Submitted to *Transportation Science*
manuscript (Please, provide the manuscript number!)

General edge assembly crossover driven memetic search for split delivery vehicle routing

Pengfei He and Jin-Kao Hao* (Corresponding author)
Université d'Angers, LERIA, 2 Boulevard Lavoisier, 49045 Angers, France,
pengfeihe606@gmail.com, jin-kao.hao@univ-angers.fr.

Transportation Science, https://doi.org/10.1287/trsc.2022.1180

The split delivery vehicle routing problem is a variant of the well-known vehicle routing problem, where each customer can be visited by several vehicles. The problem has many practical applications, but is computationally challenging. This paper presents an effective memetic algorithm for solving the problem with a fleet of limited or unlimited vehicles. The algorithm features a general edge assembly crossover to generate promising offspring solutions from the perspective of assembling suitable edges and an effective local search to improve each offspring solution. The algorithm is further reinforced by a feasibility-restoring procedure, a diversification-oriented mutation and a quality-and-distance pool updating technique. Extensive experiments on 324 benchmark instances indicate that our algorithm is able to update 143 best upper bounds in the literature and match the best results for 156 other instances. Additional experiments are presented to obtain insights into the roles of the key search ingredients of the algorithm. The method was ranked second at the 12th DIMACS Implementation Challenge on Vehicle Routing - SDVRP Track.

Key words: Split delivery vehicle routing; Vehicle routing; Heuristics; Edge assembly crossover, Hybrid search.

History: Submission December 2021, revised June 2022, September 2022, accepted 07 September 2022

---

1. Introduction

The split delivery vehicle routing problem (SDVRP) (Dror and Trudeau 1989, 1990) is a variant of the conventional vehicle routing problem (VRP). Unlike the VRP where each customer is visited exactly by one vehicle, the SDVRP allows a customer's demand to be split and served by several homogeneous capacitated vehicles starting and finishing at the depot.

Formally, let $\mathcal{G}=(\mathcal{V},\mathcal{E})$ be an undirected graph where $\mathcal{V}=\{0,1,\cdots,n\}$ is the vertex set with 0 being the depot and $\mathcal{N}=\{1,\cdots,n\}$ representing n customers and $\mathcal{E}$ is the edge set. Each customer $i\in\mathcal{N}$ is associated with an integer demand $d_i\in\mathcal{Z}^+$. Let $\mathcal{C}=(c_{ij})$ be a non-negative cost (distance) matrix associated with $\mathcal{E}$ satisfying the triangle inequality ($c_{ij} + c_{jk} > c_{ik}$ for all $i, j, k \in \mathcal{V}$ and $i \neq j \neq k$). Given a set of K identical vehicles with capacity Q available at the depot, the SDVRP is to find K routes (K can be limited or unlimited) such that 1) each route starts at the depot to serve a number of customers and ends at the depot without exceeding the vehicle capacity Q, 2) the demand d_i of customer $i \in \mathcal{N}$ can be split and served by more than one vehicle, and 3) the total traveling distance of the K routes is minimized. According to the number K of the available vehicles (fleet size), the problem is called the SDVRP-LF (for limited fleet size) if K is fixed or the SDVRP-UF (for unlimited fleet size) otherwise. For the SDVRP-LF, K is fixed to $K_{min} = \lceil(\sum_{i=1}^{n} d_i/Q)\rceil$ to ensure the feasibility of the solution. A mathematical formulation of both problems is shown in Appendix A.

Like the conventional VRP, the SDVRP has a number of applications such as determining routes and schedules for newspaper delivery (Song, Lee, and Kim 2002) and waste collection (Archetti and Speranza 2004). Meanwhile, the SDVRP has been much less investigated compared to the VRP and its variants such as the capacitated VRP, the VRP with time windows and the VRP with profits. Still, since the introduction of the SDVRP, a number of algorithms using exact and heuristic approaches have been proposed. Representative exact approaches are based on various formulations (Belenguer, Martinez, and Mota 2000, Ozbaygin, Karasan, and Yaman 2018) and the branch-and-cut framework (Archetti, Bianchessi, and Speranza 2014, Munari and Savelsbergh 2022). These exact approaches are able to provide the optimal solutions for some small or medium-sized instances with up to some 100 customers. For larger instances, heuristics and metaheuristics are preferred to find suboptimal solutions with a reasonable time, as reviewed in Section 2.

This work aims to advance the state-of-the-art for solving large SDVRP instances effectively and efficiently. The contributions of this paper are summarized as follows.

- We propose a memetic algorithm (SplitMA)[1] that combines several complementary search components including a general edge assembly crossover (gEAX) to generate promising offspring solutions and a local search associated with a maximum splits strategy to improve offspring solutions. The gEAX crossover transmits common edges from parent solutions to offspring solutions while reassembling non-common edges of parent solutions. The local search exploits both VRP neighborhood operators and SDVRP neighborhood operators reinforced by the maximum splits strategy, which ensures that a customer will not be served by too many vehicles. The algorithm additionally integrates dedicated repairing techniques to ensure the feasibility of offspring solutions, a mutation to diversify each new solution, and an advanced updating strategy to maintain a healthy population.

[1] The SplitMA algorithm was ranked second at the 12th DIMACS Implementation Challenge on Vehicle Routing - SDVRP Track `http://dimacs.rutgers.edu/programs/challenge/vrp/`.

- We illustrate the competitiveness of the algorithm on four sets of 324 instances of the SDVRP-LF and SDVRP-UF problems compared to the state-of-the-art algorithms. In particular, we report 143 new best upper bounds that can be useful for future studies. We investigate the underlying algorithmic components to shed light on their contributions to the performance of the algorithm. Specifically, we provide insights about why the gEAX crossover works well on the SDVRP and present for the first time experimental evidences that high-quality solutions are close to each other and are also close to optimal solutions.
- This work shows the interest of the general idea of the edge assembly crossover. The gEAX crossover, which generalizes the popular EAX crossover for the TSP (Nagata 1997, Nagata and Kobayashi 2013), provides a powerful solution recombination mechanism that can be advantageously applied not only to the SDVRP, but also to other routing problems where the associated graphs of candidate solutions do not necessarily have the same degree for their vertices.

The remainder of this paper is organized as follows. Section 2 provides a literature review on solution approaches for the SDVRP. Section 3 presents the details of the proposed algorithm. Section 4 shows computational results and comparisons. Section 5 investigates key ingredients of the proposed algorithm. Section 6 draws conclusions with research perspectives.

2. Literature review

A comprehensive review of exact and heuristic solution approaches until 2012 can be found in Archetti and Speranza (2012). In this section, we focus on a literature review of heuristic approaches, while mentioning some representative studies on exact approaches developed since 2014. Table 1 summarizes the methods discussed in this section.

Archetti, Bianchessi, and Speranza (2014) presented two branch-and-cut (B&C) algorithms, where the first uses the formulation of Belenguer, Martinez, and Mota (2000) and the other adopts a commodity-flow formulation. The methods solved 17 instances to optimality (one instance with 100 customers). Ozbaygin, Karasan, and Yaman (2018) created a compact vehicle-indexed flow formulation and presented computational results including optimal solutions for instances with 76 customers. Munari and Savelsbergh (2022) proposed three compact formulations and developed a B&C algorithm, which solved 91 instances to proven optimality (with up to 80 customers). For larger instances, heuristics/metaheuristics such as neighborhood-based local search and population-based search are used to find suboptimal solutions with a reasonable time.

The first local search algorithm for solving the SDVRP was presented by Dror and Trudeau (1989, 1990). Two neighborhood operators, namely *k-Split* and *RouteAddition*, were combined into the local search. The *k-Split* operator divides the demand of a customer and inserts the divided demand into different routes with an enough residual capacity. On the contrary, the *RouteAddition* operator

Table 1 Representative exact and heuristic algorithms for the SDVRP

Literature	Framework	Problem Solved
Exact algorithms		
Archetti, Bianchessi, and Speranza (2014)	B&C	Both
Ozbaygin, Karasan, and Yaman (2018)	Vehicle indexed flow formulation	Both
Munari and Savelsbergh (2022)	B&C	Both
Heuristic methods		
Dror and Trudeau (1989, 1990)	Local search	SDVRP-UF
Derigs, Li, and Vogel (2010)	Local search	SDVRP-UF
Archetti, Speranza, and Hertz (2006)	Tabu search	SDVRP-UF
Aleman and Hill (2010)	Tabu search	SDVRP-UF
Berbotto, García, and Nogales (2014)	Tabu search	SDVRP-LF
Zhang et al. (2015)	Tabu search	SDVRP-UF
Chen et al. (2017)	Priori split strategy	SDVRP-UF
Aleman, Zhang, and Hill (2010)	Variable neighborhood descent	SDVRP-LF
Han and Chu (2016)	Variable neighborhood descent	SDVRP-UF
Silva, Subramanian, and Ochi (2015)	Iterated local search	Both
Mota, Campos, and Corberán (2007)	Scatter search algorithm	SDVRP-LF
Campos, Corberán, and Mota (2008)	Scatter search algorithm	SDVRP-UF
Shi et al. (2018)	Particle swarm optimization	SDVRP-UF
Chen, Golden, and Wasil (2007)	Hybrid algorithm/matheuristic	SDVRP-UF
Archetti, Speranza, and Savelsbergh (2008)	Hybrid algorithm/matheuristic	SDVRP-UF
Jin, Liu, and Eksioglu (2008)	Hybrid algorithm/matheuristic	SDVRP-UF
Boudia, Prins, and Reghioui (2007)	Memetic algorithm	SDVRP-UF
Wilck and Cavalier (2012)	Genetic algorithm	SDVRP-LF

tries to remove a split customer from all routes and create a new route to serve the customer. These two operators were widely used in follow-up studies. To better handle the problem and cope with the complexity of the SDVRP, other neighborhood operators were presented. Boudia, Prins, and Reghioui (2007) proposed two new operators where two or three customers in two routes are swapped with the possibility of splitting their demands. Derigs, Li, and Vogel (2010) introduced a new relocation operator where three routes were manipulated to explore neighboring solutions.

The tabu search metaheuristic was adapted to the SDVRP by Archetti, Speranza, and Hertz (2006) for the first time, where a neighboring solution was obtained by removing a customer from a set of routes in which it was currently visited and inserting it either into a new route or into an existing route with an enough residual capacity. This algorithm outperformed significantly Dror and Trudeau's algorithms (Dror and Trudeau 1989, 1990). Then, Aleman and Hill (2010) proposed a so-called tabu search with vocabulary building approach (TSVBA). An initial set of solutions was constructed firstly and attractive solution attributes were summarized to explore new solutions. Solutions in the set evolved along with the searching progress. The random granular tabu search (RGTS) was proposed by Berbotto, García, and Nogales (2014), where a heuristic prunning technique is used to filter non-promising neighborhood solutions and speed up the neighborhood search. Another tabu search algorithm, namely forest-based tabu search (FBTS), was introduced by

Zhang et al. (2015), where the forest structure is used to represent each solution. Several dedicated operators based on the forest structure were also designed, and the experimental results showed that the FBTS algorithm was competitive with existing algorithms.

Mota, Campos, and Corberán (2007) proposed a scatter search heuristic to address the SDVRP-LF for the first time. Campos, Corberán, and Mota (2008) introduced another scatter search for the SDVRP-LF with two distinct procedures for generating initial populations. Han and Chu (2016) presented a multi-start solution approach for solving the SDVRP-UF. Aleman, Zhang, and Hill (2010) proposed an adaptive memory algorithm for the SDVRP-LF, which uses a constructive procedure for initial solution generation and a variable neighborhood descent (VND) for solution improvement. The constructive procedure builds an initial solution by greedily inserting customers with a mechanism called route angle control. The VND procedure follows to seek improved solutions by exploring three commonly used neighborhoods. Silva, Subramanian, and Ochi (2015) presented a multi-start iterated local search (SplitILS) for both cases of limited and unlimited fleet. SplitILS is composed of an efficient perturbation procedure and a randomized variable neighborhood descent exploring numerous VRP neighborhood operators and SDVRP neighborhood operators. Extensive experiments indicated that SplitILS performed remarkably well and dominated previous algorithms. Chen et al. (2017) introduced a novel and efficient approach to solve the SDVRP-UF, where each customer's demand was split into small pieces in advance and then the SDVRP was solved by applying leading VRP algorithms (Groër, Golden, and Wasil 2010). Shi et al. (2018) proposed the first particle swarm optimization for the SDVRP-UF and reported some new upper bounds, even though its performance is generally worse than SplitILS (Silva, Subramanian, and Ochi 2015).

In addition to these local search approaches, two hybrid population-based approaches were investigated. Boudia, Prins, and Reghioui (2007) presented the memetic algorithm with population management, which used the giant tour crossover (Prins 2004) and a local search procedure including two new swap moves. The algorithm performed competitively compared to the tabu search of Archetti, Speranza, and Hertz (2006) on a number of benchmark instances. Wilck and Cavalier (2012) proposed another hybrid genetic algorithm that reproduced offspring solutions using route-by-route methods and reported competitive results with previous algorithms, though its results were significantly improved by SplitILS (Silva, Subramanian, and Ochi 2015) later.

Our review shows that the algorithms in Silva, Subramanian, and Ochi (2015), Zhang et al. (2015), Berbotto, García, and Nogales (2014), Aleman, Zhang, and Hill (2009), Campos, Corberán, and Mota (2008), Wilck and Cavalier (2012), Aleman and Hill (2010), Boudia, Prins, and Reghioui (2007), Derigs, Li, and Vogel (2010) hold the best-known results for the SDVRP-LF and SDVRP-UF. Thus, we use these approaches as our reference algorithms for the comparative study.

3. General edge assembly crossover driven memetic algorithm

Population-based evolutionary algorithms have been successfully applied to the traveling salesman problem (Nagata 1997, Nagata and Kobayashi 2013) and several vehicle routing problems (Potvin 2009, Nagata and Bräysy 2009, Nagata, Bräysy, and Dullaert 2010, Prins 2004, Vidal et al. 2012, 2013, 2014). The proposed SplitMA algorithm for the SDVRP is a population-based hybrid algorithm that uses a dedicated edge assembly crossover to generate new solutions and an effective local optimization to improve the offspring solutions. SplitMA also applies a mutation to diversify each offspring solution and an advanced pool updating strategy to manage the population.

Algorithm 1: The memetic algorithm for the SDVRP

Input: Instance I;

Output: The best solution φ^* found so far;

1 **begin**
2 $\mathcal{P} \leftarrow PopulationInitial(I)$; `/* Initializing the population` $\mathcal{P}$`, Section 3.1 */`
3 $\varphi^* \leftarrow \arg\min\{f(\varphi_i) | i = 1, 2, \cdots, |\mathcal{P}|\}$; `/*` φ^* `Record the best solution found so far */`
4 **while** *Stopping condition is not met* **do**
5 $\{\varphi_A, \varphi_B\} \leftarrow ParentSelection(\mathcal{P})$; `/* Selecting two parental solutions randomly */`
6 $\{\varphi_O^1, \varphi_O^2, \cdots, \varphi_O^\beta\} \leftarrow gEAX(\varphi_A, \varphi_B)$; `/* Generating offspring solutions, Section 3.2 */`
7 **for** $i = 1$ *to* β **do**
8 $\varphi_O^i \leftarrow RestoringFeasibility(\varphi_O^i)$; `/* Restoring feasibility, Section 3.3 */`
9 $\varphi_O^i \leftarrow Mutation(\varphi_O^i)$; `/* Generating mutation, Section 3.4 */`
10 $\varphi_O^i \leftarrow LocalSearch(\varphi_O^i)$; `/* Improving the offspring solution, Section 3.5 */`
11 **if** *SDVRP-LF* **then**
12 $\varphi_O^i \leftarrow EmptyRoute(\varphi_O^i)$; `/* Reducing routes to` K_{min}`, Section 3.5.3 */`
13 **end**
14 **if** $f(\varphi_O^i) < f(\varphi^*)$ **then**
15 $\varphi^* \leftarrow \varphi_O^i$;
16 **end**
17 $\mathcal{P} \leftarrow PoolUpdating(\mathcal{P}, \varphi_O^i)$; `/* Managing the population, Section 3.6 */`
18 **end**
19 **end**
20 **return** φ^*;
21 **end**

The general scheme of SplitMA is outlined in Algorithm 1. SplitMA starts from an initial population $\mathcal{P}$ constructed by the population initialization procedure (Line 2 of Algorithm 1). Then the algorithm evolves the population through a number of generations by applying the gEAX crossover, the local optimization procedure and the population updating procedure (Lines 4-19). Of particular interest is the general edge assembly crossover operator (gEAX) (Line 6) that creates at each generation β offspring solutions by assembling the edges of two parent solutions. After restoring the feasibility of each offspring solution in terms of customer demand and vehicle capacity

(Line 8), the solution is diversified by the mutation operator (Line 9) and then submitted to local optimization for quality improvement (Line 10). Finally, each improved solution is used to update the population by the pool updating strategy (Line 17). For the SDVRP-LF where the fleet size is set to K_{min}, the number of the used vehicles is reduced to this fleet size by emptying some routes if needed (Lines 11-13). During the search, the best solution found so far φ^* is updated each time a solution better than it is discovered (Lines 14-16). The algorithm stops and returns the best solution φ^* when a predefined stopping condition is met (e.g., a maximum cutoff time or maximum number of generations).

3.1. Population initialization

SplitMA starts its evolution from an initial population $\mathcal{P}$, whose size varies between p_{min} and p_{max} ($p_{max} > p_{min}$) during the search process. Similar to Vidal (2022), $4 \times p_{min}$ solutions are first constructed and subsequently improved by the local search (Section 3.5), and then inserted into $\mathcal{P}$ one by one. Once $|\mathcal{P}| = p_{max}$, the surviving strategy (Section 3.6) is triggered to shrink the population $\mathcal{P}$ to p_{min} solutions.

The construction process of each solution works as follows. First, $K_{min} = \lceil (\sum_{i=1}^{n} d_i/Q) \rceil$ routes are created where each route is initialized by the depot and a random customer. Then, for each newly routed customer i, a random unrouted customer j from the δ-nearest neighborhood (see Section 3.5) is selected and inserted into the route after the customer i without split. This insertion process stops when no customer can be inserted into the solution without violating the capacity constraint. Finally, if there are unrouted customers, these customers are dividedly inserted into routes in a greedy way such that the insertions lead to the minimum increase of the objective value (i.e., the total traveling distance). Once all customers are routed, a complete solution is obtained.

3.2. The general edge assembly crossover operator

Crossover is a key component of memetic algorithms and constitutes one leading force to explore the search space (Hao 2012). In this section, we introduce the gEAX crossover for the SDVRP that generalizes the edge assembly crossover (EAX), which was initially designed for the TSP (Nagata 1997, Nagata and Kobayashi 2013) and adapted to the VRP (Nagata and Bräysy 2009). The basic idea of EAX for the TSP and the VRP is to preserve the common edges shared by the parent solutions and assemble non-common edges, based on the knowledge that high-quality solutions of these problems always share a high number of common edges and these common edges form a stable backbone that is highly likely to be part of the optimal solution.

The main difficulty of applying EAX to the SDVRP lies in the fact that EAX assumes that each customer is served by exactly one vehicle. Indeed, for a given TSP and VRP instance defined on a graph $\mathcal{G}$, a candidate solution can be identified by a partial graph of $\mathcal{G}$. Given two parent solutions, each customer vertex necessarily has the same degree of two and EAX uses this property to assemble edges from the parents. However, for the SDVRP, given that each customer can be served by several vehicles, a solution corresponds to a multigraph where parallel edges may exist between some vertices (see Definition 1). Indeed, given the assumption that triangle inequality holds, there is an optimal solution in which each edge between customers is traversed at most once. However, each edge between the depot and a customer may still be traversed several times. Without loss of generality, we use the term 'vertex' to denote both 'depot' and 'customer' in this paper. As a result, the same customer vertex may have different degrees in the multigraphs of the parent solutions, making the EAX crossover inoperative. On the other hand, the idea of assembling specific (promising) edges from the routes of high-quality solutions is highly appealing from the perspective of solution recombination. The general edge assembly crossover gEAX that we introduce in this work benefits from the basic idea of assembling suitable edges and gets around the aforementioned difficulty related to the EAX crossover.

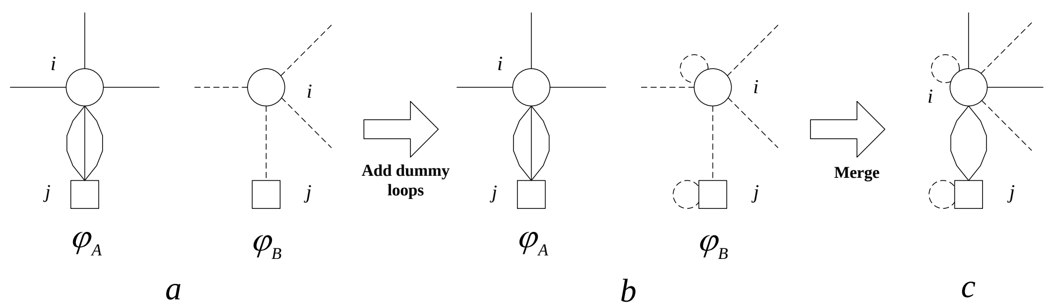


Figure 1 **Illustration of adding dummy edges. (a) A portion of the multigraphs $\mathcal{G}_{\mathcal{A}}$ and $\mathcal{G}_{\mathcal{B}}$ associated to solutions φ_A and φ_B. (b) multigraph $\mathcal{G}_{\mathcal{A}}$ and extended multigraph $\mathcal{G}_{\mathcal{B}}$ with two dummy loops. (c) Joint multigraph of $\mathcal{G}_{\mathcal{A}}$ and extended $\mathcal{G}_{\mathcal{B}}$.**

The key idea of the gEAX crossover is to ensure that each vertex has the same degree in the multigraphs of the parent solutions by introducing dummy edges, rendering it possible to apply the edge assembling operations. To describe the gEAX crossover, we first introduce the following notations.

For a SDVRP instance on graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, let φ be a solution composed of K routes. Following the notation used in Appendix A, let x_{ij}^k be a Boolean variable such that $x_{ij}^k = 1$ if route (or vehicle) k goes from vertex i to vertex j and $x_{ij}^k = 0$ otherwise. Given that edge (i,j) is traversed in the solution φ, then $x_{ij}(\varphi) = \sum_{k=1}^{K} x_{ij}^k$ is the number of times edge (i,j) is traversed in φ and $x_{ij}(\varphi) \geq 1$. For example, in Fig. 1(a) (the square is the depot j and the circle represents customer i), three vehicles (say k_1, k_2 and k_3) of solution φ_A (solid lines) go through the edge (i,j). These three distinct traversals on (i,j) are identified as $x_{ij}^{k_1} = 1$, $x_{ij}^{k_2} = 1$ and $x_{ij}^{k_3} = 1$. Thus $x_{ij}(\varphi_A) = 3$.

For solution φ_B (dot lines), there is only one route k passing through the edge (i,j), thus $x_{ij}^k = 1$ and $x_{ij}(\varphi_B) = 1$.

Definition 1. For a solution φ of the SDVRP instance on graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, we define its corresponding multigraph $\mathcal{G}_\varphi = (\mathcal{V}, \mathcal{E}_\varphi)$ with the multiset of parallel edges $\mathcal{E}_\varphi$ such that for an edge (i,j) of $\mathcal{E}$, there are $x_{ij}(\varphi)$ parallel edges in $\mathcal{E}_\varphi$.

Fig. 1(a) shows a portion of the multigraphs associated to solutions φ_A and φ_B. For solution φ_A, there are three parallel edges between the depot j and the customer i, because three vehicles traverse edge (i,j).

Definition 2. Given two solutions φ_A and φ_B, let $\mathcal{G}_\mathcal{A} = (\mathcal{V}, \mathcal{E}_\mathcal{A})$ and $\mathcal{G}_\mathcal{B} = (\mathcal{V}, \mathcal{E}_\mathcal{B})$ be the corresponding multigraphs. The *degree difference* of vertex i in $\mathcal{G}_\mathcal{A}$ and $\mathcal{G}_\mathcal{B}$ is $\Delta_i = |deg_\mathcal{A}(i) - deg_\mathcal{B}(i)|$ where $deg_\varphi(i)$ denotes the degree of vertex i in solution φ. For a vertex i, if $\Delta_i \neq 0$, $\mathcal{G}_\mathcal{A}$ or $\mathcal{G}_\mathcal{B}$ is extended by adding one or more dummy loops (i,i) to the vertex to render $\Delta_i = 0$.

In the example of Fig. 1(a), $\Delta_i = |deg_\mathcal{A}(i) - deg_\mathcal{B}(i)| = 6 - 4 = 2$ and $\Delta_j = |deg_\mathcal{A}(j) - deg_\mathcal{B}(j)| = 3 - 1 = 2$. Thus, $\mathcal{G}_\mathcal{B}$ is extended by dummy loops (i,i) and (j,j) as shown in see Fig. 1(b). In what follows, an edge $e \in \mathcal{E}_\mathcal{A} \cup \mathcal{E}_\mathcal{B}$ is called a common edge of φ_A and φ_B if $e \in \mathcal{E}_\mathcal{A} \cap \mathcal{E}_\mathcal{B}$; otherwise, e is a non-common edge.

Definition 3. Given two solutions φ_A and φ_B, let $\mathcal{G}_\mathcal{A} = (\mathcal{V}, \mathcal{E}_\mathcal{A})$ and $\mathcal{G}_\mathcal{B} = (\mathcal{V}, \mathcal{E}_\mathcal{B})$ be their extended multigraphs such that $\Delta_i = 0$ holds for each vertex i, we define the joint multigraph $\mathcal{G}_{\mathcal{AB}} = (\mathcal{V}, \{\mathcal{E}_\mathcal{A} \cup \mathcal{E}_\mathcal{B}\} \backslash \{\mathcal{E}_\mathcal{A} \cap \mathcal{E}_\mathcal{B}\})$ by the symmetric difference of $\mathcal{E}_\mathcal{A}$ and $\mathcal{E}_\mathcal{B}$.

Fig. 1(c) shows the joint multigraph $\mathcal{G}_{\mathcal{AB}}$ associated to two solutions φ_A and φ_B.

Given two solutions φ_A, φ_B as well as their corresponding multigraphs $\mathcal{G}_\mathcal{A} = (\mathcal{V}, \mathcal{E}_\mathcal{A})$ and $\mathcal{G}_\mathcal{B} = (\mathcal{V}, \mathcal{E}_\mathcal{B})$, the proposed gEAX crossover generates several offspring solutions in five steps (see Fig. 2 for an illustrative example).

1. **Addition of dummy loops and generation of graph** $\mathcal{G}_{\mathcal{AB}} = (\mathcal{V}, \mathcal{E}_{\mathcal{AB}})$. At the beginning, dummy loops are added to make the *degree difference* become 0 for all vertices in the multigraphs $\mathcal{G}_\mathcal{A}$ and $\mathcal{G}_\mathcal{B}$. Specifically, for each vertex i, the number of added dummy loops (i,i) is $\frac{|deg_\mathcal{A}(i) - deg_\mathcal{B}(i)|}{2}$. If $deg_\mathcal{A}(i) > deg_\mathcal{B}(i)$, dummy loops are added into $\mathcal{E}_\mathcal{B}$, and vice versa, as illustrated in Fig. 1(b). Once the *degree difference* becomes 0 for all vertices in the multigraphs $\mathcal{G}_\mathcal{A}$ and $\mathcal{G}_\mathcal{B}$, we create the joint multigraph $\mathcal{G}_{\mathcal{AB}} = (\mathcal{V}, \mathcal{E}_{\mathcal{AB}})$ with $\mathcal{E}_{\mathcal{AB}} = \{\mathcal{E}_\mathcal{A} \cup \mathcal{E}_\mathcal{B}\} \backslash \{\mathcal{E}_\mathcal{A} \cap \mathcal{E}_\mathcal{B}\}$. In the example of Fig. 2, four dummy loops are added.

2. **Generation of *AB-cycles***. From the joint multigraph $\mathcal{G}_{\mathcal{AB}}$, a number of *AB-cycles* are generated where each new *AB-cycle* is constructed as follows. A random vertex is selected to initialize an empty *AB-cycle*; then edges from $\mathcal{E}_\mathcal{A}$ and $\mathcal{E}_\mathcal{B}$ are traced alternatively to extend the ongoing *AB-cycle*, and each traced edge is removed from $\mathcal{G}_{\mathcal{AB}}$; the *AB-cycle* is constructed successfully when the traced edges lead to a cycle. After the construction of the current *AB-cycle*, if $\mathcal{G}_{\mathcal{AB}}$ is

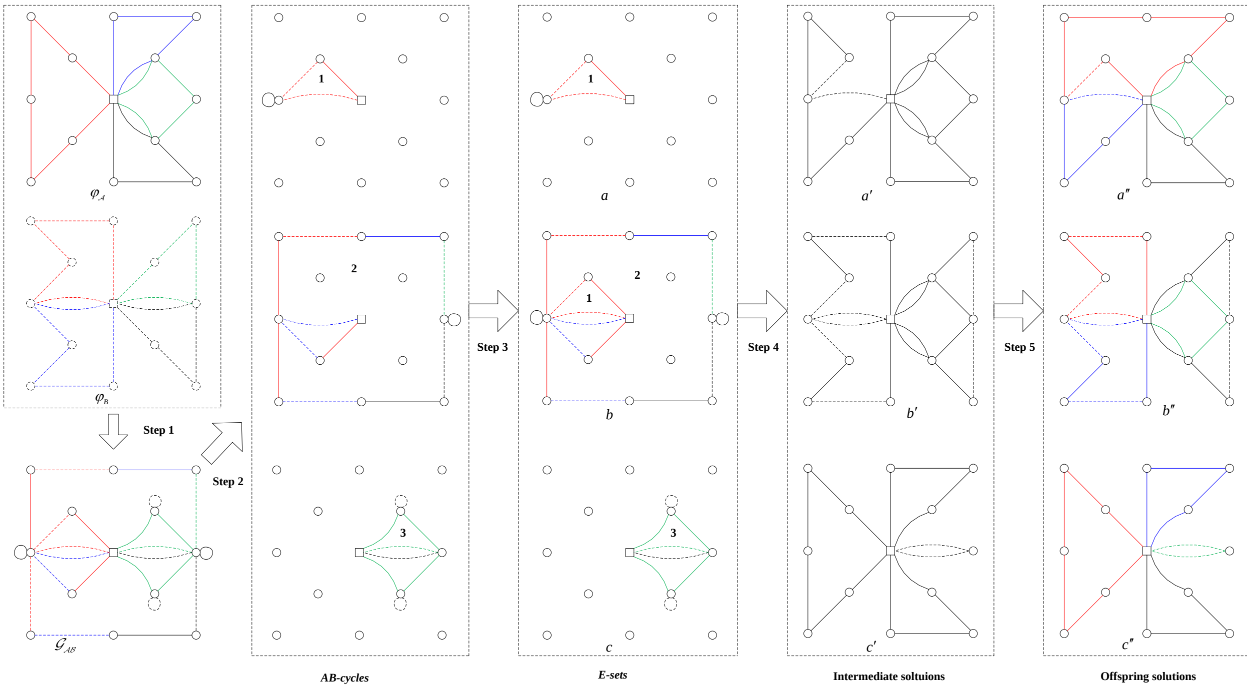


Figure 2 Illustration of the gEAX crossover.

not empty, the process continues to build the next *AB-cycle*. The process stops and returns all *AB-cycles* once $\mathcal{G}_{\mathcal{AB}}$ becomes empty. As shown in Fig. 2, three *AB-cycles* are generated from $\mathcal{G}_{\mathcal{AB}}$. One notices that each *AB-cycle* contains at least four edges. Let C denote the set of m *AB-cycles* obtained from this step.

3. **Generation of *E-sets***. From the set of m *AB-cycles* $C = \{\mathcal{C}_1, \mathcal{C}_2, \cdots, \mathcal{C}_m\}$, a set of *E-sets* is created, where an *E-set* is an union of *AB-cycles*. Each new *E-set* $\mathcal{E}_i$ is initialized by an *AB-cycle* $\mathcal{C}'$ in C and $\mathcal{C}'$ is removed from C. Then, each remaining *AB-cycle* $\mathcal{C}''$ of C are checked. If $\mathcal{C}''$ shares at least one vertex with $\mathcal{E}_i$, $\mathcal{C}''$ is added to $\mathcal{E}_i$ and removed from C. A complete *E-set* ($\mathcal{E}_i$) is achieved when no *AB-cycles* can be added into $\mathcal{E}_i$. This process stops when no *AB-cycle* is left (i.e., C becomes empty). In the example of Fig. 2, the three *AB-cycles* should be combined to form one single *E-set* since the depot is shared. However, for illustrative purpose of steps 4 and 5 below, we suppose three *E-sets* as shown in Fig. 2. Let E denote the set of *E-sets* obtained from this step.

4. **Generation of intermediate solutions**. For each *E-set* $\mathcal{E}_i$ of E, an intermediate solution is generated by using a random parent (say φ_A) as the basic solution. The dummy loops in the *E-sets* $\mathcal{E}_i$ are first removed. Then, the intermediate solution φ_i' is constructed based on φ_A by removing from it the edges of $\mathcal{E}_\mathcal{A}$ shared with $\mathcal{E}_i$ and adding the edges of $\mathcal{E}_\mathcal{B}$ shared with $\mathcal{E}_i$, that is, $\varphi_i' \leftarrow (\mathcal{E}_\mathcal{A} \setminus (\mathcal{E}_i \cap \mathcal{E}_\mathcal{A})) \cup (\mathcal{E}_i \cap \mathcal{E}_\mathcal{B})$. Such a strategy guarantees that all common edges in φ_A and φ_B are necessarily inherited by intermediate solutions. Moreover, all edges in intermediate solutions come from parent solutions. Fig. 2$(a' - c')$ illustrate the three intermediate solutions from this step.

5. **Elimination of isolated subtours**. An intermediate solution may include one or more isolated subtours, such as the triangle subtour in the upper left corner of Fig. 2(a'). The 2-opt* heuristic (Potvin and Rousseau 1995) is then adopted to eliminate these subtours. For each randomly selected subtour, an edge is removed from the subtour and an edge is removed from another route. Then two new edges are introduced to connect two routes. This process is exactly the same as the M8 and M9 neighborhood operators introduced in Section 3.5.1. Fig. 2(a'') illustrates the offspring solution after subtour elimination from the intermediate solution of Fig. 2(a').

The complexity of gEAX can be summarized as follows. Suppose without loss of generality that $|\mathcal{E}_{\mathcal{A}}| \geq |\mathcal{E}_{\mathcal{B}}|$. In the first four steps, there are $|\mathcal{E}_{\mathcal{A}}|+|\mathcal{E}_{\mathcal{B}}|$ edges involved, leading to a time complexity of $|\mathcal{E}_{\mathcal{A}}|$. For the fifth step, the time complexity of 2-opt* is $O(n \times \delta)$, where δ is a parameter (Introduced in Section 3.5). Thus, the time complexity of gEAX is $O(n \times \delta)$. Moreover, $|\mathcal{E}_{\mathcal{A}}|$ edges are invoked and thus the space complexity is $O(|\mathcal{E}_{\mathcal{A}}|)$.

The gEAX crossover follows the idea of the EAX crossover initially designed for the VRP (Nagata and Bräysy 2009) and inherits its advantages, while relaxing the customer demand and capacity constraints. A pair of solutions can generate a variety of offspring solutions with relatively short edges from the parent solutions. More importantly, gEAX overcomes the limitation of EAX that parent solutions (precisely their multigraphs) need to possess the same degree for each vertex. As we show in Sections 4 and 5.1, gEAX significantly contributes to the performance of the proposed algorithm. In Section 5.2, we provide experimental evidences to understand why gEAX is a meaningful crossover for the SDVRP. Finally, the idea behind gEAX also provides a basis for designing meaningful edge assembly crossovers for other rich routing problems such as team orienteering, location routing as well as arc routing.

3.3. Restoring the feasibility of offspring solutions

The customer demand and vehicle capacity are ignored during the gEAX crossover process. As such, an offspring solution may be infeasible in terms of these constraints. This section describes how the feasibility of an offspring solution is restored.

3.3.1. Restoring customers' demand When the routes from the parent solutions are recombined by gEAX, the total amount of served demand of a customer in an offspring solution can be different from the customer's demand. Suppose that $d_i(r_k)$ is the served demand of customer i by route r_k. For example, for the offspring b'' of Fig. 3, customer i (denoted by the red dot) is visited by two routes r_3 and r_4 with the total amount of served demand $d_i(r_3) + d_i(r_4)$. However, since route r_4 in solution b'' entirely comes from φ_A that serves the full demand d_i already, we have $d_i(r_3) + d_i(r_4) > d_i$. Thus, for each customer i, we need to adjust the demand distribution among the routes visiting the customer and make sure that $\sum_{k=1}^{K} d_i(r_k) = d_i$.

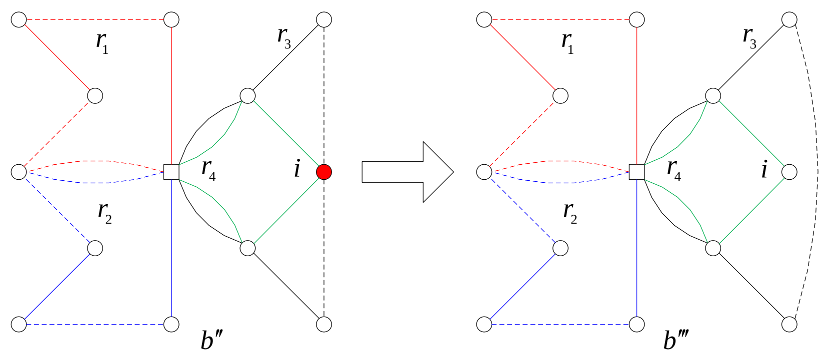


Figure 3 Illustration of balancing demands

We distinguish two cases (i) $\sum_{k=1}^{K} d_i(r_k) > d_i$, and (ii) $\sum_{k=1}^{K} d_i(r_k) < d_i$. Let d_{r_k} be the total load of route r_k. For the first case, the capacity excess $d_{r_k} - Q$ (Q is the vehicle capacity) of each route r_k visiting customer i is calculated, and the resulting values are sorted from the largest to the smallest. Then, the route r_k with the largest capacity excess is identified. If $\sum_{k=1}^{K} d_i(r_k) - d_i > d_i(r_k)$, the customer i is removed from route r_k. Otherwise the amount of demand $d_i(r_k) - (\sum_{k=1}^{K} d_i(r_k) - d_i)$ is removed from route r_k, and the demand of customer i is restored, that is $\sum_{k=1}^{K} d_i(r_k) = d_i$. This process is looped until the demand of all customers is restored. For the second case, the process is similar and operates with the residual capacity of $Q - d_{r_k}$.

3.3.2. Restoring the capacity constraints In addition to the customer demand, the offspring solutions generated by the gEAX crossover may violate the capacity constraint as well. To restore the capacity feasibility of an offspring solution, we apply two well-known inter-route move operators (i.e., insert* and 2-opt*).

Specifically, let φ be an infeasible offspring solution and $f_c(\varphi)$ be its fitness as defined by $f_c(\varphi) = f(\varphi) + p_c \times f_p(\varphi)$, where $f(\varphi)$ is the traveling cost, $f_p(\varphi)$ is the total overcapacity in solution φ, and p_c is a penalty parameter initialized to be the ratio between the longest edge and the largest demand. The repair process operates on an overcapacitated route r and uses insert* (Archetti, Speranza, and Hertz 2006) and 2-opt* (2-opt* corresponds to M8 and M9 of Section 3.5.1) to repair the route. During this process, a tabu list is used to prevent a performed move from being reversed. After each repair operation involving two routes, the set of infeasible routes $\mathcal{R}_{inf}$ is updated. The penalty parameter p_c is multiplied by 10 if no feasible move can be found while there are still infeasible routes ($\mathcal{R}_{inf} \neq \emptyset$). The procedure continues until all routes becomes feasible ($\mathcal{R}_{inf} = \emptyset$), and returns the repaired solution φ.

3.4. Mutation

Given that an offspring solution inherits exclusively the edges of its parents, it may resemble much the parents even after the feasibility restoring operations. To introduce some diversity into an offspring solution, we modify the solution with a probability p_m with the removal operator presented in Shaw (1998). Basically, this operator deletes some customers from their routes and then greedily reinserts these customers into the solution while respecting the capacity constraint.

Specifically, the mutation removes a number of customers that are similar with respect to a predefined characteristic (e.g., location or demand). In this work, we use the distance between customers to define the similarity. The mutation works in two steps as follows. Firstly, a random customer i in route r_k with its served demand $d_i(r_k)$ is selected to initialize set $\mathcal{S}$. Then, the similarity between customer i and other customers $(\mathcal{N} \setminus \mathcal{S})$ is calculated and sorted in ascending order, where the first customer has the maximum similarity. A customer with its served demand in the route is selected with the roulette-wheel selection and saved in set $\mathcal{S}$ subsequently. For each selected customer i, if it is visited by more than one route, a random route is retained. The first step terminates when l customers are considered ($|\mathcal{S}| = l$) (l is a parameter called the mutation length). More details about this step can be found in Ropke and Pisinger (2006). The second step reinserts greedily the removed customers of set $\mathcal{S}$. For each customer $i \in \mathcal{S}$, a customer $j \in \mathcal{N} \setminus \mathcal{S}$ from its δ-nearest neighborhood is selected, and the customer i is inserted after the customer j with respect to the capacity constraint and the minimum traveling distance. This procedure terminates when all customers in $\mathcal{S}$ are inserted into the solution. The worst-case time complexity of the mutation is $O(l \times \delta)$.

3.5. Local search

Local search is among the core components of the state-of-the-art heuristic algorithms for several related VRPs. Enriched neighborhood operators, exploration strategies, and speed-up techniques have been developed to allow the local search to attain high-quality solutions within a limited time. The local search procedure of SplitMA for the SDVRP adopts nine popular VRP neighborhood operators used in Vidal (2022), including eight inter-route and one intra-route structures. To reinforce its search capacity, our local search additionally employs four tailored SDVRP neighborhood operators proposed in Boudia, Prins, and Reghioui (2007) and Dror and Trudeau (1989, 1990). These 13 operators are explored under the framework of variable neighborhood descent according to the order in which they are presented in the forthcoming subsections.

Before introducing the neighborhood operators, we first present three application rules. The first rule is that once an improvement occurs with an inter-route structure, the procedure checks whether a vehicle visits some customers twice. If so, the duplicated visits with the largest distance reduction are removed. The second rule defines the neighborhood of each customer with the δ-nearest vertices, where δ ($\delta < |\mathcal{N}|$) is the granularity threshold restricting the search to nearby vertices. This rule aims to avoid the examination of non-promising neighboring solutions and speeds up the local search. The last rule is that the first improvement strategy is adopted to explore each neighborhood.

To present the different neighborhood operators, we adopt the following notations. $r(u)$ and $r(v)$ denote the routes which visit vertices u and v, respectively. Let v be a neighbor of u, and x and y the successors of u in $r(u)$ and v in $r(v)$, respectively. (u,x) is the substring from vertex u to x, and (v,y) is the substring from vertex v to y.

3.5.1. VRP neighborhood operators We first summarize the nine commonly used VRP neighborhood operators, named as M1–M9. Detailed presentations of these operators are provided in Vidal (2022). Basically, M1–M3 are based on the insertion operation and M4-M6 use the interchange (or swap) operation. M7 is the classical 2-opt for intra-route move, while M8 and M9 apply 2-opt* (Potvin and Rousseau 1995) for inter-route optimization.

- M1: If u is a customer visit, remove u from route $r(u)$ and place u after v;
- M2: If u and x are customer visits, remove them from route $r(u)$ and place (u,x) after v;
- M3: If u and x are customer visits, remove them from route $r(u)$ and place (x,u) after v;
- M4: Interchange u and v if they are customer visits;
- M5: Interchange (u,x) and v if they are customer visits;
- M6: Interchange (u,x) and (v,y) if they are customer visits;
- M7: This is 2-opt. If $r(u)=r(v)$, replace (u,x) and (v,y) by (u,v) and (x,y);
- M8: This is 2-opt*. If $r(u)\neq r(v)$, replace (u,x) and (v,y) by (u,v) and (x,y);
- M9: This is 2-opt*. If $r(u)\neq r(v)$, replace (u,x) and (v,y) by (u,y) and (v,x).

3.5.2. SDVRP inter-route neighborhood operators We describe now the four inter-route neighborhood operators M10–M13 specifically designed for the SDVRP (Boudia, Prins, and Reghioui 2007, Dror and Trudeau 1989).

- M10: This operator extends M4 by modifying the amounts to be delivered to customers with respect to the capacity constraint. Suppose that customers u and v (customer v is a neighbor of customer u) are visited on two distinct routes, that is $r(u)\neq r(v)$. There are two cases: (i) if $d_u(r(u))>d_v(r(v))$, then customer v with demand $d_v(r(v))$ is inserted before or after customer u in route $r(u)$, and a copy of u with $d_v(r(v))$ is inserted into route $r(v)$ at the position of customer v; (ii) if $d_u(r(u))<d_v(r(v))$, customer u with $d_u(r(u))$ is inserted before or after customer v, while a copy of v with $d_u(r(u))$ is removed from route $r(v)$ and repositioned at the position of customer u in route $r(u)$. Please refer to Boudia, Prins, and Reghioui (2007), Silva, Subramanian, and Ochi (2015) for a detailed description and illustration.

- M11: It extends M5 by adjusting the amounts to be delivered to customers while satisfying the capacity constraint. Suppose that customers u and v come from two different routes. Two cases are considered: (i) if $d_u(r(u)) + d_x(r(u)) > d_v(r(v))$ and $d_u(r(u)) < d_v(r(v))$, then customer u with $d_u(r(u))$ and a copy of x with $d_v(r(v)) - d_u(r(u))$ are interchanged with customer v with $d_v(r(v))$; (ii) if $d_u(r(u)) + d_x(r(u)) < d_v(r(v))$, customers u, x are inserted before or after v in route $r(v)$, and a copy of customer v with $d_u(r(u)) + d_x(r(u))$ is removed from $r(v)$ and replaced at the position of u in route $r(u)$. One notices that if $d_u(r(u)) + d_x(r(u)) = d_v(r(v))$, M11 becomes M5. A detailed description of M11 can be found in Boudia, Prins, and Reghioui (2007), Silva, Subramanian, and Ochi (2015).

- M12 (RouteAddition): This operator was introduced by Dror and Trudeau (1989). Firstly, suppose that a customer u is served by two routes $r(u)$ and $r'(u)$, and the customer u is removed from the routes and inserted in a new empty route. Then, four subtours of routes $r(u)$ and $r'(u)$ split by customer u are considered. The best component of combining these four route segments together with customer u is constructed to minimize the traveling cost, and three new routes are generated. Following Dror and Trudeau (1989), we only consider the customer u involved in two or three routes to limit the computational complexity of exploring this neighborhood. For example, if customer u is visited by two routes, there are 9 components; however, if customer u is visited by three routes, there are 19 components.
- M13 (k-Split): This operator was also introduced by Dror and Trudeau (1989). It splits a customer and inserts the split demands into different routes with respect to the minimum move gain and capacity constraint. A greedy heuristic is adopted to find the best move quickly. For a detailed description, please refer to Silva, Subramanian, and Ochi (2015).

3.5.3. Route elimination For the SDVRP-LF, feasible solutions are limited to K_{min} vehicles. However, this constraint is relaxed during the mutation and local search with different neighborhood operators. In order to obtain feasible solutions after the local search, the k-Split neighborhood operator is employed to eliminate the least loaded route one by one until the number of routes equals K_{min}. For route elimination, we adopt the *EmptyRoutes* procedure presented in Silva, Subramanian, and Ochi (2015).

3.5.4. Maximum splits per customer Intuitively, to minimize the objective function, it is not desirable to split too much a customer's demand. As a result, in SplitMA, for each customer i, a maximum number of splits s_i is determined by $s_i = max\{s_{min}, \lceil \theta \times \frac{d_i}{Q} \rceil\}$, where θ is a control parameter and s_{min} sets the minimum of s_i, which prevents the maximum splits per customer from becoming too small. In SplitMA, we experimentally set $\theta = 50$ and $s_{min} = 5$, and apply the maximum splits strategy in neighborhood operators M10, M11 and M13. The benefits of this strategy are investigated in Section 5.4.

3.6. Population management

Population management is known as an important ingredient of successful memetic algorithms. SplitMA adopts a variable population scheme inspired by that used in Vidal et al. (2012).

The number of individuals in $\mathcal{P}$ varies between p_{min} and p_{max} ($p_{min} < p_{max}$) during the evolution process. Unlike the population management strategy used in Vidal et al. (2012), clone individuals are not allowed. Along with the evolution, the size of $\mathcal{P}$ increases since offspring individuals are progressively added to the population. Once $|\mathcal{P}| > p_{max}$, the surviving selection is triggered to remove p_{max} - p_{min} individuals by considering their contributions to the diversify of the population and traveling cost. Similar to Boudia, Prins, and Reghioui (2007), the normalized Hamming distance h_{AB} between φ_A and φ_B is defined as the ratio between the number of non-common edges and the number of total edges in φ_A and φ_B, $h_{AB} = \frac{|\{\mathcal{E}_{\mathcal{A}} \cup \mathcal{E}_{\mathcal{B}}\} \setminus \{\mathcal{E}_{\mathcal{A}} \cap \mathcal{E}_{\mathcal{B}}\}|}{|\mathcal{E}_{\mathcal{A}} \cup \mathcal{E}_{\mathcal{B}}|}$. Then, the biased fitness of each solution is calculated with respect to its initial fitness and diversity rank in $\mathcal{P}$.

If the best solution found so far φ^* cannot be improved during γ consecutive iterations, the algorithm restarts by generating a totally new population.

4. Computation Results and Comparisons

In this section, we report extensive experiments to evaluate the performance of SplitMA on popular benchmark instances in comparison with the state-of-the-art SDVRP algorithms in the literature.

4.1. Benchmark instances

Four sets of commonly tested instances are used in the experiments.

- Set I. It was proposed by Belenguer, Martinez, and Mota (2000) and consists of 25 instances with 22–101 customers. The set has been widely tested by almost all SDVRP algorithms. This set considers two cost matrices (i.e., unrounded and rounded costs), leading to 50 distinct instances.
- Set II. This set was generated by Campos, Corberán, and Mota (2008) following the procedure provided by Archetti, Speranza, and Hertz (2006). It includes 49 test-instances with up to 199 customers. These instances are divided into 7 groups such that the instances of a group have the same cost matrix and distinct demands. This set was also used to evaluate some algorithms' performances, such as SplitILS (Silva, Subramanian, and Ochi 2015), Aleman and Hill (2010) and Aleman, Zhang, and Hill (2009).
- Set III. The set was presented by Archetti, Speranza, and Savelsbergh (2008) following the same approach of Archetti, Speranza, and Hertz (2006). The set is composed of 6 groups including 42 instances with 50–199 customers, and the instances in each group have the same cost matrix and distinct demands.
- Set IV. This set was provided by Chen, Golden, and Wasil (2007). It includes 21 instances with 8–288 customers. These instances have the particularity that customers are concentrically distributed around the depot.

Table 2 Parameter tuning results.

Parameter	Section	Description	Considered values	Final value
p_{min}	3.1 and 3.6	minimal size of population	$\{10, 15, 20, 25, 30\}$	30
p_{max}	3.1 and 3.6	maximal size of population	$\{45, 50, 55, 60, 65, 70, 75\}$	60
p_m	3.4	mutation probability	$\{0, 0.05, 0.1, 0.15, 0.2, 0.25, 0.3\}$	0.2
l	3.4	length of mutation	$\{0.05, 0.1, 0.15, 0.2, 0.25\}$	0.05
δ	3.5	granularity threshold	$\{10, 15, 20, 25, 30\}$	20
γ	3.6	maximum iterations without improvement	$\{5000, 10000, 15000, 20000, 25000\}$	10000

All these 162 instances are used in our experiments to evaluate the performance of the proposed SplitMA algorithm. The instances and the best solutions obtained by SplitMA are available online at `https://github.com/pengfeihe-angers/SplitMA`.

4.2. Experimental protocol and reference algorithms

Parameter setting. The SplitMA algorithm involves six main parameters: the minimal population size p_{min}, the maximal population size p_{max}, the mutation probability p_m, the mutation length l, the granularity threshold δ and the maximum iterations without improvement γ. To tune these parameter, we applied the automatic parameter tuning package Irace (López-Ibáñez et al. 2016), leading to the setting shown in Table 2. This setting can be considered as the default setting of the SplitMA algorithm and is consistently used for our experiments.

Reference algorithms. Following the review of Section 2, we adopt the following references for the comparative study.

- BKS. This indicates the best known solutions (best upper bounds) that are compiled from all reference heuristic and exact approaches (Munari and Savelsbergh 2022, Ozbaygin, Karasan, and Yaman 2018, Archetti, Bianchessi, and Speranza 2014).
- SplitILS. This multistart iterated local search algorithm was proposed by Silva, Subramanian, and Ochi (2015) for solving the SDVRP-LF and SDVRP-UF. It remains one of the current best SDVRP algorithms. The algorithm was implemented in the C++ language and executed on an Intel Core i7 2.93 GHz with 8.0 GB of RAM memory running Linux. Each instance was executed 20 times with distinct seeds under the single thread. The stopping condition is the maximum iterations given by $\min\{K_{min} \times n, 5000\} \times 10$.
- iVNDiv. The algorithm was proposed by Aleman and Hill (2010) for solving the SDVRP-LF only. The algorithm was implemented in the C# language and executed on a Pentium 4, 2.8 GHz with 512 MB of RAM. The stopping condition is a maximum number of iterations.
- RGTS. This random granular tabu search algorithm was proposed by Berbotto, García, and Nogales (2014) for solving the SDVRP-LF and SDVRP-UF. It was written in C++ and executed on a personal computer with 2.10 GHz and 4 GB RAM. The algorithm stops when the given number of non-improving moves is met.

- SS. This scatter search algorithm was proposed by Campos, Corberán, and Mota (2008) for solving the SDVRP-LF only. It was encoded by C and executed on a Pentium IV, 2.4 GHz, 1 GB RAM. The algorithm stops when the reference set remains unchanged after combining all the solutions or the maximum number of iterations is reached.
- HGA. The hybrid genetic algorithm was presented by Wilck and Cavalier (2012) and tested on some instances of Set I and Set IV. It was implemented in FORTRAN 95 and executed on an Intel Xeon 2.94 GHz with 8 GB RAM.
- TSVBA. The tabu search with vocabulary building approach was proposed by Aleman and Hill (2010) for solving the SDVRP-UF. It was implemented in C# and run on a Pentium 4, 2.8 GHz, 512 MB of RAM. The algorithm stops when a predefined number of iterations without improving is reached.
- FBTS. The forest-based tabu search was proposed by Zhang et al. (2015) for solving the SDVRP-UF. It was written in C++ and executed on an Intel i5-2410 2.3 GHz, 4 GB RAM. The algorithm terminates when the number of non-improvement steps is met.
- MAPM. The memetic algorithm with population management was proposed by Boudia, Prins, and Reghioui (2007) for solving the SDVRP-UF. The algorithm was implemented in Delphi and executed on a 3.0 GHz personal computer. The algorithm stops when a maximum number of iterations is reached.
- ABHC. The attribute based hill climber heuristic was proposed by Derigs, Li, and Vogel (2010) for solving the SDVRP-UF. It was executed on a 3 GHz personal computer with 2 GB RAM.

Among these references, the BKS values can be considered as the most reliable because they are the best results ever reached by an existing SDVRP algorithm in the literature. On the other hand, the results of the cited algorithms enable an assessment of the proposed algorithm compared to the current state-of-the-art methods. We contacted the authors of the reference algorithms, and obtained the source codes of RGTS (Berbotto, García, and Nogales 2014) and FBTS (Zhang et al. 2015). Unfortunately, for RGTS when we ran it with large scale instances such as p03-100D4, the program terminated with unknown errors. For FBTS, when we compiled the C++ code with g++ on our computer, there were several errors. Furthermore, two studies (Chen et al. 2017, Shi et al. 2018) are excluded for our comparative experiments because they report inconsistent results. For several instances, their results are even better than the proven optimal values reported in Archetti, Bianchessi, and Speranza (2014), Munari and Savelsbergh (2022).

Experimental setting and stopping criterion. The SplitMA algorithm was implemented in C++ and compiled using the g++ compiler with the -O3 option[2]. Experiments were executed on a

[2] Upon the publication of the paper, the code of our algorithm will be made available at `https://github.com/pengfeihe-angers/SplitMA`

computer with a Xeon E5-2670 processor of 2.5 GHz and 2 GB RAM running Linux with a single thread. The algorithm was executed 20 times for each instance with distinct random seeds. In order to provide a good compromise between computing time and solution quality, the SplitMA algorithm terminates when it reaches a maximum of 40,000 iterations. Since each application of the gEAX crossover produces β offspring solutions, each iteration means an offspring solution is constructed and improved by the local search subsequently. On our computer, one run of SplitMA under this stopping condition corresponds to a maximum of 0.04 to 4470.13 seconds (only one instance requires this longest time) according to the instance size, which is quite reasonable compared to the time reported by most reference algorithms in the literature.

4.3. Computational results and comparisons

In the tables presented hereafter, column *Instance* indicates the name of instances; *#Instances* is the number of instances; *LB* is the lower bound extracted from state-of-the-art exact algorithms (Belenguer, Martinez, and Mota 2000, Ozbaygin, Karasan, and Yaman 2018, Munari and Savelsbergh 2022, Archetti, Bianchessi, and Speranza 2014); *Best* and *Avg.* are the best and average results obtained by the corresponding algorithm in the column header, respectively; *Gap* is calculated as $Gap = 100 \times (f_{best} - BKS)/BKS$, where f_{best} is the best objective value of SplitMA. Since the SDVRP is a minimization problem, a negative *Gap* (in bold) indicates an improved upper bound. *Time* is the average time in seconds of 20 executions. *TMB* is the average time needed by the algorithm to hit its best solution. Furthermore, the dark gray color indicates that the corresponding algorithm obtains the best result among all compared algorithms on the corresponding instance; the medium gray color displays the second best results, and so on.

We also provide the summarizing information as follows. *Average* is the average value over the instances of a benchmark set. *#Best* is the number of instances for a set where an algorithm gets the best objective value. Finally, to access the statistically significant difference between SplitMA and each reference algorithm, the *p-value* is shown in each table and it is the result of the Wilcoxon signed-rank test with a confidence level of 0.05. If the *p-value* is less than 0.05, the null hypothesis is rejected.

In the following subsections, we present the results obtained by SplitMA on all the benchmark instances and compare them with the reference algorithms.

4.3.1. Comparative results on the SDVRP-LF Table 3 summarizes the results of the SplitMA algorithm for the SDVRP-LF (upper part) compared to the reference algorithms in terms of the best objective values while Tables 7 - 11 show the detailed results on the 162 instances. From these tables, the following observations can be made. First, as shown in Table 3, SplitMA finds 70 new upper bounds out of the 162 instances (43%), matches the BKS values for 75 other

Table 3 Summary of comparative results between SplitMA and reference algorithms in terms of the best objective values.

Pair algorithms	#Instances	*Best*				*Avg.*			
		#Wins	#Ties	#Losses	*p-value*	#Wins	#Ties	#Losses	*p-value*
SDVRP-LF	162	-	-	-	-	-	-	-	-
SplitMA vs. BKS	162	70	75	17	4.28E-09	-	-	-	
SplitMA vs. SplitILS	162	76	74	12	1.11E-12	97	29	36	7.42E-09
SplitMA vs. iVNDiv	99	92	7	0	3.15E-17	-	-	-	-
SplitMA vs. RGTS	88	78	9	1	2.15E-14	79	8	1	2.76E-14
SplitMA vs. SS	49	44	5	0	1.74E-09	-	-	-	-
SplitMA vs. HGA	21	12	8	1	3.09E-03	-	-	-	-
SDVRP-UF	162	-	-	-	-	-	-	-	-
SplitMA vs BKS	162	73	81	8	2.08E-12	-	-	-	-
SplitMA vs. SplitILS	162	82	76	4	4.35E-16	112	33	17	6.24E-18
SplitMA vs. TSVBA	120	105	13	2	8.69E-20	-	-	-	-
SplitMA vs. FBTS	67	67	0	0	1.12E-12	-	-	-	-
SplitMA vs. MAPM	74	62	12	0	1.72E-12	-	-	-	-
SplitMA vs. ABHC	36	34	2	0	1.83E-07	-	-	-	-

instances (46%) and only misses 17 BKS values (10%). This performance can be considered as remarkable given that the BKS values are the best results compiled from all existing algorithms. Furthermore, compared to the most effective heuristic SplitILS, SplitMA obtains 76 and 97 better results in terms of the best and average values, respectively, while the reverse is true for 12 and 36 cases. For the remaining reference algorithms, the dominance of SplitMA is even more evident by achieving the best results for the vast majority of the instances. According to the Wilcoxon signed-rank test, the small *p-values* ($\ll 0.05$) between SplitMA and the competitors indicate that the performance differences are statistically significant.

From the detailed results shown in Tables 7 - 11, we have several observations. First, for each benchmark set, SplitMA competes favorably with the corresponding reference algorithms in terms of the best and average results. Second, in terms of running time, SplitMA spends a little more time to obtain slightly better results compared to SplitILS for Set I with both rounded and unrounded costs. For the three remaining Sets, SplitMA finds better results than SplitILS with less computation time. Some algorithms, such as RGTS, show very short times, but their results are much worse than SplitMA (and SplitILS). It is worth saying that given the reference algorithms were programmed in different languages and performed on different computers under different stopping conditions, the comments on running times are provided for indicative purposes only.

4.3.2. Comparative results on the SDVRP-UF Table 3 summarizes the results of the SplitMA algorithm for the SDVRP-UF (lower part) compared to the reference algorithms in terms of the best objective values while Tables 12 - 16 show the detailed results on the 162 instances. One notices that our algorithm updates 73 BKS values (new upper bounds) and matches 81 other BKS values. Compared to the best reference algorithm SplitILS, our algorithm reports 82 better, 76 equal and 8 worse results, respectively. For the average results, SplitMA obtains 112 better results compared to SplitILS. SplitMA performs much better than the other reference algorithms (weaker

Table 4 Summary of comparative results of SplitMA compared to the results of SplitGiant (using the giant tour crossover) and SplitMA1 (without any crossover).

Pair algorithms	*Best*				*Avg.*			
	#Wins	#Ties	#Losses	*p-value*	#Wins	#Ties	#Losses	*p-value*
SplitMA vs. SplitGiant	46	28	0	3.52E-09	54	13	7	3.52E-12
SplitMA vs. SplitMA1	64	10	0	4.63E-10	68	6	0	7.64E-13

than SplitILS) by obtaining the best results for the vast majority of the instances. The small *p-values* ($\ll 0.05$) from the Wilcoxon signed-rank test indicate that the performance differences between SplitMA and the reference algorithms are statistically significant.

5. Analysis

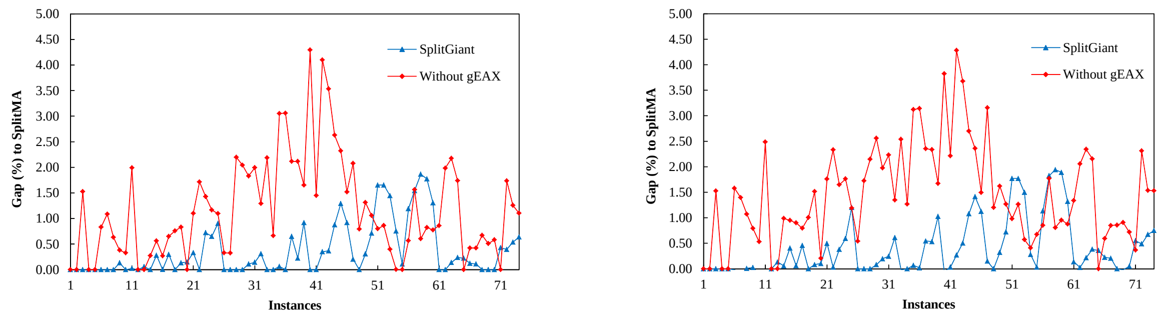
In this section, we conduct additional experiments to assess the contributions of two key components of the SplitMA algorithm, that is gEAX and local search. For this, we focus on the SDVRP-UF and the 74 instances of Sets I and II.

5.1. Significance of the gEAX crossover

To assess the interest of the gEAX crossover, we create two variants of SplitMA as follows. The first variant (SplitGiant) replaces in SplitMA the gEAX crossover by the popular giant tour crossover, which has been very successful for solving routing problems (Potvin 2009, Vidal et al. 2014) as well as the SDVRP (Boudia, Prins, and Reghioui 2007). To implement this variant, we faithfully follow the description of Boudia, Prins, and Reghioui (2007) and adopt the source code of the split procedure from Vidal (2022). The second variant (SplitMA1) just disables the gEAX crossover of SplitMA. To ensure a fair comparison, we use the average running time of SplitMA shown in Tables 12 and 14 as the stopping condition of these two variants to solve each instance. Like SplitMA, each variant is run 20 times independently on each instance. The summarized results are shown in Table 4 while the detailed results are illustrated in Fig. 4 where the results of SplitMA are used as the basis and the results of SplitGiant and SplitMA1 are presented related to this basis.

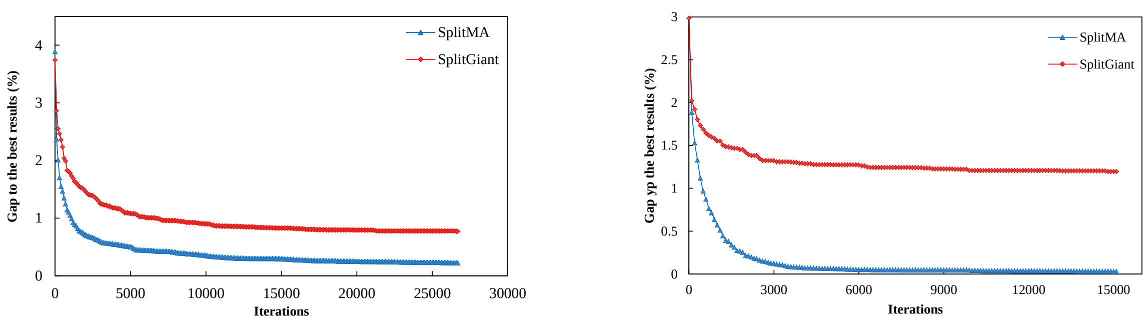
From Table 4 and Fig. 4, one observes that SplitMA outperforms SplitGiant (using the giant tour crossover) in terms of both the best and average values, by reaching 46 better results and 28 equal results out of the 74 instances. Furthermore, when the gEAX crossover is removed from SplitMA, the results become much worse since SplitMA1 (without gEAX) can only matches 10 and 6 best solutions in terms of the best and average results.

To further compare SplitMA and SplitGiant, we investigate their convergence behaviors. Specifically, we obtain the running profiles of these algorithms on two representative instances (S101D3 and S101D5). Each algorithm is run 20 times with the same time budget and the best results were recorded during the process. The results of this experiment are shown in Fig. 5. One observes that SplitMA converges not only faster than SplitGiant, but also converges better.



(a) Best results (b) Average results

Figure 4 Performance gaps of SplitGiant (with the giant tour crossover) and SplitMA1 (with the gEAX crossover disabled) compared to SplitMA on the 74 instances of Sets I and II (a positive gap indicates a deteriorating result) in terms of the best results and average results.



(a) S101D3 (b) S101D5

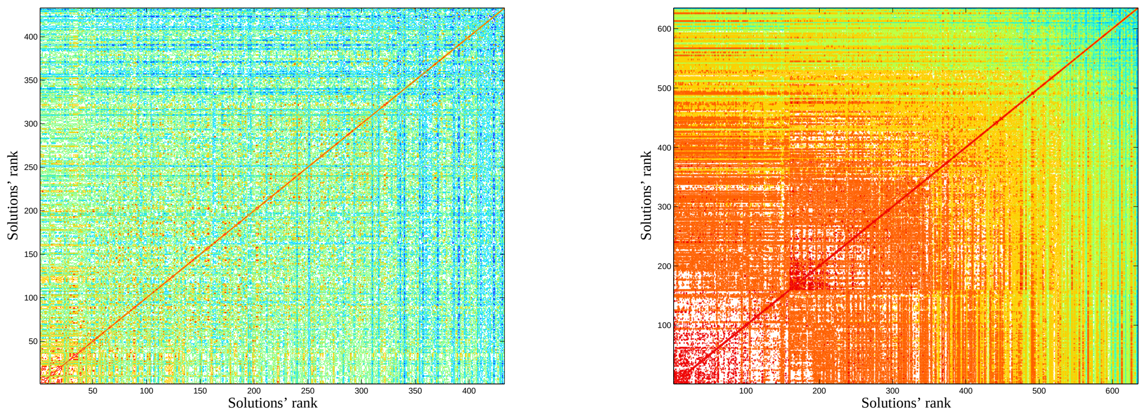
Figure 5 Convergence charts of SplitMA and SplitGiant for solving two representative instances.

We conclude that gEAX is not only a critical search operator contributing greatly to the performance of SplitMA, but also a more suitable crossover compared to the giant tour crossover.

5.2. Rationale behind the crossover

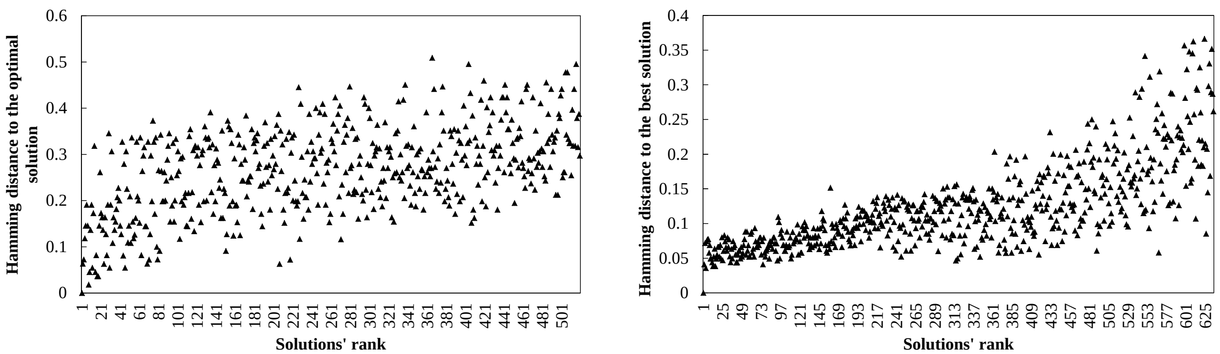
To shed insights on why the gEAX crossover is a suitable operator for the SDVRP, we investigate the relationship between high-quality local optimal solutions in terms of the Hamming distance. Indeed, relevant studies on the TSP (Mühlenbein 1990, Nagata and Kobayashi 2013) and VRP (Arnold and Sörensen 2019, Nagata and Bräysy 2009) have found that high-quality solutions share many common edges, which form the backbone of optimal solutions. EAX thus benefits from this property to construct promising offspring solutions by inheriting the backbone information while introducing a certain degree of diversity (Nagata and Kobayashi 2013). In this section, we show experimentally that the same property remains valid for the SDVRP.

For our experiment, we select two representative instances: eil51 whose optimal value is known and S101D5 whose best result is shown in Table 12. We run SplitMA on these two instances



(a) eil51 (b) S101D5

Figure 6 Hamming distance between each pair of local optimal solutions



(a) eil51 (b) S101D5

Figure 7 Hamming distance between solutions and the best/optimal solution

and record a large number of high-quality solutions whose objective value is within 5% of the best/optimal value. As such, 501 solutions for eil51 and 625 solutions for S101D5 are collected. Then, we calculate the normalized Hamming distance (see the definition in Section 3.6) between each pair of the solutions. Informally, this distance indicates the percentage of the non-common edges between two solutions over the total edges of the two solutions. A value close to 0 means that the two solutions are very similar and vice versa. The results are showed in the two dimensional heat map of Fig. 6. The abscissa and ordinate represent the rank of solutions from smallest (best) to largest (worst) with respect to the objective value. Each colored pixel corresponds to the normalized Hamming distance between two solutions. Hot colors show small Hamming distances, corresponding to pairs of similar (or close) solutions, while cold colors indicate large Hamming distances, thus pairs of distant solutions.

As one observes in Fig. 6, hot colors are around the bottom left corner of both figures, while cold colors are around the upper right corner. This indicates that the higher the quality of the solutions, the more they are similar to each other and vice versa. Furthermore, Fig. 7 illustrates the Hamming distance between high-quality solutions and the best/optimal solution. Once again, one notes that high-quality solutions are closer to the best/optimal solution compared to less good solutions. This is particularly true for S101D5, for which high-quality solutions are very close to the best known solution (with more than 90% common edges).

These findings explain why the gEAX crossover performs well for the SDVRP. Indeed, gEAX transmits the common edges from parents (high-quality solutions) to offspring and conserves the backbone information of high-quality solutions while reassembling non-common edges. It is worth noting that these findings are fully consistent with the cases of the TSP and VRP, which motivated the design of the EAX crossover.

5.3. Benefits of the local search and mutation

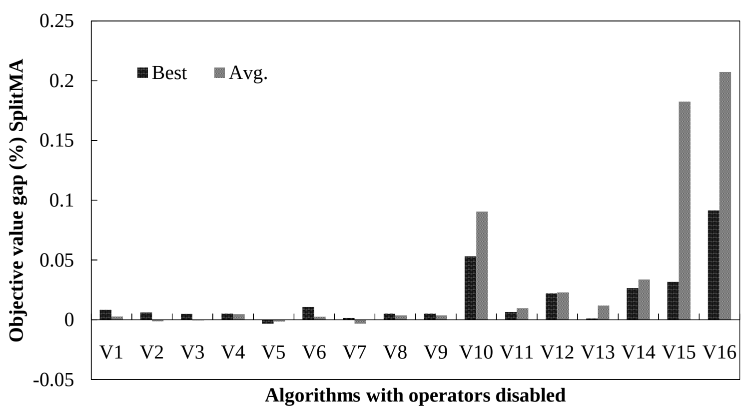


Figure 8 **Illustration of the effects of the neighborhood operators and the mutation operator in terms of the gap with respect to the results of the SplitMA algorithm with all neighborhoods and the mutation operator.**

SplitMA uses thirteen neighborhood operators in its local search procedure and one mutation operator. It is interesting to know how each of these operators contributes to the performance of the algorithm. For this purpose, we create fourteen SplitMA variants (named V1 to V14) by disabling each of these operators. For example, variant V1 is the SplitMA algorithm with the M1 neighborhood being removed from the local search procedure and V14 is SplitMA without the mutation operator. To assess the contributions of the nine VRP neighborhoods (M1-M9) and the four SDVRP neighborhoods (M10-M13), we create two additional SplitMA variants V15 and V16 where M1-M9 and M10-M13 are disabled, respectively. For each of these variants, we compare its best and average results with those obtained by SplitMA. The gaps between these variants and

Table 5 Effect of each neighborhood and the mutation operator.

Pair algorithms	*Best*				*Avg.*			
	#Wins	#Ties	#Losses	*p-value*	#Wins	#Ties	#Losses	*p-value*
SplitMA vs. V1	18	42	14	5.88E-01	31	20	23	9.01E-01
SplitMA vs. V2	13	48	13	6.94E-01	25	21	28	8.49E-01
SplitMA vs. V3	14	44	16	9.92E-01	29	22	23	9.67E-01
SplitMA vs. V4	14	45	15	9.66E-01	32	22	20	3.16E-01
SplitMA vs. V5	13	46	15	4.73E-01	27	22	25	9.75E-01
SplitMA vs. V6	18	44	12	1.53E-01	32	22	20	8.04E-02
SplitMA vs. V7	13	45	16	3.36E-01	26	22	26	2.70E-01
SplitMA vs. V8	14	45	15	9.31E-01	32	21	21	4.08E-01
SplitMA vs. V9	14	45	15	9.31E-01	32	21	21	4.08E-01
SplitMA vs. V10	32	39	3	1.05E-05	49	21	4	1.70E-09
SplitMA vs. V11	16	44	14	5.30E-01	32	21	21	3.40E-02
SplitMA vs. V12	15	42	17	4.54E-01	29	20	25	4.36E-01
SplitMA vs. V13	12	45	17	2.39E-01	39	16	19	7.13E-03
SplitMA vs. V14	26	40	8	3.76E-03	44	21	9	2.70E-07
SplitMA vs. V15	24	41	9	2.17E-02	55	14	5	1.02E-10
SplitMA vs. V16	35	36	3	3.71E-07	58	14	2	2.56E-11

SplitMA are shown in Fig. 8, and a positive gap implies a deteriorating performance with respect to the original SplitMA algorithm.

From the results of Table 5 and Fig. 8, the contribution of each operator can be summarized as follows. First, all operators influence the algorithm with variable impacts. Specifically, M10 can be considered as the most critical neighborhood operator since SplitMA deteriorates significantly its performance if M10 is disabled. Meanwhile, the roles of M2 and M9 are rather marginal. Second, each of the four tailored SDVRP neighborhood operators (M10–M13) plays an important role for the local search. Third, the mutation operator (V14) cannot be ignored since it considerably influences the performance of SplitMA for the best and average results. Finally, both V15 (without the VRP neighborhoods) and V16 (without the SDVRP neighborhoods) perform very badly, confirming that both types of neighborhoods are indispensable for the local search. Meanwhile, we observe that the SDVRP neighborhoods are more critical than the VRP neighborhoods. In summary, all the neighborhoods and mutation contribute to the performance of the SplitMA algorithm, even if their contributions vary significantly.

5.4. Benefits of the maximum splits per customer

We now study how the maximum splits strategy contributes to the performance of SplitMA. For this purpose, we create 10 SplitMA variants with different values of θ, which controls the number of maximum splits per customer (the larger θ, the higher the allowed maximum splits). For example, variant MaxS30 uses $\theta = 30$. For each of these variants, we compare its best and average results with those obtained by SplitMA ($\theta = 50$). This experiment follows the same experimental protocol as before and the results are summarized in Table 6.

From Table 6, we find that SplitMA performs significantly better than MaxS150 and MaxS200 in terms of the best results. Indeed, the value of θ used in variant MaxS200 is four times larger than SplitMA ($\theta = 50$). Furthermore, if the maximum splits strategy is removed from SplitMA,

Table 6 Effect of the maximum splits per customer.

Pair algorithms	*Best*				*Avg.*			
	#Wins	#Ties	#Losses	*p-value*	#Wins	#Ties	#Losses	*p-value*
SplitMA vs. MaxS20	11	46	17	4.25E-01	19	29	26	3.07E-01
SplitMA vs. MaxS30	14	47	13	8.29E-01	20	31	23	4.69E-01
SplitMA vs. MaxS40	12	53	9	2.97E-01	22	32	20	5.28E-01
SplitMA vs. MaxS60	13	56	5	8.54E-02	22	32	20	9.70E-01
SplitMA vs. MaxS70	13	57	4	5.52E-02	25	31	18	8.75E-01
SplitMA vs. MaxS80	12	56	6	1.33E-01	26	30	18	6.12E-01
SplitMA vs. MaxS90	12	46	16	5.24E-01	19	28	27	2.92E-01
SplitMA vs. MaxS100	13	57	4	5.52E-02	27	29	18	2.29E-01
SplitMA vs. MaxS150	13	57	4	4.94E-02	33	26	15	1.05E-01
SplitMA vs. MaxS200	14	55	5	3.29E-02	34	25	15	3.81E-02

the results we obtain are nearly the same as with the variant MaxS200. Thus, the maximum splits strategy positively contributes to the performance of SplitMA. On the other hand, SplitMA is marginally better than the other variants except two cases for these 74 SDVRP-UF instances, which indicates that SplitMA performs similarly well when the maximum splits per customer are limited to a reasonable range.

6. Conclusions

The split delivery vehicle routing problem is a useful model for a broad range of applications in various domains. This work introduced a new memetic algorithm SplitMA that features a general edge assembly crossover for creating promising offspring solutions and an effective local search for solution refinement. It also employs dedicated repairing techniques to ensure the feasibility of offspring solutions, a mutation to diversify new offspring solutions, and an advanced quality-and-distance strategy for maintaining a healthy population.

Extensive experiments on four sets of 324 commonly used instances demonstrate that our algorithm significantly outperforms all existing SDVRP algorithms available in the literature. The algorithms discovers 143 new upper bounds (70 for the SDVRP with a fleet of limited vehicles and 73 cases for the SDVRP with a fleet of unlimited vehicles) and matches the best known results for the majority of the remaining instances. Additional experiments are shown to understand the contributions of main algorithmic components including the gEAX crossover, local search neighborhoods and mutation.

For further work, several directions can be envisaged. First, the local search is the most time-consuming component. To improve the computational efficiency of the local search, it would be interesting to investigate speed-up techniques, such as static move descriptors designed for the CVRP (Accorsi and Vigo 2021). Second, the gEAX crossover is accompanied by the offspring feasibility restoring operations with respect to customer demand and vehicle capacity constraints (Section 3.3), while the local search reestablishes the fleet constraint (for the SDVRP-LF) by route elimination (Section 3.5.3). As such, the algorithm basically explores feasible solutions. Meanwhile, as discussed in (Glover and Hao 2011), for constrained problems, a controlled exploration of infeasible solutions may facilitate discover high-quality feasible solutions that are difficult to reach if the search is limited to the feasible region. This approach has been successfully applied to several routing problems (Gendreau, Hertz, and Laporte 1994, Vidal et al. 2012, Chen, Hao, and Glover 2016, Schneider and Löffler 2019). It is worth studying mixed search approaches allowing the examination of both feasible and infeasible solutions. Finally, this work confirms the interest of the general idea of assembling promising edges from elite parents. This idea together with the design principle of the gEAX can benefit the design of meaningful crossovers for other routing problems such as location routing and arc routing.

Acknowledgments

We are grateful to the reviewers for their insightful and constructive comments, which helped us to significantly improvement the paper. We also would like to thank the following colleagues: Dr. M.M. Silva for sharing the benchmark instances of Silva, Subramanian, and Ochi (2015) and answering our questions, Dr. S. García for sharing the source code of their RGTS algorithm (Berbotto, García, and Nogales 2014), Dr. Z. Zhang for sharing the source code of their FBTS algorithm (Zhang et al. 2015) and Dr. Hill for answering our questions on their work (Aleman and Hill 2010). Support from the China Scholarship Council (CSC, grant No. 201906850087) for the first author is acknowledged.

Appendix A: Mathematical model

In this section a mixed integer programming formulation for the SDVRP based on Archetti, Speranza, and Hertz (2006) is provided.

Given a undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with the vertex set $\mathcal{V} = \{0, 1, \cdots, n\}$ where 0 is the depot and $\mathcal{N} = \{1, \cdots, n\}$ represents n customers, and the edge set $\mathcal{E}$. Let $d_i \in \mathcal{Z}^+$ be the demand of customer $i \in \mathcal{N}$ and $\mathcal{C} = (c_{ij})$ a non-negative cost (distance) matrix associated with $\mathcal{E}$ satisfying the triangle inequality ($c_{ij} + c_{jk} > c_{ik}$ for all $i, j, k \in \mathcal{V}$ and $i \neq j \neq k$). Let Q be the capacity of K identical vehicles. The formulation of the SDVRP is based on two decision variables. Binary variable x_{ij}^k takes the value of 1 if vehicle k traverses edge (i, j), and it takes the value of 0 otherwise. Variable y_{ik} is the quantity of the demand of customer i delivered by the kth vehicle. The mathematical model for the SDVRP-UF is described as follows.

$$Min\ f = \sum_{k=1}^{K} \sum_{i=0}^{n} \sum_{j=0}^{n} c_{ij} x_{ij}^k \tag{1}$$

subject to:

$$\sum_{k=1}^{K} \sum_{i=0}^{n} x_{ij}^k \geq 1 \quad j = 0, \cdots, n \tag{2}$$

$$\sum_{i=0}^{n} x_{ip}^k - \sum_{j=0}^{n} x_{pj}^k = 0 \quad p = 0, \cdots, n;\ k = 1, \cdots, K \tag{3}$$

$$\sum_{i \in \mathcal{S}} \sum_{j \in \mathcal{S}} x_{ij}^k \leq |\mathcal{S}| - 1 \quad k = 1, \cdots, K;\ \mathcal{S} \subseteq \mathcal{N} \tag{4}$$

$$y_{ik} \le d_i \sum_{j=0}^{n} x_{ij}^k \quad k=1,\cdots,K;\ i=1,\cdots,n \tag{5}$$

$$\sum_{k=1}^{K} y_{ik} = d_i \quad i=1,\cdots,n \tag{6}$$

$$\sum_{i=1}^{n} y_{ik} \le Q \quad k=1,\cdots,K \tag{7}$$

$$x_{ij}^k \in \{0,1\} \quad i=0,\cdots,n;\ i=0,\cdots,n;\ k=1,\cdots,K \tag{8}$$

$$y_{ik} \ge 0 \quad i=1,\cdots,n; k=1,\cdots,K \tag{9}$$

Constraint (2) imposes that each vertex has to be visited at least once. Constraint (3) is the flow conservation constraint while constraint 4 is used to eliminate subtours. The first three constraints are classical constraints used in routing problems. Constraints (5)–(7) are related to the allocation of the demands of customers among vehicles. Constraint (5) indicates that customer i can be served by vehicle k only when k visits it. Constraint (6) guarantees that the total demand of each customer must be met. Constraint (7) imposes that the capacity for each vehicle cannot be exceeded.

Finally, since the SDVRP-LF limits the number of vehicles K to the minimum possible $K_{min} = \lceil(\sum_{i=1}^{n} d_i/Q)\rceil$, this extra constraint ($K = K_{min}$) needs to be added into the model.

Appendix B: Computational results

Detailed comparative results between the proposed SplitMA and the reference algorithms on the four sets of benchmark instances are provided in Tables 7–16. Following (Silva, Subramanian, and Ochi 2015), we provide for the instances of Set I the results using both real and rounded costs (the distance matrices of these instances with round costs are obtained from `http://dimacs.rutgers.edu/programs/challenge/vrp/vrpsd/`). For the other benchmark sets, we report real value costs like in the literature.

Table 7 Results for the SDVRP-LF on the instances of Set I.

Instances	LB	BKS	iVNDiv		RGTS			SplitILS			SplitMA				
			Best	Time	*Best*	*Avg.*	Time	*Best*	*Avg.*	Time	*Best*	*Avg.*	*Gap*(%)	Time	TMB
eil22	-	375.28	375.28	4.19	375.28	375.28	0.00	375.28	375.28	0.14	375.28	375.28	0.00	0.13	0.02
eil23	525.65	568.56	569.75	3.42	598.56	568.56	0.00	568.56	568.56	0.12	568.56	568.56	0.00	0.11	0.04
eil30	-	512.72	512.72	14.47	519.70	525.33	210.00	512.72	512.72	0.32	512.72	512.72	0.00	0.22	0.10
eil33	-	837.06	853.10	14.03	843.64	843.64	29.00	837.06	837.06	0.45	837.06	837.06	0.00	46.98	0.41
eil51	518.26	524.61	524.61	54.91	524.93	531.24	11.00	524.61	524.61	1.63	524.61	524.61	0.00	0.50	0.49
eilA76	809.67	823.89	851.24	83.28	860.86	-	37.00	823.89	825.22	27.25	823.89	823.89	0.00	122.50	19.77
eilB76	985.42	1009.04	1059.57	79.00	1023.23	1023.32	23.00	1009.04	1011.19	44.98	1009.04	1011.20	0.00	140.38	52.55
eilC76	723.55	738.67	753.29	148.20	746.34	774.20	23.00	738.67	739.83	15.68	738.67	738.67	0.00	122.60	13.84
eilD76	672.54	687.60	699.35	140.83	702.26	702.26	31.00	687.60	688.37	9.92	686.70	687.24	**-0.13**	117.77	27.89
eilA101	803.62	826.14	852.74	319.33	849.98	851.23	61.00	826.14	826.26	36.59	826.14	826.70	0.00	148.83	29.26
eilB101	1055.40	1076.26	1139.27	185.84	1112.15	1112.29	73.00	1076.26	1078.58	101.26	1076.01	1076.93	**-0.02**	169.50	74.25
S51D1	457.10	459.50	471.92	40.53	459.50	459.93	12.00	459.50	459.50	1.07	459.50	459.50	0.00	0.35	0.33
S51D2	700.40	708.42	731.01	28.34	723.97	723.32	1.00	708.42	709.54	9.98	708.42	708.60	0.00	98.05	23.32
S51D3	938.50	948.01	1001.22	14.70	970.67	970.89	4.00	948.01	949.96	14.15	947.97	947.97	**0.00**	104.49	9.85
S51D4	1549.70	1561.01	1680.66	16.53	1614.10	1614.90	14.00	1561.01	1563.25	59.96	1560.88	1561.21	**-0.01**	246.21	140.36
S51D5	1326.61	1333.67	1389.40	13.94	1381.68	1385.31	3.00	1333.67	1333.85	32.41	1333.67	1334.47	0.00	145.26	40.99
S51D6	2165.64	2169.10	2218.23	16.83	2213.93	2215.41	2.00	2169.10	2174.71	83.79	2169.10	2170.60	0.00	275.95	94.04
S76D1	592.60	598.94	606.47	476.27	629.62	629.62	101.00	598.94	598.98	4.54	598.94	598.94	0.00	101.67	4.38
S76D2	1071.30	1087.99	1143.36	46.94	1113.43	1113.43	10.00	1087.99	1089.69	74.51	1087.40	1088.53	**-0.05**	147.69	66.14
S76D3	1407.54	1427.81	1490.08	53.34	1459.96	1461.20	15.00	1427.81	1429.01	88.72	1425.73	1428.31	**-0.15**	164.97	58.95
S76D4	2059.80	2079.76	2173.61	51.84	2103.05	2103.05	14.00	2079.76	2080.76	173.55	2079.74	2079.84	**0.00**	217.01	104.34
S101D1	716.80	726.59	749.19	2125.58	791.21	791.55	123.00	726.59	728.44	14.16	726.59	726.62	0.00	135.55	28.87
S101D2	1358.90	1383.35	1443.44	217.91	1415.92	1417.40	21.00	1383.35	1386.45	129.94	1377.89	1384.74	**-0.39**	198.08	88.04
S101D3	1853.10	1876.97	1988.78	146.61	1907.92	1907.92	19.00	1876.97	1881.26	277.62	1874.84	1880.13	**-0.11**	245.04	142.05
S101D5	2767.60	2792.01	2984.48	104.05	2896.00	2898.50	14.00	2792.01	2795.73	696.64	2789.81	2798.39	**-0.08**	876.06	646.71
Average	-	1085.32	1130.51	176.04	1113.516	-	-	1085.32	1086.75	75.98	**1084.77**	**1086.03**	-	153.03	66.68
Best#	-	-	0	-	0	0	-	0	4	-	**10**	**15**	-	-	-
p-value	-	5.46E-03	2.67E-05	-	2.70E-05	2.35E-05		5.46E-03	1.01E-02	-	-	-	-	-	-

Table 8 Results for the SDVRP-LF on the instances of Set I with rounded costs.

Instances	LB	BKS	iVNDiv		SplitILS			SplitMA				
			Best	Time	*Best*	*Avg.*	Time	*Best*	*Avg.*	*Gap(%)*	Time	TMB
eil22	375.00	375	375	4.19	375	375	0.13	375	375.00	0.00	33.38	0.04
eil23	569.00	569	570	3.42	569	569	0.09	569	569.05	0.00	31.77	0.16
eil30	510.00	510	510	14.47	510	510	0.3	510	510.00	0.00	44.28	0.12
eil33	834.70	835	851	14.03	835	835	0.39	835	835.00	0.00	43.80	0.13
eil51	521.00	521	521	54.91	521	521.55	1.63	521	521.50	0.00	61.05	1.13
eilA76	807.60	818	847	83.28	818	820.45	25.68	818	824.30	0.00	96.83	49.36
eilB76	981.40	1002	1055	79	1002	1005.8	38.05	1002	1006.90	0.00	106.99	38.11
eilC76	717.80	733	746	148.2	733	733.55	15.17	733	737.40	0.00	88.71	20.60
eilD76	666.10	681	695	140.83	681	683	11.02	682	685.50	0.15	87.72	15.16
eilA101	799.80	814	843	319.33	815	815.85	32.7	817	819.30	0.37	106.37	12.39
eilB101	1040.60	1061	1122	185.84	1061	1065.4	75.43	1061	1075.10	0.00	120.25	61.02
S51D1	454.40	458	466	40.53	458	458	1.21	458	458.00	0.00	55.13	0.28
S51D2	694.20	703	725	28.34	703	704.65	8.32	703	703.00	0.00	81.68	14.26
S51D3	935.17	942	994	14.7	943	944.2	13.58	942	942.00	0.00	95.09	21.03
S51D4	1547.00	1551	1672	16.53	1552	1555.55	47.34	1551	1551.00	0.00	353.89	87.78
S51D5	1325.34	1328	1385	13.94	1328	1329.15	33.46	1328	1328.00	0.00	194.67	36.55
S51D6	2153.00	2153	2211	16.83	2163	2165.7	65.68	2156	2156.11	0.14	280.29	124.06
S76D1	592.00	592	600	476.27	592	592.45	4.75	592	593.25	0.00	76.83	2.87
S76D2	1061.10	1081	1138	46.94	1081	1083.35	59.2	1081	1081.80	0.00	139.35	40.61
S76D3	1395.90	1419	1485	53.34	1419	1422.05	8.07	1420	1420.20	0.07	162.39	57.44
S76D4	2046.10	2071	2160	51.84	2071	2074.3	148.48	2072	2072.95	0.05	265.37	127.91
S101D1	716.00	716	740	2125.58	716	718.4	14.17	716	719.00	0.00	91.57	13.08
S101D2	1337.10	1364	1426	217.91	1364	1370.95	116.33	1360	1369.74	**-0.29**	140.02	75.89
S101D3	1832.20	1859	1974	146.61	1859	1868.75	233.36	1858	1862.95	**-0.05**	199.14	114.72
S101D5	2737.10	2770	2970	104.05	2772	2779.65	579.68	2767	2775.56	**-0.11**	989.28	877.67
Avgerage	-	1077.04	1123.24	176.04	1077.64	1080.07	61.37	**1077.08**	**1079.70**	-	157.83	71.70
Best#	-	-	0	-	**3**	9	-	**3**	**12.00**	-	-	-
p-value	-	8.52E-01	4.00E-05	-	3.42E-01	4.34E-01	-	-	-	-	-	-

Table 9 Results for the SDVRP-LF on the instances of Set II.

Instances	BKS	SS		iVNDiv		SplitILS			SplitMA				
		Best	Time	*Best*	Time	*Best*	*Avg.*	Time	*Best*	*Avg.*	*Gap(%)*	Time	TMB
p01-50	524.61	524.61	49.70	524.61	54.91	524.61	524.61	1.87	524.61	524.61	0.00	73.29	0.62
p01-50D1	460.79	460.79	51.80	471.92	33.70	460.79	460.79	1.16	460.79	460.79	0.00	59.53	0.35
p01-50D2	741.06	741.06	66.40	766.19	19.77	741.06	741.26	9.87	741.06	741.06	0.00	88.95	2.19
p01-50D3	982.77	997.83	87.10	1039.89	18.16	982.77	983.70	18.44	982.77	982.77	0.00	110.82	29.23
p01-50D4	1456.00	1554.38	92.60	1522.43	16.36	1456.00	1456.87	46.74	1456.00	1456.00	0.00	162.84	14.05
p01-50D5	1467.47	1532.19	92.40	1540.39	15.33	1467.47	1467.47	48.93	1467.47	1467.47	0.00	145.46	13.50
p01-50D6	2154.21	2312.48	5.80	2215.34	18.70	2154.21	2154.51	83.85	2154.21	2154.63	0.00	289.48	144.16
p02-75	823.89	829.01	166.50	851.24	83.28	823.89	824.77	30.84	823.89	823.89	0.00	119.36	19.36
p02-75D1	596.25	596.99	144.00	597.46	303.77	596.25	596.25	5.00	596.25	596.25	0.00	97.13	5.57
p02-75D2	1064.49	1071.87	143.80	1099.47	73.05	1064.49	1066.87	53.42	1064.49	1065.41	0.00	145.60	50.85
p02-75D3	1393.11	1463.60	126.80	1478.67	67.80	1393.11	1393.11	101.77	1393.11	1393.18	0.00	159.46	48.46
p02-75D4	2081.38	2182.34	119.90	2200.51	71.11	2081.38	2084.62	219.74	2074.57	2080.37	**-0.33**	268.47	130.09
p02-75D5	2112.19	2228.90	11.10	2238.98	80.30	2112.19	2113.38	267.72	2104.37	2112.46	**-0.37**	272.09	150.94
p02-75D6	3179.20	3387.86	10.50	3304.24	58.05	3179.20	3181.30	441.77	3173.48	3178.53	**-0.18**	446.40	323.80
p03-100	826.14	829.45	276.10	852.74	319.33	826.14	826.39	40.81	826.14	826.70	0.00	147.77	28.78
p03-100D1	726.81	726.81	272.10	745.35	2194.23	726.81	730.01	17.72	726.81	726.81	0.00	140.47	32.32
p03-100D2	1376.09	1397.50	305.10	1425.90	190.53	1376.09	1380.28	182.16	1373.85	1381.60	**-0.16**	199.30	103.36
p03-100D3	1823.17	1908.02	225.20	1956.13	154.47	1823.17	1827.47	326.55	1822.25	1826.76	**-0.05**	219.96	101.25
p03-100D4	2751.13	2894.21	177.90	2865.86	126.52	2751.13	2754.52	629.59	2745.81	2750.08	**-0.19**	335.31	212.80
p03-100D5	2813.82	2986.33	17.00	2941.64	103.94	2813.82	2817.05	737.35	2812.04	2814.61	**-0.06**	344.25	197.63
p03-100D6	4294.12	4576.13	38.30	4429.21	94.98	4294.12	4298.50	731.49	4291.58	4294.52	**-0.06**	577.05	378.35
p04-150	1024.59	1045.22	527.10	1074.11	1361.16	1024.59	1026.60	251.66	1023.23	1024.32	**-0.13**	228.02	127.47
p04-150D1	866.31	871.26	743.30	891.98	3461.44	866.31	866.31	119.63	866.31	866.31	0.00	204.75	17.56
p04-150D2	1861.63	1937.20	326.60	1978.01	878.55	1861.63	1866.48	1055.54	1862.22	1869.21	0.03	294.79	187.48
p04-150D3	2528.51	2649.97	21.30	2671.62	625.83	2528.51	2531.79	1514.55	2525.51	2530.36	**-0.12**	399.21	269.29
p04-150D4	3988.06	4062.88	50.40	4165.18	671.36	3988.06	3997.49	1986.49	3980.33	3988.04	**-0.19**	886.49	686.40
p04-150D5	3986.49	4185.68	23.00	4165.18	675.39	3986.49	3996.85	2076.38	3980.33	3988.04	**-0.15**	889.04	688.41
p04-150D6	6231.01	6479.46	30.50	6482.11	584.84	6231.01	6233.76	1660.06	6225.41	6238.81	**-0.09**	2093.71	2013.53
p05-199	1289.40	1324.73	588.30	1368.67	3284.64	1289.40	1296.37	1594.46	1287.51	1295.99	**-0.15**	316.82	199.75
p05-199D1	1017.30	1023.14	1874.80	1073.55	15505.22	1017.30	1018.40	438.21	1017.28	1018.42	**0.00**	292.70	153.93
p05-199D2	2307.82	2433.17	32.10	2464.65	1457.16	2307.82	2313.37	2440.32	2306.31	2317.16	**-0.07**	425.89	353.13
p05-199D3	3153.01	3291.96	31.20	3411.38	2173.84	3153.01	3163.89	3895.07	3147.31	3160.63	**-0.18**	806.19	774.46
p05-199D4	4844.58	5074.57	50.70	5184.57	3650.59	4844.58	4855.82	3806.84	4840.46	4849.00	**-0.09**	1410.50	1283.91
p05-199D5	5061.25	5265.01	327.30	5363.65	3026.22	5061.25	5070.77	4570.46	5061.31	5069.56	0.00	1811.51	1669.95
p05-199D6	8045.18	8323.72	215.00	8329.55	2124.66	8045.18	8047.68	4718.09	8022.22	8030.53	**-0.29**	3920.84	3863.79
p06-120	1037.88	1042.12	270.30	1201.83	3414.41	1037.88	1043.41	90.06	1037.88	1037.88	0.00	171.02	46.75
p06-120D1	975.96	976.57	370.90	1087.80	3952.67	975.96	976.42	46.16	975.96	975.96	0.00	173.93	26.43
p06-120D2	2703.75	2742.60	380.80	2806.92	558.56	2703.75	2708.51	762.81	2702.50	2705.77	**-0.05**	277.22	165.90
p06-120D3	3907.27	3979.67	329.00	4026.53	358.56	3907.27	3910.03	1543.98	3906.96	3911.81	**0.01**	444.97	347.11
p06-120D4	6201.66	6357.33	20.60	6364.87	458.91	6201.66	6215.87	1975.24	6194.24	6197.24	**-0.12**	936.03	763.29
p06-120D5	6372.58	6481.09	20.50	6545.50	469.17	6372.58	6375.64	2289.79	6328.42	6330.36	**-0.69**	1165.84	931.02
p06-120D6	10001.95	10158.32	20.40	10302.16	636.72	10001.95	10005.18	2209.90	10001.70	10006.23	**0.00**	2832.47	2639.81
p07-100	819.56	819.56	192.40	824.78	126.08	819.56	819.56	27.99	819.56	819.56	0.00	136.50	1.34
p07-100D1	632.63	636.00	166.50	673.54	1207.42	632.63	633.11	14.38	632.63	632.63	0.00	150.36	16.89
p07-100D2	1413.85	1418.81	206.30	1428.27	123.00	1413.85	1413.91	130.70	1413.85	1413.91	0.00	185.34	40.73
p07-100D3	1967.41	1995.34	266.50	2007.11	107.47	1967.41	1967.93	260.65	1967.41	1967.47	0.00	236.64	91.15
p07-100D4	3087.75	3166.31	272.70	3156.31	96.98	3087.75	3088.96	435.09	3088.23	3088.78	0.02	319.06	159.89
p07-100D5	3125.47	3248.76	16.00	3225.63	110.05	3125.47	3126.22	619.62	3125.39	3125.81	**0.00**	369.35	184.86
p07-100D6	4902.81	5065.26	13.80	5028.78	178.19	4902.81	4907.00	823.19	4901.06	4902.75	**-0.04**	647.66	432.65
Average	2591.93	2678.74	201.40	2701.48	1130.15	2591.93	2595.12	925.59	**2588.92**	**2592.27**	-0.08	539.39	410.71
Best#	-	0	-	0	-	3	10	-	**26**	**32**	-	-	-
p-value	2.23E-06	1.74E-09	-	1.18E-09	-	2.23E-06	2.75E-04	-	-	-	-	-	-

Table 10 Results for the SDVRP-LF on the instances of Set III.

Instances	LB	BKS	RGTS			SplitILS			SplitMA				
			Best	*Avg.*	Time	*Best*	*Avg.*	Time	*Best*	*Avg.*	*Gap(%)*	Time	TMB
p01-50	-	524.61	529.23	535.39	13.00	524.61	524.61	1.83	524.61	524.61	0.00	72.67	0.60
p01-50D1	459.50	459.50	466.86	473.32	18.00	459.50	459.50	1.21	459.50	459.50	0.00	59.96	0.29
p01-50D2	756.71	756.71	784.60	789.83	3.00	756.71	760.52	14.55	756.71	758.02	0.00	95.68	44.36
p01-50D3	996.93	1005.75	1025.04	1036.50	1.00	1005.75	1005.93	21.48	1005.75	1005.75	0.00	113.82	19.37
p01-50D4	1485.00	1488.27	1503.33	1538.25	1.00	1488.27	1489.05	52.71	1487.18	1487.62	-0.07	184.82	61.84
p01-50D5	1474.10	1481.71	1503.21	1513.15	8.00	1481.71	1484.62	42.90	1481.71	1481.89	0.00	157.79	70.90
p01-50D6	2149.05	2156.14	2195.67	2202.50	4.00	2156.14	2160.60	84.35	2155.80	2155.80	**-0.02**	245.13	49.59
p02-75	-	823.89	864.64	879.35	10.00	823.89	824.39	30.31	823.89	823.89	0.00	118.45	19.08
p02-75D1	616.58	617.85	629.08	637.00	21.00	617.85	620.19	5.72	617.85	617.85	0.00	103.20	4.01
p02-75D2	1093.56	1110.43	1146.21	1161.75	8.00	1110.43	1112.70	54.11	1109.24	1110.48	**-0.11**	146.02	75.07
p02-75D3	1483.17	1502.05	1550.35	1584.59	12.00	1502.05	1503.42	110.02	1502.05	1503.52	0.00	161.38	59.92
p02-75D4	2270.44	2301.61	2398.40	2412.78	14.00	2301.61	2304.89	283.77	2302.12	2306.06	0.02	321.12	174.07
p02-75D5	2192.25	2219.52	2240.04	2251.50	13.00	2219.52	2222.58	261.25	2219.11	2220.02	**-0.02**	238.88	112.85
p02-75D6	3192.10	3223.06	3259.36	-	6.00	3223.06	3226.79	377.02	3217.51	3221.30	**-0.17**	416.63	282.12
p03-100	-	826.14	845.98	858.20	34.00	826.14	826.45	42.16	826.14	826.70	0.00	147.76	28.85
p03-100D1	753.12	760.00	804.86	834.16	130.00	760.00	760.70	22.00	760.00	760.00	0.00	151.50	50.93
p03-100D2	1435.23	1458.46	1491.82	1497.82	32.00	1458.46	1462.37	200.43	1458.46	1460.90	0.00	202.81	128.34
p03-100D3	1971.43	1997.76	2062.53	2019.50	51.00	1997.76	2001.83	366.31	1996.76	2002.79	**-0.05**	243.98	113.71
p03-100D4	3043.27	3090.65	3171.59	3182.40	54.00	3090.65	3094.91	746.44	3085.69	3088.52	**-0.16**	381.22	260.34
p03-100D5	2945.76	2991.22	3091.25	3111.23	54.00	2991.22	2991.89	756.18	2986.27	2991.17	**-0.17**	364.90	237.24
p03-100D6	4316.42	4387.32	4465.03	4474.00	75.00	4387.32	4389.19	719.27	4378.33	4384.70	**-0.20**	656.07	438.25
p04-150	-	1023.87	1059.71	1069.89	457.00	1023.87	1026.48	243.93	1023.23	1024.32	**-0.06**	228.72	128.20
p04-150D1	896.03	921.47	979.72	998.25	424.00	921.47	923.74	164.58	921.20	921.79	**-0.03**	224.56	116.43
p04-150D2	1986.79	2017.00	2093.21	2102.50	159.00	2017.00	2021.78	1156.99	2016.93	2025.54	**0.00**	313.34	180.33
p04-150D3	2811.64	2849.66	2943.54	2979.02	184.00	2849.66	2856.41	1699.36	2849.59	2853.04	**0.00**	431.38	315.14
p04-150D4	4474.18	4543.18	4652.10	4610.04	255.00	4543.18	4550.63	2467.51	4533.82	4547.78	**-0.21**	1094.75	992.48
p04-150D5	4269.77	4336.80	4460.22	4508.16	252.00	4336.80	4342.45	2366.91	4332.75	4341.43	**-0.09**	918.83	796.82
p04-150D6	6287.09	6396.68	6511.46	6511.46	200.00	6396.68	6402.63	2180.59	6378.28	6393.31	**-0.29**	2175.27	1853.53
p05-199	-	1285.79	1368.81	1401.30	698.00	1285.79	1292.79	1672.72	1287.51	1295.99	0.13	315.59	198.30
p05-199D1	1042.37	1074.18	1158.06	1151.59	989.00	1074.18	1080.65	629.08	1073.57	1081.15	-0.06	294.34	165.21
p05-199D2	2423.64	2481.44	2570.97	2570.97	324.00	2481.44	2487.28	2846.16	2478.37	2488.61	**-0.12**	497.44	402.65
p05-199D3	3420.17	3472.79	3592.77	3578.04	225.00	3472.79	3481.37	3015.92	3469.90	3479.66	**-0.08**	621.69	548.97
p05-199D4	5425.69	5526.28	5798.39	5798.39	220.00	5526.28	5530.56	5799.52	5521.61	5531.00	**-0.08**	3511.05	3433.92
p05-199D5	5306.11	5404.44	5556.01	5556.01	198.00	5404.44	5415.31	5706.50	5398.15	5414.40	-0.12	3460.31	3427.53
p05-199D6	8062.24	8188.47	8319.35	8319.35	241.00	8188.47	8195.06	3528.41	8181.44	8197.54	**-0.09**	4548.54	4470.13
p11-120	-	1037.88	1043.89	1080.30	1231.00	1037.88	1038.68	85.14	1037.88	1037.88	0.00	172.08	46.63
p11-120D1	1023.37	1043.19	1099.30	1120.10	1176.00	1043.19	1043.21	93.35	1042.80	1042.94	**-0.04**	169.38	76.68
p11-120D2	2879.63	2899.91	2939.41	2952.60	99.00	2899.91	2905.28	898.52	2898.25	2902.33	**-0.06**	318.13	226.02
p11-120D3	4162.99	4219.01	4301.53	4308.53	176.00	4219.01	4220.59	2260.39	4215.98	4218.70	**-0.07**	474.16	322.23
p11-120D4	6808.07	6856.11	6967.53	6967.53	301.00	6856.11	6863.96	3363.54	6849.73	6858.08	**-0.09**	1374.52	1220.46
p11-120D5	6584.11	6674.97	6770.14	6770.14	148.00	6674.97	6678.58	2306.51	6639.95	6645.59	**-0.52**	1072.92	800.01
p11-120D6	10111.11	10132.50	10132.50	10133.20	42.00	10215.90	10218.78	2006.24	10192.00	10196.90	0.59	2644.68	2223.42
Average	-	2799.24	2865.42	2865.38	203.83	2801.23	2804.84	1159.19	**2797.56**	**2802.12**	-	701.08	575.64
Best#	-	-	1	1	-	3	10	-	**25**	**29**	-	-	-
p-value	-	6.37E-05	7.86E-08	9.69E-08	-	5.01E-06	3.23E-04	-	-	-	-	-	-

Table 11 Results for the SDVRP-LF on the instances of Set IV.

Instances	LB	BKS	HGA		RGTS			SplitILS			SplitMA				
			Best	Time	*Best*	*Avg.*	Time	*Best*	*Avg.*	Time	*Best*	*Avg.*	*Gap(%)*	Time	TMB
SD1	228.28	228.28	228.28	0.27	228.28	228.28	0.00	228.28	228.28	0.05	228.28	228.28	0.00	10.98	0.03
SD2	708.28	708.28	708.28	1.95	708.28	708.28	0.00	708.28	708.28	0.58	708.28	708.28	0.00	53.74	0.06
SD3	430.40	430.40	430.58	1.94	430.58	430.58	0.00	430.58	430.58	0.59	430.58	430.58	0.04	41.80	0.06
SD4	631.05	631.05	631.05	6.24	633.98	633.98	0.00	631.05	631.05	2.16	631.05	631.05	0.00	84.20	0.22
SD5	1390.57	1390.57	1390.57	14.20	1401.28	1401.72	3.00	1390.57	1390.57	5.90	1390.57	1390.57	0.00	168.71	0.44
SD6	831.24	831.24	833.58	14.97	846.16	861.12	2.00	831.24	831.24	5.62	831.24	831.24	0.00	121.49	0.48
SD7	3640.00	3640.00	3640.00	28.61	3640.00	3640.00	3.00	3640.00	3640.00	13.74	3640.00	3640.00	0.00	237.60	0.26
SD8	5068.28	5068.28	5068.28	48.26	5068.28	5068.28	2.00	5068.28	5068.28	24.07	5068.28	5068.28	0.00	221.65	2.12
SD9	2044.19	2044.20	2054.84	48.91	2044.73	2058.03	1.00	2044.20	2044.43	35.86	2044.20	2044.20	0.00	215.89	2.73
SD10	2684.88	2684.88	2746.54	114.16	2701.55	2709.12	6.00	2684.88	2684.88	81.76	2684.88	2684.88	0.00	298.34	4.87
SD11	13275.00	13280.00	13280.00	231.64	13280.00	13280.00	15.00	13280.00	13280.00	136.43	13280.00	13280.00	0.00	455.49	5.45
SD12	7175.80	7213.61	7279.97	227.11	7213.62	7213.62	19.00	7213.61	7216.34	179.19	7213.61	7213.61	0.00	436.65	22.08
SD13	10053.60	10110.57	10110.57	421.95	10129.52	10129.52	61.00	10110.58	10110.58	168.07	10110.60	10110.60	0.00	526.65	12.42
SD14	10588.20	10715.53	10786.52	718.65	10783.00	10783.00	41.00	10715.53	10722.73	432.26	10715.50	10716.32	**0.00**	666.98	415.02
SD15	14908.50	15093.85	15160.04	1278.35	15151.06	15158.30	110.00	15093.85	15102.85	658.54	15089.60	15091.21	**-0.03**	939.34	559.89
SD16	3379.33	3379.33	3433.83	1225.88	3481.21	3481.21	54.00	3395.11	3395.16	580.27	3381.25	3381.26	0.06	1163.46	500.74
SD17	26317.20	26493.56	26559.92	1722.20	26512.51	26512.51	130.00	26493.56	26499.23	484.43	26493.60	26493.60	0.00	1090.71	247.79
SD18	14029.20	14197.80	14302.22	1735.83	14293.49	14293.49	61.00	14197.80	14202.85	676.77	14194.70	14203.31	**-0.02**	865.62	550.21
SD19	19707.20	19989.95	20152.53	3093.17	20131.94	20154.32	310.00	19989.95	20000.54	1261.95	19991.90	20003.86	0.01	1160.30	813.57
SD20	39252.80	39641.91	39706.51	6208.16	39701.96	39703.32	560.00	39641.91	39648.42	1518.12	39635.50	39638.21	**-0.02**	2500.54	1487.16
SD21	11271.00	11271.00	11461.20	10565.70	11365.16	11369.31	371.00	11344.96	11357.62	4326.99	11281.90	11315.20	0.10	2393.82	2242.46
Average	-	**9002.11**	9045.97	1319.44	9035.55	9038.95	83.29	9006.39	9009.23	504.45	9002.17	9004.98	-	650.19	327.05
Best#	-	-	1	-	0	0	-	2	3	-	**4**	**8**	-	-	-
p-value	-	3.66E-04	3.09E-03	-	5.35E-04	5.35E-04		3.33E-01	1.15E-01	-	-	-	-	-	-

Table 12 Results for the SDVRP-UF on the instances of Set I.

Instances	LB	BKS	TSVBA		FBTS		SplitILS			SplitMA				
			Best	Time	*Best*	Time	*Best*	*Avg.*	Time	*Best*	*Avg.*	*Gap*(%)	Time	TMB
eil22	375.28	375.28	375.28	2.58	375.30	4.00	375.28	375.28	0.15	375.28	375.28	0.00	0.13	0.02
eil23	568.56	568.56	569.75	1.59	568.60	4.00	568.56	568.56	0.13	568.56	568.56	0.00	0.11	0.04
eil30	505.01	505.01	505.01	7.45	519.00	7.00	505.01	505.01	0.24	505.01	505.01	0.00	0.23	0.23
eil33	837.05	837.06	843.64	8.38	837.10	10.00	837.06	837.06	0.51	837.06	837.06	0.00	44.48	0.37
eil51	524.61	524.61	527.67	49.84	528.00	23.00	524.61	524.61	1.79	524.61	524.61	0.00	0.62	0.62
eilA76	809.58	823.89	853.20	145.78	842.70	191.00	823.89	824.92	30.76	823.89	823.89	0.00	104.54	12.03
eilB76	984.13	1009.04	1034.21	91.36	1017.10	289.00	1009.04	1012.07	51.83	1009.04	1011.22	0.00	118.58	64.89
eilC76	722.76	738.67	761.55	151.13	754.30	73.00	738.67	739.89	16.96	738.67	738.67	0.00	102.90	15.59
eilD76	674.17	687.60	695.96	122.52	701.10	57.00	687.60	689.36	11.16	686.70	687.43	**-0.13**	98.84	20.79
eilA101	804.40	826.14	844.21	295.22	838.80	194.00	826.14	826.58	38.90	826.14	826.70	0.00	123.61	37.67
eilB101	1055.59	1076.26	1112.15	173.13	1096.10	280.00	1076.26	1079.15	110.61	1076.26	1077.47	0.00	146.98	72.87
S51D1	459.50	459.50	468.79	13.56	464.80	13.00	459.50	459.50	1.24	459.50	459.50	0.00	0.33	0.31
S51D2	708.41	708.42	718.69	31.66	711.90	121.00	709.29	709.49	11.20	708.42	708.51	0.00	86.84	29.75
S51D3	941.03	947.97	969.78	18.75	952.80	215.00	948.06	950.12	15.74	947.97	947.97	0.00	94.70	7.83
S51D4	1560.87	1560.88	1628.20	19.77	1587.80	134.00	1562.01	1563.29	56.28	1560.88	1561.05	0.00	194.41	113.89
S51D5	1333.66	1333.67	1362.19	15.39	1348.80	127.00	1333.67	1333.67	36.69	1333.67	1333.85	0.00	136.04	34.41
S51D6	2163.22	2169.10	2236.16	14.38	2202.20	81.00	2169.10	2177.78	62.55	2169.10	2170.32	0.00	278.82	130.03
S76D1	598.93	598.94	613.70	252.28	615.90	33.00	598.94	598.94	4.86	598.94	598.94	0.00	85.69	6.07
S76D2	1066.88	1087.40	1128.15	60.44	1103.60	329.00	1087.40	1089.45	69.36	1087.40	1088.99	0.00	131.38	76.99
S76D3	1406.85	1427.86	1472.92	51.13	1449.80	314.00	1427.86	1429.26	96.50	1426.78	1429.05	**-0.07**	148.34	69.86
S76D4	2053.66	2079.76	2180.13	53.56	2108.60	299.00	2079.76	2081.16	188.38	2079.74	2079.77	**0.00**	201.13	83.08
S101D1	716.92	726.59	749.93	860.31	745.70	223.00	726.59	728.45	15.93	726.59	726.59	0.00	109.53	16.20
S101D2	1356.78	1378.43	1409.03	219.52	1394.60	327.00	1378.43	1386.03	151.66	1377.01	1383.72	**-0.10**	172.17	120.33
S101D3	1845.07	1874.81	1947.62	132.19	1913.30	325.00	1874.81	1880.62	317.29	1874.65	1880.39	**-0.01**	222.64	141.64
S101D5	2758.21	2791.22	2910.71	131.16	2858.80	374.00	2791.22	2795.36	572.13	2789.61	2791.59	**-0.06**	318.60	161.87
Average	-	1084.67	1116.75	116.92	1101.47	161.88	1084.75	1086.62	74.51	**1084.46**	**1085.45**	-	116.87	48.70
Best#	-	-	0	-	0	-	0	1	-	**6**	**16**	-	-	-
p-value	-	5.51E-02	2.07E-05	-	1.23E-05	-	4.82E-03	5.46E-04	-	-	-	-	-	-

Table 13 Results for the SDVRP-UF on the instances of Set I with rounded costs.

Instances	LB	BKS	MAPM		TSVBA		SplitILS			SplitMA				
			Best	Time	*Best*	Time	*Best*	*Avg.*	Time	*Best*	*Avg.*	*Gap*(%)	Time	TMB
eil22	375.00	375	375	4.11	375	2.58	375	375.00	0.15	375	375.00	**0.00**	43.72	0.02
eil23	569.00	569	569	5.47	570	1.59	569	569.00	0.11	569	569.00	0.00	41.61	0.24
eil30	503.00	503	503	5.7	503	7.45	503	503.00	0.23	503	503.00	0.00	51.83	0.08
eil33	835.00	835	835	5.19	844	8.38	835	835.00	0.45	835	835.00	0.00	58.09	0.09
eil51	521.00	521	521	7.28	526	49.84	521	521.00	1.75	521	521.00	0.00	79.40	9.86
eilA76	792.71	818	828	35.94	847	145.78	818	821.75	24.63	818	820.60	0.00	129.57	32.56
eilB76	957.60	1002	1019	13.09	1027	91.36	1002	1007.05	37.68	1002	1005.90	0.00	144.39	67.67
eilC76	714.24	733	738	14.75	754	151.13	733	733.75	14.75	733	733.35	0.00	117.30	34.35
eilD76	667.93	682	682	23.12	691	122.52	682	683.05	10.39	680	682.70	**-0.29**	111.44	51.22
eilA101	792.40	814	818	25.25	834	295.22	814	816.20	32.61	814	816.65	0.00	131.91	50.99
eilB101	1017.77	1061	1082	21.81	1104	173.13	1061	1064.00	78.42	1061	1063.60	0.00	156.53	74.92
S51D1	458.00	458	458	8.77	465	13.56	458	458.00	1.17	458	458.00	0.00	73.38	0.37
S51D2	703.00	703	707	7.44	715	31.66	703	704.75	8.12	703	703.15	0.00	109.07	20.49
S51D3	933.07	943	945	7.84	966	18.75	943	944.05	13.06	942	942.00	**-0.11**	131.09	16.09
S51D4	1547.44	1553	1578	11.98	1621	19.77	1553	1556.50	39.25	1551	1551.00	**-0.13**	538.46	71.39
S51D5	1326.73	1328	1351	16.72	1357	15.39	1328	1329.25	32.07	1328	1328.00	0.00	292.13	32.91
S51D6	2153.00	2153	2182	9.92	2228	14.38	2163	2166.15	52.95	2156	2156.00	0.14	515.77	111.52
S76D1	592.00	592	592	15.23	606	252.28	592	592.30	4.75	592	592.20	0.00	97.51	13.56
S76D2	1040.67	1082	1089	30.5	1124	60.44	1082	1083.15	53.6	1080	1081.30	**-0.18**	192.15	65.90
S76D3	1379.57	1420	1427	12.89	1466	51.13	1420	1423.05	67.81	1418	1420.00	**-0.14**	235.31	97.06
S76D4	2034.70	2073	2117	8.76	2170	53.56	2073	2074.95	144.89	2071	2071.90	**-0.10**	389.04	142.81
S101D1	714.87	716	717	49.75	741	860.31	716	718.35	14.76	716	718.05	0.00	116.78	56.03
S101D2	1301.93	1366	1372	31.72	1398	219.52	1366	1371.40	112.47	1360	1365.65	**-0.44**	202.81	103.46
S101D3	1803.51	1864	1891	33.98	1936	132.19	1864	1868.05	236.05	1858	1862.10	**-0.32**	295.32	183.00
S101D5	2709.48	2770	2854	18.66	2897	131.16	2770	2779.10	439.49	2765	2767.80	**-0.18**	595.48	310.54
Average	-	1077.36	1090.00	17.03	1110.60	116.92	1077.76	1079.91	56.86	**1076.36**	**1077.72**	-	194.00	61.88
Best#		-	0	-	0	-	0	1	-	**9**	**18**	-	-	-
p-value	-	2.54E-02	1.96E-04	-	2.69E-05	-	1.95E-03	3.40E-04	-	-	-	-	-	-

Table 14 Results for the SDVRP-UF on the instances of Set II.

Instances	BKS	MAPM		TSVBA		SplitILS			SplitMA				
		Best	Time	*Best*	Time	*Best*	*Avg.*	Time	*Best*	*Avg.*	*Gap(%)*	Time	TMB
p01-50	524.61	524.61	8.53	527.67	49.84	524.61	524.61	1.82	524.61	524.61	0.00	65.81	0.52
p01-50D1	460.79	460.79	12.38	466.74	19.69	460.79	460.79	1.17	460.79	460.79	0.00	52.48	0.31
p01-50D2	741.06	751.41	10.22	753.98	23.17	741.06	741.26	9.72	741.06	741.06	0.00	80.89	1.85
p01-50D3	982.79	988.31	12.49	1023.24	17.72	982.79	983.59	18.04	982.77	983.13	**0.00**	99.79	14.50
p01-50D4	1456.00	1467.06	21.42	1530.81	19.11	1456.00	1457.37	42.86	1456.00	1456.00	0.00	154.69	8.24
p01-50D5	1467.47	1477.01	24.53	1505.38	19.09	1467.47	1467.47	49.42	1467.47	1467.47	0.00	137.18	10.08
p01-50D6	2150.97	2154.35	22.91	2219.32	24.41	2150.97	2152.95	51.94	2150.00	2150.15	**-0.05**	235.87	115.36
p02-75	823.89	823.89	35.72	853.20	145.78	823.89	824.80	30.52	823.89	823.89	0.00	104.77	12.02
p02-75D1	596.25	600.06	18.75	614.09	136.14	596.25	596.25	4.98	596.25	596.25	0.00	83.10	4.51
p02-75D2	1064.49	1074.46	34.14	1085.70	97.17	1064.49	1066.36	53.25	1064.49	1065.37	0.00	125.18	58.26
p02-75D3	1393.11	1413.80	37.38	1458.59	67.66	1393.11	1393.11	97.13	1393.11	1393.20	0.00	139.04	18.85
p02-75D4	2081.38	2102.58	46.11	2164.74	61.81	2081.38	2084.91	191.06	2076.92	2082.27	**-0.21**	251.73	135.55
p02-75D5	2111.83	2132.16	51.78	2182.33	55.17	2111.83	2114.03	212.38	2105.15	2112.19	**-0.31**	222.52	133.15
p02-75D6	3178.47	3200.35	27.48	3278.33	86.27	3178.47	3181.28	412.86	3175.61	3177.67	**-0.09**	420.01	234.40
p03-100	826.14	829.44	34.59	844.21	295.22	826.14	826.39	40.01	826.14	826.70	0.00	125.41	37.52
p03-100D1	726.81	726.81	37.12	741.60	1944.09	726.81	730.80	16.67	726.81	726.81	0.00	115.70	29.32
p03-100D2	1376.22	1392.85	78.06	1416.35	160.95	1376.22	1380.23	187.76	1373.85	1380.81	**-0.17**	171.11	81.05
p03-100D3	1823.58	1845.30	28.39	1886.70	145.05	1823.58	1827.81	313.83	1823.29	1826.80	**-0.02**	197.78	86.17
p03-100D4	2749.53	2780.95	84.38	2874.86	125.28	2749.53	2753.99	647.44	2745.64	2749.10	**-0.14**	314.73	210.32
p03-100D5	2813.52	2858.87	100.16	2929.29	134.84	2813.52	2817.05	737.91	2811.62	2815.30	**-0.07**	327.31	196.40
p03-100D6	4294.12	4312.95	55.75	4435.56	185.55	4294.12	4299.40	737.85	4292.05	4296.29	**-0.05**	573.55	416.70
p04-150	1023.66	1042.37	103.69	1079.55	2217.67	1023.66	1026.89	250.37	1023.23	1024.44	**-0.04**	190.31	85.11
p04-150D1	866.31	875.61	100.27	891.10	2640.95	866.31	866.31	120.92	866.31	866.31	0.00	168.47	19.19
p04-150D2	1861.63	1878.71	147.89	1929.91	755.08	1861.63	1866.95	1041.64	1865.12	1869.29	0.19	255.43	176.79
p04-150D3	2527.96	2561.65	224.89	2647.17	470.34	2527.96	2531.50	1445.25	2523.87	2530.27	**-0.16**	341.16	263.57
p04-150D4	3988.64	4045.87	244.91	4151.90	451.95	3988.64	3996.55	1901.05	3979.53	3985.62	**-0.23**	610.69	517.32
p04-150D5	3985.76	4045.87	244.86	4151.90	449.34	3985.76	3995.25	1836.14	3979.53	3985.62	**-0.15**	607.89	515.09
p04-150D6	6232.37	6267.48	401.62	6416.12	678.94	6232.37	6234.56	1543.69	6223.33	6235.82	**-0.14**	1396.79	1306.96
p05-199	1286.92	1311.59	353.84	1339.49	4514.28	1286.92	1293.71	1298.10	1283.27	1293.19	**-0.28**	258.95	174.95
p05-199D1	1017.28	1018.71	356.22	1069.24	11.215.52	1017.28	1018.59	431.97	1017.28	1018.99	0.00	236.00	76.64
p05-199D2	2305.70	2340.14	347.14	2408.16	1544.36	2305.70	2313.04	2296.08	2301.06	2316.42	**-0.20**	371.31	309.70
p05-199D3	3156.02	3191.25	436.20	3296.69	1216.69	3156.02	3163.26	3316.93	3146.79	3156.56	**-0.29**	532.41	438.04
p05-199D4	4843.83	4941.22	725.69	5066.24	108.63	4843.83	4855.49	3739.98	4836.17	4843.71	**-0.16**	1390.29	1251.93
p05-199D5	5063.89	5155.36	749.94	5281.55	119.04	5063.89	5072.74	4222.85	5054.50	5065.53	**-0.18**	1322.26	1218.41
p05-199D6	8037.88	8081.58	571.70	8333.61	153.12	8037.88	8048.57	4616.79	8022.89	8033.28	**-0.19**	3832.93	3738.14
p06-120	1037.88	1041.20	50.92	1051.24	1944.19	1037.88	1039.13	81.50	1037.88	1037.98	0.00	141.58	43.15
p06-120D1	975.96	976.57	72.98	990.59	2736.34	975.96	976.57	44.82	975.96	975.96	0.00	141.15	29.09
p06-120D2	2707.52	2720.38	144.19	2744.74	463.97	2707.52	2710.15	704.44	2702.26	2705.74	**-0.19**	243.22	193.95
p06-120D3	3907.27	3934.39	163.14	4010.80	340.53	3907.27	3909.28	1487.97	3909.11	3910.94	0.05	411.09	331.21
p06-120D4	6195.37	6318.37	196.14	6308.76	418.98	6195.37	6219.01	1805.91	6194.55	6197.79	**-0.01**	894.16	787.48
p06-120D5	6373.24	6424.71	271.39	6511.08	436.80	6373.24	6376.25	2303.70	6329.30	6331.17	**-0.68**	1103.03	948.83
p06-120D6	10003.99	10063.47	298.08	10186.06	30.32	10003.99	10005.29	2161.59	10003.80	10006.26	**0.00**	2985.43	2721.87
p07-100	819.56	819.56	42.89	819.60	75.33	819.56	819.56	27.67	819.56	819.56	0.00	121.23	1.19
p07-100D1	632.63	649.73	34.97	658.99	461.75	632.63	636.76	12.28	632.63	633.90	0.00	115.49	28.64
p07-100D2	1413.85	1417.28	43.27	1441.48	98.31	1413.85	1413.99	128.52	1413.85	1413.85	0.00	163.04	43.11
p07-100D3	1967.41	1994.59	51.31	2010.00	84.50	1967.41	1968.09	265.71	1967.41	1968.08	0.00	213.56	85.31
p07-100D4	3088.47	3113.72	52.13	3157.48	97.58	3088.47	3089.41	416.34	3087.93	3088.73	**-0.02**	305.53	178.16
p07-100D5	3125.47	3155.69	91.31	3200.62	96.39	3125.47	3125.98	568.97	3125.29	3125.53	**-0.01**	339.51	157.26
p07-100D6	4903.00	4919.48	180.11	4996.88	152.92	4903.00	4906.56	799.40	4901.06	4903.27	**-0.04**	614.40	429.07
Average	2591.68	2616.83	152.73	2672.32	553.59	2591.68	2595.18	872.02	2588.43	2591.03	-0.08	744.91	611.60
Best#	-	0	-	0	-	2	9	-	**28**	**34**	-	-	-
p-value	1.81E-06	1.74E-09	-	1.11E-09	-	1.81E-06	3.63E-05	-	-	-	-	-	-

Table 15 Results for the SDVRP-UF on the instances of Set III.

Instances	LB	BKS	ABHC	FBTS		SplitILS			SplitMA				
				Best	Time	*Best*	*Avg.*	Time	*Best*	*Avg.*	*Gap(%)*	Time	TMB
p01-50	-	524.61	524.61	532.00	18.00	524.61	524.61	1.88	524.61	524.61	0.00	65.49	0.61
p01-50D1	459.50	459.50	-	461.00	31.00	459.50	459.50	1.16	459.50	459.50	0.00	54.10	0.24
p01-50D2	754.45	757.15	776.42	759.80	307.00	757.15	761.12	13.24	756.71	758.08	**-0.06**	88.35	38.35
p01-50D3	999.06	1005.75	1012.56	1026.50	210.00	1005.75	1005.75	20.70	1005.75	1005.75	0.00	105.14	23.10
p01-50D4	1487.16	1487.18	1489.64	1552.10	134.00	1488.58	1488.89	43.04	1487.18	1488.01	0.00	152.96	49.01
p01-50D5	1474.34	1481.71	1488.28	1498.10	151.00	1481.71	1483.36	44.11	1481.71	1482.30	0.00	148.07	82.21
p01-50D6	2149.42	2155.80	2174.54	2191.41	107.60	2156.14	2161.31	78.49	2155.80	2155.80	0.00	229.40	71.22
p02-75	-	823.89	829.89	827.40	320.00	823.89	825.06	29.69	823.89	823.89	0.00	104.76	12.03
p02-75D1	612.45	617.85	-	637.00	44.00	617.85	619.59	5.70	617.85	617.85	0.00	86.26	3.36
p02-75D2	1095.65	1109.62	1123.97	1118.10	325.00	1109.62	1112.11	53.26	1109.24	1110.45	**-0.03**	130.05	45.96
p02-75D3	1482.50	1502.05	1508.73	1525.70	318.00	1502.05	1503.57	108.37	1502.05	1502.91	0.00	145.45	70.57
p02-75D4	2272.05	2298.58	2340.09	2358.80	322.00	2298.58	2301.85	207.22	2296.98	2298.01	**-0.07**	266.02	177.53
p02-75D5	2195.44	2219.97	2243.93	2280.30	406.00	2219.97	2224.06	265.00	2217.63	2219.51	**-0.11**	221.39	134.90
p02-75D6	3192.55	3223.40	3266.78	3259.00	200.00	3223.40	3226.20	378.95	3216.67	3219.88	**-0.21**	410.77	268.82
p03-100	-	826.14	826.14	847.40	188.00	826.14	826.45	40.52	826.14	826.70	0.00	124.75	37.31
p03-100D1	749.42	760.00	-	792.20	87.00	760.00	760.46	21.65	760.00	760.02	0.00	124.23	35.02
p03-100D2	1437.78	1458.46	1478.59	1476.90	326.00	1458.46	1462.68	179.69	1458.46	1461.61	0.00	170.89	69.89
p03-100D3	1971.34	1996.76	2035.91	2023.20	318.00	1996.76	2002.23	362.02	1996.76	2002.33	0.00	223.39	112.41
p03-100D4	3042.93	3085.69	3145.33	3181.30	339.00	3085.69	3093.64	736.52	3085.69	3089.11	0.00	366.24	274.55
p03-100D5	2945.42	2989.30	3014.08	3044.10	325.00	2989.30	2992.75	742.88	2990.34	2991.23	0.03	355.86	162.60
p03-100D6	4334.44	4387.32	4447.47	4441.70	300.00	4387.32	4389.43	675.01	4378.33	4384.69	**-0.20**	659.74	364.76
p04-150	-	1023.87	1028.42	1081.60	426.00	1023.87	1027.28	233.73	1023.23	1024.44	**-0.06**	188.78	84.70
p04-150D1	895.46	921.91	-	953.00	369.00	921.91	923.69	161.90	921.20	922.06	**-0.08**	183.29	75.86
p04-150D2	1986.34	2016.97	2055.18	2060.40	375.00	2016.97	2021.36	1109.88	2016.93	2026.05	**0.00**	268.00	166.31
p04-150D3	2811.98	2849.66	2912.08	2910.80	394.00	2849.66	2857.28	1518.82	2849.66	2853.02	0.00	374.35	261.46
p04-150D4	4474.92	4545.46	4638.74	4681.70	389.00	4545.46	4550.85	2410.48	4537.82	4548.81	**-0.17**	884.75	768.45
p04-150D5	4267.33	4334.71	4435.95	4483.40	372.00	4334.71	4341.15	2357.52	4328.77	4339.85	**-0.14**	722.75	662.86
p04-150D6	6284.76	6395.41	6467.17	6459.80	300.00	6395.41	6402.15	1926.20	6380.51	6391.51	**-0.23**	1463.21	1320.52
p05-199	-	1289.89	1302.89	1342.50	477.00	1289.89	1293.24	1355.25	1287.18	1293.19	**-0.21**	258.40	174.50
p05-199D1	1042.37	1074.18	-	1126.60	449.00	1074.18	1080.64	626.12	1074.06	1081.46	**-0.01**	237.69	154.62
p05-199D2	2423.99	2478.40	2540.06	2525.00	418.00	2478.40	2486.54	2661.25	2476.06	2485.54	**-0.09**	389.77	335.06
p05-199D3	3420.23	3471.41	3581.66	3542.50	429.00	3471.41	3480.76	3014.82	3469.18	3477.51	**-0.06**	559.50	477.57
p05-199D4	5422.95	5521.57	5669.26	5700.70	500.00	5521.57	5529.06	4349.61	5515.50	5519.99	**-0.11**	1666.05	1477.85
p05-199D5	5304.09	5409.76	5541.09	5585.10	438.00	5409.76	5417.75	4524.33	5398.71	5409.49	**-0.20**	1386.87	1318.93
p05-199D6	8062.14	8192.03	8297.71	8255.40	300.00	8192.03	8195.67	3258.29	8176.30	8190.21	**-0.19**	3462.49	3422.14
p11-120	-	1037.88	1042.12	1048.30	177.00	1037.88	1043.38	84.84	1037.88	1037.98	0.00	141.91	43.34
p11-120D1	1023.39	1043.19	-	1119.20	344.00	1043.19	1043.22	109.76	1042.80	1042.88	**-0.04**	139.01	66.96
p11-120D2	2867.79	2898.50	2913.09	2953.10	344.00	2898.50	2907.07	895.11	2898.25	2900.15	**-0.01**	294.04	199.85
p11-120D3	4156.68	4219.01	4270.38	4298.40	345.00	4219.01	4220.79	1957.37	4216.10	4219.56	**-0.07**	447.07	355.21
p11-120D4	6780.19	6854.09	6890.39	7206.20	358.00	6854.09	6865.23	3442.31	6850.78	6856.60	**-0.05**	1412.36	1171.14
p11-120D5	6593.28	6658.52	6671.04	6858.10	354.00	6673.95	6678.11	2354.02	6639.96	6645.18	**-0.28**	1019.33	780.82
p11-120D6	10113.55	10204.81	10233.37	10285.70	300.00	10204.81	10216.80	2279.57	10193.20	10197.45	**-0.11**	2718.38	2305.72
Average	-	2800.28	-	2864.56	300.82	2800.69	2804.92	1062.86	**2797.27**	**2801.08**	-	534.55	420.44
Best#	-	-	0	0	-	1	4	-	**25**	**35**	-	-	-
p-value	-	9.46E-06	-	1.65E-08	-	4.20E-06	8.68E-07	-	-	-	-	-	-

Table 16 Results for the SDVRP-UF on the instances of Set IV.

Instances	LB	BKS	TSVBA		SplitILS			SplitMA				
			Best	Time	*Best*	*Avg.*	Time	*Best*	*Avg.*	*Gap(%)*	Time	TMB
SD1	228.28	228.28	228.28	0.00	228.28	228.28	0.05	228.28	228.28	0.00	7.34	0.41
SD2	708.28	708.28	708.28	0.02	708.28	708.28	0.63	708.28	708.28	0.00	37.30	0.06
SD3	430.58	430.58	430.58	0.03	430.58	430.58	0.62	430.58	430.58	0.00	32.63	0.06
SD4	631.05	631.05	631.05	0.08	631.05	631.05	2.26	631.05	631.05	0.00	73.23	0.22
SD5	1390.57	1390.57	1390.57	0.13	1390.57	1390.57	6.07	1390.57	1390.57	0.00	149.36	0.64
SD6	831.24	831.24	831.24	0.14	831.24	831.24	5.81	831.24	831.24	0.00	118.25	0.54
SD7	3639.97	3640.00	3640.00	0.09	3640.00	3640.00	14.12	3640.00	3640.00	0.00	215.58	0.25
SD8	5068.28	5068.28	5068.28	0.14	5068.28	5068.28	24.93	5068.28	5068.28	0.00	208.20	2.62
SD9	2044.18	2044.20	2071.03	0.36	2044.20	2044.20	38.78	2044.20	2044.20	0.00	204.59	2.75
SD10	2684.86	2684.88	2747.83	0.89	2684.88	2684.88	101.10	2684.88	2684.88	0.00	279.78	5.62
SD11	13280.00	13280.00	13280.00	0.41	13280.00	13280.00	152.42	13280.00	13280.00	0.00	445.85	5.24
SD12	7135.27	7213.61	7213.62	0.84	7213.61	7216.60	210.71	7213.61	7213.61	0.00	431.13	50.94
SD13	9992.74	10110.58	10110.58	1.20	10110.58	10110.58	189.45	10110.60	10110.60	0.00	507.67	10.88
SD14	10502.76	10717.53	10802.87	2.31	10717.53	10723.79	479.85	10715.50	10716.60	**-0.02**	602.92	310.92
SD15	14787.05	15094.48	15153.45	3.20	15094.48	15105.90	731.98	15089.60	15091.78	**-0.03**	863.71	500.17
SD16	3379.33	3379.33	3446.43	7.59	3381.26	3394.48	930.72	3381.25	3381.25	0.06	1100.32	458.89
SD17	26166.80	26493.56	26493.56	7.27	26496.06	26499.32	577.29	26493.60	26493.96	0.00	934.31	350.03
SD18	13892.74	14202.53	14323.04	27.95	14202.53	14205.07	834.60	14194.70	14203.06	**-0.06**	803.83	560.56
SD19	19584.84	19995.69	20157.10	11.95	19995.69	20007.52	1524.67	19991.30	20004.14	**-0.02**	1058.17	856.28
SD20	38901.37	39635.51	39722.86	11.02	39635.51	39647.61	1563.38	39635.50	39637.02	**0.00**	1601.22	1057.46
SD21	11254.83	11271.06	11458.76	111.56	11345.68	11365.37	5034.56	11294.50	11307.54	0.21	2156.59	2087.82
Average	-	9002.44	9043.31	8.91	9006.20	9010.17	591.62	**9002.74**	**9004.62**	-	563.43	298.21
Best#	-	-	0	-	0	1	-	**5**	**9**	-	-	-
p-value	-	8.53E-01	1.62E-02	-	3.47E-02	1.14E-02	-	-	-	-	-	-

References

Accorsi L, Vigo D, 2021 *A fast and scalable heuristic for the solution of large-scale capacitated vehicle routing problems. Transportation Science* 55(4):832–856.

Aleman RE, Hill RR, 2010 *A tabu search with vocabulary building approach for the vehicle routing problem with split demands. International Journal of Metaheuristics* 1(1):55–80.

Aleman RE, Zhang X, Hill RR, 2009 *A ring-based diversification scheme for routing problems. International Journal of Mathematics in Operational Research* 1(1-2):163–190.

Aleman RE, Zhang X, Hill RR, 2010 *An adaptive memory algorithm for the split delivery vehicle routing problem. Journal of Heuristics* 16(3):441–473.

Archetti C, Bianchessi N, Speranza MG, 2014 *Branch-and-cut algorithms for the split delivery vehicle routing problem. European Journal of Operational Research* 238(3):685–698.

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

Archetti C, Speranza MG, 2012 *Vehicle routing problems with split deliveries. International Transactions in Operational Research* 19(1-2):3–22.

Archetti C, Speranza MG, Hertz A, 2006 *A tabu search algorithm for the split delivery vehicle routing problem. Transportation Science* 40(1):64–73.

Archetti C, Speranza MG, Savelsbergh MW, 2008 *An optimization-based heuristic for the split delivery vehicle routing problem. Transportation Science* 42(1):22–31.

Arnold F, Sörensen K, 2019 *What makes a vrp solution good? the generation of problem-specific knowledge for heuristics. Computers & Operations Research* 106:280–288.

Belenguer JM, Martinez M, Mota E, 2000 *A lower bound for the split delivery vehicle routing problem. Operations Research* 48(5):801–810.

Berbotto L, García S, Nogales FJ, 2014 *A randomized granular tabu search heuristic for the split delivery vehicle routing problem. Annals of Operations Research* 222(1):153–173.

Boudia M, Prins C, Reghioui M, 2007 *An effective memetic algorithm with population management for the split delivery vehicle routing problem. International Workshop on Hybrid Metaheuristics*, 16–30 (Springer).

Campos V, Corberán A, Mota E, 2008 *A scatter search algorithm for the split delivery vehicle routing problem. Advances in computational intelligence in transport, logistics, and supply chain management*, 137–152 (Springer).

Chen P, Golden B, Wang X, Wasil E, 2017 *A novel approach to solve the split delivery vehicle routing problem. International Transactions in Operational Research* 24(1-2):27–41.

Chen S, Golden B, Wasil E, 2007 *The split delivery vehicle routing problem: Applications, algorithms, test problems, and computational results. Networks: An International Journal* 49(4):318–329.

Chen Y, Hao JK, Glover F, 2016 *A hybrid metaheuristic approach for the capacitated arc routing problem. European Journal of Operational Research* 253(1):25–39.

Derigs U, Li B, Vogel U, 2010 *Local search-based metaheuristics for the split delivery vehicle routing problem. Journal of the Operational Research Society* 61(9):1356–1364.

Dror M, Trudeau P, 1989 *Savings by split delivery routing. Transportation Science* 23(2):141–145.

Dror M, Trudeau P, 1990 *Split delivery routing. Naval Research Logistics* 37(3):383–402.

Gendreau M, Hertz A, Laporte G, 1994 *A tabu search heuristic for the vehicle routing problem. Management Science* 40(10):1276–1290.

Glover FW, Hao J, 2011 *The case for strategic oscillation. Annals of Operations Research* 183(1):163–173.

Groër C, Golden B, Wasil E, 2010 *A library of local search heuristics for the vehicle routing problem. Mathematical Programming Computation* 2(2):79–101.

Han AFW, Chu YC, 2016 *A multi-start heuristic approach for the split-delivery vehicle routing problem with minimum delivery amounts. Transportation Research Part E: Logistics and Transportation Review* 88:11–31.

Hao J, 2012 *Memetic algorithms in discrete optimization.* Neri F, Cotta C, Moscato P, eds., *Handbook of Memetic Algorithms*, volume 379 of *Studies in Computational Intelligence*, 73–94 (Springer).

Jin M, Liu K, Eksioglu B, 2008 *A column generation approach for the split delivery vehicle routing problem. Operations Research Letters* 36(2):265–270.

López-Ibáñez M, Dubois-Lacoste J, Cáceres LP, Birattari M, Stützle T, 2016 *The irace package: Iterated racing for automatic algorithm configuration. Operations Research Perspectives* 3:43–58.

Mota E, Campos V, Corberán Á, 2007 *A new metaheuristic for the vehicle routing problem with split demands. European Conference on Evolutionary Computation in Combinatorial Optimization*, 121–129 (Springer).

Mühlenbein H, 1990 *Evolution in time and space - the parallel genetic algorithm.* Rawlins GJE, ed., *Proceedings of the First Workshop on Foundations of Genetic Algorithms. Bloomington Campus, Indiana, USA, July 15-18 1990*, 316–337 (Morgan Kaufmann).

Munari P, Savelsbergh M, 2022 *Compact formulations for split delivery routing problems. Transportation Science* .

Nagata Y, 1997 *Edge assembly crossover: A high-power genetic algorithm fot the traveling salesman problem. Proceedings of the 7th International Conference on Genetic Algorithms, 1997.*

Nagata Y, Bräysy O, 2009 *Edge assembly-based memetic algorithm for the capacitated vehicle routing problem. Networks: An International Journal* 54(4):205–215.

Nagata Y, Bräysy O, Dullaert W, 2010 *A penalty-based edge assembly memetic algorithm for the vehicle routing problem with time windows. Computers & Operations Research* 37(4):724–737.

Nagata Y, Kobayashi S, 2013 *A powerful genetic algorithm using edge assembly crossover for the traveling salesman problem. INFORMS Journal on Computing* 25(2):346–363.

Ozbaygin G, Karasan O, Yaman H, 2018 *New exact solution approaches for the split delivery vehicle routing problem. EURO Journal on Computational Optimization* 6(1):85–115.

Potvin JY, 2009 *State-of-the art review-evolutionary algorithms for vehicle routing. INFORMS Journal on Computing* 21(4):518–548.

Potvin JY, Rousseau JM, 1995 *An exchange heuristic for routeing problems with time windows. Journal of the Operational Research Society* 46(12):1433–1446.

Prins C, 2004 *A simple and effective evolutionary algorithm for the vehicle routing problem. Computers & Operations Research* 31(12):1985–2002.

Ropke S, Pisinger D, 2006 *An adaptive large neighborhood search heuristic for the pickup and delivery problem with time windows. Transportation Science* 40(4):455–472.

Schneider M, Löffler M, 2019 *Large composite neighborhoods for the capacitated location-routing problem. Transportation Science* 53(1):301–318.

Shaw P, 1998 *Using constraint programming and local search methods to solve vehicle routing problems. International Conference on Principles and Practice of Constraint Programming*, 417–431 (Springer).

Shi J, Zhang J, Wang K, Fang X, 2018 *Particle swarm optimization for split delivery vehicle routing problem. Asia-Pacific Journal of Operational Research* 35(02):1840006.

Silva MM, Subramanian A, Ochi LS, 2015 *An iterated local search heuristic for the split delivery vehicle routing problem. Computers & Operations Research* 53:234–249.

Song SH, Lee KS, Kim GS, 2002 *A practical approach to solving a newspaper logistics problem using a digital map. Computers & Industrial Engineering* 43(1-2):315–330.

Vidal T, 2022 *Hybrid genetic search for the cvrp: Open-source implementation and swap* neighborhood. Computers & Operations Research* 140:105643.

Vidal T, Crainic TG, Gendreau M, Lahrichi N, Rei W, 2012 *A hybrid genetic algorithm for multidepot and periodic vehicle routing problems. Operations Research* 60(3):611–624.

Vidal T, Crainic TG, Gendreau M, Prins C, 2013 *A hybrid genetic algorithm with adaptive diversity management for a large class of vehicle routing problems with time-windows. Computers & Operations Research* 40(1):475–489.

Vidal T, Crainic TG, Gendreau M, Prins C, 2014 *A unified solution framework for multi-attribute vehicle routing problems. European Journal of Operational Research* 234(3):658–673.

Wilck JI, Cavalier T, 2012 *A genetic algorithm for the split delivery vehicle routing problem. American Journal of Operations Research* 2:207–216.

Zhang Z, He H, Luo Z, Qin H, Guo S, 2015 *An efficient forest-based tabu search algorithm for the split-delivery vehicle routing problem*. Bonet B, Koenig S, eds., *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, January 25-30, 2015, Austin, Texas, USA*, 3432–3438 (AAAI Press).