Improved Approximation Algorithms for Capacitated Location Routing

Jingyang Zhao, **Mingyu Xiao*** and **Shunwang Wang**
University of Electronic Science and Technology of China, Chengdu, China
{jingyangzhao1020, myxiao, wangshunwang123}@gmail.com

Abstract

The Capacitated Location Routing Problem is an important planning and routing problem in logistics, which generalizes the capacitated vehicle routing problem and the uncapacitated facility location problem. In this problem, we are given a set of depots and a set of customers where each depot has an opening cost and each customer has a demand, and we need to use minimum cost to open some depots and route capacitated vehicles in the opened depots to satisfy all customers' demand. In this paper, we propose a 4.169-approximation algorithm for this problem, improving the best-known 4.38-approximation ratio (Transportation Science 2013). Moreover, if the demand of each customer is allowed to be delivered by multiple tours, we propose a more refined 4.092-approximation algorithm. Experimental study on benchmark instances shows that the quality of our computed solutions is better than that of the previous algorithm and is also much closer to optimality than the provable approximation factor.

1 Introduction

In the fields of logistics, vehicle routing and facility location are two major problems that have been widely studied in both theory and application. Given a set of depots, vehicle routing aims to route capacitated vehicles in the depots to deliver goods for customers to satisfy their demand using the minimum cost. Facility location concerns the opening cost of depots and the connection cost between customers and the opened depots. Location routing can be seen as a combination of these two problems, and is more natural in real life. It involves first opening a set of depots at some cost, and then routing the vehicles from the opened depots. Location routing problems have been studied for decades since the idea of combining vehicle routing and facility location was introduced in [Von Boventer, 1961; Maranzana, 1964; Webb, 1968]. Recent surveys of location routing problems can be found in [Prodhon and Prins, 2014; Drexl and Schneider, 2015].

In Capacitated Location Routing (CLR), we are given an undirected complete graph $G = (V \cup U, E, w, \phi, d, k)$, where V is the set of customers, U is the set of depots (or facilities), and there is one travel cost function $w : E \to \mathbb{R}_{\geq 0}$ on edges, an opening cost function $\phi : U \to \mathbb{R}_{\geq 0}$ on depots, and a demand function $d : V \to \mathbb{R}_{\geq 0}$ on customers. Each depot $u \in U$ has an opening cost $\phi(u)$ and contains an unbounded fleet of vehicles with the same capacity $k > 0$, each customer $v \in V$ has a demand $d(v) > 0$, and we need to determine a set of depots $O \subseteq U$ to open and a set of tours $\mathcal{I}$ such that (1) each tour starts and ends at the same (opened) depot, (2) each tour delivers at most k of demand to customers on the tour, and (3) the union of tours in $\mathcal{I}$ satisfies all customers' demand. The total cost is defined as $\sum_{I \in \mathcal{I}} w(I) + \sum_{u \in O} \phi(u)$, where $w(I)$, the cost of tour I, is defined to be the total cost of edges in I, i.e., $w(I) = \sum_{e \in I} w(e)$. We consider two variants of CLR: *splittable* and *unsplittable*, where the demand of each customer is allowed to be delivered by multiple tours (resp., only one tour) in splittable (resp., unsplittable) CLR. This definition also captures the case where each tour incurs an extra depot-dependent fixed cost [Tuzun and Burke, 1999; Barreto *et al.*, 2007], i.e., each vehicle departing from depot $u \in U$ incurs an additional cost of $F_u \in \mathbb{R}_{\geq 0}$. This can be represented by adding $F_u/2$ to the cost of all edges incident to u, as each tour originating at u uses only two of these edges.

CLR generalizes many famous NP-hard problems. If $\phi \equiv 0$, we can open all depots in U at no cost, and in this case CLR is the Multidepot Capacitated Vehicle Routing Problem (MCVRP). If $\phi \equiv 0$ and $k = \infty$, CLR becomes the metric m-depot traveling salesman problem (TSP). Hence, CLR also includes the (single depot) Capacitated Vehicle Routing Problem (CVRP) and metric TSP as special cases. Moreover, CLR with k being the greatest common divisor of the demands also generalizes Uncapacitated Facility Location (UFL), where we choose to open some depots and assign each customer v to its nearest opened depot u with a cost of $d(v)w(v, u)$. These problems have been extensively studied both in terms of approximation algorithms and experimental algorithms [Toth and Vigo, 2014; Montoya-Torres *et al.*, 2015; An *et al.*, 2017; Zhang *et al.*, 2015; Xin *et al.*, 2021; Zhou *et al.*, 2023].

1.1 Related Work

We focus on approximation algorithms. Next, we give a brief review of literature on TSP, Vehicle Routing, UFL, and CLR.

*Contact Author

TSP. For metric TSP, the Christofides-Serdyukov algorithm [1976; 1978] is a well-known 1.5-approximation algorithm, and the ratio has been recently improved to $1.5-10^{-36}$ by Karlin *et al.* [2021; 2023]. For metric m-deopot TSP, Rathinam *et al.* [2007] proposed a simple 2-approximation algorithm, and Xu *et al.* [2011] improved the ratio to $2-1/m$. There are also some works for the case that m is fixed [Xu and Rodrigues, 2015; Traub *et al.*, 2022; Deppert *et al.*, 2023].

Vehicle Routing. For the case of single depot, Haimovich and Kan [1985] proposed a 2.5-approximation algorithm for splittable CVRP, and Altinkemer and Gavish [1987] proposed a 3.5-approximation algorithm for unsplittable CVRP. For the case of multidepot, Li and Simchi-Levi [1990] proposed a 4-approximation algorithm for splittable MCVRP, and Harks *et al.* [2013] proposed a 4-approximation algorithm for unsplittable MCVRP. These results got improved only very recently. For CVRP, Blauth *et al.* [2022] improved the ratio to $2.5-\frac{1}{3000}$ for the splittable case, and Friggstad *et al.* [2022] improved the ratio to about 3.164 for the unsplittable case. For MCVRP, Zhao and Xiao [2023] obtained a ratio of $4-\frac{1}{1500}$ for the splittable case and a ratio of $4-\frac{1}{50000}$ for the unsplittable case.

UFL. UFL has a rather rich history on approximation algorithms (see the book [Williamson and Shmoys, 2011]), where many new techniques were developed. We mention the following results: Jain *et al.* [2003] proposed a practical 1.861-approximation algorithm based on the greedy method with a running time of $O(nm\log nm)$; Byrka and Aardal [2010] proposed a 1.5-approximation algorithm and a bifactor approximation algorithm by modifying the LP rounding method in [Chudak and Shmoys, 2003]; The current best result is a 1.488-approximation algorithm [Li, 2013].

CLR. Since CLR is more challenging, there are only a few results on approximation algorithms. Harks *et al.* [2013] proposed a 4.38-approximation algorithm for both unsplittable and splittable CLR, and showed that unsplittable CLR cannot be approximated better than 1.5 unless $P=NP$. They also extended their algorithm to derive approximation algorithms for three settings of CLR: prize-collecting, grouping, and cross-docking. Recently, Heine *et al.* [2023] proposed a bifactor approximation algorithm for a variant of CLR, where each depot is also capacitated.

1.2 Our Results

In this paper, we propose two improved approximation algorithms for CLR. The first, denoted as Tree-Alg, is a 4.169-approximation algorithm for both unsplittable and splittable CLR, which improves the previous 4.38-approximation algorithm [Harks *et al.*, 2013]. Note that the previous approximation ratio has been kept for a decade. The second, denoted as Path-Alg, achieves a better ratio of 4.092 for splittable CLR. The details of our improvements are as follows.

First, we obtain two stronger lower bounds for CLR, which are essential to our results. Second, by refining the previous 4.38-approximation algorithm, we obtain our Tree-Alg, and the main idea is to obtain tours by splitting trees. At last, motivated by the cycle-splitting method used in vehicle routing problems, we develop our Path-Alg, focusing on splitting paths. Although paths are simpler than trees, the analysis uses more techniques. The main reason is that we may use an approximation algorithm of TSP to compute paths which are more expensive, and then we cannot even obtain a better approximation ratio by a straightforward analysis.

In practice, our algorithms are easy to implement and run very fast. Experimental study on benchmark instances shows that the quality of our computed solutions is better than that of the previous algorithm and is also much closer to optimality than the provable approximation factor.

1.3 Paper Organization

The remaining parts of the paper are organized as follows. In Section 2, we introduce some notations and the formal definitions of CLR and UFL. In Section 3, we propose two stronger lower bounds for CLR. In Section 4, we give our Tree-Alg for both unsplittable and splittable CFL, and in Section 5, we give our Path-Alg for splittable CFL. In Section 6, we present the experimental study of our algorithms. At last, we make the concluding remarks in Section 7.

Due to limited space, the proofs of lemmas and theorems marked with ♣ were omitted and they can be found in the full version of this paper.

2 Preliminaries

In CLR, we use $G=(V\cup U,E,w,\phi,d,k)$ to denote the input complete graph. The cost function w is a metric function, i.e., it is symmetric and satisfies the triangle inequality. In UFL, the input graph is the same as CLR, except for the absence of the parameter k, and we use $G=(V\cup U,E,w,\phi,d)$ to denote it.

For any function $f:X\rightarrow\mathbb{R}_{\geq 0}$, we always define $f(Y)=\sum_{x\in Y}f(x)$ for any $Y\subseteq X$. For any subgraph S of G, we use $V(S)$, $U(S)$, and $E(S)$ to denote the customer set, the depot set, and the edge set of S, respectively. Furthermore, we define $w(S)=\sum_{e\in E(S)}w(e)$, $\phi(S)=\sum_{u\in U(S)}\phi(u)$, and $d(S)=\sum_{v\in V(S)}d(v)$.

A *walk* in a graph, denoted by $v_1v_2\dots v_l$, is a sequence of vertices $v_1,v_2,\dots,v_l$, where a vertex may appear more than once and each consecutive pair of vertices is connected by an edge. A *path* in a graph is a walk where no vertex appears more than once. The first and the last vertices of a path are referred to as its *terminals*. A *closed* walk is a walk where the first and the last vertices are the same, and a *cycle* is a walk where only the first and the last vertices are the same. Given a closed walk, we can skip repeated vertices along the walk to get a cycle, and such an operation is called *shortcutting*. During shortcutting, if a specific vertex v is consistently skipped, we refer to it as *shortcutting* v.

A *constrained spanning forest* in G is a forest that spans (i.e., covers) all vertices in V and each tree in it contains only one depot. A *constrained spanning path-packing* in G is a set of vertex-disjoint paths that spans all vertices in V, each depot is contained in at most one path in it, and each path in it contains only one depot and the depot is one of its terminals. A *constrained spanning cycle-packing* in G is a set of vertex-disjoint cycles or paths that spans all vertices in V, each cycle in it contains only one depot, and each path in it contains only two depots and the depots are its terminals.

A *tour* is a walk that starts and ends at the same depot and does not pass through any other depot. We may only consider simple and minimal tours with each containing only one depot, i.e., each tour is a cycle containing only one depot.

2.1 Formal Problem Definition

Definition 1 (**CLR**). *Given an undirected complete graph* $G = (V \cup U, E, w, \phi, d, k)$*, we need to find a set of depots* $O \subseteq U$ *and a set of tours* $\mathcal{I}$ *with a demand assignment* $x : V \times \mathcal{I} \to \mathbb{R}_{\geq 0}$ *such that*

- $U(I) \cap O \neq \emptyset$ *for any* $I \in \mathcal{I}$*,*
- $\sum_{v \in V(I)} x_{vI} \leq k$ *for any* $I \in \mathcal{I}$*,*
- $\sum_{v \in V \setminus V(I)} x_{vI} = 0$ *for any* $I \in \mathcal{I}$*,*
- $\sum_{I \in \mathcal{I}} x_{vI} = d(v)$ *for any* $v \in V$*,*

and $\sum_{I \in \mathcal{I}} w(I) + \sum_{u \in O} \phi(u)$ *is minimized.*

In the above definition, if each customer is allowed to be satisfied by using multiple tours, we call it as *splittable* CLR. For *unsplittable* CLR, it requires that each customer must be satisfied by using only one tour. Clearly, unsplittable CLR admits a feasible solution only if it holds $d(v) \leq k$ for any $v \in V$. In any solution $(O, \mathcal{I})$, we will refer to $\sum_{I \in \mathcal{I}} w(I)$ as the *routing cost* and $\sum_{u \in O} \phi(u)$ as the *opening cost*.

Definition 2 (**UFL**). *Given an undirected complete graph* $G = (V \cup U, E, w, \phi, d)$*, we need to find a set of depots* $O \subseteq U$ *such that the cost* $\sum_{v \in V} d(v) \cdot \min_{o \in O} w(o, v) + \sum_{u \in O} \phi(u)$ *is minimized, i.e., for each* $v \in V$ *we directly assign* $d(v)$ *of demand to* v *from its nearest depot* o *in* O *with a connection cost of* $d(v)w(o, v)$*.*

In Definition 2, $\sum_{v \in V} d(v) \cdot \min_{o \in O} w(o, v)$ is called the *connection cost* and $\sum_{u \in O} \phi(u)$ is called the *opening cost*.

3 Lower Bounds

In this section, we prove two lower bounds of the optimal solution that holds for both unsplittable and splittable CLR. These bounds are crucial for us to prove the approximation ratio.

Given an instance $G = (V \cup U, E, w, \phi, d, k)$ of CLR, we construct an UFL instance $G = (V \cup U, E, \widetilde{w}, \widetilde{\phi}, d)$ as follows. The sets of depots and customers with their demand remain the same as in CLR, but we set the costs of edges to $\widetilde{w} \coloneqq (2/k)w$ and the costs of depots to $\widetilde{\phi} \coloneqq \alpha \cdot \phi$, where α is a fixed constant that will be defined later. Note that α is an important parameter newly proposed by us. In contrast, Harks *et al.* [2013] only focused on $\widetilde{\phi} \coloneqq \phi$.

Let OPT (resp., OPT′) denote the cost of an optimal solution for CLR (resp., UFL). We let ψ^* and ϕ^* to denote the routing cost of vehicles and the opening cost of facilities with respect to the optimal solution of CLR, respectively. Note that we have $\text{OPT} = \psi^* + \phi^*$. We have the following bound.

Lemma 3 (♣)**.** *It holds that* $\mathit{OPT}' + (1 - \alpha) \cdot \phi^* \leq \mathit{OPT}$*.*

Our bound in Lemma 3 is more general since the previous paper [Harks *et al.*, 2013] only obtained the bound under $\alpha = 1$. Next, we consider the second lower bound.

Algorithm 1 An improved approximation algorithm for unsplittable and splittable CLR (Tree-Alg)

Input: An instance of CLR.
Output: A feasible solution to CLR.

1: Create an UFL instance with edge costs $\widetilde{w} = (2/k)w$ and depot costs $\widetilde{\phi} = \alpha \cdot \phi$ as in Lemma 3.
2: Apply Byrka and Aardal's bifactor approximation algorithm [Byrka and Aardal, 2010] with a parameter of $\gamma > 0$ on the UFL instance, and let O_1 be the set of depots opened in the resulting UFL solution.
3: Compute a constrained spanning forest $\mathcal{T}$ in G as in Lemma 4, and let $O_2 = U(\mathcal{T})$ be the set of depots contained in some $T \in \mathcal{T}$.
4: Open all depots in $O \coloneqq O_1 \cup O_2$.
5: Obtain a set of feasible tours $\mathcal{I}$ by calling the tree-splitting procedure in Algorithm 2.
6: Return $(O, \mathcal{I})$.

It was shown in [Harks *et al.*, 2013] that one can use polynomial time to find a constrained spanning forest $\mathcal{T}$ in G such that $2w(\mathcal{T}) + \phi(\mathcal{T}) \leq 2 \cdot \text{OPT}$. We propose a better result.

Lemma 4 (♣)**.** *There is a polynomial-time algorithm to find a constrained spanning forest* $\mathcal{T}$ *such that*

$$2w(\mathcal{T}) + \phi(\mathcal{T}) + \phi^* \leq 2 \cdot \mathit{OPT}.$$

4 General CLR

In this section, we introduce our Tree-Alg, which is a 4.169-approximation algorithm for unsplittable and splittable CLR.

4.1 The Algorithm

Our algorithm uses the framework of the 4.38-approximation algorithm in [Harks *et al.*, 2013]. It uses a bifactor approximation algorithm to compute a solution (i.e., a set of depots) for the constructed instance of UFL in Lemma 3. Then, it computes the constrained spanning forest $\mathcal{T}$ in Lemma 4. Let O_1 be the set of opened depots in the solution of UFL and O_2 be the set of depots contained in $\mathcal{T}$. The algorithm will open all depots in $O \coloneqq O_1 \cup O_2$. Based on splitting the trees in $\mathcal{T}$ (the tree-splitting procedure in [Harks *et al.*, 2013]), the algorithm will return a feasible solution in polynomial time. The framework of the algorithm is shown in Algorithm 1. Note that the 4.38-approximation algorithm in [Harks *et al.*, 2013] corresponds to our algorithm under $\alpha = 1$.

Given a set of opened depots O and a constrained spanning forest $\mathcal{T}$, the tree-splitting procedure is to obtain a set of feasible tours using only depots in O, which is also equivalent to solving an instance of MCVRP. Note that it also works for the case of splittable demand. Hence, we may assume w.l.o.g. that a customer can have a demand of more than k. The main idea is to repeatedly find a sub-tree S of trees in $\mathcal{T}$ such that $k \geq d(S) > k/2$, and then construct a tour for (part of) customers in $V(S)$ by doubling edges in $E(S) \cup \{e_S\}$ and shortcutting, where e_S is a minimum weight edge between one customer in $V(S)$ and one depot in O.

The details are as follows. First, for each customer $v \in V$ with $d(v) > k$, we construct $\lceil \frac{d(v)}{k} \rceil$ tours for v by connecting v with its nearest opened depot in O and regard v as a zero-demand customer in the following. Then, we consider a tree $T_u \in \mathcal{T}$ rooted at $u \in O$, and satisfy all non-zero-demand customers in $V(T_u)$ by splitting T_u. Denote the sub-tree rooted at v and the set of v's children by T_v and Q_v, respectively.

- If $d(T_u) \leq k$, we construct a tour for all non-zero-demand customers in $V(T_u)$ by doubling all edges in $E(T_u)$ and then shortcutting. Note that $d(T_u)$ may have a demand of less than $k/2$, but T_u is special since it contains an opened depot.
- Else, we find a minimal sub-tree T_v such that $d(T_v) > k$ and $d(T_{v'}) \leq k$ for every $v' \in Q_v$. Consider sub-trees in $\mathcal{T}_v := \{v\} \cup \{T_{v'} \mid v' \in Q_v\}$. We greedily partition them into sets $\mathcal{T}_0, \ldots, \mathcal{T}_l$ such that $d(\mathcal{T}_i) \leq k$ for each i and $d(\mathcal{T}_i) > k/2$ for each $i > 0$. For each $\mathcal{T}_i$ with $i > 0$, we combine trees in $\mathcal{T}_i$ into a sub-tree S by adding the edges joining v and each tree in $\mathcal{T}_i$ and v (if $\{v\} \notin \mathcal{T}_i$). Then, we find a minimized cost edge e_S connecting one depot in O with one vertex in $V(S) \cup U(S)$. By doubling edges in $E(S) \cup \{e_S\}$ and shortcutting, we construct a tour for all non-zero-demand customers in $V(\mathcal{T}_i)$. At last, we update T_u by removing $V(S) \setminus \{v\}$ and $E(S)$ from T_u, and regard v as a zero-demand customer in the following if $\{v\} \in \mathcal{T}_i$.

The tree-splitting procedure used in Algorithm 1 is shown in Algorithm 2. The tours have the following properties.

Lemma 5. [Harks *et al.*, 2013]. *Given a set of opened depots O and a constrained spanning forest $\mathcal{T}$, the set of tours $\mathcal{I}$ returned by the tree-splitting procedure holds that $(O, \mathcal{I})$ is a feasible solution for unsplittable and splittable UFL, and $w(\mathcal{I}) \leq 2w(\mathcal{T}) + \sum_{v \in V}(4/k)d(v) \cdot \min_{u \in O} w(v, u)$.*

4.2 The Analysis

In this subsection, we show that by carefully setting the values of α and γ we can obtain a 4.169-approximation ratio.

For UFL with any constant $\gamma \geq 1.678$, Byrka and Aardal [2010] proposed a (bifactor) $(1 + 2e^{-\gamma}, \gamma)$-approximation algorithm. It returns a solution whose connection cost is at most $(1 + 2e^{-\gamma}) \cdot \psi_{LP}$ and whose opening cost is at most $\gamma \cdot \phi_{LP}$, where ψ_{LP} and ϕ_{LP} are the values of the connection cost and the opening cost of an initially computed optimal fractional LP solution, respectively. In Step 2 of Algorithm 1, we apply Byrka and Aardal's algorithm on the UFL instance to open a set of depots O_1. Hence, the connection cost and the opening cost in the solution satisfies $\sum_{v \in V} d(v) \cdot \min_{u \in O_1} \widetilde{w}(v, u)$ and $\widetilde{\phi}(O_1)$, respectively. Therefore, we have the following bounds.

Lemma 6. *It holds that $\sum_{v \in V} d(v) \cdot \min_{u \in O_1} \widetilde{w}(v, u) \leq (1 + 2e^{-\gamma}) \cdot \psi_{LP}$ and $\widetilde{\phi}(O_1) \leq \gamma \cdot \phi_{LP}$.*

Recall that OPT$'$ is the cost of an optimal solution on the UFL instance in Lemma 3. The cost of the optimal fractional LP solution is at most the cost of an optimal solution. Hence, we have $\psi_{LP} + \phi_{LP} \leq$ OPT$'$. By Lemma 3, we have

Lemma 7. *It holds that $\psi_{LP} + \phi_{LP} \leq OPT' \leq \psi^* + \alpha \cdot \phi^*$.*

Theorem 8. *For unsplittable and splittable CLR, Tree-Alg is a polynomial-time 4.169-approximation algorithm.*

Algorithm 2 The tree-splitting procedure for CLR

Input: An instance of CLR, a set of opened depots O, and a constrained spanning forest $\mathcal{T}$.
Output: A set of feasible tours $\mathcal{I}$ to CLR.

Initialize $\mathcal{I} = \emptyset$.
for all $v \in V$ with $d(v) > k$ **do**
  Construct $\lceil \frac{d(v)}{k} \rceil$ tours for v by connecting v with its nearest opened depot in O.
  Add the tours to $\mathcal{I}$ and regard v as a zero-demand customer in the following.
end for
for all $T_u \in \mathcal{T}$ **do**
  while $d(T_u) > k$ **do**
    Find $v \in V(T_u)$ with $d(T_v) > k$ and $d(T_{v'}) \leq k$ for each $v' \in Q_v$.
    Greedily partition trees in $\mathcal{T}_v := \{v\} \cup \{T_{v'} \mid v' \in Q_v\}$ into sets $\mathcal{T}_0, \ldots, \mathcal{T}_l$ such that $d(\mathcal{T}_i) \leq k$ for each i and $d(\mathcal{T}_i) > k/2$ for each $i > 0$.
    for $i \in \{1, \ldots, l\}$ **do**
      Combine trees in $\mathcal{T}_i$ into a sub-tree S by adding the edges joining v and each tree in $\mathcal{T}_i$, and v (if $\{v\} \notin \mathcal{T}_i$).
      Find an edge e_S with minimized cost connecting one depot in O with one vertex in $V(S) \cup U(S)$.
      Construct a tour for all non-zero-demand customers in $V(\mathcal{T}_i)$ by doubling edges in $E(S) \cup \{e_S\}$ and shortcutting.
      Add the tour to $\mathcal{I}$, update T_u by removing $V(S) \setminus \{v\}$ and $E(S)$ from T_u, and regard v as a zero-demand customer in the following if $\{v\} \in \mathcal{T}_i$.
    end for
  end while
  Construct a tour for all non-zero-demand customers in $V(T_u)$ by doubling edges in $E(T_u)$ and shortcutting.
  Add the tour to $\mathcal{I}$.
end for
Return $\mathcal{I}$.

Proof. By Algorithm 1, it returns a solution $(O, \mathcal{I})$ such that

$$
\begin{aligned}
& w(\mathcal{I}) + \phi(O) \\
& \leq 2w(\mathcal{T}) + \sum_{v \in V}(4/k)d(v) \cdot \min_{u \in O} w(v, u) + \phi(O_1) + \phi(O_2) \\
& \leq 2w(\mathcal{T}) + \sum_{v \in V} 2d(v) \cdot \min_{u \in O_1} \widetilde{w}(v, u) + \frac{1}{\alpha} \cdot \widetilde{\phi}(O_1) + \phi(\mathcal{T}) \\
& \leq 2\psi^* + \phi^* + 2(1 + 2e^{-\gamma}) \cdot \psi_{LP} + (1/\alpha) \cdot \gamma \cdot \phi_{LP},
\end{aligned}
$$

where the first inequality follows from $w(\mathcal{I}) \leq 2w(\mathcal{T}) + \sum_{v \in V}(4/k)d(v) \cdot \min_{u \in O} w(v, u)$ by Lemma 5 and $\phi(O) \leq \phi(O_1) + \phi(O_2)$, the second inequality follows from $\min_{u \in O} w(v, u) \leq \min_{u \in O_1} w(v, u)$, $\widetilde{w} = (2/k)w$, $\widetilde{\phi} = \alpha \cdot \phi$, and $\phi(O_2) = \phi(\mathcal{T})$, and the last inequality follows from OPT $= \psi^* + \phi^*$ and Lemmas 4 and 6.

Let $f_\gamma(\alpha) := \max\{2(1 + 2e^{-\gamma}), (1/\alpha) \cdot \gamma\}$. Since OPT $= \psi^* + \phi^*$, the approximation ratio is bounded by

$$
\begin{aligned}
& \frac{2\psi^* + \phi^* + f_\gamma(\alpha) \cdot (\psi_{LP} + \phi_{LP})}{\psi^* + \phi^*} \\
& \leq \frac{2\psi^* + \phi^* + f_\gamma(\alpha) \cdot (\psi^* + \alpha \cdot \phi^*)}{\psi^* + \phi^*} \\
& \leq \max\{2 + f_\gamma(\alpha), 1 + \alpha \cdot f_\gamma(\alpha)\},
\end{aligned}
$$

where the first inequality follows from Lemma 7.

Setting $\alpha = 1.461$ and $\gamma = 3.168$, we get $f_\gamma(\alpha) \leq 2.169$ and $\max\{2 + f_\gamma(\alpha), 1 + \alpha \cdot f_\gamma(\alpha)\} \leq 4.169$. Hence, the approximation ratio of Algorithm 1 is at most 4.169.

Since the bifactor approximation algorithm [Byrka and Aardal, 2010] and the tree-splitting procedure [Harks *et al.*, 2013] run in polynomial time, it is easy to see that Tree-Alg also runs in polynomial time. □

5 Splittable CLR

In this section, we introduce our Path-Alg. As asked in [Harks *et al.*, 2013], an open problem is whether a more tour-specific approach could lead to better approximation ratios. We answer this question partially by showing that for splittable CLR our Path-Alg is a 4.092-approximation algorithm. The approach is inspired by the cycle-splitting method used for MCVRP in [Li and Simchi-Levi, 1990].

The main idea of the cycle-splitting method proposed in [Li and Simchi-Levi, 1990] is to obtain a Hamiltonian cycle by using a δ-approximation algorithm for metric TSP (recall that $\delta \approx 1.5$), and then obtain tours based on splitting the cycle. Since we can transform a cycle into a path by deleting an edge from it, instead of splitting a cycle, we focus on splitting a path. Although it seems that there is no big difference between cycles and paths, they may lead to different approximation ratios. This is because the edges incident to depots may have additional costs, and we need to carefully control the number of edges incident to depots.

A path, as a special case of tree, has a simpler structure: for the case of splittable demand, one can greedily divide it into several segments of demand k and possibly leave the last segment (containing the depot) with a demand of less than k. Recall that the tree-splitting procedure is mainly to obtain a sub-tree S such that $k \geq d(S) > k/2$ repeatedly. Therefore, based on a path-splitting procedure, we may reduce the connection part of the routing cost from $\sum_{v\in V}(4/k)d(v) \cdot \min_{u\in O} w(v, u)$ to $\sum_{v\in V}(2/k)d(v)\cdot\min_{u\in O} w(v, u)$. However, it is not straightforward to obtain better approximation ratios for splittable CLR since we need to use twice cost of the paths obtained from a δ-approximation algorithm for metric TSP that is more expensive than the constrained spanning forest computed in Lemma 4. We will obtain a good approximation ratio based on this idea with novel analysis.

5.1 The Algorithm

Compared to Tree-Alg, Path-Alg has two main modifications: firstly, it computes a constrained spanning path-packing rather than a constrained spanning forest, and secondly, it constructs a set of tours through a path-splitting procedure instead of the tree-splitting procedure.

To compute a constrained spanning path-packing, we construct two new graphs G' and H. Given $G = (V \cup U, E, w, \phi)$, we obtain a new graph $G' = (V \cup U, E, w', \phi)$ such that $w'(v, v') = w(v, v')$ for any $v, v' \in V$ and $w'(u, v) = w(u, v)+\theta\cdot\phi(u)$ for any $v \in V$ and $u \in U$, where θ is a constant defined later. Then, we obtain another graph $H = (V \cup \{r\}, F, c)$ such that $c(r, v) = \min_{u\in U} w'(u, v)$ and $c(v, v') = \min\{w'(v, v'), c(r, v) + c(r, v')\}$. One can also think that H is obtained by contracting all depots in U as a super-depot r and then taking a metric closure of G'. Note that the edge weight functions in new graphs G' and H are still metric functions.

Algorithm 3 An improved approximation algorithm for splittable CLR (Path-Alg)

Input: An instance of CLR.
Output: A feasible solution to CLR.
1: Create an UFL instance with edge costs $\widetilde{w} = (2/k)w$ and depot costs $\widetilde{\phi} = \alpha \cdot \phi$ as in Lemma 3.
2: Apply Byrka and Aardal's bifactor approximation algorithm [Byrka and Aardal, 2010] with a parameter of $\gamma > 0$ on the UFL instance and let O_1 be the set of depots opened in the resulting UFL solution.
3: Compute a constrained spanning path-packing $\mathcal{P}$ in G as in Lemma 9 and let $O_2 = U(\mathcal{P})$ be the set of depots contained in some $P \in \mathcal{P}$.
4: Open all depots in $O \coloneqq O_1 \cup O_2$.
5: Obtain a set of feasible tours $\mathcal{I}$ by calling the path-splitting procedure in Algorithm 4.
6: Return $(O, \mathcal{I})$.

Lemma 9. *Given a δ-approximation algorithm for metric TSP, there is a polynomial-time algorithm to compute a constrained spanning path-packing $\mathcal{P}$ in G.*

Proof. By applying a δ-approximation algorithm for metric TSP in H, we can obtain a Hamiltonian cycle C in H. Note that C corresponds to a subgraph of G' where each vertex in V has an even degree. Therefore, we can obtain a constrained spanning cycle-packing $\mathcal{C}$ in G' by shortcutting. Note that the shortcutting can ensure that each depot is contained in at most one cycle or path in $\mathcal{C}$.

Then, we can transform $\mathcal{C}$ into a constrained spanning path-packing $\mathcal{P}$ by doing: (1) for each cycle $uv_1 \dots v_iu \in \mathcal{C}$ we delete the edge with a smaller cost from uv_1 and uv_i, and (2) for each path $uv_1 \dots v_iu' \in \mathcal{C}$ we delete the edge incident to the depot with a smaller opening cost from uv_1 and $u'v_i$. We can see each depot is contained in at most one path in $\mathcal{P}$. □

The framework of our Path-Alg for splittable CLR can be seen in Algorithm 3.

The path-splitting procedure works as follows. First, for each customer $v \in V$ with $d(v) > k$, we construct $\lceil \frac{d(v)}{k} \rceil - 1$ tours for v (with each delivering k of demand) by connecting v with its nearest opened depot in O and update $d(v) \coloneqq d(v) - k \cdot (\lceil \frac{d(v)}{k} \rceil - 1)$. In the following, the demand of each customer v holds that $0 < d(v) \leq k$. Then, we consider a path $P_u \in \mathcal{P}$ rooted at $u \in O$, and satisfy all customers in $V(P_u)$ by splitting P_u. Denote the sub-path rooted at v and v's the only children by P_v and $q(v)$, respectively. We consider the following two cases.

- If $d(P_u) \leq k$, we get a tour for all customers in $V(P_u)$ by doubling all edges in $E(P_u)$ and then shortcutting.
- Otherwise, we can do the following repeatedly until it satisfies that $d(P_u) \leq k$. First, we can find a customer $v \in V(P_u)$ such that $d(P_v) > k$ and $d(P_{q(v)}) \leq k$.

Algorithm 4 The path-splitting procedure for splittable CLR

Input: An instance of CLR, a set of opened depots O, and a constrained spanning path-packing $\mathcal{P}$.
Output: A set of feasible tours $\mathcal{I}$ to CLR.

```
Initialize 𝓘 = ∅.
for all v ∈ V with d(v) > k do
    Construct ⌈d(v)/k⌉ − 1 tours for v by connecting v with its nearest opened depot in O (each delivers k of demand).
    Add the tours to 𝓘 and update d(v) := d(v) − k·(⌈d(v)/k⌉ − 1).
end for
for all P_u ∈ 𝓟 do                      ▷ P_u is rooted at the depot u ∈ O
    while d(P_u) > k do
        Find v ∈ V(P_u) with d(P_v) > k and d(P_{q(v)}) ≤ k.   ▷ q(v) is v's the only children
        Find an edge e_{P_v} with minimized cost connecting one depot in O with one vertex in V(P_v).
        Construct a tour I for all customers in V(P_v) by doubling edges in E(P_v) ∪ {e_{P_v}} and shortcutting with demand assignments: x_{vI} = k − d(P_{q(v)}) and x_{v'I} = d(v') for each v' ∈ V(P_{q(v)}).
        Add the tour to 𝓘, update P_u by removing V(P_{q(v)}) and E(P_v) from P_u, and update d(v) := d(v) − x_{vI}.
    end while
    Construct a tour for all customers in V(P_u) by doubling edges in E(P_u) and shortcutting.
    Add the tour to 𝓘.
end for
Return 𝓘.
```

Consider the sub-path P_v. Then, we find an edge e_{P_v} with minimized cost connecting one depot in O with one vertex in $V(P_v)$. By doubling edges in $E(P_v) \cup \{e_S\}$ and shortcutting, we construct a tour I for all customers in $V(P_v)$ with demand assignments: $x_{vI} = k - d(P_{q(v)})$ and $x_{v'I} = d(v')$ for each $v' \in V(P_{q(v)})$. At last, we update P_u by removing $V(P_{q(v)})$ and $E(P_v)$ and setting $d(v) \coloneqq d(v) - x_{vI}$.

The path-splitting procedure used in Algorithm 3 is formally shown in Algorithm 4.

Lemma 10 (♣). *Given a set of opened depots O and a constrained spanning path-packing $\mathcal{P}$, the path-splitting procedure can use polynomial time to obtain a set of tours $\mathcal{I}$ such that $(O, \mathcal{I})$ is a feasible solution for splittable UFL and $w(\mathcal{I}) \leq 2w(\mathcal{P}) + \sum_{v \in V}(2/k)d(v) \cdot \min_{u \in O} w(v,u)$.*

5.2 The Analysis

Based on a straightforward analysis, we may obtain $2w(\mathcal{P}) + \phi(\mathcal{P}) \leq 2\delta \cdot \psi^* + 2\delta \cdot \phi^*$, where δ is the approximation ratio of the algorithm used for metric TSP. However, it cannot lead to a better-than-4.169-approximation ratio. In this subsection, we will prove a better bound based on a tighter analysis, and obtain a 4.092-approximation ratio.

Let C^* be a minimum cost Hamiltonian cycle in graph H.

Lemma 11. *It holds that $c(C^*) \leq \psi^* + 2\theta \cdot \phi^*$.*

Proof. Consider an optimal solution $(O, \mathcal{I})$ of CLR. For each depot $o \in O$, there is a set of tours $\mathcal{I}_o \subseteq \mathcal{I}$ with each in it containing o. By shortcutting all tours in $\mathcal{I}_o$, we obtain a cycle $C_o = ov_1 \dots v_i o$ such that $w'(C_o) \leq w(\mathcal{I}_o) + 2\theta \cdot \phi(o)$. Note that H is obtained by contracting all depots in U as a super-depot r and then taking a metric closure of G', and $\{C_o\}_{o \in O}$ corresponds to an Eulerian graph in H. Therefore, by shortcutting $\{C_o\}_{o \in O}$, we obtain a Hamiltonian cycle in H with a cost of at most $\sum_{o \in O} w'(C_o)$. Since C^* is the minimum cost Hamiltonian cycle in H, we can get that

$$
\begin{aligned}
c(C^*) &\leq \sum_{o \in O} w'(C_o) \\
&\leq w(\mathcal{I}) + 2\theta \cdot \phi(O) = \psi^* + 2\theta \cdot \phi^*,
\end{aligned}
$$

which finishes the proof. □

Lemma 12. *Given a δ-approximation algorithm for metric TSP, the computed constrained spanning path-packing $\mathcal{P}$ satisfies that $2w(\mathcal{P}) + \phi(\mathcal{P}) \leq 2\delta \cdot \psi^* + \delta \cdot \phi^*$ under $\theta = \frac{1}{4}$.*

Proof. Given a δ-approximate Hamiltonian cycle in H, the algorithm in Lemma 9 simply computes a constrained spanning cycle-packing $\mathcal{C}$ in G' by shortcutting the corresponding subgraph in G'. Hence, we have $w'(\mathcal{C}) \leq \delta \cdot c(C^*)$. Let $\mathcal{C} = \mathcal{C}_1 \cup \mathcal{C}_2$, where $\mathcal{C}_1$ and $\mathcal{C}_2$ denote the set of cycles and paths in $\mathcal{C}$, respectively.

For each cycle $uv_1 \dots v_i u \in \mathcal{C}_1$ (resp., path $uv_1 \dots v_i u' \in \mathcal{C}_2$), the algorithm deletes the edge with a smaller cost from uv_1 and uv_i (resp., incident to the depot with a smaller opening cost from uv_1 and $u'v_i$). Therefore, the constrained spanning path-packing $\mathcal{P}$ satisfies that $w'(\mathcal{P}) \leq w'(\mathcal{C}) - \theta \cdot \phi(\mathcal{C}_1) - \frac{1}{2}\theta \cdot \phi(\mathcal{C}_2)$ and $\phi(\mathcal{P}) \leq \phi(\mathcal{C}_1) + \frac{1}{2}\phi(\mathcal{C}_2)$. Note that $w'(\mathcal{P}) = w(\mathcal{P}) + \theta \cdot \phi(\mathcal{P})$ since each path in $\mathcal{P}$ contains only one edge incident to the depot. Moreover, under $\theta = \frac{1}{4}$, we also have $1 - 2\theta = 2\theta > 0$. Hence, we have

$$
\begin{aligned}
2w(\mathcal{P}) + \phi(\mathcal{P}) &= 2w'(\mathcal{P}) + (1 - 2\theta) \cdot \phi(\mathcal{P}) \\
&\leq 2w'(\mathcal{P}) + (1 - 2\theta) \cdot \left(\phi(\mathcal{C}_1) + \frac{1}{2}\phi(\mathcal{C}_2)\right) \\
&= 2w'(\mathcal{P}) + 2\theta \cdot \left(\phi(\mathcal{C}_1) + \frac{1}{2}\phi(\mathcal{C}_2)\right) \\
&\leq 2w'(\mathcal{C}) \\
&\leq 2\delta \cdot c(C^*).
\end{aligned}
$$

So, we get $2w(\mathcal{P}) + \phi(\mathcal{P}) \leq 2\delta \cdot \psi^* + \delta \cdot \phi^*$ by Lemma 11. □

Using the well-known Christofides-Serdyukov algorithm [1976; 1978] for metric TSP, we have $\delta = \frac{3}{2}$, and then we have $2w(\mathcal{P}) + \phi(\mathcal{P}) \leq 3\psi^* + \frac{3}{2}\phi^*$ by Lemma 12. Note that this result may only lead to a 4.143-approximation algorithm for splittable CLR. We can obtain a further improvement if using the property of their algorithm. Let T^* be a minimum spanning tree in H. We have the following lemma.

Lemma 13 (♣). *It holds that $c(T^*) \leq \psi^* + \theta \cdot \phi^*$.*

Lemma 14 (♣). *Given the Christofides-Serdyukov algorithm for metric TSP, the computed constrained spanning path-packing $\mathcal{P}$ satisfies that $2w(\mathcal{P}) + \phi(\mathcal{P}) \leq 3\psi^* + \phi^*$ if $\theta = \frac{1}{4}$.*

Theorem 15 (♣). *For splittable CLR, Path-Alg is a polynomial-time 4.092-approximation algorithm.*

6 Experimental Results

We conduct experiments to compare our two algorithms, Tree-Alg and Path-Alg, with the previous approximation algorithm in [Harks *et al.*, 2013]. Next, we introduce the benchmark instances of CLR, the implementations of our algorithms, and the results, respectively.

Instances. Harks *et al.* [2013] tested their approximation algorithm on 45 CLR benchmark instances in total, including 36 instances from [Tuzun and Burke, 1999] and 9 instances from [Barreto *et al.*, 2007]. They compared their results with the *previous best known solutions* (pbks) obtained by heuristic approaches [Prins *et al.*, 2007; Baldacci *et al.*, 2009; Barreto *et al.*, 2007; Tuzun and Burke, 1999] by computing the gaps between their results and the pbks. Although some results of these benchmark instances have been further slightly improved [Baldacci *et al.*, 2011; Contardo *et al.*, 2014], for the sake of comparison, we test our algorithms on these instances and still compute the gaps between our results and the pbks.

Implementations. We present the detailed implementations of our algorithms.

On one hand, given a sub-tree, instead of finding tours by the method of doubling and shortcutting, we obtained a tour by finding a minimum cost matching on the odd-degree vertices contained in the sub-tree and then shortcutting. This was motivated by its ability to guarantee a 1.5-approximation for metric TSP [Christofides, 1976; Serdyukov, 1978], whereas the doubling-and-shortcutting method may only achieve a 2-approximation [Williamson and Shmoys, 2011]. Note that in our Path-Alg we used this method to implement the 1.5-approximation algorithm of metric TSP as well.

On the other hand, as in [Harks *et al.*, 2013], we used the practical 1.861-approximation algorithm [Jain *et al.*, 2003] of UFL to open a set of depots O_1 instead of using the bifactor approximation algorithm [Byrka and Aardal, 2010] since the latter algorithm involves solving an LP, which is not practical. So, there was no need to consider the setting of γ. Moreover, after opening a set of depots using the greedy algorithm, suggested in [Harks *et al.*, 2013], we regarded their opening cost as zero in the following. This does not impact the approximation ratio. For α, we tested different values. This choice was made because for a range of values of α the implementations can always guarantee an approximation ratio. See the following lemma. Harks *et al.* [2013] only considered $\alpha = 1$ and showed the implementation had a ratio of 5.722.

Lemma 16 (♣). *For any* $0.5 \leq \alpha \leq 1.26$*, our implementation of Tree-Alg has an approximation ratio of 5.722; For any* $1 \leq \alpha \leq 2.07$*, our implementation of Path-Alg has an approximation ratio of 4.861; Moreover, both implementations have a running time of* $O(n^3 + nm \log nm)$.

Our algorithms are implemented in C++ on a desktop computer with an AMD Ryzen 5 PRO 4650G with Radeon Graphics (3.70 GHz, 32.0 GB RAM) using Windows Subsystem for Linux (WSL). The detailed information of our algorithms, the 45 tested instances and the pbks can be found in https://github.com/JingyangZhao/CLR.

α	**Tree-Alg**	**Path-Alg**
0.1	0.1996	0.2621
0.2	0.1503	0.2098
0.3	0.1265	0.1883
0.4	**0.1164**	0.1755
0.5	0.1211	0.1755
0.6	0.1202	0.1677
0.7	0.1219	**0.1676**
0.8	0.1270	0.1733
0.9	0.1330	0.1834
1.0	0.1363	0.1874
1.1	0.1408	0.1872
1.2	0.1420	0.1888
1.3	0.1480	0.1932
1.4	0.1566	0.1998
1.5	0.1589	0.2024

Table 1: The average gaps between our results and the pbks for Tree-Alg and Path-Alg under different values of α

Results. A summary of our results can be found in Table 1, where we list the average gaps between our results of Tree-Alg (resp., our results of Path-Alg) and the pbks under different settings of α. Note that our algorithms run very fast. The running time of each instance on average is about 0.08s for both Tree-Alg and Path-Alg, which is almost unchanged under different values of α.

The previous algorithm in [Harks *et al.*, 2013] achieves an average gap of 18.8%. Under $\alpha = 0.1$, our algorithms get a bigger average gap, which means our results are worse. The reason may be that under $\alpha = 0.1$ the greedy algorithm opens almost all depots for each instance which is too expensive. For other values of α in Table 1, we can achieve a better ratio. Notably, as we increase the value of α, the gaps exhibit a significant decrease potentially because the greedy algorithm opens less depots. Especially, when setting $\alpha = 0.4$ and $\alpha = 0.7$, the gaps become 11.64% and 16.76% for Tree-Alg and Path-Alg, respectively, both surpassing the performance of 18.8%. But, when further increasing α, the gaps start to increase: the number of opened depots may be too small.

We conclude that when setting α suitably, our algorithms achieve much better results than the previous approximation algorithm, and the quality of solutions is also much closer to optimality than the provable approximation ratio. Although Path-Alg has a better approximation ratio in theory, Tree-Alg is slightly better on the tested instances in practical. The reason may be that in the worst-case, Tree-Alg delivers only about $k/2$ of demand on each tour but it delivers close to k of demand on each tour in practical.

7 Conclusion

In this paper, we propose two improved approximation algorithms for CFL: one is a 4.169-approximation algorithm for both the unsplittable case and the splittable case, and another is a 4.092-approximation algorithm for the splittable case. Our algorithms may be extended to some variants of CLR in [Harks *et al.*, 2013]. A remained open problem is to design a better-than-4-approximation algorithm for CLR.

Acknowledgements

This work is supported by National Natural Science Foundation of China under grants 62372095 and 62172077, and Natural Science Foundation of Sichuan Province of China under grant 2023NSFSC0059.

References

[Altinkemer and Gavish, 1987] Kemal Altinkemer and Bezalel Gavish. Heuristics for unequal weight delivery problems with a fixed error guarantee. *Operations Research Letters*, 6(4):149–158, 1987.

[An *et al.*, 2017] Hyung-Chan An, Mohit Singh, and Ola Svensson. LP-based algorithms for capacitated facility location. *SIAM Journal on Computing*, 46(1):272–306, 2017.

[Baldacci *et al.*, 2009] R Baldacci, A Mingozzi, and R Wolfer-Calvo. The capacitated location routing problem. *Presented at ROUTE 2009*, 2009.

[Baldacci *et al.*, 2011] Roberto Baldacci, Aristide Mingozzi, and Roberto Wolfler Calvo. An exact method for the capacitated location-routing problem. *Operations research*, 59(5):1284–1296, 2011.

[Barreto *et al.*, 2007] Sérgio Barreto, Carlos Ferreira, Jose Paixao, and Beatriz Sousa Santos. Using clustering analysis in a capacitated location-routing problem. *European journal of operational research*, 179(3):968–977, 2007.

[Blauth *et al.*, 2022] Jannis Blauth, Vera Traub, and Jens Vygen. Improving the approximation ratio for capacitated vehicle routing. *Mathematical Programming*, pages 1–47, 2022.

[Byrka and Aardal, 2010] Jaroslaw Byrka and Karen I. Aardal. An optimal bifactor approximation algorithm for the metric uncapacitated facility location problem. *SIAM J. Comput.*, 39(6):2212–2231, 2010.

[Christofides, 1976] Nicos Christofides. Worst-case analysis of a new heuristic for the travelling salesman problem. Technical report, Carnegie-Mellon University, 1976.

[Chudak and Shmoys, 2003] Fabián A. Chudak and David B. Shmoys. Improved approximation algorithms for the uncapacitated facility location problem. *SIAM J. Comput.*, 33(1):1–25, 2003.

[Contardo *et al.*, 2014] Claudio Contardo, Jean-François Cordeau, and Bernard Gendron. An exact algorithm based on cut-and-column generation for the capacitated location-routing problem. *INFORMS Journal on Computing*, 26(1):88–102, 2014.

[Deppert *et al.*, 2023] Max Deppert, Matthias Kaul, and Matthias Mnich. A $(3/2 + \varepsilon)$-approximation for multiple TSP with a variable number of depots. In *ESA 2023*, volume 274 of *LIPIcs*, pages 39:1–39:15. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2023.

[Drexl and Schneider, 2015] Michael Drexl and Michael Schneider. A survey of variants and extensions of the location-routing problem. *Eur. J. Oper. Res.*, 241(2):283–308, 2015.

[Friggstad *et al.*, 2022] Zachary Friggstad, Ramin Mousavi, Mirmahdi Rahgoshay, and Mohammad R. Salavatipour. Improved approximations for capacitated vehicle routing with unsplittable client demands. In *IPCO 2022*, volume 13265 of *Lecture Notes in Computer Science*, pages 251–261. Springer, 2022.

[Haimovich and Kan, 1985] Mordecai Haimovich and Alexander H. G. Rinnooy Kan. Bounds and heuristics for capacitated routing problems. *Mathematics of Operations Research*, 10(4):527–542, 1985.

[Harks *et al.*, 2013] Tobias Harks, Felix G. König, and Jannik Matuschke. Approximation algorithms for capacitated location routing. *Transportation Science*, 47(1):3–22, 2013.

[Heine *et al.*, 2023] Felipe Carrasco Heine, Antonia Demleitner, and Jannik Matuschke. Bifactor approximation for location routing with vehicle and facility capacities. *European Journal of Operational Research*, 304(2):429–442, 2023.

[Jain *et al.*, 2003] Kamal Jain, Mohammad Mahdian, Evangelos Markakis, Amin Saberi, and Vijay V. Vazirani. Greedy facility location algorithms analyzed using dual fitting with factor-revealing LP. *J. ACM*, 50(6):795–824, 2003.

[Karlin *et al.*, 2021] Anna R. Karlin, Nathan Klein, and Shayan Oveis Gharan. A (slightly) improved approximation algorithm for metric TSP. In *STOC 2021*, pages 32–45. ACM, 2021.

[Karlin *et al.*, 2023] Anna R. Karlin, Nathan Klein, and Shayan Oveis Gharan. A deterministic better-than-3/2 approximation algorithm for metric TSP. In *IPCO 2023*, volume 13904 of *Lecture Notes in Computer Science*, pages 261–274. Springer, 2023.

[Li and Simchi-Levi, 1990] Chung-Lun Li and David Simchi-Levi. Worst-case analysis of heuristics for multidepot capacitated vehicle routing problems. *INFORMS Journal on Computing*, 2(1):64–73, 1990.

[Li, 2013] Shi Li. A 1.488 approximation algorithm for the uncapacitated facility location problem. *Inf. Comput.*, 222:45–58, 2013.

[Maranzana, 1964] FE Maranzana. On the location of supply points to minimize transport costs. *OR*, pages 261–270, 1964.

[Montoya-Torres *et al.*, 2015] Jairo R Montoya-Torres, Julián López Franco, Santiago Nieto Isaza, Heriberto Felizzola Jiménez, and Nilson Herazo-Padilla. A literature review on the vehicle routing problem with multiple depots. *Computers & Industrial Engineering*, 79:115–129, 2015.

[Prins *et al.*, 2007] Christian Prins, Caroline Prodhon, Angel Ruiz, Patrick Soriano, and Roberto Wolfler Calvo. Solving the capacitated location-routing problem by a cooperative lagrangean relaxation-granular tabu search heuristic. *Transportation science*, 41(4):470–483, 2007.

[Prodhon and Prins, 2014] Caroline Prodhon and Christian Prins. A survey of recent research on location-routing problems. *Eur. J. Oper. Res.*, 238(1):1–17, 2014.

[Rathinam *et al.*, 2007] Sivakumar Rathinam, Raja Sengupta, and Swaroop Darbha. A resource allocation algorithm for multivehicle systems with nonholonomic constraints. *IEEE Transactions on Automation Science and Engineering*, 4(1):98–104, 2007.

[Serdyukov, 1978] Anatolii Ivanovich Serdyukov. Some extremal bypasses in graphs. *Upravlyaemye Sistemy*, 17:76–79, 1978.

[Toth and Vigo, 2014] Paolo Toth and Daniele Vigo. *Vehicle routing: problems, methods, and applications*. SIAM, 2014.

[Traub *et al.*, 2022] Vera Traub, Jens Vygen, and Rico Zenklusen. Reducing path TSP to TSP. *SIAM Journal on Computing*, 51(3):20–24, 2022.

[Tuzun and Burke, 1999] Dilek Tuzun and Laura I Burke. A two-phase tabu search approach to the location routing problem. *European journal of operational research*, 116(1):87–99, 1999.

[Von Boventer, 1961] Edwin Von Boventer. The relationship between transportation costs and location rent in transportation problems. *Journal of Regional Science*, 3(2):27–40, 1961.

[Webb, 1968] MHJ Webb. Cost functions in the location of depots for multiple-delivery journeys. *OR*, pages 311–320, 1968.

[Williamson and Shmoys, 2011] David P Williamson and David B Shmoys. *The design of approximation algorithms*. Cambridge university press, 2011.

[Xin *et al.*, 2021] Liang Xin, Wen Song, Zhiguang Cao, and Jie Zhang. Multi-decoder attention model with embedding glimpse for solving vehicle routing problems. In *AAAI 2021*, pages 12042–12049. AAAI Press, 2021.

[Xu and Rodrigues, 2015] Zhou Xu and Brian Rodrigues. A 3/2-approximation algorithm for the multiple tsp with a fixed number of depots. *INFORMS Journal on Computing*, 27(4):636–645, 2015.

[Xu *et al.*, 2011] Zhou Xu, Liang Xu, and Brian Rodrigues. An analysis of the extended christofides heuristic for the k-depot TSP. *Operations Research Letters*, 39(3):218–223, 2011.

[Zhang *et al.*, 2015] Zizhen Zhang, Huang He, Zhixing Luo, Hu Qin, and Songshan Guo. An efficient forest-based tabu search algorithm for the split-delivery vehicle routing problem. In *AAAI 2015*, pages 3432–3438. AAAI Press, 2015.

[Zhao and Xiao, 2023] Jingyang Zhao and Mingyu Xiao. Improved approximation algorithms for multidepot capacitated vehicle routing. In *COCOON 2023*, volume 14423 of *Lecture Notes in Computer Science*, pages 378–391. Springer, 2023.

[Zhou *et al.*, 2023] Jianan Zhou, Yaoxin Wu, Wen Song, Zhiguang Cao, and Jie Zhang. Towards omni-generalizable neural methods for vehicle routing problems. In *ICML 2023*, volume 202 of *Proceedings of Machine Learning Research*, pages 42769–42789. PMLR, 2023.