# An algorithm for the metric multiple depots capacitated vehicle routing problem with restocking and capacity two[*]

Chao Xu[†] Yichen Yang[‡] Qian Zhang[§]

**Abstract**

The capacitated vehicle routing problem (CVRP) is one of the most well known NP-hard combinatorial optimization problems. Single depot CVRP with a general metric is NP-hard even for fixed capacity 3, while polynomial time solvable for fixed capacity 2. We consider the variant of CVRP where restocking is available. We show that if there is a constant number of depots, then the problem can be solved in polynomial time when capacity is 2.

## 1 Introduction

In this paper, we study a variant of the capacitated vehicle routing problem (CVRP). In the CVRP problem, there is a single depot, and multiple customers each have unit demand. Each vehicle has a capacity of $Q$. Each vehicle travels from the depot, delivers the inventory to at most $Q$ customers, and returns to the depot. We consider the variant that allows restocking. After completing a trip, the vehicle can return to any depot, and restock the inventory for more delivery. The objective is to minimize the total distance traveled over all vehicles.

We now introduce the case for multiple depots. In the *multiple depots capacitated vehicle routing with restocking problem* (MDCVRRP), there are $k$ depots with enough inventory and $n$ customers with unit demand. There are vehicles with capacity $Q$. A depot that owns at least one vehicle is called a *base*. Each vehicle is associated with a base. A vehicle starts from its base, and can restock at any depot before any further delivery. We are interested in finding a tour for each vehicle so that the demand of all the customers can be satisfied, each vehicle returns to its base after it *completes all deliveries*, and the total travel distance is minimized.

Since the traveling salesman problem (TSP) is a special case of MDCVRRP when the number of depot equals one and the capacity of the vehicle is unlimited, MDCVRRP is strongly NP-hard even in the Euclidean plane [8]. MDCVRRP is closely related to the classic CVRP. Almost all the special cases of CVRP studied in the literature are NP-hard. The only polynomial solvable special case of CVRP that relates to our result is the case when $Q \leq 2$ and the number of depots equals one [2]. However, for $Q \leq 2$, it is unknown if the multiple depot case can be solved in polynomial time. For a general metric, CVRP is shown to be APX-complete for any fixed $Q \geq 3$ [2].

Studying the capacity 2 case is more than a theoretical curiosity. The practical application concerns with large containers (skips), where each vehicle can only pick up one or two containers. A typical example is waste collection from the collection area to landfill [1].

It turns out the number of non-base depots, not the number of depots, is the most important parameter of the computational complexity. Intuitively, one approach to the problem is to generate tours by looking at the graph instead of looking at each vehicle. This would give us the optimal result when every depot is a base. However, because we disregard vehicles, there might be a tour that contains only non-base depots, which cannot be completed by any vehicle. Therefore, we need to take into account if a non-base depot is used in a tour, there is a base matched to that tour. It is unclear how to handle this constraint other than trying all of them, and hence blowing up the running time significantly as the number of non-base depots grows.

**Our Contribution** We show that if there are a constant number of depots, then MDCVRRP with capacity 2 is solvable in polynomial time. In fact, we prove a stronger result. Even if the number of depots is large, as long as the number of the non-base depots is a constant, then MDCVRRP with capacity 2 can be solved in polynomial time.

[*]This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

[†]Yahoo! Research, New York, NY 10003, USA chao.xu@verizonmedia.com

[‡]Department of Mathematics, East China University of Science and Technology, Shanghai, China agamemnon314@hotmail.com

[§]School of Civil and Environmental Engineering, Cornell University, Ithaca, NY 14853, USA qz283@cornell.edu

**Organization** The rest of this paper is organized as follows. In section 2, we introduce some notations and notions we used in our work, and describe the problem. In section 3, we introduce $T$-bones, and solving the minimum cost $T$-bone problem. In section 4, we introduce constrained forests and find how many there can be. In section 5, we describe an algorithm for MDCVRRP with capacity 2 which runs in polynomial time for a constant number of non-base depots.

## 2 Preliminaries

Let $G=(V,E)$ be an undirected complete graph. Often $V$ is partitioned into two sets of vertices $D$ and $U$, hence we will abuse the notation and write $G=(D,U,E)$ to denote $G=(V,E)$, where $V=D\cup U$. The vertices in $D$ are called *depots*. We fix a set $B\subseteq D$ of depots to be the set of *bases*. We define $\overline{B}=\overline{B}$, the set of *non-base* depots. The set $U$ represents the set of $n$ *customers*. Each customer has a unit demand, and each vehicle has capacity 2. Let $c:E\to\mathbb{R}^+$ be a non-negative function over the edges, which we call the *cost*. The cost function $c$ is a *metric* if $c$ is non-negative and satisfies the triangle inequality, that is, $c(uw)+c(wv)\geq c(uv)$ for every $u,v,w\in V$. A closed *walk* is a sequence of vertices and edges, $v_1,e_1,v_2,\ldots,e_{n-1},v_n$, such that $v_1=v_n$, $e_i\in E$ and is precisely an edge containing $v_i$ and $v_{i+1}$.

Let $F\subseteq E$ be a set of edges. $F$ *covers* a vertex $v\in V$ if $v\in e$ for some $e\in F$. We use $V(F)=\bigcup_{e\in F}e$ to denote the vertex set of $F$, which is the set of vertices covered by $F$. We use $c(F)=\sum_{uv\in F}c(uv)$ to denote the total cost of $F$. We use $F[B]$ to denote the set of edges in $F$ that have both endpoints in $B$. Let $G$ be a graph, then $G[B]$ is the *induced subgraph* whose vertex set is $B$ and whose edge set is $E[B]$. If a statement applies to graphs, then we extend a statement to a set of edges $F$ by considering the graph $(V(F),F)$. For any $F\subseteq E$, we use $Odd(F)$ to denote the set of odd degree vertices. Note that $|Odd(F)|$ is always even. A *forest* is a set of edges without cycles. The *degree* of a vertex $v$ for a set of edges $F$, denoted as $\deg_F(v)$, is the number of edges in $F$ incident to $v$. A maximal set of connected vertices is called a *component*. A component with at least two vertices is called a *non-trivial component*.

If $uv$ and $vw$ are edges, the *splitting-off* operation [7] on $v$ removes the edge $uv$ and $vw$, and add an edge $uw$. This operation maintains the parity of the degree of $v$, and preserves the degree of all other vertices.

We consider edges with multiplicities. When we write $S\subseteq E$, we are allowing duplicate edges in $S$, even when the corresponding edge appeared only once in $E$. Now, we define the abstract problem we try to solve in detail.

**Problem 1 (2-MDCVRRP'($p$))**
Input: *A complete graph $G=(D,U,E)$, $B\subseteq D$ such that $|\overline{B}|\leq p$, and a metric cost $c:E\to\mathbb{R}$.*
Output: *A minimum cost set of closed walks $\mathcal{W}$ on $G$ under the following constraints:*

1. *each closed walk in $\mathcal{W}$ contains a depot in $B$;*
2. *each closed walk in $\mathcal{W}$ contains at most 2 successive vertices in $U$;*
3. *every vertex in $U$ appears in some walk.*

Instead of a set of closed walks, we just have to find the set of edges (with multiplicity) used by the set of closed walks. Indeed, by the Eulerian theorem [4], there is a linear time algorithm to decompose the edges into a set of closed walks given a union of the edges of some closed walks. Due to the fact we have a metric, there exists an optimal solution where no vertex in $U$ appeared more than once in the walk. Hence, we can solve the following problem instead, which transform the problem on walks into finding a collection of edges.

**Problem 2 (2-MDCVRRP($p$))**
Input: *A complete graph $G=(D,U,E)$, $B\subseteq D$ such that $|\overline{B}|\leq p$, and a metric cost $c:E\to\mathbb{R}$.*
Output: *A minimum cost collection of edges $W$ in $G$ (allow duplicates) such that:*

1. *each non-trivial component of $W$ contains a depot in $B$;*
2. *$\deg_W(v)$ is even for every vertex, and equals 2 if $v\in U$;*
3. *$W[U]$ is a matching.*

In particular, each non-trivial component of a solution $W$ of 2-MDCVRRP($p$) requires only a single vehicle to traverse.

Note the problem 2-MDCVRRP($p$) captures the case when there are at most $p$ depots without a vehicle. We define 2-MDCVRRP to be 2-MDCVRRP($\infty$). Namely, there is no bound on the size of $|\overline{B}|$.

A *matching* is a set of disjoint edges. For a set $T \subseteq V$ of even cardinality. A set of edges $J \subseteq E$ (with multiplicity) is called a $T$*-join* if $\deg_J(v)$ is odd if and only if $v \in T$.

**Lemma 2.1** ( [6]) *Given an undirected graph $G = (V, E)$ and a set $T \subseteq V$ of even cardinality, there exists a $T$-join in $G$ if and only if $|C \cap T|$ is even for each component $C$ of $G$.*

**Theorem 2.2** ( [5]) *Given an undirected graph $G = (V, E)$ and a set $T \subseteq V$ of even cardinality, if the edge costs are nonnegative, the minimum cost $T$-join can be found in $O(|V|^3)$ time.*

We emphasize the algorithm are split into two different parts, where they are independent of each other, and should be read independently.

## 3 $T$-Bones

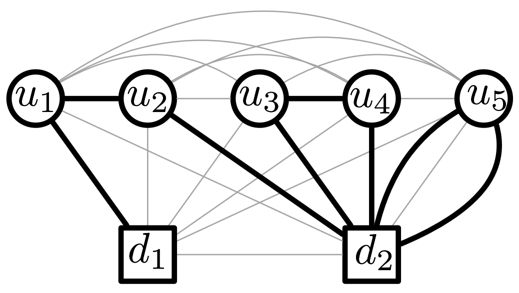
Consider a complete graph $G = (D, U, E)$ with a metric cost $c$. For $T \subseteq D$ such that $|T|$ is even, a set of edges $J$ is a $T$*-bone*, if $J$ is a $T$-join, $\deg_J(u) = 2$ for all $u \in U$, and $J[U]$ is a matching. Note we do allow parallel edges in $J$.

Note in particular, each non-trivial component in a $T$-bone $J$ contains at least one vertex in $D$. Indeed, consider a non-trivial component of $J$ that contains a vertex $u \in U$. Because $J[U]$ is a matching and $\deg_J(u) = 2$, then $u$ has a neighbor that is not in $U$, therefore must be an element in $D$.

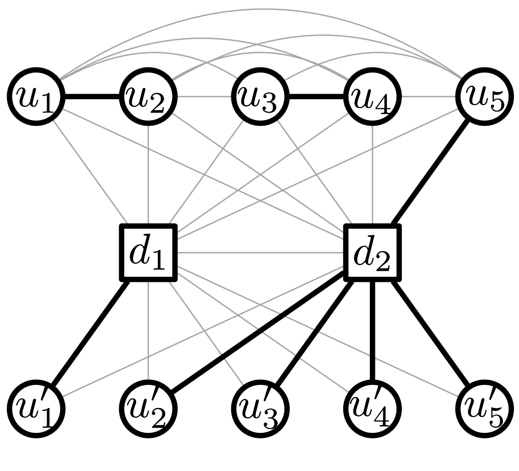
We will show that finding the minimum cost $T$-bone in $G$ can reduce to solving a minimum cost $T$-join problem on the auxiliary graph of $G$. First, we describe the construction of the auxiliary graph $G'$ of $G$:

1. for each $u \in U$, duplicate a vertex $u'$. Let $U' = \{u' | u \in U\}$.
2. let $V' = U \cup U' \cup D$;
3. let $E' = E \cup \{u'v | u' \in U', v \in D\}$;
4. let $G' = (D, U \cup U', E')$;

See Figure 3.1b for an example of the auxiliary graph. We consider a cost function $c' : E' \to \mathbb{R}$ as follows. $c'(uv) = c(uv)$ for all $u, v \in V$, and $c'(u'v) = c(uv)$ for all $u \in U$ and $v \in D$.



(a) The original graph and a $T$-bone overlay. Note we highlight that there are two parallel edges between $u_5$ and $d_2$ in the T-bone, but the original graph does not.



(b) The auxiliary graph and the corresponding $T \cup U \cup U'$-join overlay.

Figure 3.1: An example of $T$-bone computation where $|D| = 2$, $|U| = 5$ and $T = D$. The gray edges are the underlying graph, and the black edges are the overlay.

**Lemma 3.1** *For any $T'$ such that $U \cup U' \subseteq T'$ and $|T'|$ is even, the minimum cost $T'$-join $J^*$ in $G'$ such that $\deg_{J^*}(u) = 1$ for all $u \in U \cup U'$ can be found in $O(|V|^3)$.*

**Proof:** Here we present a constructive proof. First, find a minimum cost $T'$-join $J^*$ in $O(|V|^3)$ time. Because the graph is complete, it has only one component. Additionally, $|T'|$ is even. Therefore $J^*$ exists by Lemma 2.1. If there exists $u \in U \cup U'$ such that $\deg_{J^*}(u) > 1$, we can apply the splitting-off operation on $u$ with respect to $J^*$ until we get only one edge incident to $u$. Indeed, if $u \in U$, then it only incidents to vertices in $U \cup D$, and $G'[D \cup U]$ is a complete graph, so splitting-off is feasible. If $u \in U'$, then it only incidents to vertices in $D$, and $G'[D]$ is a complete graph, so splitting-off is also feasible. This operation can be applied at most $|E|$ times, as the number of edges decreases by one after each splitting-off operation. Due to the triangle inequality, splitting-off does not increase the total cost of the $T'$-join. Hence, we can find the desired structure in $O(|V|^3)$ time. □

**Lemma 3.2** *The minimum cost $T$-bone in $G$ can be found in $O(|V|^3)$.*

**Proof:** Let $T' = T \cup U \cup U'$. Consider an arbitrary minimum cost $T'$-join $J^{*\prime}$ in $G'$ such that $\deg_{J^*}(u) = 1$ for all $u \in U \cup U'$. Shrink $J^{*\prime}$ to $J$ by contracting each vertex $u \in U$ with the corresponding vertex $u' \in U'$. It is easy to check that $J$ is a $T$-bone in $G$ with $c'(J^{*\prime}) = c(J)$.

On the other hand, consider an arbitrary minimum cost $T$-bone $J^*$ in $G$, we can construct a $T'$-join $J'$ in $G'$ as follows:

1. start with $J' = J^*$;
2. for each vertex $u \in U$, find an edge $ud \in J'$ for some $d \in D$. Delete edge $ud$ from $J'$ and add edge $u'd$ to $J'$.

It is easy to check that $J'$ is a $T'$-join in $G'$ and $c'(J') = c(J^*)$.

From the above, the minimum cost $T$-bone can be found by solving the minimum cost $T'$-join problem on $G'$, which implies the minimum cost $T$-bone can be found in $O(|V|^3)$. □

```
MinCostTBone($G = (D, U, E), c, T$)
    $U' \leftarrow \{u' | u \in U\}$
    $V' \leftarrow D \cup U \cup U'$
    $E' \leftarrow E \cup \{u'v | u' \in U', v \in D\}$
    $G' \leftarrow (V', E')$
    $T' \leftarrow T \cup U \cup U'$
    $c'$ a function s.t. $c'(u'v) = c(uv)$ if $u \in U, v \in D$ and $c'(uv) = c(uv)$ if $u, v \in V$
    $J' \leftarrow$ minimum cost $T'$-join in $G'$ with cost $c'$
    while $\deg_{J'}(u) > 1$ for some $u \in U \cup U'$:
        splitting off $u$ arbitrarily in $J'$
    $J \leftarrow$ identify each $u$ and $u'$ in $J'$, where $u \in U$
    return $J$
```

Figure 3.2: The algorithm for computing a min-cost $T$-bone

## 4 Counting Constrained Forests

Recall we have a graph $G = (D, U, E)$, with metric cost function $c$ and a set of bases $B \subseteq D$. A *constrained forest* $F$ is a forest such that each non-trivial component of $F$ has exactly one vertex in $B$, $\deg_F(v) = 2$ for all $v \in U \cap V(F)$, and $F[U]$ is a matching.

We show a useful property of the constrained forests.

**Lemma 4.1** *Let $F$ be a constrained forest, each non-trivial component in the forest contains at least one vertex in $\overline{B}$.*

**Proof:** A non-trivial component of $F$ is a tree. A tree with at least 2 vertices contains at least 2 leaves. No vertex in $U$ can be a leaf because $\deg_F(u) = 2$ for all $u \in U \cap V(F)$. Hence all leaves are in $D$. Exactly one leaf is in $B$, hence at least one leaf has to be in $\overline{B}$. □

Intuitively, each non-trivial component in a constrained forest tries to match non-bases to a base. In the context of the vehicle routing, a constrained forest tries to specify a vehicle that visits a non-base depot. As a consequence, there can be at most $|\overline{B}|$ non-trivial components. See Figure 5.1a for an example of a constrained forest.

We proceed with a sequence of lemmas that eventually counts the number of constrained forests.

**Lemma 4.2** *Let $Y_1, \dots, Y_t$ be disjoint subsets of $D$ such that each set contains exactly one vertex in $B$ and at least one vertex in $\overline{B}$. Let $\overline{B}' = (\bigcup Y_i) \cap \overline{B}$ and $q = |\overline{B}'|$. The number of constrained forests $F$ with non-trivial components $C_1, \dots, C_t$, such that $C_i \cap D = Y_i$ is at most $(q+1)^{q-t} n^{2q}$.*

**Proof:** Let $F$ be a constrained forest with non-trivial components $C_1, \dots, C_t$ such that $C_i \cap D = Y_i$. We look at tree $T_i = F[C_i]$. $T_i$ contains $|Y_i|$ vertices in $D$, and at most $2(|Y_i|-1)$ vertices in $U$. Recall Cayley's formula, which states the number of labeled trees on $n$ vertices is $n^{n-2}$ [3]. We can fix a spanning tree on $Y_i$, and assign at most 2 vertices in $U$ on each edge. Since each vertex in $U$ has degree 2, we obtain a bijection from these labeled spanning trees to a non-trivial component in a constrained forest. Therefore, there can be no more than $|Y_i|^{|Y_i|-2}|U|^{2(|Y_i|-1)}$ trees that can act as the $i$th non-trivial component of a constrained forest satisfies the property in the lemma. Each non-trivial component is independent, hence we take their product.

$$\begin{aligned}
&\prod_{i=1}^{t} |Y_i|^{|Y_i|-2}|U|^{2(|Y_i|-1)} \\
\le &\prod_{i=1}^{t} (q+1)^{|Y_i|-2} n^{2|Y_i|-1} \\
= &(q+1)^{\sum_{i=1}^{t}(|Y_i|-2)} n^{2\sum_{i=1}^{t}|Y_i|-1} \\
= &(q+1)^{q-t} n^{2q}.
\end{aligned}$$

□

**Lemma 4.3** *Let $X \subseteq \overline{B}$. For $X_1, \dots, X_t$ a partition of $X$. The number of constrained forests $F$ with non-trivial components $C_1, \dots, C_t$, such that $C_i \cap \overline{B} = X_i$ is $O(k^t (q+1)^{q-t} n^{2q})$, where $q = |X|$.*

**Proof:** We pick a sequence of $t$ distinct vertices in $B$, $s_1, \dots, s_t$, there are $O(k^t)$ choices. For each of the choices, by the Lemma 4.2, we take $Y_i = X_i \cup \{s_i\}$, and obtain the number of constrained forests $F$ satisfies the condition of this lemma is $O(k^t (q+1)^{q-t} n^{2q})$. □

**Lemma 4.4** *Let $X \subseteq \overline{B}$ a set of $q$ elements. The number of constrained forests $F$ such that $V(F) \cap \overline{B} = X$ is $O(k^q (q+1)^{2q} n^{2q})$.*

**Proof:** There can be no more than $q^t$ ways to partition $X$ to $t$ classes. By Lemma 4.3, the total number of constrained forest is bounded by

$$\sum_{t=1}^{q} q^t k^t (q+1)^{q-t} n^{2q} = O(k^q (q+1)^{2q} n^{2q}).$$

□

**Lemma 4.5** *There are at most $O(k^p n^{2p})$ different constrained forests, where $p = |\overline{B}|$ and is a constant.*

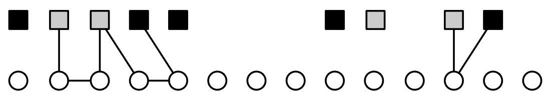
**Proof:** Take $X$ over all subsets of $\overline{B}$. Using Lemma 4.4, we have The number of constrained forests is therefore at most $\sum_{q=0}^{p} \binom{p}{q} O(k^q (q+1)^q n^{2q}) = O(2^p k^p (p+1)^p n^{2p}) = O(k^p n^{2p})$. □

## 5 The Algorithm for 2-MDCVRRP($p$)

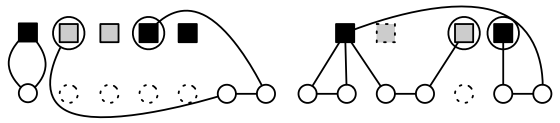
In this section, we solve 2-MDCVRRP($p$). Intuitively, the algorithm tries to first specify which vehicle goes to which non-base depots (and also which path it takes), and then ask for the optimal solution satisfies the constraints. Since we do not know which vehicle goes to the non-base depots, we will try all possibilities. A constrained forest encodes such information.

We define a particular induced subgraph $G_F$ of $G$. The new graph $G_F$ is obtained from $G$ by retaining the base depots $B$, non-base depots in $V(F)$, and remove all covered $U$ vertices in $V(F)$. Formally, let $D_F = B \cup (D \cap V(F))$, $U_F = U \setminus V(F)$, $V_F = D_F \cup U_F$, $E_F = E[V_F]$. We define $G_F = G[V_F] = (D_F, U_F, E_F)$. See Figure 5.1b for an example of $V_F$.

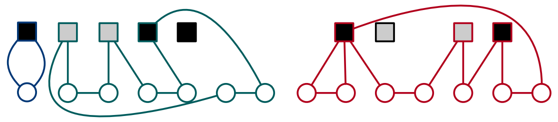
We say a constrained forest $F$ *spans* a feasible solution $W$ of 2-MDCVRRP($p$), if $F \subseteq W$, and $F$ covers the same set of non-base depots as $W$, i.e. $V(F) \cap \overline{B} = V(W) \cap \overline{B}$. Our algorithm exploits the fact that each feasible solution to 2-MDCVRRP($p$) is a union of a constrained forest $F$ and a $T$-bone for some suitably defined $T$ in an $G_F$. See Figure 5.1 for an example. This is the main goal of the next lemma.



(a) A constrained forest $F$ for graph $G$.



(b) Corresponding vertices in $V_F$ and the $T$-bone $J$. The dotted elements are not in $V_F$. The elements contained in a circle are elements in $T$.



(c) A feasible solution $W$ obtained as the union of constrained forest $F$ and $T$-bone $J$.

Figure 5.1: An example of how a solution is obtained. White circle are elements in $U$, black square are elements in $B$, and gray square are elements in $\overline{B}$.

**Lemma 5.1** *Every feasible solution $W$ contains a constrained forest that spans $W$.*

**Proof:** A path is *elementary* if it starts and ends in a vertex in $D$, and contains no other vertices in $D$. Let $P_1, \ldots, P_m$ be all the elementary paths in $W$. The fact that $\deg_W(u) = 2$ for all $u \in U$ shows $P_1, \ldots, P_m$ partitions the edges of $W$. We define $W_0 = W$, and

$$W_i = \begin{cases} W_{i-1} & \text{if } \overline{B} \cap V(W_{i-1} \setminus P_i) \neq \overline{B} \cap V(W) \\ W_{i-1} \setminus P_i & \text{otherwise} \end{cases}.$$

Intuitively, we remove paths such that the set of non-base depots covered does not change. Let $F = W_m$, we want to show it is a constrained forest and it is contained in $W$. Note that $\deg_{W_i}(v) = 2$ for all $v \in U \cap V(W_i)$, $V(W_i) \cap \overline{B} = V(W) \cap \overline{B}$, and $W_i[U]$ is a matching for all $i$. We show that each non-trivial component of $F$ has exactly one vertex in $B$.

Note that if $P_i \subseteq F$, then $V(F \setminus P_i) \cap \overline{B} \subsetneq V(F) \cap \overline{B}$. Because otherwise, $P_i$ would not be present in $W_i$. Assume that $F$ has a component $C$ that contains more than one vertex in $B$, say $b$ and $b'$. Consider a path in $F$ from $b$ to $b'$, which can be decomposed into a sequence of elementary paths, let one of them be $P_i$. Now, we claim $\overline{B} \cap V(F \setminus P_i) = \overline{B} \cap V(F)$. Once we remove $P_i$, any vertices in $C \setminus \overline{B}$ is still in $V(F \setminus P_i)$. Any vertex in $C \cap \overline{B}$ can reach either $b$ or $b'$ after removal of $P_i$. Hence $V(F) \cap \overline{B} = V(F \setminus P_i) \cap \overline{B}$, a contradiction. Similarly one can show there is no cycles in $F$. Assume there is a cycle and it contains some $P_i$, then one can see $V(F \setminus P_i) \cap \overline{B} = V(F) \cap \overline{B}$. A contradiction. □

**Lemma 5.2** *Let $W$ be a feasible solution to 2-*MDCVRRP*$(p)$ of $G = (D, U, E)$. Let $F$ be a constrained forest that spans $W$. Let $J = W \setminus F$, then $J$ is a $T$-bone of $G_F$, where $T = Odd(F)$.*

**Proof:** All 3 properties of $J$ as a $T$-bone of $G_F$ is satisfied. Indeed, $J$ is a $T$-join because all vertices in $W$ has even degree, thus $Odd(W \setminus F) = Odd(F) = T$. $\deg_J(u) = \deg_W(u) = 2$ for all $u \in U_F$. $J[U_F]$ is a matching because $W[U]$ is a matching. □

Finally, we extend the above lemma to characterize the optimal solution of 2-MDCVRRP$(p)$.

**Lemma 5.3** *Let $W^*$ be an optimal solution to 2-*MDCVRRP*$(p)$ of $G = (D, U, E)$. Let $F$ be a constrained forest that spans $W^*$. Let $T = Odd(F)$. Let $J$ be a minimum cost $T$-bone of $G_F$, then $W = J \cup F$ is an optimal solution to 2-*MDCVRRP*$(p)$ on $G$.*

**Proof:** By Lemma 5.2, $W^* \setminus F$ is a $T$-bone of $G_F$. Hence we have $c(J) \leq c(W^*) - c(F)$. Note that $J \cup F$ is a feasible solution to 2-MDCVRRP$(p)$ of $G$. $W$ is also a solution to 2-MDCVRRP$(p)$ of $G$, therefore $c(W^*) \leq c(W)$. It follows that

$$c(W^*) \leq c(W) = c(J) + c(F) \leq c(W^*) - c(F) + c(F) = c(W^*)$$

Therefore we have shown $W$ is an optimal solution. □

```
SOLVE($G = (D,U,E), c$)
  for $F$ a constrained forest in $G$:
      $V_F \leftarrow B \cup (\overline{B} \cap V(F)) \cup (U \setminus V(F))$
      $T \leftarrow Odd(F)$
      $G_F \leftarrow G[V_F]$
      $J \leftarrow$ MINCOSTTBONE($G_F, c|_{E[V_F]}, T$)
      $W \leftarrow J \cup F$
      add $W$ into the list of candidates
  return the minimum cost candidate
```

Figure 5.2: The algorithm for solving the 2-MDCVRRP.

The minimum cost $T$-bone in the previous theorem always exists because the graph is a complete graph and $|T|$ is even. At this point, we obtain the algorithm in Figure 5.2. Next, we analyze its running time.

**Theorem 5.4 (Main Theorem)** *2-MDCVRRP(p) can be solved in $O(|V|^3 \cdot k^p n^{2p})$ time, where k is the number of depots and n is the number of customers.*

**Proof:** Let $\mathcal{F}$ be the set of constrained forests of $G$. By Lemma 5.3, the optimal solution can be found by enumerating all the possible constrained forests and finding minimum cost $Odd(F)$-join in $G_F$. Hence Figure 5.2 is correct. The for-loop ran $|\mathcal{F}|$ time. The running time inside the for-loop is dominated by finding the $T$-bone, which takes $O(|V|^3)$ time. Together, we obtain the desired running time of $O(|V|^3|\mathcal{F}|)$ time. By Lemma 4.5, there are at most $O(k^p n^{2p})$ different constrained forests, and we obtain the desired running time. □

In particular, because $p \leq k-1$, this implies if the number of depots $k$ is a constant, we also have a polynomial time algorithm.

**Theorem 5.5** *Let $k$ be a constant. 2-MDCVRRP can be solved in $O(n^{2k+1})$ time if there is at most $k$ depots.*

This matches the previous best result for a single depot [2]. We can also show if the number of non-base depots is unbounded, the problem is NP-hard.

**Theorem 5.6** *2-MDCVRRP is NP-hard in general.*

**Proof:** Indeed, every TSP instance can transform into a 2-MDCVRRP instance. For each point in TSP, we create a depot and two customers at the same location. Let there be exactly one base. It's easy to see the optimal 2-MDCVRRP tour would reflect an optimal tour in the TSP. □

## References

[1] C. Archetti and M G Speranza. Vehicle routing in the 1-skip collection problem. *Journal of the Operational Research Society*, 55(7):717–727, Jul 2004.

[2] Tetsuo Asano, Naoki Katoh, Hisao Tamaki, and Takeshi Tokuyama. Covering points in the plane by $k$-tours: Towards a polynomial time approximation scheme for general $k$. In *Proceedings of the Twenty-ninth Annual ACM Symposium on Theory of Computing*, STOC '97, pages 275–283, New York, NY, USA, 1997. ACM.

[3] A. Cayley. A theorem on trees. *Quart. J. Math.*, 23:376–378, 1889.

[4] William J. Cook, William H. Cunningham, William R. Pulleyblank, and Alexander Schrijver. *Combinatorial Optimization*. John Wiley & Sons, Inc., 2011.

[5] Jack Edmonds and Ellis L. Johnson. Matching, euler tours and the chinese postman. *Mathematical Programming*, 5(1):88–124, Dec 1973.

[6] Bernhard Korte and Jens Vygen. *Combinatorial Optimization: Theory and Algorithms*. Springer, 2007.

[7] L Lovász. *Combinatorial Problems and Exercises*. Elsevier, 1994.

[8] Christos H. Papadimitriou. The Euclidean travelling salesman problem is NP-complete. *Theoretical Computer Science*, 4(3):237 – 244, 1977.