

Capacitated Vehicle Routing with Target Geometric Constraints

Kai Gao

Jingjin Yu

Abstract—We investigate the capacitated vehicle routing problem (CVRP) under a robotics context, where a vehicle with limited payload must complete delivery (or pickup) tasks to serve a set of geographically distributed customers with varying demands. In classical CVRP, a customer location is modeled as a point. In many robotics applications, however, it is more appropriate to model such “customer locations” as 2D regions. For example, in aerial delivery, a drone may drop a package anywhere in a customer’s lot. This yields the problem of CVRG (Capacitated Vehicle Routing with Target Geometric Constraints). Computationally, CVRP is already strongly NP-hard; CVRG is therefore more challenging. Nevertheless, we develop fast algorithms for CVRG, capable of computing high quality solutions for hundreds of regions. Our algorithmic solution is guaranteed to be optimal when customer regions are convex. Numerical evaluations show that our proposed methods significantly outperform greedy best-first approaches. Comprehensive simulation studies confirm the effectiveness of our methods.

I. INTRODUCTION

The Capacitated Vehicle Routing Problem (CVRP) naturally arises in many robotics applications, e.g., autonomous truck routing on road networks [1], aerial delivery [2], [3], clutter removal [4], and so on. In CVRP, a vehicle starting at a *depot* with limited load capacity is tasked to deliver (resp., pick up) goods to (resp., from) geographically scattered *customers*. The goal is to minimize the total distance that the vehicle must travel to complete all delivery/pickup tasks. CVRP is closely related to the *vehicle routing problem* (VRP) [5] and the *traveling salesperson problem* (TSP) [6], both of which are NP-hard problems that have been studied extensively [7], [8]. In the literature, effective solutions to CVRP have mainly been based on *branch-and-bound*, *branch-and-cut*, and related methods [9]–[13].

In this work, we examine a variant of CVRP when customer locations have non-trivial geometry and study the resulting optimality structure. In this CVRP variant, instead of having a point location, each customer now occupies a contiguous 2D region. We denote the problem as **Capacitated Vehicle Routing with (non-trivial target) Target Geometric Constraints (CVRG)**. CVRG accurately models multiple real-world application scenarios. For example, in aerial delivery (or pickup), a parcel may be dropped (picked up) anywhere inside a given region (Fig. 1). Similarly, optimal algorithms for CVRG can provide the most desirable solutions for helicopter-based rescuing missions where people are isolated on “islands” during floods or other natural disasters.

Our algorithmic attack on the CVRG problem seeks to decouple the CVRP element and the additional constraint

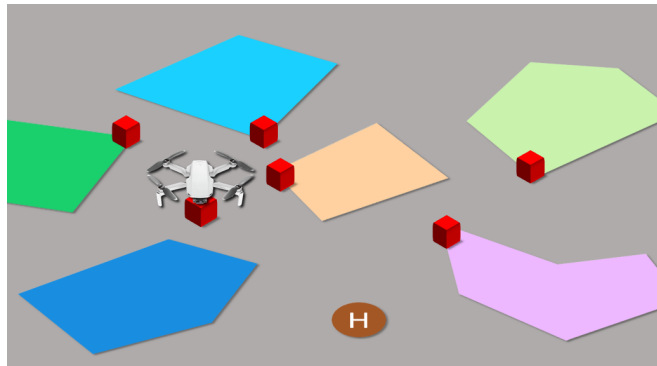


Fig. 1: An CVRG application where an aerial vehicle with limited load capacity is tasked to deliver packages to multiple regions. For a given region, the package may be dropped at any point inside the region. To make fast deliveries to all regions, the vehicle must carefully plan multiple tours with optimized delivery locations, which is computationally intractable. “H” denotes the depot.

induced by non-trivial customer location geometry. On the CVRP side, we examine existing state-of-the-art linear programming based solvers and develop a combinatorial algorithm that uses dynamic programming (DP) to chain together individual tours (a *tour* is a round trip of the robot starting from the depot; generally, multiple tours are required to solve a CVRP). On the geometric constraint side, we prove that optimizing the delivery/pickup locations over a tour with fixed customer sequence induces an optimization problem that can be solved efficiently when customer regions are convex, allowing us to construct efficient subroutines for computing the corresponding optimal tour. A hierarchical combination of the CVRP subroutines and the geometric optimization subroutines prove to be highly effective when compared with greedy best-first approaches, yielding solutions with much higher quality. At the same time, we note that CVRP is strongly NP-hard even when customer demand is lower bounded, which we prove via a reduction from **3-PARTITION** [14]. The effective algorithmic solutions developed for CVRG, in particular an optimal solution method for convex customer regions and a fast, high-quality finite horizon heuristic algorithm based on DP, form the main contributions of this work. The algorithms developed in this paper are thoroughly evaluated in simulation studies, including a physics engine based drone delivery scenario, which confirm their effectiveness.

Beside its relevance to CVRP, CVRG is closely related to research on object rearrangement in robotics. A diverse set of methods have been applied to tackle object rearrangement tasks, including search based approaches [15] and symbolic reasoning based approaches [16]. In contrast, in [17], [18], more focus is put on taming the combinatorial explosion

caused by the large number of objects. There, even the seemingly simple problem of rearranging unlabeled objects turns out to be NP-hard if an optimal solution is sought after [18], which echoes the computational challenge we face in the current study. From an application perspective, our work applies to scenarios including aerial delivery [2], [3], disaster response [4], [19], [20], among others. CVRG can also model the truck behavior in the truck-drone collaborative delivery problems [21] [22] [23]. For each delivery, the truck only needs to reach the neighborhood of the customer and let the drone do the last-mile delivery.

Organization. The rest of the paper is organized as follows. In Section II, CVRG is stated, followed by a preliminary structural analysis and a hardness proof. In Section III, we describe in detail our proposed algorithmic solutions for CVRG, which are thoroughly evaluated and compared in Section IV. We conclude in Section V.

II. PRELIMINARIES

A. CVRP with Target Geometric Constraints (CVRG)

In a standard *capacitated vehicle routing problem* (CVRP) [24], a robot (vehicle) with a fixed load capacity is tasked to transport goods to multiple customers from a depot. More formally, let $d \in \mathbb{R}^2$ be the location of a depot where a robot may carry a maximum load of W and deliver different loads to a set of n customers located at c_1, \dots, c_n , respectively, with $c_i \in \mathbb{R}^2$ for $1 \leq i \leq n$. Each customer has a demand “weight” of w_i that must be satisfied through delivery by the robot.¹ The robot returns to the depot after all deliveries are finished. As a problem that is always feasible, the goal in solving CVRP is to minimize the total distance traveled by the robot. The problem of picking up goods is symmetrical to the problem of delivering goods. In presenting this work, we mainly use the delivery setup but note that the pickup setup is also used when appropriate and is equivalent.

CVRP generally requires a customer’s demand to be met by a single delivery. This makes sense as each delivery operation itself will incur a non-trivial overhead (for both parties). In robotics operations, the overhead can be dropping of packages from air or grasping an object. We mention that the results that we develop also apply if the environment is modeled as a graph, e.g., as a road network. When it comes to robotics applications, e.g., in aerial delivery, the customer location could have non-trivial geometry. We model such constraints by treating each delivery location as a simply-connected polygon.

Problem 1 (CVRP with Target Geometric Constraints (CVRG)). *Let $d \in \mathbb{R}^2$ be the location of a depot with unlimited supply. There are n customers located in P_1, \dots, P_n , where $P_i \subset \mathbb{R}^2$ is a simply-connected polygon. Customer i has a demand of $w_i \in (0, 1]$, which is satisfied as a single supply of w_i is delivered to any point $p \in P_i$. Find the minimum total distance required for a robot with unit capacity to complete all deliveries, starting from and ending at the depot.*

¹Because the absolute values of W and w_i do not matter, we set $W = 1$ in this paper, which means that $w_i \in (0, 1]$.

In CVRG, it is not required that for $1 \leq i, j \leq n$, $P_i \cap P_j = \emptyset$, i.e., P_i and P_j may overlap. Whereas the non-overlapping case is suitable for applications like aerial delivery, the overlapping setup can be more suitable for applications like picking up objects which may fall on one another. We explicitly address the case where elements of $\{P_i\}$ overlap.

B. Strong NP-Hardness of CVRP/CVRG

The classical VRP problem is computationally intractable for large instances as the NP-hard TSP readily reduces to it [7]. The same can be shown for CVRP using similar arguments. However, solvers exist that can solve very large instances of TSP near optimally very fast. CVRP, on the other hand, proves much more challenging. We observe that CVRP is in fact strongly NP-hard, even when customer demand is lower bounded, i.e., the customer’s demand is no smaller than $1/k$ for some integer k . Setting k to be 4 is sufficient to show strong NP-hardness. To show this, we first introduce the strongly NP-hard 3-PARTITION problem [14].

PROBLEM: 3-PARTITION

INSTANCE: A finite set A of $3m$ elements, a bound $B \in \mathbb{Z}^+$, and a “size” $s(a) \in \mathbb{Z}^+$ for each $a \in A$, such that each $s(a)$ satisfies $B/4 < s(a) < B/2$ and $\sum_{a \in A} s(a) = mB$.

QUESTION: Is there a partition of S into m disjoint subsets S_1, \dots, S_m such that for $1 \leq i \leq m$, $\sum_{a \in S_i} s(a) = B$?

Theorem II.1. *CVRP is strongly NP-hard.*

Proof. Given a 3-PARTITION instance with $3m$ elements, we construct a CVRP instance as follows. In \mathbb{R}^2 , let $d = (0, 0)$ be the depot and let there be $3m$ customers all located close to $(1, 0)$. That is, for any customer i , $1 \leq i \leq 3m$, $c_i \in B_\varepsilon(1, 0)$, which is the ε ball around the point $(1, 0)$ for some positive $\varepsilon \ll 1/m$. Let the robot has a unit capacity and let customer i has a demand of $s(a_i)/B$. Since $s(a) > B/4$, each delivery can supply at most 3 customers. To show that CVRP is NP-hard, we will show that the 3-PARTITION problem admits an optimal partition if and only if the CVRP problem admits a total travel distance of no more than $2m + 6m\varepsilon$.

For the “only if” part, if the 3-PARTITION problem admits a partition of the $3m$ elements into m sets S_1, \dots, S_m of three elements each such that $\sum_{a \in S_i} s(a) = B$, then a robot can complete all deliveries using a total of m tours based on the partition. For the i -th tour, the robot may travel in a straight line to a customer in S_i , which incurs a distance of no more than $1 + \varepsilon$. Then, the robot will travel along straight lines to the other two customers in S_i , one after the other. These tours will incur a cost of no more than 2ε each. Finally, the robot can return to the depot with a distance of no more than $1 + \varepsilon$. The total cost is then no more than $2m + 6\varepsilon$.

For the “if” part, if the CVRP problem admits a solution with a cost of no more than $2m + 6m\varepsilon$. Since $\varepsilon \ll 1/m$, $2m + 6m\varepsilon < 2m + 1$ and the robot can only make no more than m tours from the depot and return. Since the robot can move at most a unit of supply to the customers per round tour, the robot can move at most a total supply of m . As the CVRP problem is solved, this means that the robot must complete

delivery to exactly 3 customers since (1) partial deliveries are not allowed and (2) at most three customers' demand can be fulfilled with a unit of supply. \square

With CVRP (and subsequently, CVRG) being strongly NP-hard, it does not admit an FPTAS (fully polynomial-time approximation scheme) unless $P = NP$ [25]. That is, it is unlikely that efficient algorithms exist for solving CVRP/CVRG approximately optimally.

III. OPTIMALLY SOLVING CVRG

Our algorithmic solution for CVRG has two components: a *dynamic programming* (DP) based algorithm for CVRP (Section. III-A) and a geometric optimization subroutine (Section. III-B) that efficiently compute optimal single tours in the presence of the geometric constraint. Our DP algorithm for CVRG is guaranteed to be optimal when the customer regions are convex. We also examine the relevant scenario when regions represent physical objects to be picked up and the objects have overlaps (Section. III-C).

A. Exact and Finite-Horizon DP Algorithms for CVRP

Exact combinatorial algorithm. Our exact algorithm for CVRP proceeds in two phases. In the first phase, we exhaustively compute the optimal costs c_s of all valid tours starting from the depot that visits a subset of customers s . It is easy to see that there are at most 2^n tour combinations, many of which will be invalid as they will exceed the capacity limit. For each valid tour combination, the optimal cost is computed using an exact TSP solver [26]. The costs are then stored for the second phase of the computation.

In the second phase, to select the optimal set of disjoint tours, a straightforward application of dynamic programming (DP) is used. Let C be the set of all customers. Let $I \subset C$ be the set of customers that have not been served and let S_I be the set of customer subsets that can be visited in a single tour without violating the capacity constraint. Let J_I be the optimal cost (i.e., the minimum total path length) to satisfy the demands of all of I , the standard DP recursion is

$$J_I = \min_{s \in S_I} (J_{I-s} + c_s). \quad (1)$$

Note that c_s is provided by the first phase through a direct look-up. The DP algorithm provides significant computational savings by storing the optimal cost of all possible I 's. For n customers, there are at most 2^n such I 's. The algorithm, denoted DP-CVRP, is straightforward to implement: we simply create a large enough table to hold the 2^n entries and then iteratively populate the entries.

Algorithm analysis. In the first phase, there are at most 2^n tours to examine. Since each tour visits no more than n elements, obtaining the optimal cost of a tour takes $O(T(n))$ time, where $T(n) = O(2^n)$ even with a brute force TSP solver. The first phase then takes time $O(2^n T(n))$.

In the second phase, the incremental DP computation needs to go through 2^n possible I 's, starting from $I = \emptyset$ and eventually reach $I = C$. For each I , S_I contains $O(2^{|I|})$ potential tours that need to be checked. The second phase

then requires $O(2^n 2^n)$, which dominates the computation time of the first phase. The overall computational complexity of DP-CVRP is then bounded by $O(2^{2n}) = O(4^n)$.

With the application of DP cutting down the computation from a naive $O(n!f(n))$ (where $f(n)$ is some polynomial function of n for computing the optimal cost of a single sequence) to $O(4^n)$, significantly larger CVRP instances can be solved optimally. As we will demonstrate, DP-CVRP is exact. More importantly, DP-CVRP readily allows the integration of additional geometric constraints in the CVRG formulation.

Finite-horizon heuristic. While DP-CVRP computes exact solutions with decent performance, the computation time is still exponential with respect to n . To address this issue, we introduce a finite horizon heuristic which restricts the number of customers examined at any given time. That is, some fixed h ($h \leq n$) customers are selected on which DP-CVRP is run. Because high quality solutions to CVRP are generally clustered (see, e.g., Fig. 13 and Fig. 14), these h candidates are selected in a way to facilitate such clustering. From running DP-CVRP on the h candidates, we pick two most "convenient" tours with the least average cost (path length divided by the number of served customers), serve these customers, and repeat the process on the remaining customers. The heuristic algorithm, which we denote as FH-DP-CVRP, has an apparent complexity of $O(n4^h)$, which is polynomial.

B. Optimal Subtours Crossing Multiple Regions

For the first phase of DP-CVRP, the optimal cost for a single tour is computed by solving a TSP with DP, where a certain DP property holds: given five customers c_1 to c_5 , and two paths $c_1c_2c_3c_4c_5$ and $c_1c_2c_3c_5c_4$, the optimal cost of $c_1c_2c_3$ is the same in both and the computation for this part only needs to be done once. The property breaks as we compute an optimal tour *passing through* a set of convex polygons, as is required by CVRG. As an example, consider the setup in Fig. 2 with six rectangles and assume the robot starts from P_1 and end at P_6 . For the visiting sequence $P_1P_2P_3P_4P_5P_6$, the blue solid path is the optimal local path. On the other hand, for the visiting sequence $P_1P_2P_3P_5P_4P_6$, the red dashed path is optimal. We observe that the partial path between P_1 and P_3 cannot be reused.

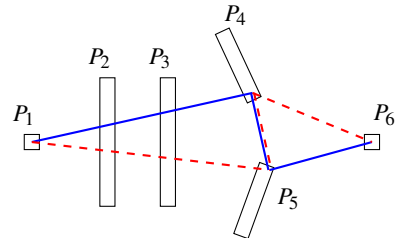


Fig. 2: Dynamic programming property existing in TSP tour computation fails to hold as the vertices become polygons.

Though there is a lack of partial ordering in optimal tour computation for CVRG, for a tour with k sites where k is not large, which is the case due to limited robot capacity, sifting through all $k!$ sequences is doable provided that the optimal

tour based on each sequence can be computed efficiently. This turns out to be the case when the regions are convex; all that is required is to iteratively improve a tour locally, as outlined in Algorithm 1. The sub-routine LOCAL-IMPROV attempts to move p_i on the polygon P_i to shorten the existing tour length. We call the algorithm an *elastic improvement* as the process acts as fixing two endpoints of an elastic band and let a point in the middle “slide” along a line to a low energy configuration.

Algorithm 1: ELASTIC-IMPROV (Iterative Elastic Improvement)

Input : P_1, \dots, P_k : polygon sequence; d : depot location
Output : τ : the optimal tour, initially empty

```

1 for each  $P_i$ ,  $1 \leq i \leq k$  do
2    $p_i \leftarrow$  a random point on  $P_i$ ;
3    $\tau \leftarrow \tau + p_i$ ; % ``+''' denotes appending
4  $\tau' = \emptyset$ ;
5 while  $\tau \neq \tau'$  do
6    $\tau' \leftarrow \tau$ ;  $\tau \leftarrow []$ ;
7   for  $1 \leq i \leq k$  do
8      $\tau \leftarrow \tau + \text{LOCAL-IMPROV}(i, \tau, \tau', P_i, d)$ ;
9 return  $\tau$ ;
```

To see that Algorithm 1 is globally optimal when the polygons P_1, \dots, P_k are convex, we first examine the case where each P_i is a line segment L_i . We start with defining the terms *refraction*, *reflection*, *crossing*, and *mirror-reflection*. Without loss of generality, we assume that a tour T never overlaps with a line segment L_i as the probability of a shortest tour overlapping with a line segment is zero.

Definition III.1 (Refraction, Reflection, Crossing, and Mirror-Reflection). Let a line segment L and a tour T intersect at p . We say T is *refracted* by L at p if the two line segments ℓ_1 and ℓ_2 of T meeting at p lie on different sides of L . If ℓ_1 and ℓ_2 are co-linear, then the refraction is a *crossing*. Otherwise, if ℓ_1 and ℓ_2 lie on the same side of L , T is *reflected* by L . A reflection is a *mirror-reflection* if the bisector of the angle between ℓ_1 and ℓ_2 is perpendicular to L .

These definitions are illustrated in Fig. 3.

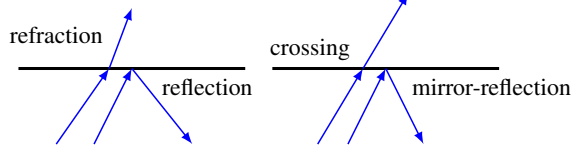


Fig. 3: Illustrations of refraction, reflection, crossing, and mirror-reflection. The bold horizontal line is a line segment L and the line segments with arrows are part of a tour T .

Definition III.2 (Elastic Band). Given a point $d \in \mathbb{R}^2$ and a set of line segments L_1, \dots, L_k , we say a tour T starting at d and passing through L_1, \dots, L_k in that order, at locations p_1, \dots, p_k , is an elastic band if for any i , $1 \leq i \leq k$, one of the following holds: 1) T is mirror-reflected by L_i at p_i , 2) T crosses L_i at p_i , or 3) p_i is an end point of L_i and the angle formed by the two line segment of T meeting at p_i and enclosing L_i is no less than π .

The conditions specified in the definition of elastic bands

are necessary conditions for a tour to be locally optimal (shortest). We note that a tour T may intersect an L_i at more than one point. However, since the visiting order of L_1, \dots, L_k is fixed, p_i 's are uniquely defined. It is clear that a shortest tour T going through L_1, \dots, L_k , in that order, must be an elastic band: if at some p_i the elastic band properties are not satisfied, T can be readily shortened.

Lemma III.1. A shortest tour T starting at a depot d and going through line segments L_1, \dots, L_k , in that order, at points p_1, \dots, p_k , must be an elastic band.

Next, we will show uniqueness of elastic bands, which yields global optimality. The main idea behind the proof of uniqueness is to establish that two different optimal elastic bands going through the same set of line segments cannot meet back at the starting point (depot d). Showing this requires detailed cases analysis. We start with some necessary terminologies, definitions, and intermediate lemmas.

Let T be a tour intersecting line segment L_i at p . As shown in Fig. 4, pp' is part of L_i , where p may be an endpoint of L_i . If T is an elastic band, then we have $\theta_1 \leq \theta_2$ in the setup given in Fig. 4a and $\theta_3 \leq \theta_4$ in the setup given in Fig. 4b. Otherwise, p can be moved toward the middle of pp' to reduce the length of T . Specifically, $\theta_1 < \theta_2$ and $\theta_3 < \theta_4$ only when p is an endpoint of L_i .

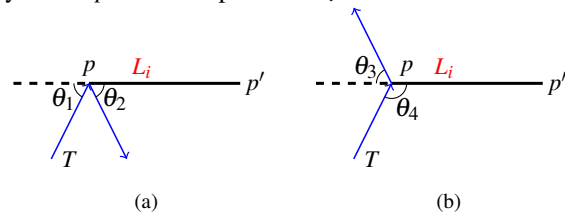


Fig. 4: Given an elastic band T_1 and the solid line is part of the line segment L_i , we have $\theta_1 \leq \theta_2$, $\theta_3 \leq \theta_4$.

Given an elastic band T_j starting at d and intersecting with line segments L_1, \dots, L_k in the given order at $p_{j,1}, \dots, p_{j,k}$, respectively, we define a ray $R_{j,i}$ as follows.

$$R_{j,i} = \begin{cases} \text{ray } \overrightarrow{dp_{j,1}}, & \text{for } i = 0 \\ \text{ray } \overrightarrow{p_{j,i}p_{j,i+1}}, & \text{for } 0 < i < k \\ \text{ray } \overrightarrow{p_{j,k}d}, & \text{for } i = k \end{cases}$$

For discussing the relationship between two elastic bands, we introduce a definition *convergence angle* v for rays.

Definition III.3. Let R_1 and R_2 be two rays with direction vectors \vec{n}_1 and \vec{n}_2 , respectively. If R_1 and R_2 intersect, then the *convergence angle* v is defined to be the angle between \vec{n}_1 and \vec{n}_2 , i.e. $\arccos(\vec{n}_1, \vec{n}_2)$, as θ_1 in Fig. 5a. On the other hand, if the two rays do not intersect, then the convergence angle v is defined as $-\arccos(\vec{n}_1, \vec{n}_2)$ (e.g., $-\theta_2$ in Fig. 5b).

We note that the later case in the definition of the convergence angle includes two sub cases: two rays intersecting on their extensions (along the negative directions of the rays) or one ray intersecting the extension of another ray. According to the definition, two different rays intersect if and only if they have a positive convergence angle.

Definition III.4. Let T_1 and T_2 be two elastic bands going

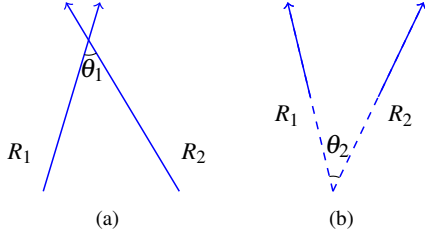


Fig. 5: Two possible arrangements of two rays. In the second case, the intersection may be between one ray and the extension of a second ray.

through a line segment L_i containing rays $\overrightarrow{p_{1,i}p_{1,i+1}}$ and $\overrightarrow{p_{2,i}p_{2,i+1}}$, respectively. When the two outgoing rays emit from L_i from the opposite sides and it holds that

$$\angle p_{1,i+1}p_{1,i}p_{2,i} + \angle p_{2,i+1}p_{2,i}p_{1,i} < \pi, \quad (2)$$

we say that the rays form a *zigzag* with respect to L_i .

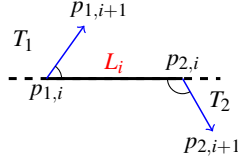


Fig. 6: A zigzag is the case when the two tours leaving a line segment from the opposite sides and $\angle p_{1,i+1}p_{1,i}p_{2,i} + \angle p_{2,i+1}p_{2,i}p_{1,i} < \pi$.

Lemma III.2. *Given two elastic bands T_1 and T_2 going through line segment L_1 containing rays $\overrightarrow{p_{1,1}p_{1,2}}$ and $\overrightarrow{p_{2,1}p_{2,2}}$, respectively. If L_1 is the first line segment in the visiting order, then the rays $\overrightarrow{p_{1,1}p_{1,2}}$ and $\overrightarrow{p_{2,1}p_{2,2}}$ cannot form a zigzag with respect to L_1 .*

Proof. Since L_1 is the first object in the visiting order, both T_1 and T_2 come from the same point d before going through L_1 . We prove the lemma by showing that d cannot exist if $\overrightarrow{p_{1,1}p_{1,2}}$ and $\overrightarrow{p_{2,1}p_{2,2}}$ form a zigzag with respect to L_1 .

Denote the intersection of elastic band T_i and line segment L_j by $p_{i,j}$. If a zigzag forms about L_1 , then the inequality (2) holds, i.e. $\angle p_{1,2}p_{1,1}p_{2,1} + \angle p_{2,2}p_{2,1}p_{1,1} < \pi$. Without loss of generality, we may assume $\angle p_{1,2}p_{1,1}p_{2,1} < \pi/2$. Due to the properties of elastic bands, d must be in the shaded area as shown in Fig. 7a, where $p'_{1,0}p_{1,1}p_{1,2}$ is a crossing and $p''_{1,0}p_{1,1}p_{1,2}$ is a mirror-reflection. Once $\angle p_{2,2}p_{2,1}p_{1,1}$ is fixed, there will be another shaded area corresponding to T_2 , and d has to be in the intersection of the two areas. To allow overlapping between the areas, $\angle p_{2,2}p_{2,1}p_{1,1}$ must be larger than $\pi/2$ and the dash area is as shown in Fig. 7b. With the inequality (2), the aforementioned two shaded areas must be disjoint, which leads to the nonexistence of d . \square

Lemma III.3. *Consider two elastic bands T_1 and T_2 , v_i is the convergence angle of the corresponding rays $R_{1,i}$ and $R_{2,i}$. It is impossible to have a zigzag for an object L_j if $v_i \in (-\pi, 0]$ holds for all $i = 1, 2, \dots, j-1$.*

Proof. Assuming the contrary, then (2) holds. Without loss of generality, we can assume $\angle p_{1,j+1}p_{1,j}p_{2,j} < \pi/2$.

- 1) When $\angle p_{2,j+1}p_{2,j}p_{1,j} < \pi/2$, by the assumption $-\pi < v_{j-1} \leq 0$, it is easy to verify that the tours

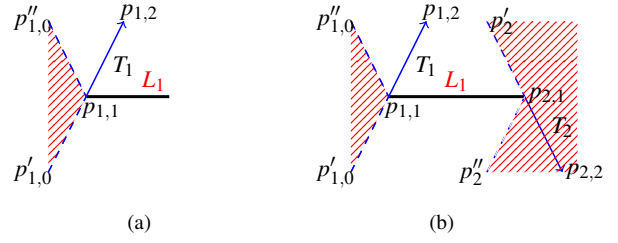


Fig. 7: If T_1 and T_2 go through L_1 with a zigzag, then there is no proper location for the depot d .

must go through object L_{j-1} with a zigzag and $\angle p_{1,j}p_{1,j-1}p_{2,j-1}$, $\angle p_{2,j}p_{2,j-1}p_{1,j-1} < \pi/2$. With the assumption that $v_i \in (-\pi, 0]$ always holds for all $i = 1, 2, \dots, j-1$, recursively, the two tours must leave the first object with a zigzag, contradicting Lemma III.2.

- 2) When $\angle p_{2,j+1}p_{2,j}p_{1,j} \geq \pi/2$, there are four cases: a) Both T_1 and T_2 pass L_j with refractions, b) Both T_1 and T_2 pass L_j with reflections, c) T_1 passes L_j with a reflection and T_2 passes L_j with a refraction, or d) T_1 passes L_j with a refraction and T_2 passes L_j with a reflection. For the first case, since $v_{j-1} \leq 0$, L_{j-1} intersects L_j as shown in Fig. 8 where $\theta_3 < \theta_1$, $\theta_4 < \theta_2$, we must have $\theta_3 + \theta_4 < \pi$. Therefore, the tours emanate from object L_{j-1} with a zigzag. Recursively, it leads to a contradiction to the statement of Lemma III.2. The other cases can be verified with similar reasoning. \square

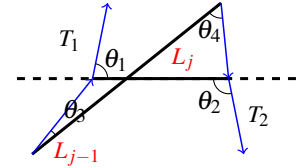


Fig. 8: When $\angle p_{2,j+1}p_{2,j}p_{1,j} \geq \pi/2$, a contradiction arises.

Theorem III.1. *For a tour starting from the depot and passing through a set of straight line segments in a fixed order, its length has a unique global minimum as realized by a elastic band.*

Proof. Assuming the contrary, then we can let T_1 and T_2 be two different elastic bands for the problem. Recall that v_i is the convergence angle of rays $\overrightarrow{p_{1,i}p_{1,i+1}}$ and $\overrightarrow{p_{2,i}p_{2,i+1}}$, where $p_{t,i}$ is the intersection between tour t and line segment L_i . Based on the fact that both tours share the same depot, T_1 and T_2 must diverge at some point and converge before the destination, i.e. $\exists(i, j)$, s.t. $i < j$, $v_i < 0 < v_j$. We show it is impossible to have the first j of this kind. By definition, the range of convergence angle is $[-\pi, \pi]$. Since the probability of tours overlapping with a line segment object is zero, we may assume that $v_i, v_j \in (-\pi, \pi)$.

There are three cases which may allow $v_j > 0$: C-1: T_1 and T_2 are from the same side of L_j and pass with refractions; C-2: T_1 and T_2 are from the same side of L_j and pass with reflections; and C-3: T_1 and T_2 are from different sides of L_j and with different types of passing.

For case C-1 (as Fig. 9), $v_{j-1} = -(\pi - \theta_2 - \theta_4) = -\pi + \theta_2 + \theta_4 \leq 0$ and $v_j = \pi - (\pi - \theta_1) - (\pi - \theta_3) = -\pi + \theta_1 + \theta_3$.

With the properties of elastic bands, we have $0 < \theta_1 \leq \theta_2$, $0 < \theta_3 \leq \theta_4$, therefore $-\pi < v_j \leq v_{j-1} \leq 0$.

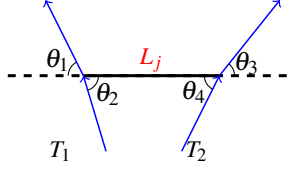


Fig. 9: When $-\pi < v_{j-1} \leq 0$, if two elastic bands T_1 and T_2 approach object L_j from the same side and form refractions, then $-\pi < v_j \leq 0$.

With similar reasoning, we can verify that $-\pi < v_j \leq 0$, when $-\pi < v_{j-1} \leq 0$ in case C-2.

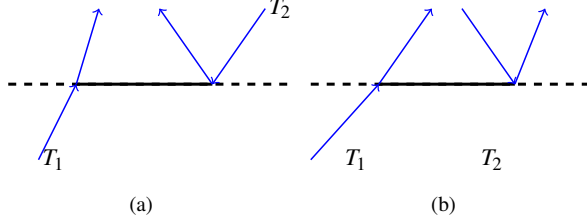


Fig. 10: Two sub-cases in case C-3.

As for case C-3, there are two sub-cases (Fig. 10a and Fig. 10b). For the first sub-case, assuming that $-\pi < v_i \leq 0$, for all $i < j$, $\overrightarrow{p_{1,j-1}p_{1,j}}$ and $\overrightarrow{p_{2,j-1}p_{2,j}}$ must be from the opposite sides of object L_{j-1} as shown in Fig. 11.

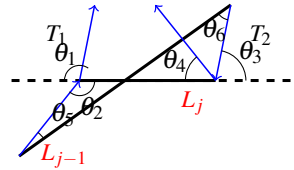


Fig. 11: Geometric analysis shows that $-\pi < v_j < 0$ in case C-3.

To allow $v_j > 0$, $\theta_4 < \theta_1$. By the properties of the elastic band, $\theta_1 \leq \theta_2, \theta_3 \leq \theta_4$. Therefore, $\theta_6 < \theta_3 \leq \theta_4 < \theta_1 \leq \theta_2$. Since $\theta_5 + \theta_6 < \theta_5 + \theta_2 < \pi$, the object L_{j-1} is gone through with a zigzag, which is impossible by Lemma III.3.

The same conclusion holds for the second sub-case under case C-3. Therefore, none of the three cases (C-1, C-2, C-3) allows $v_j > 0$. And we conclude that once T_1 and T_2 diverge from one point, they are unable to converge to the same destination. \square

When the visiting sequence consists of line segments, Theorem III.1 ensures the uniqueness of the elastic band for each instance. Since a shortest tour is an elastic band, the solution we get from Algorithm 1 is optimal. Optimality of Algorithm 1 extends when objects are convex.

Corollary III.1. *For a tour starting from the depot passing through a set of convex regions in a fixed order, its length has a unique global minimum, realized by an elastic band.*

Proof. Assuming the contrary, we can let T_1 and T_2 be two different elastic bands. Denote the intersection of elastic band T_i and convex region R_j by $p_{i,j}$. Recall that the proof of Theorem III.1 is based on the fact that segment $p_{1,j}p_{2,j}$ is part of line segment L_j . For each convex region R_j , segment $p_{1,j}p_{2,j}$ is still part of it. Therefore, the conclusion holds for

the convex case, i.e., elastic bands T_1 and T_2 should be the same tour. \square

The uniqueness property stated in Corollary III.1 breaks down without the convexity assumption. Fig. 12 shows such an instance with two different elastic bands.

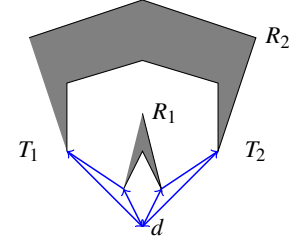


Fig. 12: An instance that has two different elastic bands when some of the regions are non-convex. R_1, R_2 are two non-convex regions and d is the depot. When the visiting sequence is fixed as P_1P_2 , T_1 and T_2 are two tours represented with blue arrows. Both of the tours are elastic bands.

C. Handling Overlaps

Polygons in CVRG may overlap (e.g., in a clutter removal scenario). For arbitrary polygonal objects i and j , if i is placed on j or the opposite, we say object i and j overlap with each other. When overlaps occur, it is necessary to consider the partial orders before we schedule the pickup sequence. When the objects form a stack, we should always first pick up the objects on the top. We developed the corresponding DP algorithm for the overlapping cases and compared it with a greedy approach. Given the partial order constraints, invalid routes and invalid ordering among routes can be eliminated. Even when there is only one overlap in the environment, it will cut off the number of task sequences by 50%. In terms of optimality, corollary III.1 continues to hold. Due to limited space, evaluation for the overlapping cases is not presented in Section IV; we mention that, besides less computation cost, the results show no difference from the non-overlapping setting.

IV. EVALUATION

We carried out extensive simulation studies to evaluate the performance of the proposed methods. These efforts are described and discussed in this section. The proposed algorithms are implemented in Python and all the experiments are executed on an Intel[®] Xeon[®] CPU at 3.00GHz.

A. Environment Setup

We selected multiple customer location and object weight distributions for realistic evaluation. More specifically, problem instances are generated in two phases. First, object weights are uniformly randomly selected with a lower and upper weight bound. For a given integer k , three ranges are used: (1) $[0, 1]$, (2) $[1/k, 1]$, and (3) $[1/k, 2/k]$. This choice models the practical setting that a lower limit on pickup/delivery weight is placed, below which it becomes uneconomical to do so. In our experiments, $k = 7$. The $[0, 1]$ case consumes the case of k being arbitrarily large.

After weights are picked, the orientation of the objects are uniformly selected in $[0, 2\pi)$. The mass center of the objects in the workspace are selected according to two distributions:

- 1) **Uniform:** The locations of objects are uniformly selected in the bounded 2D workspace.
- 2) **Gaussian:** The locations of objects follow a two-dimensional Gaussian distribution where heavy objects are closer to the mean of the distribution (Fig. 13(a)).

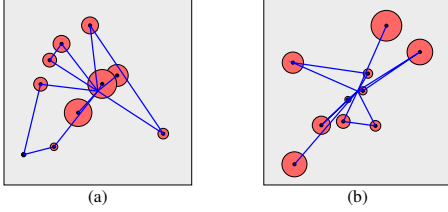


Fig. 13: Gaussian (a) and inverse-Gaussian (b) distributions of customer locations. The size of the disc signifies the objects’ size and/or weights.

We selected the Gaussian setup to model the collapsing of a large object into multiple smaller pieces (e.g., after an explosion), where it is likely for heavy pieces to be close to the epicenter. During the evaluation, we further examined the “inverse” Gaussian setting where heavy objects are more likely to be away from the Gaussian mean (Fig. 13(b)). We omit the result for this setting as it demonstrates performance characteristics similar to the uniform setting.

B. Performance Evaluation of CVRG

We evaluate the performance of four algorithms on CVRG: (1) DP-CVRG, an exact algorithm based on DP-CVRP, with tour costs computed with ELASTIC-IMPROV, (2) FH, the finite horizon version of DP-CVRG, with $h = 10$, (3) GD, a greedy best-first algorithm that picks the closest object satisfying capacity constraints, (4) BCP-CVRG, which uses a state-of-the-art branch-cut-and-price method BCP-CVRP [27] to compute a solution with centroid of the objects and then use ELASTIC-IMPROV to shorten the paths. We also include BCP-CVRP, to see ELASTIC-IMPROV’s contribution.

Our evaluation models aerial delivery applications (Fig. 1), where each customer location is a polygon. A typical solution is illustrated in Fig. 14 (the left figure), where the blue lines are the tours. In generating the polygonal regions for evaluation, we made the diameter of the polygon bounded by $\frac{1}{10}$ of the workspace side length. The result is given in Fig. 15. The solution produced by BCP-CVRG is not optimal and is about 5% worse than the solution produced by FH, which computes near optimal solution (DP-CVRG computes optimal solution when ELASTIC-IMPROV is optimal).

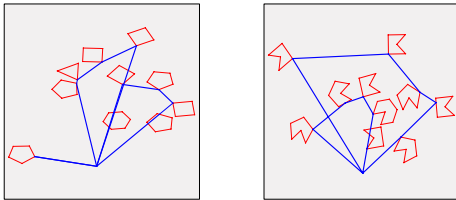


Fig. 14: (left) A (convex polygon only) CVRG instance with an optimal solution generated by DP-CVRG. (right) A CVRG instance with non-convex polygons and an optimal solution.

The second evaluation examines the Gaussian setup with all three weight distributions. The running time and solution

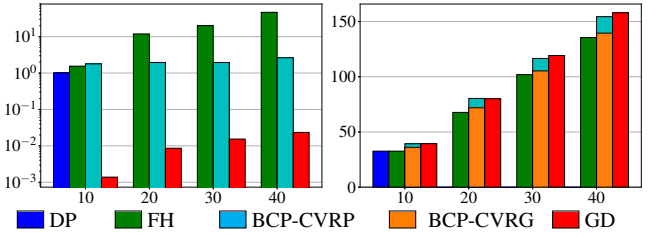


Fig. 15: Algorithm performance for CVRG under uniform customer location distribution (left: time (s); right: optimality (unitless)). BCP-CVRP and BCP-CVRG are overlaid with the shorter bar in front. For computational time, BCP variants take essentially the same amount of time as ELASTIC-IMPROV is fast.

quality results are given in Fig. 16 and Fig. 17, respectively. We also observe that FH runs much faster as we lower bound the object weight, as expected in practice. In terms of solution optimality, FH is near optimal and significantly outperforms the greedy method (by 50%) as well as BCP-CVRG (by 15% to 20% percent). This shows superiority of FH in solution quality.

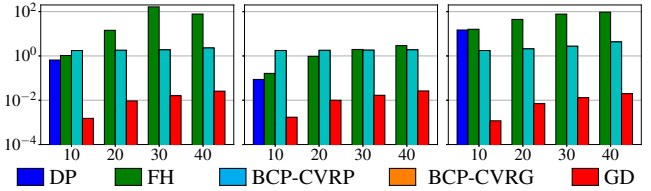


Fig. 16: Algorithm running time in seconds for CVRG under Gaussian distribution for different weight distributions: [left] $[0, 1]$. [middle] $[1/k, 1]$. [right] $[1/k, 2/k]$.

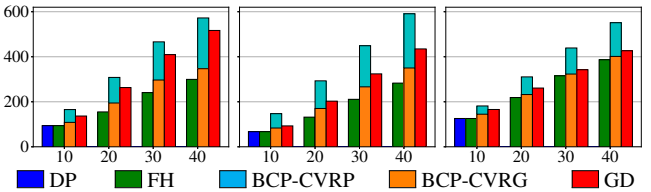


Fig. 17: Corresponding solution quality for cases in Fig. 16.

We evaluated performance of DP-CVRG and FH for cases where customer regions are non-convex. For the case where there are 10 regions, we compute the optimal solution by splitting each non-convex region into multiple convex ones and then run a modified version of DP-CVRG. The result confirms that DP-CVRG and FH both compute the same as the exact optimal solution. A typical case is illustrated in Fig. 14 (the right figure). The evaluation empirically suggests that FH is expected to compute high-quality solution even when regions are non-convex.

In addition to comprehensive numerical studies, we carried out physics based simulations of the aerial delivery scenario in the Unreal Engine to obtain a more realistic estimate of the travel and delivery time. The setup is similar to that illustrated in Fig. 1 and Fig. 14, with 10 to 20 objects. Over 10 runs of different setups for 10 objects, the time cost ratio between DP-CVRG/FH, BCP-CVRG, and GD is 1:1.07:1.17 (for this case, recall that DP-CVRG and FH compute the same optimal solutions). For 20 objects, the ratio between FH, BCP-CVRG, and GD is 1:1.02:1.16. Considering the time that is required for making deliveries,

these results largely agree with the earlier numerical results.

Our methods for CVRG have excellent scalability. In under 200 seconds, BCP-CVRG and FH readily scale to over 200 objects. Outcome of the computational experiments are given in Fig. 18, in which uniform object distribution with uniform weight distribution in $[0, 1]$ is used.

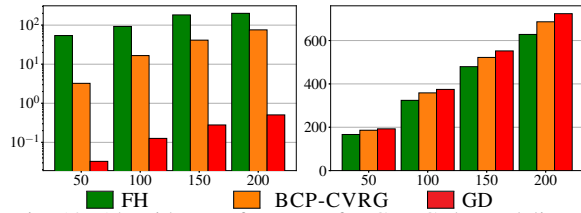


Fig. 18: Algorithm performance for CVRG drone delivery.

We observe that the trend agrees with earlier results. In particular, for CVRG, FH does considerably better than both BCP-CVRG ($\sim 10\%$) and the greedy method ($\sim 20\%$).

As a last evaluation, we attempted large scale instances for CVRG where the regions are chosen following the Gaussian distribution. As shown in Fig. 19, the trend of the solution quality agrees with previous experiments under Gaussian distribution. On the other hand, the running time for FH is comparatively large. This can be attributed to the regions with low weights tending to be the last delivery targets. When there are hundreds of regions in the instance, most of the candidates in the last rounds of DP process are with low weights. This causes the amount of candidates in the later tours to be disproportionately large, requiring more time to go through.

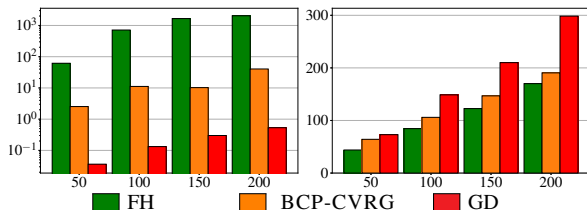


Fig. 19: Algorithm performance for CVRG drone delivery under the Gaussian distribution, both object locations and weights are uniformly distributed.

V. CONCLUSION

In this work, we examined CVRG as a CVRP variant where the target locations have non-trivial geometry, with applications toward a variety of robotics tasks including aerial delivery, rescue, clutter removal, and so on. Solving CVRG optimally requires the careful selection of exact customer locations for completing pickup or delivery tasks. We developed multiple efficient algorithms for tackling CVRG, and evaluated their performance as the number of customers, customer geometry, and customer distribution changes. In all cases, it was shown that our algorithms provide significant advantage in computing high quality solutions as compared with greedy approaches and CVRP solver based methods.

REFERENCES

[1] J. Liu, C.-L. Li, and C.-Y. Chan, "Mixed truck delivery systems with both hub-and-spoke and direct shipment," *Transportation Research Part E: Logistics and Transportation Review*, vol. 39, no. 4, pp. 325–339, 2003.

[2] T. Wen, Z. Zhang, and K. K. Wong, "Multi-objective algorithm for blood supply via unmanned aerial vehicles to the wounded in an emergency situation," *PLoS one*, vol. 11, no. 5, 2016.

[3] A. Karak and K. Abdelghany, "The hybrid vehicle-drone routing problem for pick-up and delivery services," *Transportation Research Part C: Emerging Technologies*, vol. 102, pp. 427–449, 2019.

[4] W. N. Tang and J. Yu, "Taming combinatorial challenges in clutter removal," in *Proceedings International Symposium on Robotics Research*, 2019.

[5] G. B. Dantzig and J. H. Ramser, "The truck dispatching problem," *Management science*, vol. 6, no. 1, pp. 80–91, 1959.

[6] E. L. Lawler, "The traveling salesman problem: a guided tour of combinatorial optimization," *Wiley-Interscience Series in Discrete Mathematics*, 1985.

[7] P. Toth and D. Vigo, *The vehicle routing problem*. SIAM, 2002.

[8] M. R. Garey and D. S. Johnson, *Computers and intractability*. Freeman San Francisco, 1979, vol. 174.

[9] G. Laporte and Y. Nobert, "Exact algorithms for the vehicle routing problem," in *North-Holland Mathematics Studies*. Elsevier, 1987, vol. 132, pp. 147–184.

[10] D. L. Miller, "A matching based exact algorithm for capacitated vehicle routing problems," *ORSA Journal on Computing*, vol. 7, no. 1, pp. 1–9, 1995.

[11] P. Toth and D. Vigo, "An exact algorithm for the vehicle routing problem with backhauls," *Transportation science*, vol. 31, no. 4, pp. 372–385, 1997.

[12] A. Caprara and M. Fischetti, "Branch-and-cut algorithms," *Annotated bibliographies in combinatorial optimization*, pp. 45–64, 1997.

[13] R. Baldacci, E. Hadjiconstantinou, and A. Mingozzi, "An exact algorithm for the capacitated vehicle routing problem based on a two-commodity network flow formulation," *Operations research*, vol. 52, no. 5, pp. 723–738, 2004.

[14] M. R. Garey and D. S. Johnson, "Complexity results for multiprocessor scheduling under resource constraints," *SIAM journal on Computing*, vol. 4, no. 4, pp. 397–411, 1975.

[15] J. Ota, "Rearrangement planning of multiple movable objects by a mobile robot," *Advanced Robotics*, vol. 23, no. 1-2, pp. 1–18, 2009.

[16] G. Havur, G. Ozbilgin, E. Erdem, and V. Patoglu, "Geometric rearrangement of multiple movable objects on cluttered surfaces: A hybrid reasoning approach," in *Robotics and Automation (ICRA), 2014 IEEE International Conference on*. IEEE, 2014, pp. 445–452.

[17] A. Krontiris and K. E. Bekris, "Dealing with difficult instances of object rearrangement," in *Robotics: Science and Systems*, 2015.

[18] S. D. Han, N. M. Stiffler, A. Krontiris, K. E. Bekris, and J. Yu, "Complexity results and fast methods for optimal tabletop rearrangement with overhand grasps," *The International Journal of Robotics Research*, vol. 37, no. 13-14, pp. 1775–1795, 2018.

[19] G. Pratt and J. Manzo, "The darpa robotics challenge [competitions]," *IEEE Robotics & Automation Magazine*, vol. 20, no. 2, pp. 10–12, 2013.

[20] R. R. Murphy, *Disaster robotics*. MIT press, 2014.

[21] Q. M. Ha, Y. Deville, Q. D. Pham, and M. H. Ha, "On the min-cost traveling salesman problem with drone," *Transportation Research Part C: Emerging Technologies*, vol. 86, pp. 597–621, 2018.

[22] H. D. Yoo and S. M. Chankov, "Drone-delivery using autonomous mobility: an innovative approach to future last-mile delivery problems," in *2018 IEEE international conference on industrial engineering and engineering management (ieem)*. IEEE, 2018, pp. 1216–1220.

[23] C. C. Murray and A. G. Chu, "The flying sidekick traveling salesman problem: Optimization of drone-assisted parcel delivery," *Transportation Research Part C: Emerging Technologies*, vol. 54, pp. 86–109, 2015.

[24] N. Christofides and S. Eilon, "An algorithm for the vehicle-dispatching problem," *Journal of the Operational Research Society*, vol. 20, no. 3, pp. 309–318, 1969.

[25] V. V. Vazirani, *Approximation algorithms*. Springer Science & Business Media, 2013.

[26] N. Simonetti and E. Balas, "Implementation of a linear time algorithm for certain generalized traveling salesman problems," in *International Conference on Integer Programming and Combinatorial Optimization*. Springer, 1996, pp. 316–329.

[27] A. Pessoa, R. Sadykov, E. Uchoa, and F. Vanderbeck, "A generic exact solver for vehicle routing and related problems," in *International Conference on Integer Programming and Combinatorial Optimization*. Springer, 2019, pp. 354–369.