Bi-Objective Splitting Delivery VRP with Loading Constraints and Restricted Access

Jiyuan Pei*†, Chengpeng Hu*†, Jialin Liu*†, Yi Mei‡, and Xin Yao*†
*Research Institute of Trustworthy Autonomous System
Southern University of Science and Technology
†Guangdong Provincial Key Laboratory of Brain-inspired Intelligent Computation
Department of Computer Science and Engineering
Southern University of Science and Technology
Shenzhen 518055, China
‡School of Engineering and Computer Science
Victoria University of Wellington
Kelburn 6012, New Zealand

***Abstract*—The splitting delivery vehicle routing problem with 3-dimensional loading constraints (3L-SDVRP) is an important and challenging VRP variant. The problem consists of two sub-problems: the routing and 3D bin packing, each of which is NP-hard by itself. Compared with CVRP, 3L-SDVRP is much closer to the reality. There have been many studies on 3L-SDVRP. However, to our best knowledge, no complete mathematical model has been formalised with comprehensive loading constraints and there are still some real-world factors ignored in the existing studies. In this paper, we consider a more realistic 3L-SDVRP model with restricted access provided by the 2021 HUAWEI Logistics Competition, which is different from existing problem models in two aspects. First, this problem considers the total travel cost and average loading rate simultaneously. Second, this problem has additional constraints related to certain special pickup points. These differences make existing optimisation approaches not directly applicable. A major contribution of this paper is the formal mathematical model developed for this new 3L-SDVRP. In addition, we propose a genetic algorithm with an efficient fitness evaluation. The proposed algorithm has been demonstrated to significantly outperform the baseline solver provided by the competition in solving the problem instances from the competition and the ones adapted from benchmark datasets of related problems.**

***Index Terms*—3L-SDVRP, capacitated vehicle routing problem, 3D packing, splitting delivery, combinatorial optimisation**

I. Introduction

Capacitated vehicle routing problem (CVRP) has been researched alone for decades [1]–[6], while the loading constraints of a vehicle regarding its capacity (in terms of weight and size) are not negligible in real-world applications. Consequently, research around splitting delivery vehicle routing problems with 3-dimensional loading constraints (3L-SDVRPs) grows rapidly in recent years [7]–[10]. Although the problems considered in those works share the same name, they actually have different objective functions and constraints [7]–[10]. Existing approaches for solving various 3L-SDVRPs in literature can be mainly categorised into two groups, loading-driven and route-driven approaches. The former is composed by two stages: (i) optimisation of loading plan alone, and then (ii) optimisation of routes alone while fixing the loading plan determined by stage (i). The latter iteratively optimises routing and loading plans in a sequence.

Although compared with CVRP, 3L-SDVRP is much closer to reality, it still misses some important real-world factors, making the existing methods not directly applicable to the real-world applications. In this paper, we focus on a more realistic 3L-SDVRP model that is oriented from a real logistics problem that Huawei Ltd. encounters, which was also proposed as a competition in the 11th International Conference on Evolutionary Multi-Criterion Optimization, 2021. This new problem aims at efficiently allocating multiple types of vehicles to visit a number of pickup points and collect items located at the pickup points, while minimising the total travelling distance and maximising the averaged loading rate (in terms of weight and volume), and satisfying some routing and loading constraints. In particular, a special type of pickup point, called warehouse, is considered. When entering a warehouse, the vehicle must be empty. In the real world, a warehouse can refer to a custom point, where no item can be brought in.

Compared with the existing 3L-SDVRP models, the model considered in our study is different from the following perspectives: First, our problem considers both the total travel cost and average loading rate simultaneously in the objective function, while the existing models only considered one of them. Second, our problem has additional constraints related to certain special pickup points (warehouses with restricted access). To our best knowledge, no complete mathematical model has been formalised with all the considered loading constraints and there are still some real-world factors ignored

This work was supported by the National Natural Science Foundation of China (Grant No. 61906083), the Guangdong Provincial Key Laboratory (Grant No. 2020B121201001), the Program for Guangdong Introducing Innovative and Entrepreneurial Teams (Grant No. 2017ZT07X386), the Guangdong Basic and Applied Basic Research Foundation (Grant No. 2021A1515011830), the Shenzhen Science and Technology Program (Grant No. KQTD2016112514355531), the Shenzhen Fundamental Research Program (Grant No. JCYJ20190809121403553) and the Marsden Fund of New Zealand Government under Contract VUW1614.

Corresponding author: Xin Yao (xiny@sustech.edu.cn).

in the existing studies.

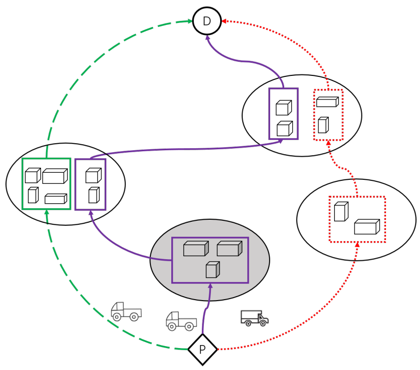

Fig. 1. Illustrative example of 3L-SDVRP with warehouses considered in this paper. The warehouses are highlighted with grey background colour. Point marked by "P" refers to the depot, while the node marked by "D" is the delivery point (e.g., a port in real life).

For 3L-SDVRPs of large problem size, the evaluation of a solution quality is computationally expensive. We propose an objective estimation based solver which estimates the evaluation values as an alternative of actual solution evaluation to improve the time efficiency. A vehicle selection strategy is introduced and embedded so that the advantage of using multiple vehicle types is exploited.

Our main contributions are as follows: (i) we formulate the mathematical model for the studied problem (Section III-B), which has more realistic loading constraints compared to existing ones and considers the quality of loading and routing plans simultaneously (detailed in Section III-C). (ii) For better investigating algorithms for tackling 3L-SDVRP, we generate a number of 3L-SDVRP benchmark instances based on well-known benchmarks of related problems. (iii) We propose a genetic algorithm, equipped with a novel efficient objective estimation method, and a vehicle selection strategy to optimise the routing and loading plans. The proposed approach has been shown to significantly outperform the baseline solver provided by the competition in solving competition instances and 3L-SDVRP instances. Besides, the proposed algorithm provides a solution set that contains several non-dominated solutions instead of one single solution.

The remainder of the paper is organised as follows. Section II reviews the related work. We describe the 3L-SDVRP considered in this work in Section III-A, formulate its mathematical model in Section III-B and discuss its key differences compared to related problems in Section III-C. Section IV briefly introduces the instances and baseline solver provided by the competition. Section V presents the generation of novel benchmark datasets. Section VI presents our approach and Section VII presents the comparison between it and baseline solver on the test instances. Section VIII concludes the paper.

II. Related Work

In this section, we review the related vehicle routing problems with 3-dimensional loading constraints. Then, existing approaches for those problems, categorised into loading-driven and route-driven approaches, are presented.

A. Related Vehicle Routing Problems

Diverse variants of capacitated vehicle routing problems (CVRPs) have been formulated from real-world applications [11], among which 3-dimensional loading constraints and splitting delivery have been considered separately.

Few works have considered simultaneously the splitting delivery and 3-dimensional loading constraints [7]–[10], [12]. Although the same name ("splitting delivery vehicle routing problem with 3-dimensional constraints", 3L-SDVRP) has been used in those works, the described problems differ from each other in the objective functions or constraints.

To our best knowledge, Gendreau *et al.* [13] was the first to formally introduce the vehicle routing problem with 3-dimensional loading constraints (3L-CVRP), assuming that the items located at the same customer points can not be splitted. This 3L-CVRP, also studied in [14] and [12], assumes that identical vehicles are used for all routes [13]. A simplified version of 3L-CVRP by removing constraints such as the ones regarding supporting area and loading direction was studied in [15]. In the aforementioned works [7], [12]–[15], no complete mathematical model of the problems was formulated. Junqueira *et al.* [16] formulated a 3L-CVRP model, assuming that all vehicles have the same capacity, in terms of size and weight. Moreover, minimising the total travel cost was the only objective [16]. [17] also assumed identical vehicles and formulated the problem with two objectives, minimising total travel cost and maximising the number of items loaded into the vehicle.

In the work of [7], a real-world problem with more realistic considerations compared to 3L-CVRP [13], including multiple vehicle types of different capacity and split delivery, was introduced. The problem was referred to as "3L-CVRP" in [7], but the term "3L-SDVRP" [8] fits it better. The 3L-SDVRP studied by [9] allows loading items from sides of vehicles, while the other definitions of 3L-SDVRP only allow loading from the rear door, which significantly affects the loading plan and the order of item collection. In [8] and [10], two objectives are considered simultaneously, minimising the sum of total travel cost and minimising the number of vehicles used. No one has considered the total travel cost and average loading rate simultaneously.

B. Related Approaches

Existing approaches for various 3L-SDVRPs can be mainly categorised into route-driven (e.g., [7]–[9]) and loading-driven (e.g., [10]) approaches. An hybrid algorithm was designed in [12] for solving 3L-CVRPs, in which a tree search algorithm (TSA) was used to optimise routing plans and a subordinate TSA (TRSA) was used to load items to be collected on each obtained routes. In the work of [7], the routing and loading plans are iteratively optimised with local search and a deepest-bottom-left-fill packing (DBLP) heuristic. In the work of [8], a data-driven three-Layer search algorithm (DTSA) was designed, in which two estimation of distribution algorithms were used to optimise routes and split items at collection points and a prediction model was trained to evaluate loading plans. Bortfeldt and Yi [10] designed a layer-based loading heuristic. The loading plan is optimised with a genetic algorithm, and then the routing plan is optimised with local search [10]. Meta-heuristics and heuristics have been proven to be effective in solving similar problems.

III. 3L-SDVRP

In this section, we summarise the problem and constraints, formulate its mathematical model, and clarify the differences between our model and existing ones.

A. Problem Description

The problem aims at efficiently allocating multiple types of vehicles to visit a number of pickup points (including warehouses with restricted access) and collect items located at the pickup points, while minimising the total travelling distance of vehicles used and maximising the averaged loading rate of vehicles, in terms of weight and volume. At each pickup point, there could be more than one item to be collected. The height, width, length and weight of any item is known. Each vehicle type is determined by a height, width, depth and capacity. All items and vehicles are of rectangular shape. Fig. 1 illustrates an example of our problem.

We categorised the constraints into routing and loading ones, summarised as follows.

a) Routing constraints:

$\mathcal{RC}_1$: each vehicle should depart from a single depot and deliver the collected items to a single destination that is different from the depot;

$\mathcal{RC}_2$: each vehicle can visit each pickup point at most once;

$\mathcal{RC}_3$: when entering a warehouse, the vehicle must be empty;

$\mathcal{RC}_4$: each item should be collected once and only once.

b) Loading constraints: At any time of the trip, inside any vehicle:

$\mathcal{LC}_1$: the total weight of its loading should not exceed its capacity;

$\mathcal{LC}_2$: the total size of its loading should not exceed its size (i.e., length, width and height);

$\mathcal{LC}_3$: items cannot overlap each other in any dimension;

$\mathcal{LC}_4$: when loading any item, at least 80% bottom area should be supported by previously loaded items;

$\mathcal{LC}_5$: for any item, it should not be placed behind, looking from the rear of a vehicle to its head, collected from any pickup point that was visited before its pickup point;

$\mathcal{LC}_6$: items cannot be re-arranged after being loaded;

$\mathcal{LC}_7$: when placing an item, it can only be rotated horizontally with 0 or 90 degrees. Its vertical orientation is fixed.

B. Problem Formulation

1) Notations:

$\mathcal{P}$: set of pickup points.

$\mathcal{P}_w$: set of special pickup points (warehouses), $\mathcal{P}_w \subset \mathcal{P}$.

N: number of pickup points, i.e., $N = |\mathcal{P}|$.

p_0: start point.

p_{N+1}: delivery (end) point.

m_k: number of items at pickup point p_k, $\forall k \in \{1, \dots, N\}$.

$\mathcal{T}_k = \{t_{k,1}, t_{k,2}, \dots, t_{k,m_k}\}$: set of items at p_k, $\forall k \in \{1, \dots, N\}$.

$\mathcal{T}$: set of all items, $\mathcal{T} = \cup_{k=1}^{N} \mathcal{T}_k$.

K: number of vehicle types.

$W_t(\cdot)$: width of a given item.

$W_v(\cdot)$: width of a given vehicle.

$H_t(\cdot)$: height of a given item.

$H_v(\cdot)$: height of a given vehicle.

$L_t(\cdot)$: length of a given item.

$L_v(\cdot)$: length of a given vehicle.

$C_t(\cdot)$: weight of a given item.

$C_v(\cdot)$: weight capacity of a given vehicle.

$\tau(\cdot)$: pickup point of a given item, $1 \le \tau(\cdot) \le N$.

$d(p_i, p_j)$: distance between any two pickup points $p_i, p_j \in \mathcal{P}$. It's notable that $d(p_i, p_i) = 0$ ($\forall i \in \{1, \dots, N\}$) and $d(p_i, p_j)$ does not necessarily equal to $d(p_j, p_i)$ due to single lanes ($\forall i, j \in \{1, \dots, N\}$).

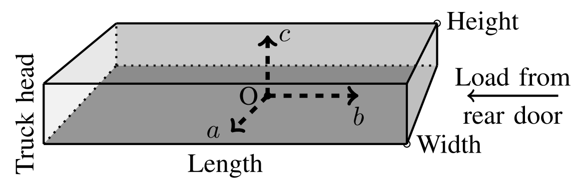

Fig. 2. Illustration of the coordinate system (a, b and c) for items with origin at the center of a truck. The coordinates are in parallel with the truck's width, length and height, respectively.

2) Solution representation: A solution is represented by a number of routes. Each route is composed of its collected items, the type of vehicle used and the loading plan. Assuming a solution of n routes, for each route r_i ($\forall i \in \{1, 2, \dots, n\}$), its aforementioned components are as follows:

a) Collected items: Let l_i denote the number of items to be collected on this route. $(x_{i,1}, x_{i,2}, \dots, x_{i,l_i})$ represents the collected item, $\forall j \in \{1, 2, \dots, l_i\}$, $x_{i,j} \in \mathcal{T}$.

b) Type of vehicle used: For each route r_i, a vehicle type v_i is specified. Consequently, its weight capacity, $C_v(v_i)$, and size, $W_v(v_i)$, $L_v(v_i)$ and $H_v(v_i)$, are determined.

c) Loading plan: A loading plan is represented by an array of 4-tuples, the length of which is the number of collected items, l_i. For each collected item $x_{i,j}$, the 4-tuple $\langle a_{i,j}, b_{i,j}, c_{i,j}, \theta_{i,j} \rangle$ determines its geometric center location and its horizontal direction. The geometric center of v_i is set as its origin, the coordinates of $a_{i,j}$, $b_{i,j}$ and $c_{i,j}$ are in parallel with v_i's width, depth and height, respectively (cf. Figure 2). According to $\mathcal{LC}_5$, an item can be placed in two horizontal directions only considering the symmetry relation. The horizontal direction $\theta_{i,j}$ is 1 if the width of item $x_{i,j}$ is in parallel with the width of vehicle; otherwise 0.

3) Mathematical model: Two objectives are considered, maximising the averaged loading rate of all vehicles and minimising the total routing distance. We formalise the objectives

$$\max f_l(\boldsymbol{x}) = \frac{1}{n}\sum_{i=1}^{n}\max\left\{\frac{\sum_{j=1}^{l_i} W_t(x_{i,j}) * L_t(x_{i,j}) * H_t(x_{i,j})}{W_v(v_i) * L_v(v_i) * H_v(v_i)}, \frac{\sum_{j=1}^{l_i} C_t(x_{i,j})}{C_v(v_i)}\right\}, \quad (1)$$

$$\min f_r(\boldsymbol{x}) = \sum_{i=1}^{n}\left(d\Big(p_0, \tau(x_{i,1})\Big) + d\Big(\tau(x_{i,l_i}), p_{N+1}\Big) + \sum_{j=1}^{l_i-1} d\Big(\tau(x_{i,j}), \tau(x_{i,j+1})\Big)\right), \quad (2)$$

where $x_{i,j}$ is the jth item of the ith route, which is served by truck v_i.

$s.t.$

$$\forall i \in \{1,\dots,n\}, \forall 1 \le z \le o \le l_i,\ \text{if } \tau(x_{i,z}) = \tau(x_{i,o}),\ \text{then for } z < j < o, \tau(x_{i,j}) = \tau(x_{i,z}), \quad (3)$$

$$\forall i \in \{1,\dots,n\},\ \forall j \in \{1,\dots,l_i-1\},\ \text{if } \tau(x_{i,j}) \notin \mathcal{P}_w,\ \text{then } \tau(x_{i,j+1}) \notin \mathcal{P}_w, \quad (4)$$

$$\forall i \in \{1,\dots,n\},\ \forall j \in \{1,\dots,l_i\},\ \text{if } \tau(x_{i,j}) \in \mathcal{P}_w,\ \text{then } \tau(x_{i,m}) = \tau(x_{i,j})\ \forall m \in \{1,\dots,j-1\}, \quad (5)$$

$$\forall i \in \{1,\dots,n\}, \forall j, j' \in \{1,\dots,l_i\},\ \text{if } j \ne j',\ \text{then } x_{i,j} \ne x_{i,j'}, \quad (6)$$

$$\cup_{i=1}^{n} Set(r_i) = \mathcal{T}, \quad (7)$$

$$\forall i, i' \in \{1,\dots,n\} \text{ and } i \ne i', Set(r_i) \cap Set(r_{i'}) = \emptyset, \quad (8)$$

$$\forall i \in \{1,\dots,n\},\ \sum_{j=1}^{l_i} C(x_{i,j}) \le C(v_i), \quad (9)$$

$$\forall i \in \{1,\dots,n\},\ \forall j \in \{1,\dots,l_i\}, \frac{W_v(v_i) - \Phi_W(x_{i,j})}{2} \ge a_{i,j} \ge -\frac{W_v(v_i) - \Phi_W(x_{i,j})}{2},$$
$$\frac{L_v(v_i) - \Phi_L(x_{i,j})}{2} \ge b_{i,j} \ge -\frac{L_v(v_i) - \Phi_L(x_{i,j})}{2}, \frac{H_v(v_i) - H_t(x_{i,j})}{2} \ge c_{i,j} \ge -\frac{H_v(v_i) - H_t(x_{i,j})}{2}, \quad (10)$$

$$\forall i \in \{1,\dots,n\},\ \forall j, j' \in \{1,\dots,l_i\} \text{ and } j \ne j',$$
$$|a_{i,j} - a_{i,j'}| \ge \frac{\Phi_W(x_{i,j}) + \Phi_W(x_{i,j'})}{2} \text{ or } |b_{i,j} - b_{i,j'}| \ge \frac{\Phi_L(x_{i,j}) + \Phi_L(x_{i,j'})}{2} \text{ or } |c_{i,j} - c_{i,j'}| \ge \frac{H_t(x_{i,j}) + H_t(x_{i,j'})}{2}, \quad (11)$$

$$\forall i \in \{1,\dots,n\},\ \forall j \in \mathcal{S}_{notBottomItems} = \left\{j | j \in \{1,\dots,l_i\}, c_{i,j} > \frac{H_t(x_{i,j})}{2} - \frac{H_v(v_i)}{2}\right\},$$
$$\text{let } \mathcal{S}_j = \left\{j' | 1 \le j' < j, c_{i,j} - c_{i,j'} = \frac{H_t(x_{i,j}) + H_t(x_{i,j'})}{2}, |a_{i,j} - a_{i,j'}| < \frac{\Phi_W(x_{i,j}) + \Phi_W(x_{i,j'})}{2}, |b_{i,j} - b_{i,j'}| < \frac{\Phi_L(x_{i,j}) + \Phi_L(x_{i,j'})}{2}\right\},$$
$$\text{then } \sum_{j' \in \mathcal{S}_j}\left(\Big(\frac{\Phi_W(x_{i,j}) + \Phi_W(x_{i,j'})}{2} - |a_{i,j} - a_{i,j'}|\Big) * \Big(\frac{\Phi_L(x_{i,j}) + \Phi_L(x_{i,j'})}{2} - |b_{i,j} - b_{i,j'}|\Big)\right) \ge 0.8 * W_t(x_{i,j}) * L_t(x_{i,j}), \quad (12)$$

$$\forall i \in \{1,\dots,n\}, \forall 1 \le j < j' \le l_i,\ \text{if } \tau(x_{i,j}) \ne \tau(x_{i,j'}) \text{ and } |a_{i,j'} - a_{i,j}| < \frac{\Phi_W(x_{i,j'}) + \Phi_W(x_{i,j})}{2},$$
$$\text{then } b_{i,j'} - b_{i,j} \ge \frac{\Phi_L(x_{i,j'}) + \Phi_L(x_{i,j})}{2} \text{ or } |c_{i,j'} - c_{i,j}| \ge \frac{H_t(x_{i,j'}) + H_t(x_{i,j})}{2}, \quad (13)$$

where $Set(r)$ denotes the set of all items in route r, and

$$\Phi_W(x_{i,j}) = \theta_{i,j} * W_t(x_{i,j}) + (1 - \theta_{i,j}) * L_t(x_{i,j}), \quad (14)$$
$$\Phi_L(x_{i,j}) = \theta_{i,j} * L_t(x_{i,j}) + (1 - \theta_{i,j}) * W_t(x_{i,j}). \quad (15)$$

Fig. 3. Mathematical model of the problem considered in this work.

and constraints in Fig. 3. The formulation of f_r (Eq. (2) in Fig. 3) implies the the routing constraint $\mathcal{RC}_1$. $\mathcal{RC}_2$ is guaranteed by Eq. (3). Eqs. (4) and (5) ensure $\mathcal{RC}_3$. $\mathcal{RC}_4$ is guaranteed by Eqs. (6), (7) and (8). Eqs. (9), (10), (11) and (12) ensure the loading constraints $\mathcal{LC}_1$, $\mathcal{LC}_2$, $\mathcal{LC}_3$ and $\mathcal{LC}_4$, respectively. Eq. (13) ensures $\mathcal{LC}_5$ and $\mathcal{LC}_6$. $\mathcal{LC}_7$ is implicitly implied in the solution representation and formulation of objectives.

C. Key Differences to Related Problems

A number of studies have investigated into various problems called "splitting delivery vehicle routing problem with 3-dimensional constraints" [7]–[10], however, those problems are actually different from our problem in terms of objective functions or detailed loading constraints. The key differences are as follows. No existing study has considered the total travel cost and average loading rate simultaneously (c.f. Eqs. (1) and (2)). No one has ever considered special types of pickup points, e.g., warehouses with restricted access in this work (c.f. Eqs. (4), (5) and $\mathcal{RC}_3$). In addition, the starting point and delivery point (destination) are distinct in our problem while the other 3L-SDVRPs assume the identical starting and delivery point. The differences originate from some widely existed real-world scenario: the collect-to-centre scenario, in which the parking lot of vehicles is the starting point and the collecting centre (e.g., train stations or airports) is the delivery point. In each instance, vehicles start from the parking lot, collect and load boxes from customers at each pickup point and delivery them to the collecting centre. A warehouse with restricted access is a bonded area, which does not allow any non-empty vehicle to enter. The total distance and loading rate are two main factors that determine the price of transport contractor who provides different types of vehicles. The problem considered in this work and its formulation reflect the aforementioned scenario.

IV. Competition Instances and Baseline Solver

As case study, we use the problem instances and solver provided by the 2021 HUAWEI Logistics Competition organised at the 11th International Conference on Evolutionary Multi-Criterion Optimization [1] as test instances and baseline solver, respectively.

A. Problem instances

250 instances were provided by the competition. However, 212 instances have no warehouse or only have one single type of vehicle. Therefore, only the 38 instances that meet our problem's definition are studied in this paper. Those instances are renamed as *CI-1* to *CI-38* and saved in JSON format[2].

B. Baseline solver

A greedy solver provided by the competition is used as a baseline in our work. For each instance, the baseline solver assumes that each vehicle visits the pickup points in the order as listed in the JSON file. The vehicle with the biggest size is used to collect items, starting from the depot, until being fulfilled (size or weight) or all items of the visited pickup point are collected. If a vehicle is fulfilled, it goes directly to the destination. Then, another vehicle with the biggest size is used to continue collecting. The above steps are repeated until all pickup points are served. The packing policy of this baseline solver is also greedy. At each visited pickup point p, for all the items of the same size at p, all possible assembled blocks of those items are computed. The resulted block with the cross section the most closed to the one of the vehicle is used. More details can be found on the competition website [18].

Besides, a Java program for checking solutions' feasibility, a Python program for calculating f_l, Eq. (1), and a tool for visualising loading plans are also provided by the competition. Fig. 7 gives examples of visualising loading plans.

V. Instance Generation

Besides the competition instances, we also generate a number of 3L-SDVRP instances based on some well-known benchmark instances of related problems for better studying 3L-SDVRP and evaluating algorithms for tackling the problem.

Specifically, we find two commonly used datasets which are closest to our problem: *Shanghai* (*Sha01 - Sha15*) instances from [10] and *SD-CSS* instances (*SD-CSS1 - SD-CSS13*) from [7]. Table I summarises the differences between those instances and our problem's definition.

The instances are adapted to our problem as follows. (i) In each instance, a randomly selected pickup point is set as a warehouse. (ii) For each dataset (Shanghai and SD-CSS), vehicle types from all the instances are collected as a set of different vehicles and reassigned into instances of the corresponding data set. Notably, if two vehicle types have no significant difference (defined as the sum of differences on wight, height, length and capacity is smaller than 5%), then we randomly omit one. (iii) In each instance, a new point is introduced as the delivery point. Its distance to any other point is randomly generated in the range of the highest and lowest distances between any two points in this instance.

Instances adapted from datasets *SD-CSS* and *Shanghai* are named as *w-SD-CSS* and *w-Sha*[2], respectively.

Algorithm 1: Routing distance estimation. Notations used for describing the problem were already defined in Section III-B1.

Input: I, problem instance
Input: ζ, permutation of pickup points
$N \leftarrow$ number of pickup points of I
$\mathcal{M} \leftarrow$ normalised distance matrix between any pickup points of I (The normalised distance between any points i and j is $(d(i,j) - minD)/(maxD - minD)$ with $minD = \min d(i',j')$ and $maxD = \max d(i',j')$ for all $i' \neq j'$)
$VolList \leftarrow$ list of the total volume of items at each pickup point of I
$p_0 \leftarrow$ starting point of I
$p_{N+1} \leftarrow$ delivery point of I
$maxVol \leftarrow$ the volume of biggest vehicle in I
$distList \leftarrow$ an empty list to save distance of routes
$v \leftarrow$ initialise the load of residual box to 0
$D \leftarrow \mathcal{M}(p_0, \zeta_1)$
for $i \in \{1, \ldots, N\}$ **do**
    $v \leftarrow v + VolList[\zeta_i]$
    if $v > maxVol$ **then**
        $ratio \leftarrow \frac{v}{maxVol}$
        while $ratio > 1$ **do**
            Add $D + \mathcal{M}(\zeta_i, p_{N+1})$ to the end of $distList$
            $ratio \leftarrow ratio - 1$
            $v \leftarrow v - maxVol$
        $D = \mathcal{M}(p_0, \zeta_i)$
    if $i > 1$ **then**
        $D \leftarrow D + \mathcal{M}(\zeta_{i-1}, \zeta_i)$
Add $D + \mathcal{M}(\zeta_N, p_{N+1})$ to the end of $distList$
$\tilde{f}_r = mean(distList)$
return Estimated routing distance $\tilde{f}_r$

VI. Proposed Approach: Genetic Solver with Objective Estimation

We propose a genetic algorithm with a new efficient routing distance estimation and a vehicle selection strategy to optimise 3L-SDVRP.

A. Estimation of Routing Distance

The baseline solver uses a constant order, determined by the instance description file, to visit pickup points, which leads to deterministic and probably sub-optimal solutions. Optimising permutations of pickup points by meta-heuristics will lead to better solutions. However, optimising with the actual evaluation as fitness will cost significant time, as the feasibility checking and the evaluation of a solution is complex and time-consuming. For reference, it costs approximately $3.2s$ to evaluate a solution for *w-SD-CSS6* with the provided evaluation program on a machine with an Intel i7-9700K CPU and 64G RAM. If at least $10,000$ solution evaluations are

[1] https://emo2021.org/index.php/competition/ or https://www.noahlab.com.hk/logistics-ranking/#/home/index.

[2] Instances available at https://github.com/PeiJY/3L-SDVRP-instances. By default, warehouses are listed before regular pickup points without restricted access.

TABLE I
DIFFERENCES BETWEEN OUR PROBLEM AND TWO WELL-KNOWN BENCHMARKS OF RELATED PROBLEMS: *Shanghai* AND *SD-CSS*.

	Vehicle type	Point type
Our problem	Multiple types.	Regular pickup points and special pickup points with restricted access. Distinct starting and delivery points.
SD-CSS	Two types in *SD-CSS5*, *SD-CSS7*, *SD-CSS8*, *SD-CSS11*. One type in the other instances.	Regular pickup points only. An identical starting and delivery point.
Shanghai	One type.	Regular pickup points only. An identical starting and delivery point.

required during search to obtain a satisfactory solution (which is often the case in optimising vehicle routing problems), it takes more than 8 hours if considering *w-SD-CSS6*.

To reduce the time consumption, we design a routing distance estimation method as an alternative of actual evaluation without considering the solution feasibility during optimisation. Algorithm 1 details how the estimated routing distance, $\tilde{f}_r(\zeta)$, is calculated, where ζ is a permutation of pickup points.

B. θ-greedy Policy for Vehicle Selection

Multiple types of vehicles with different size and capacity are available to load items. The selection of vehicle type for each route directly affects the loading structure and the two objective values. To make the best use of the diverse vehicle types, we design a dynamic vehicle selection strategy, named as θ-greedy policy, in Algorithm 2 which selects a vehicle type for each route based on the volume of unloaded items, denoted as $resBoxVolume$, and the estimated loading rate of each vehicle type, $\tilde{f}_r$.

Algorithm 2: θ-greedy policy for vehicle selection. Notations used for describing the problem were defined in Section III-B1.

```
Input: I, problem instance
Input: resBoxVolume, volume of unloaded items
Input: γ, tight coefficient
Input: θ, threshold of estimated loading rate
vehicleList ← vehicle type list of I
K ← |vehicleList|
rl ← an empty list to save the predicted loading rates of vehicles
n ← 0 // vehicle counter
foreach v ∈ vehicleList do
    f̂_l ← resBoxVolume / (W_v(v)*H_v(v)*L_v(v)*γ)
    rl.add(f̂_l)
    if f̂_l > θ then
        n++
if n/K > random(0,1) then
    v ← the vehicle with maximum volume in I
else
    choose v with Roulette Wheel Selection based on softmax(rl)
return Estimated best vehicle v
```

C. Genetic Solver with Objective Estimation

A solution is represented by a sequence consisting of indices of all pickup points, which is the order of pickup points to be visited. We use a genetic algorithm with our routing distance estimation $\hat{f}_r$ as fitness function for optimising the routes and the greedy packing policy provided by the competition for loading items into vehicles that are selected by our θ-greedy policy. For a further simplification, when encoding a solution, warehouses are excluded. After optimisation, the warehouses are re-inserted after the starting point of routes in a greedy way when actually evaluating a solution. The population is initialised uniformly at random. Rank-based parent selection, order-based crossover and inverse mutation are used. At each generation, the top ranked ones of the current population and offspring are used as the population of next generation.

VII. EXPERIMENTAL STUDY AND DISCUSSION

Our proposed approach is compared to the baseline solver provided by the competition. 30 independent optimisation trials have been performed on each of the problem instances as described in Section V. The population size, mutation rate and maximum generation number are set as 50, 0.5 and $10N$, respectively, with N the number of pickup points.

At the end of each optimisation trial, the final population is validated by the constraint checking program provided by the competition and evaluated by real objective functions, Eqs. (1) and (2). The two best solutions in terms of each objective function among the last population are recorded and denoted as ζ_r^* and ζ_l^*, respectively. All experiments are taken on the same machine with an Intel i7-9700K CPU and 64G RAM.

Figs. 5 and 6 compare solutions for each instance found by our approach and the baseline solver, in terms of the actual objective values, Eqs. (2) and (1), respectively. Reading instructions of Fig. 5 are as follows. ζ_i^* is set as the solution with shortest routing distance in the final population of the i^{th} optimisation trial. Given a solution ζ, $f_r(\zeta)$ and $f_l(\zeta)$ are its real objective evaluations for routing and loading plans, Eqs. (2) and (1), respectively. f_r^B and f_l^B refer to the objective values of solutions found by the baseline solver. The top of Fig. 5 plots, for each instance, $\frac{\frac{1}{30}\sum_{i=1}^{30} f_l(\zeta_i^*)-f_l^B}{f_l^B}$ and $\frac{f_r^B-\frac{1}{30}\sum_{i=1}^{30} f_r(\zeta_i^*)}{f_r^B}$. The bottom one plots $\frac{f_r^B-f_r(\zeta_{best}^*)}{f_r^B}$ and the corresponding $\frac{f_l(\zeta_{best}^*)-f_l^B}{f_l^B}$, with $best = \underset{1\leq i\leq 30}{\arg\min} f_r(\zeta_i^*)$. A positive y-value in Fig. 5 means that the average of our solutions or the best over the 30 trials, respectively, is better than the baseline on the instance indicated by x-axis. Reading instructions of Fig. 6 are the same as Fig. 5 with ζ_i^* set as the solution with highest loading rate in the final population.

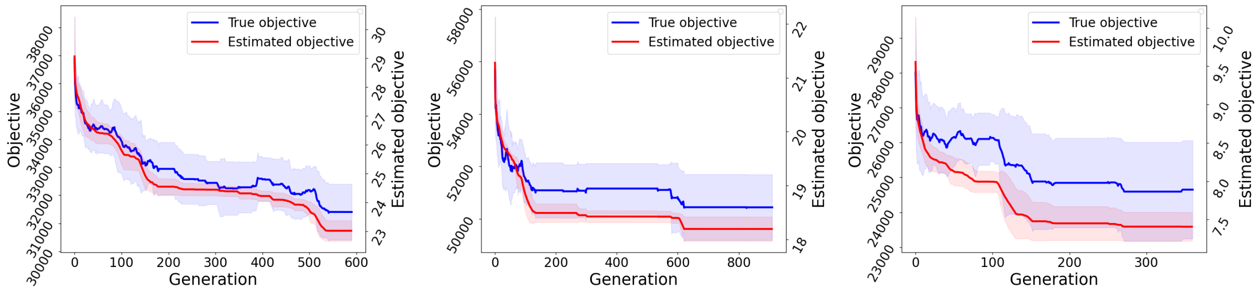

Fig. 4. Averaged actual routing distance and averaged estimated routing objective values of population during optimisation trials of instances SD-CSS10 (left), SD-CSS11 (middle) and SD-CSS4 (right). The estimated values have the similar variation tendency and decreasing trend as the actual distances.

Table II summarises the number of instances of which the averaged or the best of our solutions, in terms of route distance or loading rate, is better than the baseline on both objectives (4^{th} column), neither (5^{th} column) or one of the objectives only (6^{th} and 7^{th} columns), i.e., the solution found by our solver and the one found by baseline solver are non-dominated.

Table II and Figs. 5 and 6 clearly show that our genetic algorithm with objective estimation significantly outperforms the baseline solver. Among 66 problem instances, considering the average performance, the solution ζ_r^* (with shortest routing distance) found by our solver dominates the baseline on 48 instances, while our solution and the baseline are non-dominated on 18 instances; the solution ζ_l^* (with highest loading rate) found by our solver dominates the baseline on 59 instances on average, while our solution and the baseline are non-dominated on 7 instances.

To illustrate the effectiveness of our objective estimation method, Fig. 4 demonstrates the averaged estimated values of solutions and the averaged true objective values of same solutions during the optimisation trails of 3 randomly selected instances. The estimation is informative as the estimated values and real objective values have the similar variation tendency and decreasing trend.

Fig. 7 gives an example of solutions for instance *w-Sha02*. In this example, the loading rate and routing distance of our solution obtained in an arbitrarily chosen optimisation trial are 0.4533 and 726.7, respectively, while the baseline's solution has $f_l^B = 0.3022$ and $f_r^B = 1125.1$. The actual routes of visiting pickup points of our solution are $(7, 8, 2, 6, 1)$ and $(5, 3, 4)$, while the ones of the baseline's solution are $(7, 1, 2)$, $(2, 3, 4, 5, 6)$ and $(6, 8)$. In the solution found by the baseline solver, there are 5 items that can not be loaded into the first two vehicles, therefore one more vehicle is needed. A better visiting order of points is found by our algorithm, which leads to a better loading strategy with only two occupied vehicles. Our routing plan not only has significantly shorter travelling distance and higher loading rate compared to the solution found by the baseline solver, but also is more balanced.

TABLE II
DOMINANCE RELATIONSHIP OF SOLUTIONS FOUND BY OUR SOLVER COMPARED TO THE ONES FOUND BY BASELINE SOLVER.

Dataset			Dominate	Be dominated	Non-dominated	
		Better on	f_l and f_r	Neither	Only f_l	Only f_r
w-SD-CSS	ζ_r^*	Avg.	9	0	0	4
		Best	13	0	0	0
	ζ_l^*	Avg.	13	0	0	0
		Best	13	0	0	0
w-Sha	ζ_r^*	Avg.	14	0	0	1
		Best	15	0	0	0
	ζ_l^*	Avg.	14	0	1	0
		Best	15	0	0	0
CI	ζ_r^*	Avg.	25	0	2	11
		Best	30	0	0	8
	ζ_l^*	Avg.	32	0	6	0
		Best	37	0	1	0

VIII. CONCLUSION

In this paper, we focus on a realistic splitting delivery vehicle routing problems with 3-dimensional loading constraints (3L-SDVRP) and special pickup points (warehouse with restricted access) which considers the total travelling distance and the averaged loading rate (in terms of weight and volume) while satisfying some routing and loading constraints. The studied problem comes from the 2021 HUAWEI Logistics Competition and has more realistic constraints compared to existing 3L-SDVRP models. In this work, we first formulate its complete mathematical model with all the loading constraints considered, then we propose an objective estimation method to efficiently estimate the routing distance. The proposed method is used as surrogate objective evaluations in a genetic algorithm for optimising the problem. Besides the competition instances, more instances are generating based on well-known benchmark sets of related problems. Our proposed approach shows superior performance compared to the competition baseline solver on problem instances provided in the 2021 HUAWEI Logistics Competition benchmark and the novel instances.

As future work, we will investigate in efficient approaches for tackling large-scale 3L-SDVRP instances, in particular, designing heuristics for assembling items into blocks for reducing the dimensionality of problem. Studying more realistic 3L-SDVRPs by considering uncertainties in real-world routing scenarios is another direction [19], [20].

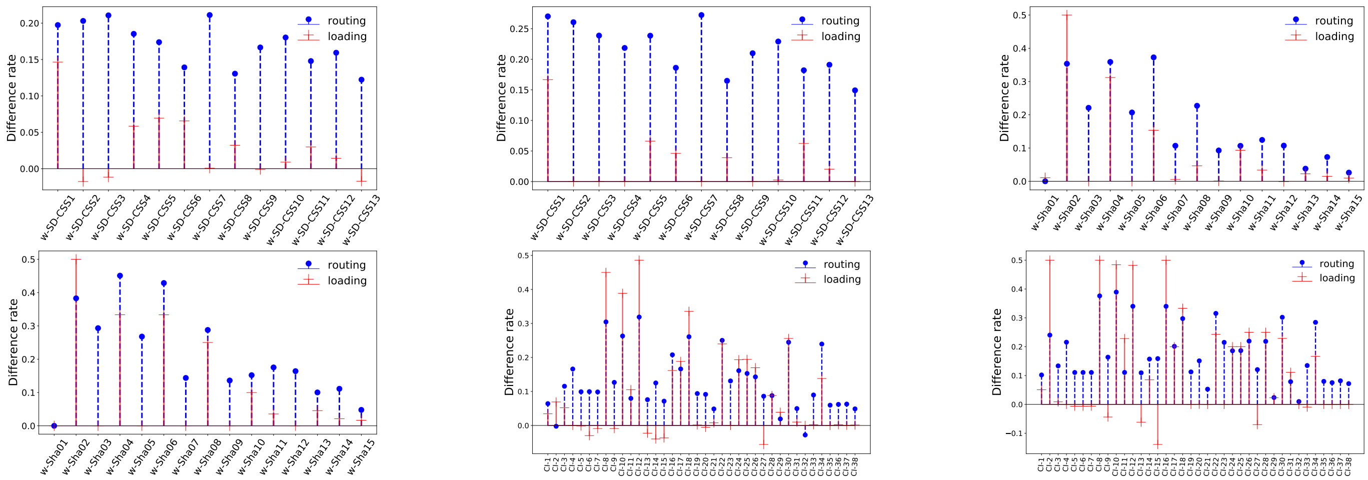

Fig. 5. Comparing our solutions with shortest routing distance in the final population of each optimisation trial to the solutions found by the baseline solver. Top: $\frac{\frac{1}{30}\sum_{i=1}^{30} f_l(\zeta_i^*)-f_l^B}{f_l^B}$ (red dot) and $\frac{f_r^B-\frac{1}{30}\sum_{i=1}^{30} f_r(\zeta_i^*)}{f_r^B}$ (blue dot) averaged over 30 trials for each instance; bottom: $\frac{f_r^B-f_r(\zeta_{best}^*)}{f_r^B}$ (blue dot) of the best trial (i.e., shortest distance) and the corresponding $\frac{f_l(\zeta_{best}^*)-f_l^B}{f_l^B}$ (red $+$) with $best = \underset{1\leq i\leq 30}{\arg\min} f_r(\zeta_i^*)$ for each instance.

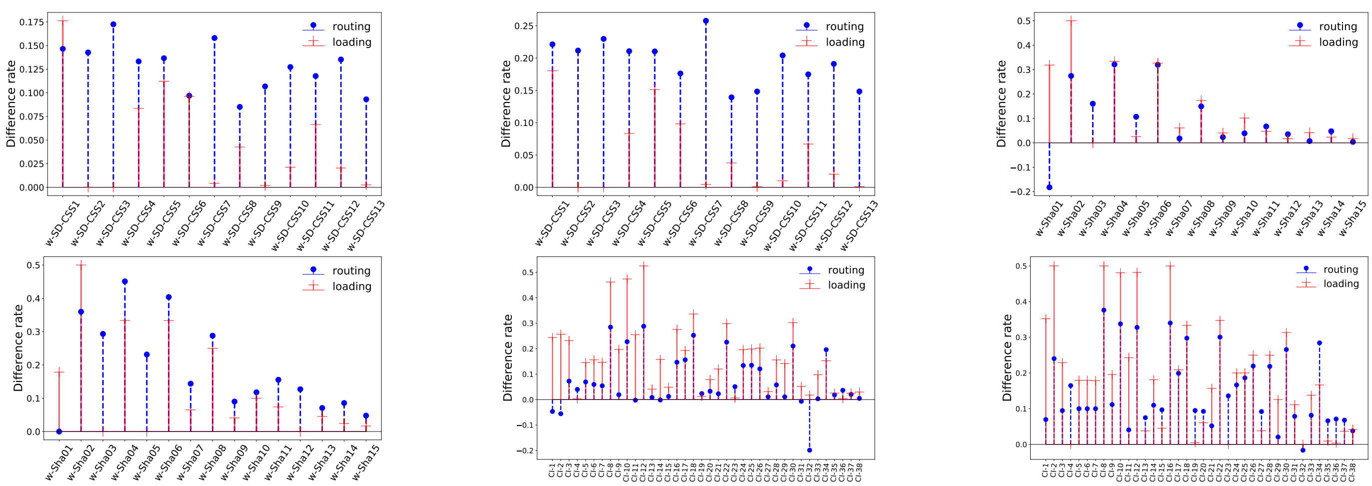

Fig. 6. Comparing our solutions with highest loading rate in the final population of each optimisation trial to the solutions found by the baseline solver.

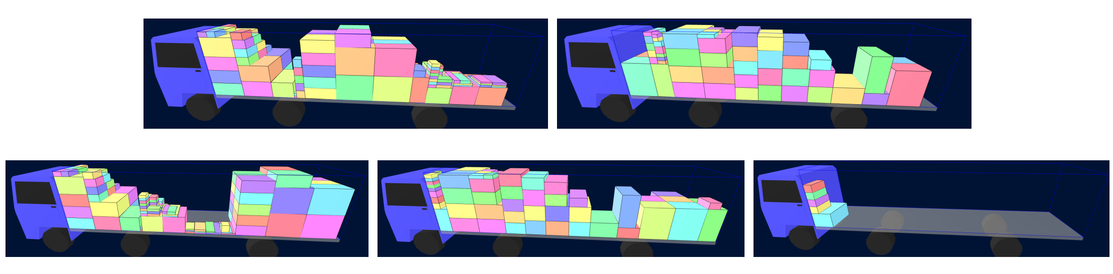

Fig. 7. Screenshots of solution visualisation for *w-Sha02*. Top: our solution composed by two routes. Bottom: baseline solution composed by three routes.

REFERENCES

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

[2] K. C. Tan, L. H. Lee, Q. Zhu, and K. Ou, "Heuristic methods for vehicle routing problem with time windows," *Artificial intelligence in Engineering*, vol. 15, no. 3, pp. 281–295, 2001.

[3] G. Laporte and F. Semet, "Classical heuristics for the capacitated VRP," in *The vehicle routing problem*. SIAM, 2002, pp. 109–128.

[4] R. Baldacci, A. Mingozzi, and R. Roberti, "Recent exact algorithms for solving the vehicle routing problem under capacity and time window constraints," *European Journal of Operational Research*, vol. 218, no. 1, pp. 1–6, 2012.

[5] K. Tang, J. Wang, X. Li, and X. Yao, "A scalable approach to capacitated arc routing problems based on hierarchical decomposition," *IEEE transactions on cybernetics*, vol. 47, no. 11, pp. 3928–3940, 2016.

[6] J. Liu and X. Yao, "Self-adaptive decomposition and incremental hyperparameter tuning across multiple problems," in *Proceedings of the 2019 IEEE Symposium Series on Computational Intelligence*, 2019, pp. 1590–1597.

[7] S. Ceschia, A. Schaerf, and T. Stützle, "Local search techniques for a routing-packing problem," *Computers & industrial engineering*, vol. 66, no. 4, pp. 1138–1149, 2013.

[8] X. Li, M. Yuan, D. Chen, J. Yao, and J. Zeng, "A data-driven three-layer algorithm for split delivery vehicle routing problem with 3D container loading constraint," in *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2018, pp. 528–536.

[9] J. Yi and A. Bortfeldt, "The capacitated vehicle routing problem with three-dimensional loading constraints and split delivery—a case study," in *Operations Research Proceedings 2016*. Springer, 2018, pp. 351–356.

[10] A. Bortfeldt and J. Yi, "The split delivery vehicle routing problem with three-dimensional loading constraints," *European Journal of Operational Research*, vol. 282, no. 2, pp. 545–558, 2020.

[11] S. Wøhlk, "A decade of capacitated arc routing," in *The Vehicle Routing Problem: Latest Advances and New Challenges*. Springer, 2008, pp. 29–48.

[12] A. Bortfeldt, "A hybrid algorithm for the capacitated vehicle routing problem with three-dimensional loading constraints," *Computers & Operations Research*, vol. 39, no. 9, pp. 2248–2257, 2012.

[13] M. Gendreau, M. Iori, G. Laporte, and S. Martello, "A tabu search algorithm for a routing and container loading problem," *Transportation Science*, vol. 40, no. 3, pp. 342–350, 2006.

[14] G. Fuellerer, K. F. Doerner, R. F. Hartl, and M. Iori, "Metaheuristics for vehicle routing problems with three-dimensional loading constraints," *European Journal of Operational Research*, vol. 201, no. 3, pp. 751–759, 2010.

[15] P. Lacomme, H. Toussaint, and C. Duhamel, "A GRASP× ELS for the vehicle routing problem with basic three-dimensional loading constraints," *Engineering Applications of Artificial Intelligence*, vol. 26, no. 8, pp. 1795–1810, 2013.

[16] L. Junqueira, J. F. Oliveira, M. A. Carravilla, and R. Morabito, "An optimization model for the vehicle routing problem with practical three-dimensional loading constraints," *International Transactions in Operational Research*, vol. 20, no. 5, pp. 645–666, 2013.

[17] Q. Ruan, Z. Zhang, L. Miao, and H. Shen, "A hybrid approach for the vehicle routing problem with three-dimensional loading constraints," *Computers & Operations Research*, vol. 40, no. 6, pp. 1579–1589, 2013.

[18] Huawei, "The EMO 2021 Huawei logistics competition," https://www.noahlab.com.hk/logistics-ranking/#/home/index, accessed: 2021-02-05.

[19] J. Liu, K. Tang, and X. Yao, "Robust optimization in uncertain capacitated arc routing problems: Progresses and perspectives," *IEEE Computational Intelligence Magazine*, vol. 16, no. 1, pp. 63–82, 2021.

[20] H. Tong, L. L. Minku, S. Menzel, B. Sendhoff, and X. Yao, "A hybrid local search framework for the dynamic capacitated arc routing problem," in *Proceedings of the Genetic and Evolutionary Computation Conference Companion*, 2021, pp. 139–140.