\bar{c}(u, v)$ of the edge $\{u, v\}$ to be the length of the shortest path between u and v in G_1.

Step 2: For the graph $\bar{G}$, find a minimum cost constrained spanning forest F, i.e., a minimum cost spanning forest that contains exactly k trees and each tree of the forest contains a distinct depot$^{⁕}$.

Step 3: Compute a minimum cost perfect matching M on the set of odd degree vertices in F. Adding M to F derives an even graph F_1 with at most k connected components. Replace each edge of F_1 by the shortest path in G_1 to derive another even graph F_2.

Step 4: In F_2, we uncontract the vertices in N as the original required edges of G to generate a graph F_3. For each edge $\{u_i, v_i\} \in R$ corresponding to $n_i \in N$, if both u_i and v_i are of odd degree in F_3, we call $\{u_i, v_i\}$ an odd edge and replicate $\{u_i, v_i\}^{\star}$. This will result in an even graph $\hat{F}$ with $p \leqslant k$ connected components.

Step 5: For $j = 1, 2, \ldots, p$, compute the Eulerian tour C_j of the j-th connected components of $\hat{F}$. Return the routes $C_1, C_2, \ldots, C_p$.

Notes: ⁕ F can be found in polynomial time, as mentioned by Xu et al.[32]. ⋆ Note that the degrees of u_i and v_i have the same parity since F_2 is an even graph.

Let C^* be the optimal solution. Next we evaluate the performance of the algorithm. We need the following result by Xu et al.[32].

Lemma 3 (Xu et al.[32]) Given a complete graph $\tilde{G}$ with edge cost obeying the triangle inequality and $k \geqslant 2$ depot vertices, if $\tilde{F}$ is the minimum cost constrained spanning forest of $\tilde{G}$ and $\tilde{M}$ is the minimum cost perfect matching on the set of odd degree vertices in $\tilde{F}$, then the total cost of $\tilde{F}$ and $\tilde{M}$ is at most $2 - 1/k$ times the total cost of any $\tilde{p} \leqslant k$ cycles covering all the vertices such that each cycle contains a distinct depot.

Now we can give an upper bound on the total cost of the routes $C_1, C_2, \ldots, C_p$.

Lemma 4 If $k \geqslant 2$,

$$\sum_{j=1}^{p} c(C_j) \leqslant (2 - \frac{1}{k})c(C^*) + \frac{1}{k}c_R(C^*).$$

Proof It can be seen that all the non-required edges in C^* correspond to a constrained spanning forest of $\bar{G}$, which implies that $c(F) \leqslant c(C^*) - c_R(C^*)$. Moreover, the non-required edges in C^* also contain at most k cycles covering all the vertices in $\bar{G}$ with each cycle including a distinct depot. Using Lemma 3, this leads to

$$c(F_2) = c(F_1) = c(F) + c(M) \leqslant \left(2 - \frac{1}{k}\right)(c(C^*) - c_R(C^*)) \quad (1)$$

We proceed to derive an upper bound on the total cost of the odd edges in R added in Step 4 in Algorithm 3. Since F_3 is obtained by uncontracting the edges in R from F_2, we have $c(F_3) \leqslant c(F_2) + c(R)$. To produce

$\hat{F}$ we add a copy of each odd edge in R, which implies

$$c(\hat{F}) \leqslant c(F_3) + c(R) \leqslant c(F_2) + 2c(R) \quad (2)$$

Therefore,

$$\sum_{j=1}^{p} c(C_j) = c(\hat{F}) \leqslant \left(2 - \frac{1}{k}\right)(c(C^*) - c_R(C^*)) + 2c(R) \leqslant \left(2 - \frac{1}{k}\right)c(C^*) + \frac{1}{k}c_R(C^*),$$

where the first inequality follows Formulas (1) and (2), and the second inequality holds by the fact that $c(R) \leqslant c_R(C^*)$. ■

Our second algorithm for the MRPP is described in Algorithm 4. An example illustrating the steps of Algorithm 4 is given in Fig. 2.

It is easy to see that Algorithm 4 runs in polynomial time and returns a feasible solution to the MRPP.

As before, we use C^* to denote the optimal solution.

Algorithm 4 ***MRPP*2**

Step 1: Construct a weighted graph $G_0 = (V \cup D', E_0)$ from G, where $D' = \{d'_1, d'_2, \ldots, d'_k\}$ is a copy of $D = \{d_1, d_2, \ldots, d_k\}$ with d'_i corresponding to d_i and $E_0 = E_1 \cup E_2$ with

$$E_1 = \{\{u, v\} \mid \text{at least one of } u, v \text{ is not from } D \cup D'\}$$

and

$$E_2 = \{\{d_i, d'_i\} \mid i = 1, 2, \ldots, k\}.$$

The weight of each edge $\{u, v\}$ equals the length of the shortest path between u and v in G, while the weight of each edge in E_2 is defined as zero.

Step 2: Find a minimum weight perfect matching M for G_0. Set $M_0 = M \cap E_2$, and $M_1 = M \setminus M_0$. Each edge $\{u, v\} \in M_1$ corresponds to a shortest path $P(u, v)$ from u to v in G. Adding $P(u, v)$ for all u and v to the graph $H_0 = (V, R)$ derives a spanning subgraph H of G. Assume that H contains q connected components $F_1, F_2, \ldots, F_q$, each of which is an Eulerian graph, and the first $l \leqslant k$ components include at least one depot.

Step 3: Find a minimum cost non-required edge subset E', such that for any vertex $v \in V \setminus D$ there is a path between v and some depot. This can be done by the following Kruskal-like procedure. Starting with $F_1, F_2, \ldots, F_q$ we keep adding the least cost edge between two different connected components, such that at least one of the components does not contain a depot. The procedure is terminated when there are exactly l connected components which constitute a spanning subgraph H_1. By doubling the edges of E' in H_1, we derive a spanning subgraph H_2 composed of l connected components, say $F'_1, F'_2, \ldots, F'_l$. Each of these components contains a depot and is Eulerian. Let C'_j $(j = 1, 2, \ldots, l)$ be the Eulerian tour of F'_j.

Step 4: Return the routes $C'_1, C'_2, \ldots, C'_l$.

The total cost of the routes produced by Algorithm 4 is bounded in the following lemma.

Lemma 5 $\sum_{j=1}^{l} c(C'_j) \leqslant 3c(C^*) - 2c_R(C^*)$.

Proof Note that there may be multiple copies of required edges of R in C^*, we fix one copy of each edge of R in C^*. Then all the paths in C^* between the end vertices of the fixed copies of required edges correspond to a perfect matching of G_0 in Step 1 in Algorithm 4. The weight of this perfect matching is at most $c(C^*) - c(R)$. Due to the optimality of M, this implies that $c(M) \leqslant c(C^*) - c(R)$. It holds that $c(H) = c(M) + c(R) \leqslant c(C^*)$ by definition. It can be verified that the non-required edges in C^* can connect the connected components $F_1, F_2, \ldots, F_q$ into at most l components, such that there is a path between any non-depot vertex and some depot. This means that $c(E') \leqslant c(C^*) - c_R(C^*)$. According to the construction of Steps 3 and 4 in Algorithm 4, one can see that

$$\sum_{j=1}^{l} c(C'_j) \leqslant c(H_2) = c(H) + 2c(E') \leqslant c(C^*) + 2(c(C^*) - c_R(C^*)) = 3c(C^*) - 2c_R(C^*).$$

■

Due to Lemmas 4 and 5, the cost of the better solution between Algorithm *MRPP*1 and Algorithm *MRPP*2 is at most

$$\min\left\{(2 - \frac{1}{k})c(C^*) + \frac{1}{k}c_R(C^*), 3c(C^*) - 2c_R(C^*)\right\} \leqslant \left(2 - \frac{1}{2k+1}\right)c(C^*) \quad (3)$$

which holds by equality if $c_R(C^*) = \dfrac{k+1}{2k+1}c(C^*)$.

Theorem 3 There is a $(2 - 1/(2k+1))$-approximation algorithm for the MRPP.

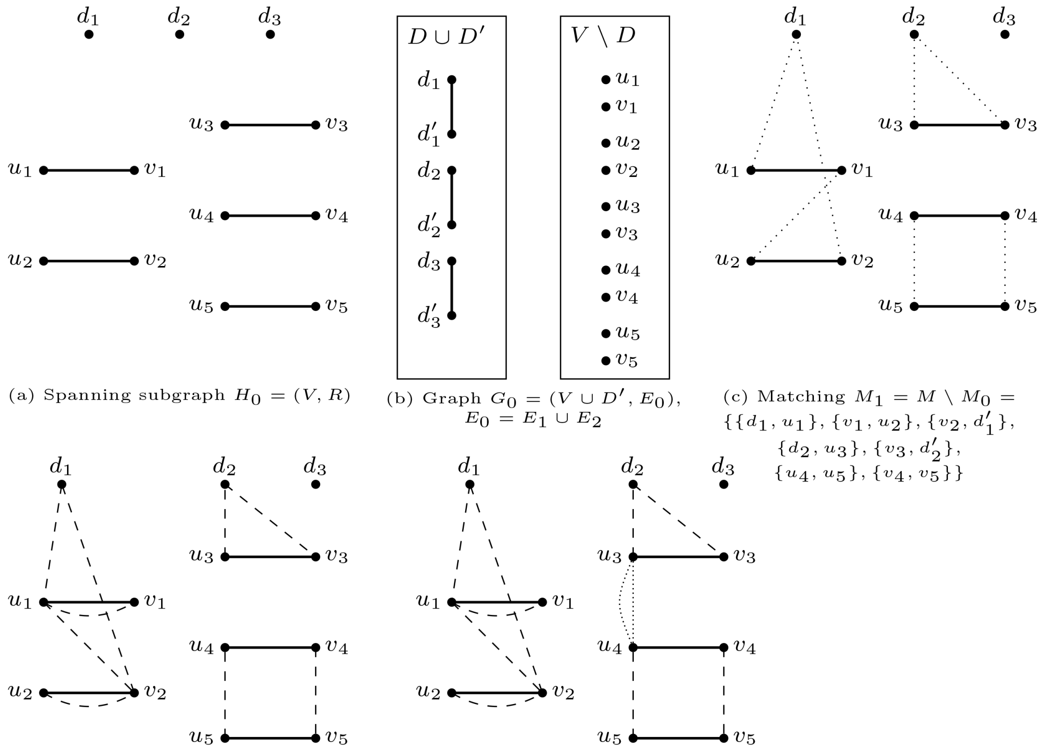
6 Multi-Depot CARP on a Line

In this section, we deal with the MCARP/MCVRP defined on a line graph. We show that the equal-demand MCARP on a line can be solved in $O(n^2)$ time. For the MCVRP on a line, we give the first 2-approximation algorithm.

Let $L = (V, E)$ with $V = \{v_1, v_2, \ldots, v_n\}$ and

$$E = \{(v_i, v_{i+1}) \mid i = 1, 2, \ldots, n-1\}$$

be the underlying line graph. Assume that the depots are given by $d_1 = v_{i_1}, d_2 = v_{i_2}, \ldots, d_k = v_{i_k}$ with $i_1 < i_2 < \cdots < i_k$. One can observe that there is an optimal solution in which the demands between d_{j-1} and d_j are always served by the vehicles located at either



(a) Spanning subgraph $H_0 = (V, R)$

(b) Graph $G_0 = (V \cup D', E_0)$, $E_0 = E_1 \cup E_2$

(c) Matching $M_1 = M \setminus M_0 = \{\{d_1, u_1\}, \{v_1, u_2\}, \{v_2, d'_1\}, \{d_2, u_3\}, \{v_3, d'_2\}, \{u_4, u_5\}, \{v_4, v_5\}\}$

(d) Spanning subgraph H (Path $P = v_1\ u_1\ v_2\ u_2$ corresponds to the matching edge $\{v_1, u_2\}$ in M_1)

(e) Spanning subgraph H_2

Fig. 2 Example for Algorithm *MRPP2*. Only edges in E_2 are depicted in (b), each vertex $d'_i \in D'$ has been merged with $d_i \in D$ and the dotted lines indicate the edges in M_1 in (c), the dashed lines denote the shortest paths corresponding to the matching edges in M_1 in (d), the dotted lines represent the edges in E' and their corresponding copies in (e).

d_{j-1} or d_j for $j = 1, 2, \ldots, k+1$ (for convenience, we define $d_0 = v_1$ and $d_{k+1} = v_n$ as two dummy depots). This is true since otherwise we can break the routes of the vehicles at the depots and reassign the new routes to the vehicles to satisfy the above property without increasing the total cost. Therefore, the MCARP on a line can be reduced to two special cases of the same problem. The first one is called the CARP on a half-line, where the single depot is located at one of the two end points of the line graph. The second one is called the 2-Depot CARP on a line, where there are only two depots located at the two end points of the line graph. As mentioned in the introduction, the CVRP on a half-line cannot be approximated within ratio 3/2 unless $P = NP$[23].

In the following, we first consider the equal-demand case. Archetti et al.[24] showed that the equal-demand CVRP on a half-line can be solved in $O(n)$ time by the longest distance rule. That is, the first vehicle serves the farthest Q vertices. After that the second vehicle serves the farthest Q unserved vertices, and so on. The last vehicle may serve less than Q vertices. Clearly, the same rule can also solve the equal-demand CARP on a half-line where the demands are located at the edges instead of the vertices.

As for the equal-demand 2-Depot CARP on a line, we observe that there is an optimal solution, such that the routes of the vehicles from depot d_1 can never intersect with the routes of the vehicles from depot d_2 (but may have at most one common vertex). This can be verified by a simple exchange argument. As a consequence, the problem can be reduced to optimally partition the line into two edge disjoint half-lines and solve two subproblems defined on these two half-lines, separately. Since there are at most n such partitions and the half-line problem is solvable in $O(n)$ time. Thus, the equal-demand 2-Depot CARP on a line can be solved in $O(n^2)$ time.

As mentioned before, the equal-demand MCARP on a line can be reduced to the equal-demand CARP on a half-line and the equal-demand 2-Depot CARP on a line. Therefore, we have the following result.

Theorem 4 The equal-demand MCARP on a line can be solved in $O(n^2)$ time.

Now we consider the general case for the MCVRP on a line. We assume that $d(v) \leqslant Q$ for any $v \in V$, otherwise there exists no feasible solution. As before, we need only focus on the CVRP on a half-line and the 2-Depot CVRP on a line. For the CVRP on a half-line, Wu and Lu[23] gave a 5/3-approximation algorithm.

To devise an approximation algorithm for the 2-Depot CVRP on a line, we first show how to solve optimally a relaxed problem, called the split-delivery 2-Depot CVRP on a line, where the demands of the vertices can be split and satisfied by different routes of the vehicles. By replacing each vertex v_i with a path consisting of $d(v_i)$ unit-demand vertices $v_i^1, v_i^2, \ldots, v_i^{d(v_i)}$ connected by $d(v) - 1$ zero-cost edges in $L = (V, E)$, we can derive a line graph $L' = (V', E')$ with $V' = \bigcup_{i=1}^{n}\{v_i^1, v_i^2, \ldots, v_i^{d(v_i)}\}$ and

$$E' = (\bigcup_{i=1}^{n}\{\{v_i^1, v_i^2\}, \{v_i^2, v_i^3\}, \ldots, \{v_i^{d(v_i)-1}, v_i^{d(v_i)}\}\}) \bigcup (\bigcup_{i=1}^{n-1}\{v_i^{d(v_i)}, v_{i+1}^1\}).$$

Then the split-delivery 2-Depot CVRP on a line defined on $L = (V, E)$ is equivalent to the equal-demand 2-Depot CVRP on a line defined on $L' = (V', E')$, which can be solved using the above-mentioned algorithm§. After that, we merge the vertices connected by zero-cost edges to obtain a set of m routes $C_1, C_2, \ldots, C_p, C_{p+1}, \ldots, C_m$ for the split-delivery problem, where $C_1, C_2, \ldots, C_p$ ($C_{p+1}, C_{p+2}, \ldots, C_m$) are the routes for the vehicles from the left (right) depot, such that the vertices served by C_j are on the left of those served by C_l for any $j < l$. Since the split-delivery problem is a relaxation of the original problem, we have $\sum_{j=1}^{m} c(C_j) \leqslant Z^*$. For each route C_j, only the first and the last vertex served by it may have split demands and the demand of each vertex is split at most once, since we use the longest distance rule and $d(v) \leqslant Q$ by assumption. Thus, for each $j = p, p-1, \ldots, 1$, we can add a route C_j' starting from the left depot to serve solely the first vertex of C_j and remove the demands of the first vertex and the last vertex (if $2 \leqslant j \leqslant p$) of C_j to generate another route C_j''. For each $j = p+1, p+2, \ldots, m$, we add a route C_j' starting from the right depot to serve solely the first vertex of C_j and remove the demands of the first vertex and the last vertex (if $p+1 \leqslant j \leqslant m-1$) of C_j to derive another route C_j''. Clearly, $\max\{c(C_j'), c(C_j'')\} \leqslant c(C_j)$ for any $j = 1, 2, \ldots, m$. Moreover, one can see that the $2m$ new routes $C_1', C_1'', C_2', C_2'', \ldots, C_m', C_m''$ form a feasible solution of the original problem whose cost is

$$\sum_{j=1}^{m}(c(C_j') + c(C_j'')) \leqslant 2\sum_{j=1}^{m} c(C_j) \leqslant 2Z^*.$$

To sum up, we obtain the following result.

Theorem 5 The MCVRP on a line admits a 2-approximation algorithm.

7 Experiment

In this section, we conduct numerical experiments on our algorithms for the MRPP. We test the algorithms on randomly generated instances to investigate the average performance, which is then compared with the theoretical worst-case performance. The experiments are performed on a desktop computer with 3.6 GHz CPU and 16 GB RAM. All the algorithms are implemented with VS 2022 (C++ 17). Besides, we use the COIN-OR LEMON library for basic graph operations. The source code can be found at https://github.com/agamemnon314/mdcarp.

7.1 Data generation

Given the number $|D|$ of depots and the number $|R|$ of required edges, we generate the random instances by the following steps:

Step 1: Generate a complete graph with $2(|D| + |R|)$ vertices;

Step 2: Randomly select $|D|$ vertices as depots and $|R|$ edges R^0;

Step 3: For each edge $\{u, v\} \in R^0$, split the graph into three new edges $\{u, u_1\}$, $\{u_1, v_1\}$, and $\{v_1, v\}$, and select the middle one as a required edge;

Step 4: For each edge e, assign an integer edge cost $c(e) \sim U(0, 100)$;

Step 5: Construct a new complete graph $G = (V, E)$ with $V = V(R) \cup D$ and the edge cost between any two vertices is defined as the length of the shortest path in the graph obtained in Step 4.

§ One may notice that this transformation may yield only a pseudo-polynomial algorithm for the split-delivery problem since there are $\sum_{v \in V} d(v)$ possible partitions of the line graph. However, we will show that we need only consider $O(n^2)$ partitions to find the optimal solution for the split-delivery problem. Given any partition π of the line graph, which can be represented by an edge $\{u, v\} \in E'$. Let $C_1, C_2, \ldots, C_p$ ($C_{p+1}, C_{p+2}, \ldots, C_m$) be the routes generated by the longest distance rule for the split-delivery problem defined on the half-line between the left end vertex and u (the right end vertex and v), such that the vertices served by C_j are on the left of those served by C_l for any $j < l$. By properly shifting the edge $\{u, v\}$ to the right we can obtain another partition π' whose corresponding routes are $C_1', C_2', \ldots, C_p'$ ($C_{p+1}', C_{p+2}', \ldots, C_{m'}'$) for the vehicles from the left (right) depot such that $m' \leqslant m$ and $\sum_{j=1}^{m'} c(C_j') \leqslant \sum_{j=1}^{m} c(C_j)$. Moreover, the routes $C_1', C_2', \ldots, C_p', C_{p+1}', C_{p+2}', \ldots, C_{m'}'$ also satisfy at least one of the following conditions: (1) C_1' serves exactly Q vertices; (2) there exists some j with $1 \leqslant j \leqslant p$, such that the first vertex served in C_j is $v_i^{d(v_i)}$ for some i; (3) there exists some j with $p+1 \leqslant j \leqslant m'$, such that the first vertex served in C_j is v_i^1 for some i. Clearly, the number of partitions π' whose corresponding routes satisfy at least one of these three conditions is at most $O(n^2)$. To sum up, the split-delivery problem can be solved in $O(n^3)$ time.

7.2 Shortcut procedure

We introduce a shortcut procedure to further improve the numerical performance of Algorithm *MRPP*1 and Algorithm *MRPP*2. Let C_{sol} be the better solution returned by these two algorithms. We perform the following steps to obtain a shortcut solution.

Go through each walk in C_{sol} one by one, let $e = \{v, w\}$ be the current visiting edge and $e' = \{u, v\}$ be the last visited edge, we consider the following cases:

Case 1: If both e and e' are non-required edges, then we make a shortcut by replacing e and e' by a new edge $e'' = \{u, w\}$;

Case 2: If e is a non-required edge and e' is a required edge already visited, then we make a shortcut by replacing e and e' by a new edge $e'' = \{u, w\}$;

Case 3: For all the other cases, no shortcut is performed.

Note that the walks after shortcut still form a feasible solution. And the total edge cost will not increase due to the triangular inequality.

7.3 Lower bounds

For large-scale MRPP instances, it is hard to calculate the optimal value. Thus, we use some theoretical lower bound on the optimal value instead.

Let C_{sol} be the solution before the shortcut, and $\hat{C}_{sol}$ be the solution after the shortcut.

By Formula (3), we have

$$(2 - \frac{1}{2k+1})c(C^*) \geqslant c(C_{sol}),$$

$$\frac{c(\hat{C}_{sol})}{(2 - \frac{1}{2k+1})c(C^*)} \leqslant \frac{c(\hat{C}_{sol})}{c(C_{sol})},$$

$$\frac{c(\hat{C}_{sol})}{c(C^*)} \leqslant (2 - \frac{1}{2k+1})\frac{c(\hat{C}_{sol})}{c(C_{sol})} \tag{4}$$

In fact, this bound for the optimal solution can be further improved. Because we can take advantage of the actual value of the trees and matchings and required edges used, which are unknown when analyzing the approximation ratio.

Let $\alpha = 2 - 1/k$. By Formula (1), we have

$$c(C^*) \geqslant \frac{c(F) + c(M)}{\alpha} + c(R).$$

To avoid confusion, we denoted the minimum cost matching in Algorithm $MRPP2$ by M'. By the proof of Lemma 5, we have $c(M') \leqslant c(C^*) - c(R)$ and $c(E') \leqslant c(C^*) - c_R(C^*) \leqslant c(C^*) - c(R)$. Thus,

$$c(C^*) \geqslant \max\{c(M'), c(E')\} + c(R),$$

$$c(C^*) \geqslant \max\{\frac{c(F) + c(M)}{\alpha}, c(M'), c(E')\} + c(R).$$

Therefore, it holds that

$$\frac{c(C_{sol})}{c(C^*)} \leqslant \frac{c(C_{sol})}{\max\{\frac{c(F)+c(M)}{\alpha}, c(M'), c(E')\} + c(R)} \tag{5}$$

and

$$\frac{c(\hat{C}_{sol})}{c(C^*)} \leqslant \frac{c(\hat{C}_{sol})}{\max\{\frac{c(F)+c(M)}{\alpha}, c(M'), c(E')\} + c(R)} \tag{6}$$

7.4 Experiment results

Table 1 shows the upper bounds of $\frac{c(C_{sol})}{c(C^*)}$ and $\frac{c(\hat{C}_{sol})}{c(C^*)}$ on instances of different sizes, where k is the number of depots and n is the number of required edges. $\hat{ub} = 2 - 1/(2k+1)$ is the theoretical upper bound by Lemma 5.

$$ub_1 = \frac{c(C_{sol})}{\max\{\frac{c(F)+c(M)}{\alpha}, c(M'), c(E')\} + c(R)}$$

is the upper bound before the shortcut in Formula (5) and

$$ub_2 = \frac{c(\hat{C}_{sol})}{\max\{\frac{c(F)+c(M)}{\alpha}, c(M'), c(E')\} + c(R)}$$

is the upper bound after the shortcut in Formula (6). The value of each entry represents the average solution value of 100 random instances with a given size.

Table 1 Average upper bounds on the approximation ratio.

n	$k=2$			$k=5$			$k=10$			$k=20$		
	ub_1	ub_2	$\hat{ub}$	ub_1	ub_2	$\hat{ub}$	ub_1	ub_2	$\hat{ub}$	ub_1	ub_2	$\hat{ub}$
50	1.48	1.13	1.8	1.54	1.15	1.91	1.52	1.14	1.95	1.54	1.14	1.98
100	1.46	1.14	1.8	1.49	1.12	1.91	1.52	1.12	1.95	1.51	1.12	1.98
150	1.44	1.13	1.8	1.49	1.12	1.91	1.52	1.13	1.95	1.52	1.13	1.98
200	1.45	1.12	1.8	1.49	1.12	1.91	1.52	1.13	1.95	1.52	1.13	1.98
250	1.44	1.12	1.8	1.50	1.13	1.91	1.51	1.12	1.95	1.51	1.12	1.98
300	1.44	1.12	1.8	1.49	1.12	1.91	1.51	1.12	1.95	1.51	1.12	1.98
350	1.45	1.12	1.8	1.49	1.12	1.91	1.50	1.12	1.95	1.51	1.12	1.98
400	1.44	1.12	1.8	1.48	1.12	1.91	1.51	1.12	1.95	1.51	1.12	1.98
450	1.44	1.12	1.8	1.48	1.12	1.91	1.50	1.12	1.95	1.51	1.12	1.98
500	1.44	1.12	1.8	1.49	1.12	1.91	1.50	1.12	1.95	1.51	1.12	1.98

By comparing the values of columns ub_1, ub_2, and $\hat{ub}$, we can find that the performance of our algorithm is much better than the theoretical upper bound on average. For instances with a larger number of required edges, ub_2 is hardly affected by the number of depots. By comparing the values of columns ub_1 and ub_2, we can see that the shortcut procedure has a great effect on the performance of the algorithm for the MRPP. Note that the only difference between ub_1 and ub_2 is the shortcut procedure.

7.5 Sensitivity analysis

In this section, we analyze how the performance of the algorithm is affected by different parameters of the inputs, which includes the number of depots, the number of required edges, and the ratio between the cost of required edges and non-required edges.

For the number of required edges, we calculate the average approximation ratio upper bounds for 100 random instances with $|R|$ ranging from 5 to 100. From Fig. 3, we can observe that the value ub_1 decreases quickly when the number of required edges increases from 5 to 20. After that, ub_1 converges to around 1.45 which is much less than the theoretical value. At the same time, we can see that ub_2 converges to around 1.1 and is always much better than ub_1.

For the number of depots, we compute the average approximation ratio upper bounds for 100 random instances with $|D|$ ranging from 2 to 20. As shown in Fig. 4, the solution without the shortcut (ub_1) is already better than the theoretical worst-case value. While the solution with the shortcut (ub_2) is always much better than the theoretical worst-case value and ub_1. Besides, we can observe that both ub_1 and ub_2 are hardly affected when the number of depots increases.

For the parameter of the ratio between the cost of non-required and required edges, we reckon the average approximation ratio upper bounds for 100 random instances with the cost ratio ranging from 0.1 to 10. As demonstrated in Fig. 5, when the ratio is closed to zero, ub_1 is close to the theoretical value. Both ub_1 and ub_2 decrease quickly with the increment of the cost ratio. Note that ub_2 almost converges to the optimal value, as expected, since both the solution of our algorithm and the optimal solution will not contain any required edge twice.

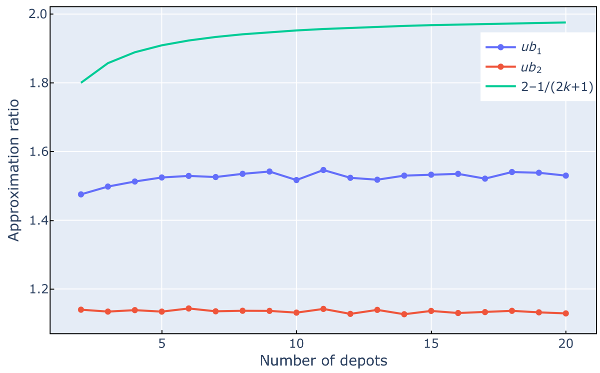


Fig. 4 Average approximation ratio according to the number of depots.

In summary, we find that the cost of the solutions before the shortcut is always close to the theoretical value from below. And the shortcut can help to improve the solution quality considerably. In addition, the key factor that affects the average solution quality after the shortcut is the ratio between required edge costs and non-required edge costs.

Acknowledgment

The authors are grateful to the anonymous referees for their valuable and constructive comments. This research was supported by the National Natural Science Foundation of China (Nos. 11671135, 11871213, and 11901255),

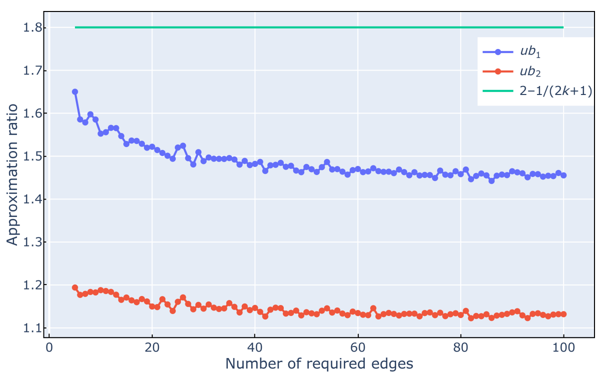


Fig. 3 Average approximation ratio according to the number of required edges.

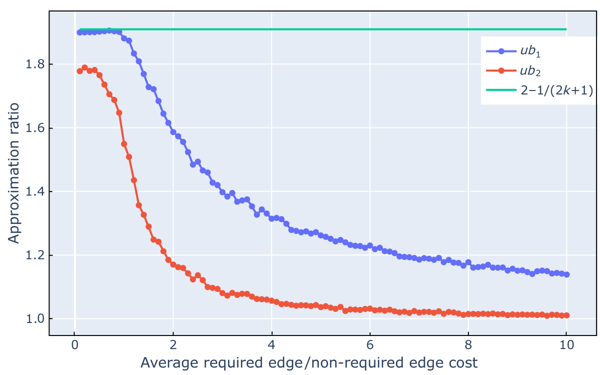


Fig. 5 Average approximation ratio according to the ratio between the cost of non-required and required edges.

and the Natural Science Foundation of Shanghai (No. 19ZR1411800).

References

[1] B. L. Golden and R. T. Wong, Capacitated arc routing problems, *Networks*, vol. 11, no. 3, pp. 305–315, 1981.

[2] H. A. Eiselt, M. Gendreau, and G. Laporte, Arc routing problems, part II: The rural postman problem, *Oper. Res.*, vol. 43, no. 3, pp. 399–414, 1995.

[3] J. Park and B. I. Kim, The school bus routing problem: A review, *Eur. J. Oper. Res.*, vol. 202, no. 2, pp. 311–319, 2010.

[4] E. Fernández, D. Fontana, and M. G. Speranza, On the collaboration uncapacitated arc routing problem, *Comput. Oper. Res.*, vol. 67, pp. 120–131, 2016.

[5] A. Hertz, G. Laporte, and M. Mittaz, A Tabu search heuristic for the capacitated arc routing problem, *Oper. Res.*, vol. 48, no. 1, pp. 129–135, 2000.

[6] E. Fernández and J. Rodríguez-Pereira, Multi-depot rural postman problems, *TOP*, vol. 25, no. 2, pp. 340–372, 2017.

[7] A. Kansou and A. Yassine, A two ant colony approaches for the multi-depot capacitated arc routing problem, in *Proc. of the 2009 Int. Conf. Computers & Industrial Engineering*, Troyes, France, 2009, pp. 1040–1045.

[8] H. Hu, T. Liu, N. Zhao, Y. Zhou, and D. Min, A hybrid genetic algorithm with perturbation for the multi-depot capacitated arc routing problem, *J. Appl. Sci.*, vol. 13, no. 16, pp. 3239–3244, 2013.

[9] H. Chen, T. Cheng, and J. Shawe-Taylor, A balanced route design for min-max multiple-depot rural postman problem (MMMDRPP): A police patrolling case, *Int. J. Geogr. Inform. Sci.*, vol. 32, no. 1, pp. 169–190, 2018.

[10] J. Zhao and F. Zhu, A multi-depot vehicle-routing model for the explosive waste recycling, *Int. J. Prod. Res.*, vol. 54, no. 2, pp. 550–563, 2016.

[11] E. Fernández, G. Laporte, and J. Rodríguez-Pereira, A branch-and-cut algorithm for the multidepot rural postman problem, *Transport. Sci.*, vol. 52, no. 2, pp. 353–369, 2018.

[12] D. Krushinsky and T. van Woensel, An approach to the asymmetric multi-depot capacitated arc routing problem, *Eur. J. Oper. Res.*, vol. 244, no. 1, pp. 100–109, 2015.

[13] T. Liu, Z. Jiang, and N. Geng, A genetic local search algorithm for the multi-depot heterogeneous fleet capacitated arc routing problem, *Flex. Serv. Manuf. J.*, vol. 26, no. 4, pp. 540–564, 2014.

[14] M. Haimovich and A. H. G. R. Kan, Bounds and heuristics for capacitated routing problems, *Math. Oper. Res.*, vol. 10, no. 4, pp. 527–542, 1985.

[15] N. Christofides, Worst-case analysis of a new heuristic for the travelling salesman problem, *Operations Research Forum*, vol. 3, p. 20, 2022.

[16] R. van Bevern and V. A. Slugina, A historical note on the 3/2-approximation algorithm for the metric traveling salesman problem, *Hist. Math.*, vol. 53, pp. 118–127, 2020.

[17] M. Haimovich, A. H. G. R. Kan, and L. Stougie, Analysis of heuristics for vehicle routing problems, in *Vehicle Routing: Methods and Studies*, B. L. Golden and A. A. Assad, eds. Amsterdam, The Netherlands: Elsevier, 1988, pp. 47–61.

[18] K. Altinkemer and B. Gavish, Technical note—Heuristics for delivery problems with constant error guarantees, *Transport. Sci.*, vol. 24, no. 4, pp. 294–297, 1990.

[19] K. Altinkemer and B. Gavish, Heuristics for unequal weight delivery problems with a fixed error guarantee, *Oper. Res. Lett.*, vol. 6, no. 4, pp. 149–158, 1987.

[20] J. Blauth, V. Traub, and J. Vygen, Improving the approximation ratio for capacitated vehicle routing, in *Proc. 22nd Conf. Integer Programming and Combinatorial Optimization*, Atlanta, GA, USA, 2021, pp. 1–14.

[21] Z. Friggstad, R. Mousavi, M. Rahgoshay, and M. R. Salavatipour, Improved approximations for capacitated vehicle routing with unsplittable client demands, in *Proc. 23rd Conf. Integer Programming and Combinatorial Optimization*, Eindhoven, The Netherlands, 2022, pp. 251–261.

[22] M. Labbé, G. Laporte, and H. Mercure, Capacitated vehicle routing on trees, *Oper. Res.*, vol. 39, no. 4, pp. 616–622, 1991.

[23] Y. Wu and X. Lu, Capacitated vehicle routing problem on line with unsplittable demands, *J. Comb. Optim.*, vol. 44, no. 3, pp. 1953–1963, 2022.

[24] C. Archetti, D. Feillet, M. Gendreau, and M. G. Speranza, Complexity of the VRP and SDVRP, *Transp. Res. Part C: Emerg. Technol.*, vol. 19, no. 5, pp. 741–750, 2011.

[25] K. Jansen, Bounds for the general capacitated routing problem, *Networks*, vol. 23, no. 3, pp. 165–173, 1993.

[26] R. van Bevern, S. Hartung, A. Nichterlein, and M. Sorge, Constant-factor approximations for Capacitated Arc Routing without triangle inequality, *Oper. Res. Lett.*, vol. 42, no. 4, pp. 290–292, 2014.

[27] S. Wøhlk, An approximation algorithm for the capacitated arc routing problem, *Open Oper. Res. J.*, vol. 2, no. 1, pp. 8–12, 2008.

[28] C. L. Li and D. Simchi-Levi, Worst-case analysis of heuristics for multidepot capacitated vehicle routing Problems, *ORSA J. Comput.*, vol. 2, no. 1, pp. 64–73, 1990.

[29] B. Bhattacharya and Y. Hu, Approximation algorithms for the multi-vehicle scheduling problem, in *Proc. 21st Int. Symp. Algorithms and Computation*, Jeju Island, Republic of Korea, 2010, pp. 192–205.

[30] A. Corberán and G. Laporte, *Arc Routing: Problems, Methods, and Applications*. Philadelphia, PA, USA: SIAM, 2015.

[31] W. Yu, Z. Liu, and X. Bao, Approximation algorithms for some min-max postmen cover problems, *Ann. Oper. Res.*, vol. 300, no. 1, pp. 267–287, 2021.

[32] Z. Xu, L. Xu, and B. Rodrigues, An analysis of the extended Christofides heuristic for the k-depot TSP, *Oper. Res. Lett.*, vol. 39, no. 3, pp. 218–223, 2011.

Wei Yu received the PhD degree from East China University of Science and Technology, China in 2010, where he is currently an associate professor. His research interests include combinatorial optimization, approximation algorithm, routing scheduling problem, vehicle routing problem, and algorithmic game theory.

Yujie Liao received the BEng degree from Anqing Normal University, China in 2019. Currently she is a master student at East China University of Science and Technology. Her research interests include combinatorial optimization, approximation algorithm, and vehicle routing problem.

Yichen Yang received the PhD degree from East China University of Science and Technology, China in 2019. Currently he is a data scientist at Sabre Lab Research Team, Sabre Inc., USA. His research interests include combinatorial optimization, approximation algorithm, flight scheduling problem, routing problem, and pricing problem.