




Combining hybrid genetic search with ruin-and-recreate for solving the capacitated vehicle routing problem

Martin Simensen^{1,2}  · Geir Hasle² · Magnus Stålhane¹

Received: 7 September 2021 / Revised: 10 May 2022 / Accepted: 24 May 2022 /

Published online: 15 October 2022

© The Author(s) 2022

Abstract

The Capacitated Vehicle Routing Problem (CVRP) has been subject to intense research efforts for more than sixty years. Yet, significant algorithmic improvements are still being made. The most competitive heuristic solution algorithms of today utilize, and often combine, strategies and elements from evolutionary algorithms, local search, and ruin-and-recreate based large neighborhood search. In this paper we propose a new hybrid metaheuristic for the CVRP, where the education phase of the hybrid genetic search (HGS) algorithm proposed by (Vidal Hybrid Genetic Search for the CVRP: Open-Source Implementation and SWAP* Neighborhood 2020) is extended by applying large neighborhood search (LNS). By performing a series of computational experiments, we attempt to answer the following research questions: 1) Is it possible to gain performance by adding LNS as a component in the education phase of HGS? 2) How does the addition of LNS change the relative importance of the local search neighborhoods of HGS? 3) What is the effect of devoting computational efforts to the creation of an elite solution in the initial population of HGS? Through a set of computational experiments we answer these research questions, while at the same time obtaining a good configuration of global parameter settings for the proposed heuristic. Testing the heuristic on benchmark instances from the literature with limited computing time, it outperforms existing algorithms, both in terms of the final gap and the primal integral.

✉ Martin Simensen
martin.simensen@sintef.no

Geir Hasle
geir.hasle@sintef.no

Magnus Stålhane
magnus.staalhane@ntnu.no

¹ Department of Industrial Economics and Technology Management, The Norwegian University of Science and Technology, Trondheim, Norway

² SINTEF Digital, P.O. Box 124, Blindern, 0314 Oslo, Norway

Keywords Vehicle Routing Problems · Hybrid Metaheuristics · Evolutionary Algorithms · Local Search · Large Neighborhood Search

1 Introduction

The *Capacitated Vehicle Routing Problem* (CVRP) was first introduced by Dantzig and Ramser (1959) as the truck dispatching problem. Since then, the CVRP has been one of the most studied problems in operations research and combinatorial optimization. It consists of finding a set of vehicle routes, starting and ending at a depot, to deliver goods to a set of geographically dispersed customers. A feasible solution requires each customer to be visited by exactly one of the identical vehicles, and the total amount of goods delivered by a vehicle cannot exceed its capacity. The objective is to design the routes in such a way that the total distance traveled by the vehicles is minimized. For a more formal description, and mathematical formulations, of the problem we refer to Laporte (2009).

Since its introduction in 1959, numerous extensions and variants of the CVRP, considering a wide range of real-world aspects, have been studied in the literature (Laporte 2009). However, the CVRP still plays a central role due to its relative simplicity, which makes it easier to test out new ideas for both exact and heuristic solution methods, allowing for algorithmic improvements that can later be applied to more complex problem variants. Therefore, the focus of this paper is to improve the heuristic solution methods for the CVRP. For further introduction to richer vehicle routing problems we refer to the taxonomy by Eksioglu et al. (2009), the survey by Vidal et al. (2013), or the book by Toth and Vigo (2014).

The CVRP and all its extensions are known to be NP-hard (Lenstra and Kan 1981), a complexity that is severely limiting the size of the problems that can be solved to proven optimality. The state-of-the-art exact solution methods are the *branch-and-cut-and-price* algorithms by Pecin et al. (2017) and Pessoa et al. (2020), which combines cut and column generation with several additional mechanisms. In general, the exact methods are able to solve most instances of up to 275 customers, although the computational time required is often several days, making the methods impractical in many settings. As a consequence, a considerable part of the literature consists of heuristic solution methods, where the most successful ones are metaheuristics.

The term “metaheuristic” was first coined by Glover (1986), and it has since been used to describe heuristics with additional strategies enabling search past any encountered local optima. A huge number of metaheuristics have been proposed for the CVRP, and they are usually classified into three main types that are neither disjoint nor exhaustive: neighborhood-centered searches, population-based methods, and hybrid metaheuristics (Laporte et al. 2014). Neighborhood-centered metaheuristics utilize a local search procedure, local improvement heuristics, and strategies for escaping local optima, to improve a single incumbent solution. Population-based methods maintain a set of solutions throughout the search process, where in each iteration, information from multiple solutions may be used to create new ones. Hybrid metaheuristics combine multiple approaches. The current state-of-the-art metaheuristics for the CVRP include: *Hybrid Iterated Local Search* (HILS) (Subramanian et al. 2013), *Knowledge*

Guided Local Search (KGLS) (Arnold and Sörensen 2019), *Slack Induction by String Removals* (SISR) (Christiaens and Vanden Berghe 2020), and *Hybrid Genetic Search* (HGS) (Vidal et al. 2012; Vidal 2020).

HILS is a hybrid metaheuristic that combines the neighborhood-centered *Iterated Local Search* (ILS) (Baxter 1981) with an exact algorithm for solving the set partitioning problem. ILS escapes local optima by perturbing the solution to achieve a new starting point for the local improvement heuristic. One iteration of HILS first uses ILS to find a locally optimal solution where some of the solution's routes are added to a route pool, before the exact algorithm finds the optimal partition of routes from the pool. Voudouris and Tsang (2003) proposed the *Guided Local Search* (GLS) metaheuristic, which penalizes "bad" features of the solution, for instance edges, effectively changing the objective function. KGLS builds upon GLS using knowledge obtained through data mining to find the most important features of optimal or near-optimal solutions, and this information determines which edges to penalize. SISR, proposed in Christiaens and Vanden Berghe (2020), is based on a local search heuristic guided by *Simulated Annealing* (Kirkpatrick et al. 1983), in addition to a mechanism for minimizing the vehicle fleet. Instead of using classical neighborhoods for the VRP, it performs large neighborhood search (LNS) defined by *ruin-and-recreate* (R&R) operators. Finally, HGS is a population-based method extending the genetic algorithm proposed by Prins (2004). It maintains a diverse population of individual solutions. In each iteration, these individuals are recombined and the resulting offspring are "educated" with a local search heuristic. Vidal et al. (2012) presented a version of HGS called *Hybrid Genetic Search with Advanced Diversity Control* (HGSADC), and an improved version specialized for the CVRP, namely HGS-CVRP, was proposed in Vidal (2020). This paper focuses on the latter and refer to it as HGS.

Several improvements have been made on metaheuristics for the CVRP during recent years, although most of the improvements have not come from new metaheuristic concepts, but rather from more efficient and focused local improvement heuristics. HGS achieved a significant performance improvement by means of an additional neighborhood in the local search, as reported by Vidal (2020). Similarly, the special R&R operators proposed in Christiaens and Vanden Berghe (2020) arguably constitute the most important factor for the success of the SISR metaheuristic. However, introducing more neighborhoods to a local search does not guarantee improvement, as it is only beneficial if the neighborhoods are complementary. The intersection between neighborhoods should be small or empty so that each neighborhood tends to find different solutions. Additionally, the size of the neighborhoods must be relatively small such that the computational complexity of exploring them is manageable. These challenges may be summarized in the fundamental balance between intensification and diversification, which is vital for a metaheuristic to be successful (Glover 1989).

The overall goal of this paper is to investigate whether combining HGS and LNS is a way to create a competitive solution method for the CVRP. To this end, we formulate the following research questions:

1. Is it possible to gain performance by adding LNS as a component in the education phase of HGS?

2. How does the addition of LNS change the relative importance of the local search neighborhoods of HGS?
3. What is the effect of devoting computational efforts to the creation of an elite solution in the initial population of HGS?

To answer these questions, we design a metaheuristic called *Hybrid Genetic Search with Ruin-and-Recreate* (HGSRR) and a sequence of computational experiments. HGSRR combines elements from the latest version of HGS (Vidal 2020) and the recently proposed SISR metaheuristic by Christiaens and Vanden Berghe (2020) that is based on LNS.

The remainder of this paper is structured as follows. Section 2 describes the proposed hybrid metaheuristic and its components. The sequence of experiments and the final benchmark test comparing the hybrid metaheuristic against the state-of-the-art are presented in Section 3. Finally, in Section 4, concluding remarks are presented in addition to suggestions for future research.

2 Hybrid genetic search with ruin-and-recreate

In short, HGSRR is a genetic algorithm that iterates a cycle consisting of seven components divided into three main phases: Recombination, Education, and Population Management. Six of the components are the same as in the original HGS, while the seventh is responsible for search in the R&R-based neighborhood from SISR. Fig. 1 shows a flowchart of the hybrid method.

The Recombination phase is made up of two components that are responsible for the recombination of individuals. Component (1) selects two parent solutions from the population, whereas component (2) performs crossover of the selected parents in order to recombine them into an offspring solution.

The Education phase is responsible for improving the offspring solution. Because the solutions in the recombination phase are represented as giant tours, a split algorithm is applied to the offspring solution in component (3). A giant tour contains all customers on a single tour without considering the capacity constraint, and to restore feasibility, the split algorithm is used to divide the giant tour into feasible routes in an optimal way. Component (4) then runs a local search on the solution until a local optimum is found, and component (5) continues to search in the R&R-based neighborhood, attempting to further improve the solution. It should be noted that even though component (3) creates feasible vehicle routes with respect to capacity, exceeding the capacity during the Education phase is permitted in exchange of a penalty, potentially leading to infeasible solutions. The penalty is dynamically adjusted based on a target fraction of ξ^{REF} feasible individuals in the population. In addition, if the solution remains infeasible after the Education phase, there is a 50% probability for the phase to restart with a ten times higher penalty coefficient.

Once the offspring solution is educated, the final Population Management phase starts. It includes two components, where component (6) first evaluates the educated offspring and calculates its fitness relative to the population, before component (7) determines if the offspring is among the individuals transferred to the next generation.

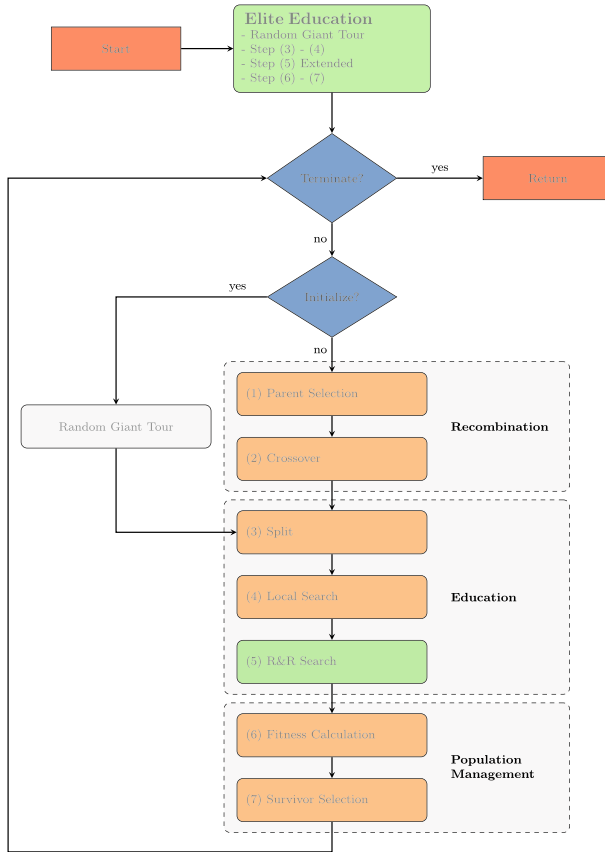


Fig. 1 Flowchart of the complete HGSRR metaheuristic. The genetic cycle consists of seven components and three phases

The three phases are then repeated and the genetic cycle continues until either the predetermined time limit T^{MAX} is reached, or N^{IT} iterations are performed without improvement. In the former case, the algorithm is terminated, while in the latter case, the algorithm is restarted.

To initialize the population, the method starts by generating μ^I individuals, where each individual is a random giant tour before the tour is split and the individual is educated, as in the genetic cycle. However, in contrast to the original HGS, a single individual goes through elite education that includes an intensified version of the R&R search in component (5). The purpose of the elite education is to quickly find a high quality solution, thus improving the convergence of the hybrid method. Unlike the local search in component (4), the search based on the R&R operators of the SISR metaheuristic may accept worsening moves, allowing it to escape local optima, and thus it can be used to continue improvement of a given solution.

The HGSRR with its elite education can be seen as a combination between the population-based HGS and the neighborhood-centered SISR. If no time is spent on

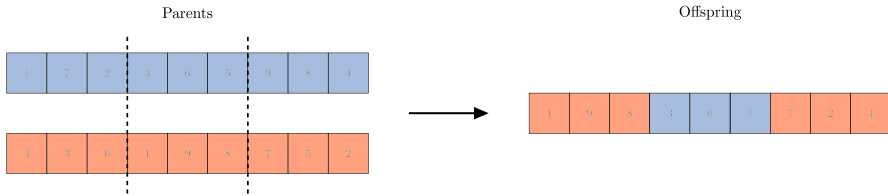


Fig. 2 Illustration of the ordered crossover operator. Two parents, represented on the left, are combined to form the offspring solution to the right. The dashed lines is the crossover region

elite education, the HGSRR is the same as an extended version of HGS, and otherwise, if the elite education takes up all the time, the HGSRR is similar to the SISR metaheuristic without the fleet minimization mechanism. The balance between the two is explored in this paper, and results from the experiments are presented in Section 3. A short description of each component used in the HGSRR is given in the following subsections.

2.1 Parent selection

A tournament-based selection scheme is used to select a parent, where $k \geq 2$ individuals are randomly selected from the population. Out of these k individuals, the one with the best fitness is selected as the tournament winner. In HGSRR, $k = 2$ is used, as higher values of k increase the likelihood of the best solutions being selected, implying a higher selection pressure, leading to less diversity. Since we use binary crossover, two parent solutions are required, and therefore, the tournament selection is performed twice.

2.2 Crossover

An ordered crossover operator (Oliver et al. 1987), where the offspring solution inherits parts of the order from each of the two parent solutions, is used to recombine the two parents using their giant tour representation. Fig. 2 illustrates the ordered crossover operator. A random crossover region is selected, in which the customers within the region are copied from the giant tour of the first parent. Starting outside the crossover region, the remaining customers are copied in order from the giant tour of the second parent, possibly wrapping around.

2.3 Split

Because individuals are represented as a giant tour in the crossover, a dynamic programming algorithm called *Split*, is used to optimally divide the giant tour into segments defining feasible vehicle routes. This approach was first presented and used by Prins (2004).

A giant tour can be represented as a directed acyclic graph (DAG), and the process of splitting the giant tour can be seen as finding the shortest path in the DAG, illustrated

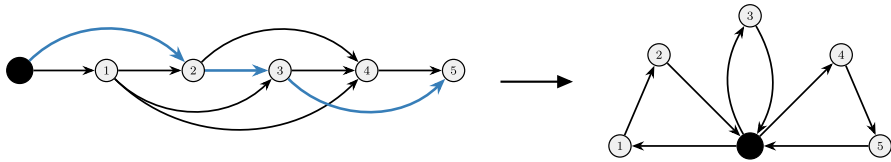


Fig. 3 Illustration of the split algorithm. On the left is the DAG created from the giant tour, where the blue arcs represent the shortest path. The corresponding vehicle routing solution is illustrated to the right

in Fig. 3. In the example, the giant tour consists of five customers in order, where the arcs represent feasible routes and the arc cost corresponds to the length of the route. We see that there is an arc from the depot to the second node, but not to the third. This implies that a route with the first two customers is feasible, and the absence of an arc to the third customer means that a route with the first three customers would violate the capacity constraint. The optimal split is given by the set of arcs making up the shortest path in the graph, and it can be found in $O(n^2)$ using Bellman's algorithm, as shown by Prins (2004). However, like in the HGS by Vidal (2020), the specific implementation relies on a linear-time split algorithm exploiting dominance rules, as described in Vidal (2016).

2.4 Local search

A local search is performed to improve the offspring solution, in which ten complementary neighborhoods are explored in order using a first-improvement strategy. I.e., the local search cycles through the different types of neighborhoods in order and immediately applies the encountered moves if they improve the solution. These neighborhoods are the same as those used in the latest version of HGS by Vidal (2020). Nine of the neighborhoods were first presented in Prins (2004) and they are denoted (M1 - M9). The neighborhoods include *relocation* of one or two vertices, *swap* of one or two vertices, in addition to the *intra-route 2-opt*, and two versions of the *inter-route 2-opt** neighborhoods. These neighborhoods are defined by the moves for all pairs of distinct vertices, and consequently, the neighborhoods can generally be explored in $O(n^2)$. However, a move is only considered if it involves a pair of vertices that are geographically close, and the granularity parameter Γ is used to restrict the neighborhoods, allowing for an exhaustive exploration in $O(\Gamma n)$.

The tenth and final neighborhood is defined by the SWAP* move proposed by Vidal (2020). Following the same notation as Prins (2004) and Vidal et al. (2012), this is denoted M10. It is defined for pairs of routes with overlapping circle sectors. For a pair of routes, one customer from each route is swapped, however, instead of inserting the customer in place of the other, the customers are optimally relocated into the opposite route. Despite generally containing $O(n^4)$ moves, the neighborhood can be explored in sub-quadratic time. For further details on the last neighborhood, the reader is referred to Vidal (2020).

2.5 R&R search

The R&R search distinguishes HGSRR from the original HGS by Vidal (2020). Its purpose is to further improve the offspring solutions after the local search. It searches in the large neighborhood defined by the new R&R operators proposed in Christiaens and Vanden Berghe (2020). A single R&R move from a solution s results in the solution s^* , and is performed by first removing a set of customers, before the customers are inserted back into the solution.

The removal operator is called *Adjacent String Removal*, and it attempts to induce two important types of slack into the solution. A sufficient number of customers must be removed from each route to induce *capacity slack* in the route, as other customers that are more suitable to be served by the route cannot be feasibly added unless the needed capacity is freed up. The second type of slack is the *spatial slack*, given by the reachable area of the vehicle operating the route. Even though the distance driven by a vehicle is unlimited in the CVRP, it can be seen as a resource like the capacity, but instead of being a constraint, it is minimized by the objective function. Removing customers from a route induces spatial slack by increasing the reachable area of the route such that other customers can be visited without exceeding the original distance. To induce both types of slack, the operator starts with a randomly chosen customer from which it removes strings of adjacent customers that are geographically close across multiple routes. The selection of the adjacent strings are controlled by two parameters where \bar{c} is the average number of customers to remove and L^{MAX} is the maximum length for a single string of customers.

The recreate operator is based on the greedy insertion approach and is called *Greedy Insertion with Blinks*. It inserts the removed customers sequentially, and the order of the sequence is determined by sorting the removed customers in one of four ways: Random, Demand, Close, and Far. The first is random order, while the second starts by inserting the removed customer with the largest demand. The last two ways sort the customers based on their distance to the depot, starting with the customer either closest or farthest away. Instead of always inserting the customer in the position with lowest additional cost, the operator searches through all feasible insertion points but may “blink”, i.e., disregard a given point, with a probability of β . The effect is that an insertion point with rank r according to cost has probability $p(r) = (1 - \beta)\beta^{(r-1)}$ of being chosen. The blinking may come across as counter-intuitive, as not inserting a customer into the best position may result in a worse solution. However, the blinking introduces some randomness to the operator, reducing the greediness compared to only choosing the best position, and Christiaens and Vanden Berghe (2020) experienced that the probabilistic blink element improved the overall performance of the SISR metaheuristic. For further details on the two operators, we refer to the original paper by Christiaens and Vanden Berghe (2020).

Like in the SISR metaheuristic, a simulated annealing algorithm with an exponential cooling schedule is used to control the search such that the probability for accepting a move to a new solution s^* is dependent on the current temperature \mathcal{T} and the difference in the objective values $\Delta z = z(s) - z(s^*)$. Let \mathcal{T} be the temperature variable, \mathcal{T}_0 the initial temperature, and \mathcal{T}_f the final temperature. The number of moves N^R in the

R&R search is a linear function of the instance size n determined by the parameter γ : $N^R = \lfloor \gamma n \rfloor$. The cooling factor c follows from N^R , \mathcal{T}_0 , and \mathcal{T}_f : $c = (\mathcal{T}_f / \mathcal{T}_0)^{1/N^R}$.

A pseudocode of the R&R component is presented in Algorithm 1. First, the input solution is saved as the best solution found during the R&R search, s^B (line 1). The temperature \mathcal{T} is set to the starting temperature \mathcal{T}_0 (line 2). For every iteration in the algorithm, solution s^* is determined by applying the R&R operators from SISR (lines 3 and 4). If the move is accepted in line (5), the current solution s is updated in line (6). Additionally, if the new solution is the best found during the search, s^B is updated (lines 7 and 8). The last step in one iteration of the search is to reduce the temperature by multiplying it with the cooling factor in line (9). Finally, in line (10), the best solution found during the R&R search is returned.

Algorithm 1: R&R Search

```

Input:  $s \leftarrow$  Offspring solution after local search
1  $s^B \leftarrow s$ 
2  $\mathcal{T} \leftarrow \mathcal{T}_0$ 
3 for  $N^R$  iterations do
4    $s^* \leftarrow$  Ruin-and-Recreate( $s$ )
5   if  $z(s^*) < z(s) - \mathcal{T} \ln(U(0, 1))$  then
6      $s \leftarrow s^*$ 
7     if  $z(s^*) < z(s^B)$  then
8        $s^B \leftarrow s^*$ 
9    $\mathcal{T} \leftarrow c\mathcal{T}$ 
10 return  $s^B$ 

```

2.6 Fitness calculation

The individual’s fitness is used to measure its quality relative to the other individuals in the population. To balance the diversification and intensification of the search, the fitness combines two measures: the objective value of the solution and the individual’s contribution to the population diversity. The latter is calculated as the average normalized Hamming distance to the other individuals in the population. Additionally, since the solutions may become infeasible after the education phase due to the relaxation of the capacity constraint, two subpopulations are maintained, i.e., one for feasible and one for infeasible solutions. Each individual $p \in \mathcal{P}$ in the subpopulation is given ranks r_p^{OBJ} and r_p^{DIV} , corresponding to the two measures, respectively. A slightly smaller weight is given to the diversity contribution to ensure that the N^E individuals in the subpopulation with the best objective values always survive. The formula for the fitness is shown Equation (1).

$$f(p) = r_p^{OBJ} + \left(1 - \frac{N^E}{|\mathcal{P}|}\right) r_p^{DIV} \tag{1}$$

2.7 Survivor selection

A generational model is used to manage the subpopulations, in which μ is defined as the minimum population size, and λ is the generation size. Therefore, both subpopulations manage between μ and $\mu + \lambda$ individuals each. After an offspring is educated, its fitness is calculated and it is inserted into the respective subpopulation based on feasibility. All the individuals in the subpopulation are kept unless the size of the subpopulation exceeds the upper limit of $\mu + \lambda$ individuals. Once the upper limit is reached, individuals are removed from the subpopulation based on two criteria. First, any clones are removed, and second, if all individuals are unique, the individual with the worst fitness is chosen for removal. Since the diversity contribution of an individual is only defined relative to the subpopulation, the fitness of all the individuals in the subpopulation must be recalculated every time the subpopulation changes. The process of removing individuals from the subpopulation and updating the survivors' fitness is repeated until the size of the subpopulation is back at μ .

3 Computational experiments

To answer the research questions posed in the introduction and obtain well performing parameter values for the HGSRR metaheuristic, we design a sequence of six experiments. The *R&R Search Experiment* investigates the effect of adding LNS to the education phase of HGS, and determines what parameter values yield good performance. In the *Neighborhood Selection Experiment*, fifteen configurations of local search and LNS neighborhoods are tested to determine whether a reduced set of operators may be used without substantial loss of performance. The *Elite Education Experiment* determines the effect of devoting computational efforts to the creation of an elite solution in the initial population, and finds parameter values that aim to strike a good balance between the efforts devoted to elite education and the remaining search components. The *Diversity Control Experiment* investigates whether the introduction of an elite solution in the initial population necessitates stronger diversity control. The HGSRR metaheuristic configured through the above four experiments is compared to high performing competitors in a *Benchmark Test*. Finally, in the *Best Known Solutions Experiment* HGSRR is run for 16 hours with the aim of finding new best known solutions.

3.1 Experimental setup

The computational experiments are conducted on the X dataset by Uchoa et al. (2017) with 100 instances, where the number of customers vary between 100 and 1000. They cover a wide range of characteristics such as route length, customer and depot positioning, and demand distribution. The web page CVRPLIB (2021) contains instance definitions and best known solutions for the X dataset and other standard CVRP benchmarks. The first four experiments are run on a subset including 15 of the 100 benchmark instances. The selected instances are chosen such that they cover a wide

range of different characteristics. The Benchmark Test is run on all 100 X-instances, whereas the Best Known Solutions Experiment is run on all X instances for which a proven optimal solution has not been found, plus a benchmark of ten very large CVRP instances due to Arnold et al. (2019).

HGSRR was implemented using version 1.49.0 of the Rust programming language, and it was compiled with the Rust compiler. The implementation uses the same parameter values as in the original papers by Vidal (2020) and Christiaens and Vanden Berghe (2020), unless explicitly stated otherwise. A complete list of the parameters and their corresponding values is found in Appendix A.

The experimental setup is close to a replication of the experimental setup in Vidal (2020), such that the HGSRR results can be compared directly against the results presented there. However, to account for differences in computing power, the time limit for each problem instance is adjusted relative to the processors' single-threaded ratings. The original time limit T^{MAX} is $\frac{240}{100}n$ seconds, corresponding to four minutes of runtime per 100 customers. All the tests are single-threaded and they are performed on a 2.3GHz Intel E5-2670v3 processor, which according to PassMark Software (2021) is 23% slower than the processor used in the setup of Vidal (2020). Consequently, the time limit is extended by 23% in our setup.

For each run, the best solution value, $z(t_i)$, is recorded after $t_i \in T$ percentages of the time limit.

$$T = \{1\%, 2\%, 5\%, 10\%, 15\%, 20\%, 30\%, 50\%, 75\%, 100\%\}$$

The results of the experiments and the benchmark test conducted in this paper are then presented as convergence profiles. In the first four experiments, one convergence profile corresponds to a specific configuration of HGSRR¹, while in the Benchmark Test presented in Section 3.6, the convergence profiles correspond to different methods. Furthermore, all the results are reported as the average of ten independent runs to increase statistical significance.

Since there is usually a trade-off between the solution quality and the CPU time, one method or configuration can, in general, only be considered better than another if it finds better solutions in a shorter amount of time. Therefore, we define two metrics from the convergence profiles, and these are used to determine which method or configuration is preferable. The first metric is called the *final gap*, denoted g^F . It is defined as the gap between the solution after 100% of the time limit, denoted z^F , and the best known solution (BKS) for the instance. The gap to the BKS after a given percentage of the time limit is defined in Equation (2). To get an accurate comparison between the HGSRR and the other metaheuristics in the benchmark test, the BKSs used are the same as the ones presented in Vidal (2020), even though better solutions have later been found for some of the instances.

$$g(t_i) = \frac{z(t_i) - z^{BKS}}{z^{BKS}} \quad (2)$$

¹ Below we will refer to several HGSRR configurations as HGS extensions, when not all HGSRR components are included.

The final gap is dependent on the predefined time limit and only represents the performance at a single point during the runtime, therefore, we also report the *average gap*, denoted g^{AVG} , which is based on the *primal integral* defined by Berthold (2013). The primal integral is a calculation of the area under the curve of the convergence profile, and since the time is normalized, the primal integral is the same as the average gap relative to the BKS. It can be seen as the average of the best solution gaps if the runs were randomly stopped at any given time up until the time limit. To approximate the primal integral, let z^{AVG} be defined as the average solution value, as shown in Equation (3), where $t_0 = 0$. The average gap is then calculated as the gap between the average solution value and the BKS using Equation (2).

$$z^{AVG} = \sum_{i=1}^{|T|} z(t_i)(t_i - t_{i-1}) \quad (3)$$

To assure that the results of the experiments presented are statistical significant, we have used a one-tailed Wilcoxon signed-rank test (Wilcoxon 1945). In the test we evaluate two hypotheses, the null-hypothesis H_0 , and the alternative hypothesis H_1 , defined as follows:

$$\begin{aligned} H_0 : z(X^{Best}) &= z(X) \\ H_1 : z(X^{Best}) &> z(X) \end{aligned}$$

where z denotes the solution value, which can be either z^F or z^{AVG} . Each parameter configuration or method is denoted X , and we define $X^{Best} = \operatorname{argmin}\{z(X)\}$, i.e., it is the configuration with the lowest z -value for the given experiment. With a significance level α of 2.5%, the tested hypotheses with p-values lower than α are rejected.

Failing to reject H_0 means the performance of the two configurations in the experiment are statistically indistinguishable. However, if H_0 is rejected, we continue to test H_1 , and rejecting it as well implies that X^{Best} performed better than X . These hypotheses are tested on the experimental results presented in this paper, and the configurations that are not significantly worse than X^{Best} for a given experiment are highlighted in bold in Tables 1 to 5.

3.2 R&R search experiment

The goal of the R&R Search Experiment is to test whether R&R search performed after local search education can improve the original HGS. Recall from Section 2.5 that the R&R search is controlled by three parameters: γ that determines the number of R&R moves, the start temperature \mathcal{T}_0 , and the final temperature \mathcal{T}_f . Together the two temperature parameters and the number of moves define the cooling schedule. We fix the final temperature, $\mathcal{T}_f = 1$, as this ensures a very low probability for accepting deteriorating moves in the final part of the search. That leaves the former two parameters, and the experiment is therefore designed to test different combinations

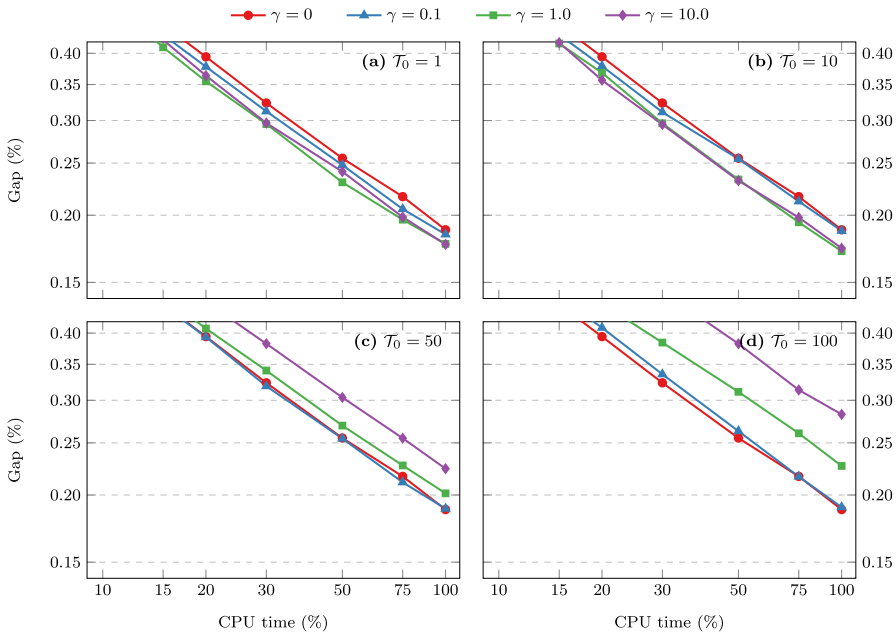


Fig. 4 Convergence profiles from the R&R Search Experiment that tests different combinations of the γ and T_0 parameters, shown on a logarithmic scale

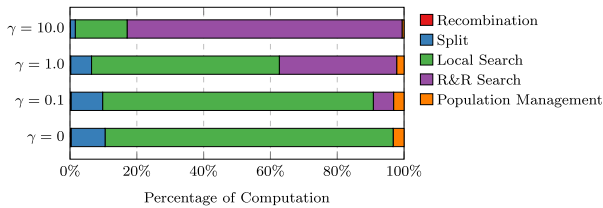
of γ and T_0 . Note that the experiment is not designed to find the precise combination of parameters performing the best, but rather to find the correct order of magnitude. The following parameter value combinations are tested: $\gamma \in \{0, 0.1, 1.0, 10.0\}$ and $T_0 \in \{1, 10, 50, 100\}$.

Figure 4 shows the results from the R&R Search Experiment. Each curve in the figure is the convergence profile of the extended HGS metaheuristic for a specific combination of γ and T_0 . Notice that in the case of $\gamma = 0$, the R&R search is not run at all, and the method is the same as the original HGS. By looking at subfigures (c) and (d) with higher start temperatures, the results indicate that the method performs worse as γ increases. Even though an exponential cooling schedule is used, the combination of high start temperature and larger γ -values make the method spend too much time accepting deteriorating moves, and it becomes increasingly difficult to find an improved solution. On the other hand, if the start temperature is lower, as shown in subfigures (a) and (b), the results indicate that most of the combinations of parameters work somewhat better than the original HGS.

Table 1 summarizes the metrics for the best parameter combinations found in the R&R Search Experiment, and compares them to our reimplemention of the original HGS. The results show that there is no statistically significant difference between any combination of $\gamma \in \{1.0, 10.0\}$, and $T_0 \in \{1, 10\}$. This indicates that the HGSRR is relatively robust with respect to the values of the γ and T_0 parameters. Further, the results show that dedicating some computing time towards R&R search leads to statistically significant better solutions than the original HGS. Even though there is

Table 1 Metrics from relevant parameter combinations in the R&R Search Experiment

\mathcal{T}_0	γ	g^F	g^{AVG}
–	0	0.1879	0.3061
1	1.0	0.1769	0.2814
1	10.0	0.1766	0.2880
10	1.0	0.1714	0.2810
10	10.0	0.1736	0.2828

**Fig. 5** Statistics on the computing time spent on each component of the extended HGS metaheuristic

no statistically significant difference between results for the four parameter configurations, we have selected to proceed with the setting that gives the lowest g^F and g^{AVG} which is $\gamma = 1.0$ and $\mathcal{T}_0 = 10$.

The proportion of time spent on each of the components in the extended HGS metaheuristic are illustrated in Fig. 5. The results show that the temperature parameters affect the performance of the R&R search, but they are considered independent of its time consumption. I.e., the time consumption is correlated with the number of moves determined only by the γ parameter and the problem size. Therefore, each of the horizontal bars shows the distribution of time consumption for the different γ parameter values in the experiment. In the original HGS, corresponding to $\gamma = 0$, the local search is clearly the most computationally expensive component taking more than 85% of the time, on average. For the best performing parameter combination, the local search is still responsible for around 55%, while the R&R search takes around 35%. Out of the ten neighborhoods in the local search, the neighborhood defined by move M10 is the most expensive to explore with around 8% of the total time. This implies that with the best parameter configuration, the R&R-based neighborhood is by far the most time-consuming neighborhood to explore.

3.3 Neighborhood selection experiment

The Neighborhood Selection Experiment explores whether adding the R&R component allows us to remove some of the ten local search neighborhoods without significant performance deterioration. As Arnold and Sörensen (2019) points out, the use of several neighborhoods is only beneficial if they complement each other. I.e., the intersection between the neighborhoods should be small or empty so that each neighborhood finds different solutions. By design, this is the case for the ten neighborhoods used in the HGS local search. However, the R&R search component may

in effect replace some of the local search neighborhoods. Thus, it might be beneficial for the extended HGS to only use a subset of the ten neighborhoods in order to avoid unnecessary computation.

Table 2 shows the metrics for different combinations of search neighborhoods and the R&R component. The first row shows the results of the HGSRR with all eleven neighborhoods enabled, using the best parameter configuration from the R&R Search Experiment presented in Section 3.2. The next eleven rows present metrics for configurations where one of the eleven neighborhoods is removed, ordered by ascending average gap. The results give some indication on the relative importance of each neighborhood, however, the relatively small differences in addition to the lack of statistical significance make it difficult to draw any strong conclusions.

Separately removing the neighborhoods defined by moves M3, M2, M7, and M4 all improve performance, and these neighborhoods may therefore be considered less important. In contrast, removing the neighborhoods defined by R&R, M8, or M10 seems to worsen performance the most, and thus one could argue that these neighborhoods are more important. Since removing M3 yielded the best average gap, the same approach is repeated in which one more neighborhood is removed at a time. Unfortunately, no combination where removing M3 and one additional neighborhood achieved a better average or final gap, although the best one of these is listed in Table 2 as Subset 1.

Subset 2 and Subset 3 are the results of attempting to remove even more neighborhoods because a method with fewer neighborhoods requires less implementation and may be considered less complex. Subset 2 is defined by simultaneously removing all the neighborhoods that individually improved the average gap when being removed. Subset 3, on the other hand, is defined by removing all the neighborhoods that, when being the only one removed, did not increase the average gap by more than 0.1 percentage points compared to the full HGSRR. Despite none of the subsets achieving the best average or final gap, it is noteworthy that Subset 3, which removes six of the neighborhoods from the local search, still performs better than the original HGS without R&R.

The convergence profiles for the relevant neighborhood combinations from Table 2 are shown in Fig. 6. Even though the R&R Search Experiment described in Section 3.2 indicates that the inclusion of the R&R-based neighborhood positively impacts the method's convergence, Fig. 6 shows that the convergence can be further improved by removing a few neighborhoods. However, notice that it is no longer the case if too many neighborhoods are removed, as in Subset 2 and Subset 3.

Although there are several combinations of neighborhoods that we cannot statistically distinguish, the best performing combination, excluding the neighborhood defined by move M3, is used throughout the following experimental investigations.

3.4 Elite education experiment

Similar to the R&R Search Experiment, the Elite Education Experiment tests combinations of parameters in the R&R search, although this time it is used for elite education. Recall that elite education consists of the same education phase as previously, but

Table 2 Metrics for different combinations of search neighborhoods included in the extended HGS metaheuristic

Name	Neighborhoods										g ^F	g ^{AVG}	
	M1	M2	M3	M4	M5	M6	M7	M8	M9	M10			R&R
All	x	x	x	x	x	x	x	x	x	x	x	0.1714	0.2810
All - M3	x	x		x	x	x	x	x	x	x	x	0.1652	0.2735
All - M2	x		x	x	x	x	x	x	x	x	x	0.1678	0.2757
All - M7	x	x	x	x	x	x	x	x	x	x	x	0.1684	0.2794
All - M4	x	x	x	x	x	x	x	x	x	x	x	0.1722	0.2823
All - M6	x	x	x	x	x	x	x	x	x	x	x	0.1782	0.2878
All - M5	x	x	x	x	x	x	x	x	x	x	x	0.1738	0.2887
All - M1		x	x	x	x	x	x	x	x	x	x	0.1867	0.2976
All - M9	x	x	x	x	x	x	x	x		x	x	0.1853	0.3005
All - R&R	x	x	x	x	x	x	x	x	x			0.1879	0.3061
All - M8	x	x	x	x	x	x	x		x	x	x	0.2130	0.3321
All - M10	x	x	x	x	x	x	x	x	x		x	0.2546	0.3896
Subset 1	x	x		x	x		x		x		x	0.1683	0.2744
Subset 2	x			x	x	x		x	x	x	x	0.1739	0.2822
Subset 3	x						x	x	x	x	x	0.1852	0.2946

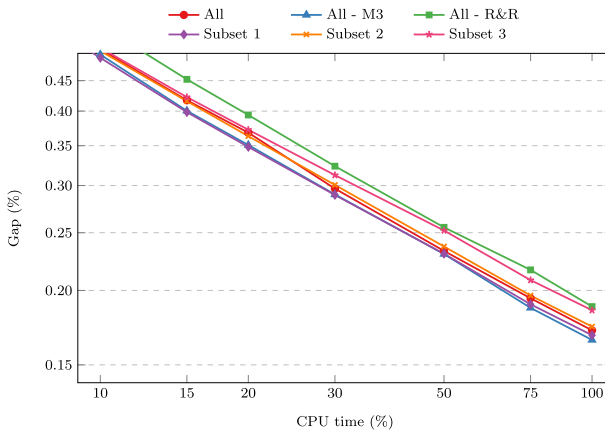


Fig. 6 Convergence profiles for the most relevant combinations of neighborhoods in the extended HGS metaheuristic, shown on a logarithmic scale. The different combinations are defined in Table 2

the R&R search is run for much longer in order to generate a single incumbent elite solution before the rest of the population is initialized. The starting temperature is denoted \mathcal{T}_0^E , and the parameter determining the number of R&R moves in the elite education is denoted γ^E instead of γ . Once again, the experiment is designed to find parameter values that yield good performance, not necessarily optimal values for our experimental setup. Thus the following parameter value combinations are tested: $\gamma^E \in \{1, 1000, 10000, 100000, 200000\}$ and $\mathcal{T}_0^E \in \{1, 10, 50, 100\}$. Note that $\gamma^E = 1$ corresponds to no elite education, as the same value is used for education of the remaining individuals. I.e., $\gamma = 1$ is used during regular education.

The convergence profiles are shown in Fig. 7. Starting with subfigure (a), in which the starting temperature is the lowest, the results show little difference as long as the γ^E parameter is relatively low. However, for the largest values, the convergence is much slower. This is likely due to the increased time spent on the elite education, leaving less time for the remaining components.

Subfigure (b) indicates that increasing the starting temperature to 10 finds somewhat better solutions early on, but has little impact on the final solution gap. Moreover, it is noteworthy that the solutions early on become better with a higher number of moves and a longer cooling schedule obtained by increasing γ^E . The results indicate that spending time on elite education of a single individual improves the convergence of the method.

The same trend can be seen in subfigures (c) and (d) for values of γ^E up to 10,000. However, a further increase in γ^E , in combination with the higher starting temperatures, seems to considerably delay the convergence. A higher starting temperature implies a greater probability of accepting deteriorating moves early on, and thus, a stronger exploration of the search space during the R&R search. Despite delaying the convergence, the results indicate that the increased exploration from a higher starting temperature and a larger number of R&R moves, tend to improve the final gap. Notice that higher values for γ^E increase the time spent on elite education and thus reduces

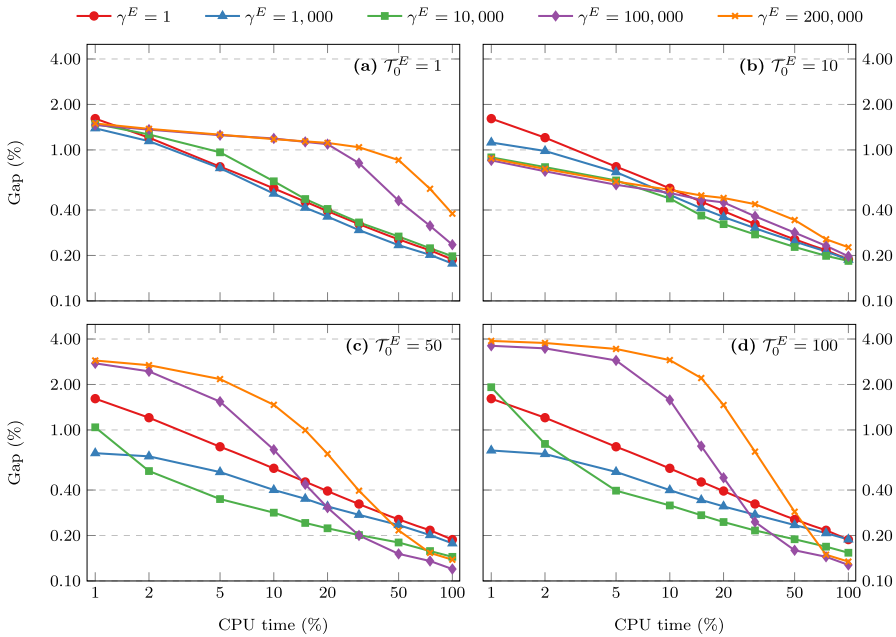


Fig. 7 Convergence profiles from different combinations of values for the γ^E and \mathcal{T}_0^E parameters for elite education, shown on a logarithmic scale

Table 3 Metrics for relevant parameter combinations in the Elite Education Experiment

\mathcal{T}_0^E	γ^E	g^F	g^{AVG}
50	10,000	0.1444	0.1953
50	100,000	0.1199	0.2864
50	200,000	0.1383	0.4344
100	10,000	0.1538	0.2209
100	100,000	0.1277	0.4240
100	200,000	0.1343	0.7082

the number of iterations of the genetic cycle that can be performed within the given time limit.

Table 3 shows the metrics for relevant parameter combinations from the Elite Education Experiment. We observe that a better final gap is achieved by increasing γ^E , thus spending more time on the elite solution, although it cannot be increased too much. The best final gap is achieved with $\mathcal{T}_0^E = 50$ and $\gamma^E = 100,000$, but increasing \mathcal{T}_0^E to 100 does not make it statistically worse. Thus, the starting temperature may not be that important for the final gap when the elite education is given more time.

However, the gains in the final gap by increasing γ^E above 10,000 seem to come at the expense of the average gap. The best average gap is achieved with $\mathcal{T}_0^E = 50$ and $\gamma^E = 10,000$, where all other configurations are statistically worse. Both of the

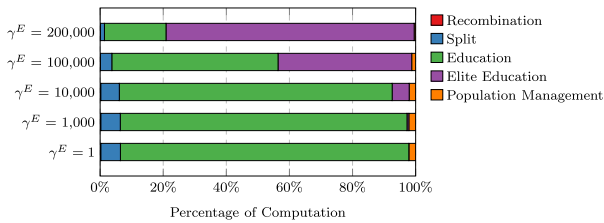


Fig. 8 Statistics on the computational time spent on each component of the HGSRR metaheuristic in the Elite Education Experiment, for different values of γ^E

best performing combinations of parameters spend a significant amount of time on the elite education.

The computational time devoted to each of the components in the HGSRR, including the elite education, are shown in Fig. 8. The local search and the R&R search within the genetic cycle are grouped together as *Education*, while *Elite Education* represents the local search and R&R search on the single incumbent elite solution. In the experiment, $\gamma^E = 10,000$ achieved the best average gap, and Fig. 8 shows that HGSRR with this setting devotes around 5% of the total computing time to finding a single elite solution. The best final gap was achieved with $\gamma^E = 100,000$, and in that case, the elite education takes more than 40% of the total time spent. Even 5% is a significant amount to spend on a single individual, as the search process may visit thousands or even millions of unique individuals.

3.5 Diversity control experiment

Injecting a single, very good solution into the population of a genetic algorithm may lead to the population losing diversity as the injected solution may become too dominant. The loss of diversity decreases the genetic algorithm's ability to explore different parts of the search space, which is crucial for performance. Recall from Section 2.6 that the HGSRR, similar to the HGS, combines the diversity contribution of a solution with its objective value in order to calculate the fitness. Hopefully, this mechanism is enough to maintain diversity even though the population is initialized with an elite individual. The Diversity Control Experiment is designed to check the importance of the mechanism, and test different values for N^E , the parameter controlling the diversity contribution in the fitness function. We test the two γ^E values that yield the best metrics values in the Elite Education Experiment, in combination with $N^E \in \{1, 2, 4, 6, 8\}$.

Convergence profiles for each of the two best performing γ^E values from the Elite Education Experiment are shown in Fig. 9. The results indicate that the choice of N^E plays a significant role in the performance of the method. Adjusting the N^E parameter is one way to control the balance between intensification and diversification. Both subfigures show that the convergence profiles are almost overlapping during the beginning of the runs, as the N^E parameter is irrelevant in the elite education phase. However, rather early in the search process, the convergence profiles diverge. The timing corresponds very well with the time spent on elite education, as presented in Fig. 8.

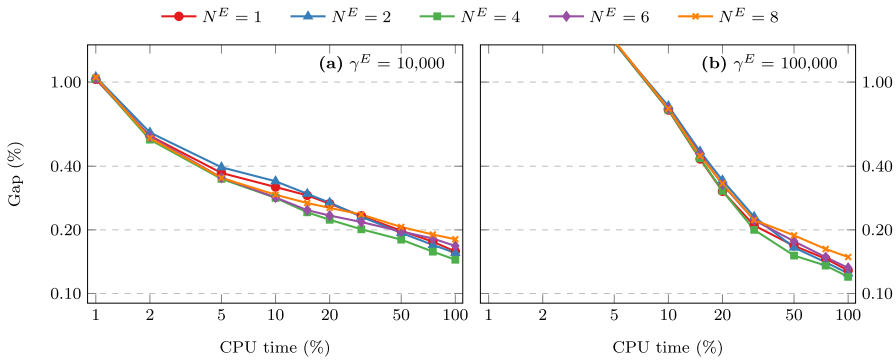


Fig. 9 Convergence profiles from the Diversity Control Experiment shown on a logarithmic scale. The experiment tests different combinations of values for the γ^E and N^E parameters

Table 4 Metrics from relevant parameter combinations in the Diversity Control Experiment. Combinations achieving either the best final or average gap are highlighted in bold

T_0^E	γ^E	N^E	g^F	g^{AVG}
50	10,000	2	0.1555	0.2166
50	10,000	4	0.1444	0.1953
50	10,000	6	0.1677	0.2134
50	100,000	2	0.1245	0.2997
50	100,000	4	0.1199	0.2864
100	100,000	6	0.1321	0.3030

Table 4 presents the metrics of the best convergence profiles from the experiment. Regardless of the value of γ^E , $N^E = 4$ yields the best performance. However, reducing N^E to 2 with $\gamma^E = 100,000$ does not statistically worsen the final gap, which is likely due to leaving less time for the genetic algorithm where the diversity is of importance. As it is preferable with a single parameter value, $N^E = 4$ is chosen, which is the same as used in Vidal (2020). Although the results show that the parameter is important for the performance, the injection of an initial elite individual into the population did not change the best parameter value. Therefore, the results indicate that the HGSRR metaheuristic does not require additional diversification to counteract the dominant elite solution. Since the best value has been used in the previous experiments, the best metrics are the same as shown in Table 3.

3.6 Benchmark test

This section presents the results of a complete benchmark test where the proposed HGSRR metaheuristic is compared against state-of-the-art metaheuristics on the full X dataset from Uchoa et al. (2017). Two different versions of HGSRR are included in the benchmark test, and these correspond to the two best configurations found in the Diversity Control Experiment. Let HGSRR1 denote the configuration achieving the best average gap ($\gamma^E = 10,000$), and HGSRR2 is the configuration with the best final

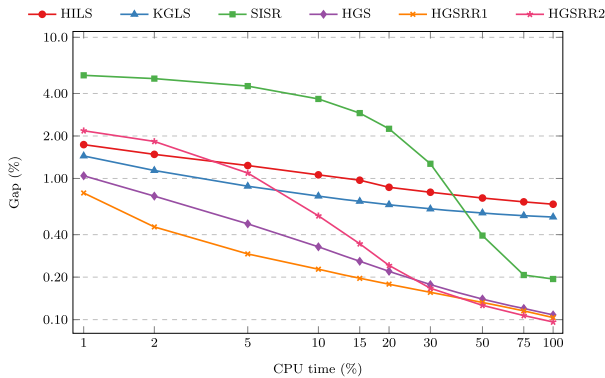


Fig. 10 Convergence profiles for the different metaheuristics in the Benchmark Test are shown on a logarithmic scale

gap ($\gamma^E = 100,000$). The two versions are compared directly against the results from four state-of-the-art metaheuristics presented in the benchmark test in Vidal (2020), namely HILS by Subramanian et al. (2013), KGLS by Arnold and Sörensen (2019), SISR by Christiaens and Vanden Berghe (2020), and HGS by Vidal (2020). A complete list with the results on each of the problem instances is found in Appendix B.

The convergence profiles of the methods in the Benchmark Test are presented in Fig. 10. Both versions of the HGSRR achieve a better final gap than the rest, although the differences are relatively small in terms of absolute values. This is not unexpected, given the hard competition. HGSRR1 achieved the best average gap in addition to a slightly better final gap than the original HGS. Furthermore, the best final gap is achieved by HGSRR2, but with a slower convergence, the method has a somewhat worse average gap.

Even though the computing times of each solution method presented in this section are adjusted based on the hardware used to produce them, other factors such as the choice of programming language, compiler, and other implementation details may still lead to an unfair comparison. To verify that we do not introduce an unfair advantage to HGSRR, we have tested our implementation of HGS in Rust against the original HGS implemented in C++. The results show that the two implementations have very similar convergence profiles, although the Rust implementation performs consistently and significantly worse, indicating that the real relative difference in performance in favor of HGSRR may be bigger than presented in this paper. Detailed results from the comparison of the two implementations of HGS may be found in Appendix C.

To get more information about the strengths and weaknesses of the different methods, their performance on various subsets of the instances is analyzed. Like in the benchmark test by Vidal (2020), the following subsets are defined:

- (a) SMALL: First 50 instances in the dataset with between 100 and 330 customers.
- (b) SHORT ROUTES: Instances of index $i = 5k + 1$ and $i = 5k + 2$ for $k \in \{0, \dots, 19\}$. These 40 instances are designed to have fewer customers per route.
- (c) LARGE: Last 50 instances in the dataset with between 335 and 1000 customers.

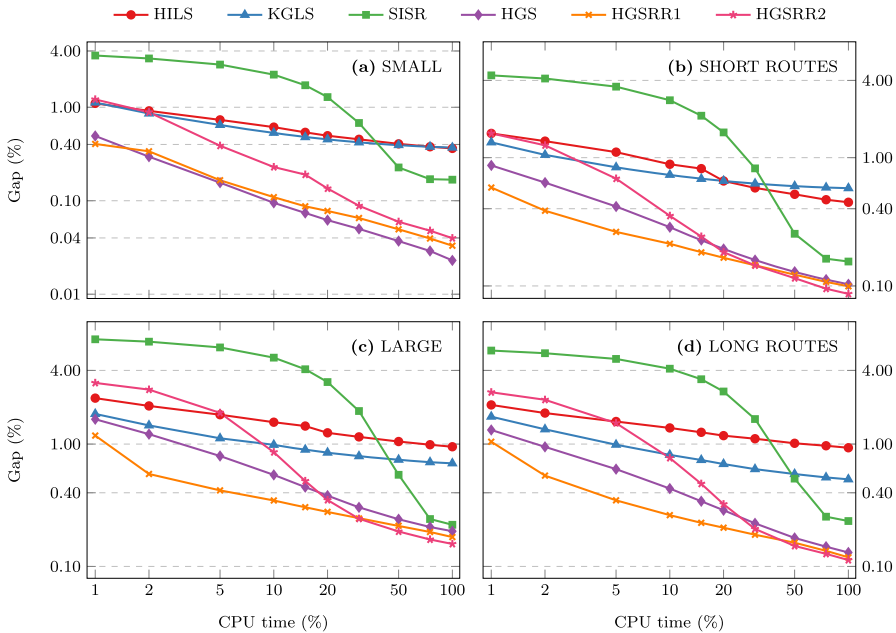


Fig. 11 Convergence profiles for the different metaheuristics in the Benchmark Test for the four different subsets of instances are shown on a logarithmic scale

Table 5 Metrics for the methods in the Benchmark Test, where the values in bold are not statistically worse than the best performing method on the respective subset

Instances	Metric	HILS	KGLS	SISR	HGS	HGSRR1	HGSRR2
Small	g^F	0.3633	0.3702	0.1685	0.0231	0.0330	0.0398
	g^{AVG}	0.4365	0.4206	0.6149	0.0497	0.0607	0.1030
Short Routes	g^F	0.4495	0.5776	0.1546	0.1029	0.0994	0.0867
	g^{AVG}	0.5718	0.6263	0.7303	0.1585	0.1364	0.1703
Large	g^F	0.9507	0.6957	0.2189	0.1937	0.1738	0.1527
	g^{AVG}	1.1127	0.7816	1.3591	0.3011	0.2354	0.3416
Long Routes	g^F	0.9331	0.5153	0.2354	0.1301	0.1186	0.1127
	g^{AVG}	1.0626	0.6104	1.1599	0.2199	0.1739	0.2819
All	g^F	0.6570	0.5329	0.1937	0.1084	0.1034	0.0962
	g^{AVG}	0.7746	0.6011	0.9870	0.1754	0.1480	0.2221

– (d) LONG ROUTES: Instances of index $i = 5k + 3$ and $i = 5k + 4$ for $k \in \{0, \dots, 19\}$. These 40 instances are designed to have a higher number of customers per route.

The convergence profiles of the methods in the benchmark on the different subsets are shown in Fig. 11, while the metrics are shown in Table 5. By inspecting the

different subfigures and the summarized metrics, the same observation as previously can be seen. In general, HGSRR1 performed significantly better in terms of average gap, meanwhile the significantly best final gap was achieved by HGSRR2. However, this is not the case for the subset with the smallest problem instances, where the original HGS achieves the significantly best final and average gaps. It is only the final gaps on the instances with long routes that are not statistically different, where HGS and the two versions of HGSRR are indistinguishable within the given confidence level used in the statistical tests.

Most of the BKSs are proven to be optimal for the smallest instances, and the solutions achieved by the best metaheuristics are either the same as the BKSs, or very close. It can be argued that heavier intensification may help identify good solutions faster, although to find the optimal or close to the optimal solution, a strong diversification is required. Recall the trade-off between intensification and diversification. Since the original HGS does not include R&R search or elite education, the genetic algorithm can run more genetic cycles, or alternatively, achieve more restarts of the algorithm. As a result, the HGS is considered heavier weighted towards diversification compared to both versions of the HGSRR. Higher degree of diversification seems to be an advantage on the smaller instances, but in contrast, it may be a drawback on the larger instances in which even the best metaheuristics are still relatively far away from the best known solutions.

Despite both versions of the HGSRR performing slightly worse on the smaller instances, their performance on the other subsets is noteworthy. Especially their performance on the larger instances should be highlighted, as the final gap of the HGSRR2 is more than 0.04 percentage points better than the HGS, and the average gap of the HGSRR1 is more than 0.06 percentage points better than the HGS. The significant improvements by the two HGSRR versions on both of the metrics on the larger instances in terms of absolute values are the strongest contributor to their overall performance. On the complete X dataset, HGSRR2 reduces the final gap by 0.0122 percentage points, and HGSRR1 reduces the average gap by 0.0274 percentage points. Although the improvements are small in absolute numbers, they correspond to a relative reduction in the final and average gaps of 11.25% and 15.62%, respectively.

Since the difference in performance seems to be related to the size of the problems, and the final gaps are correlated with the problem size, it is interesting to take a closer look at the distributions of the final gaps.

Fig. 12 shows the distributions of final gaps for each of the metaheuristics illustrated as boxplots, and the same five quartiles from the distributions are summarized in Table 6. As the solutions are closer to the BKSs on the smaller problem instances, both the 25th percentile and the median are best for HGS that performs better on the smallest instances. At the same time, the 75th percentile and the maximum are significantly lower for both versions of the HGSRR. The spread of the final gaps is clearly the lowest for HGSRR2. I.e., it has the lowest difference in the worst and the best final gaps across all the instances. A lower spread in the final gaps may indicate a more robust method in terms of lower variance in the final gaps on problem instances from the same distribution. Note that nothing can be said about the variance across multiple runs on the same problem instance, only about the expected variance across multiple instances from the same distribution as the benchmark instances.

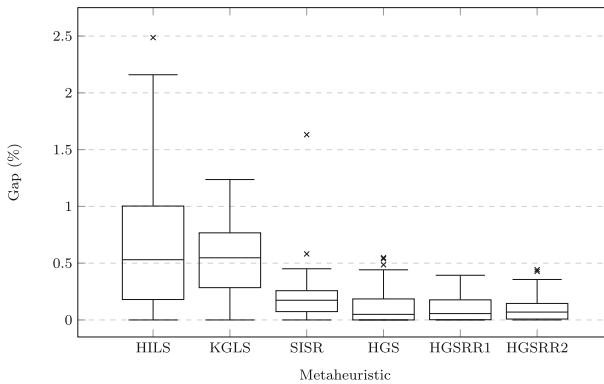


Fig. 12 The distributions of the final gaps for each of the metaheuristics in the Benchmark Test are represented by the five quartiles and shown as boxplots

Table 6 The distribution of the final gaps for each of the metaheuristics in the Benchmark Test. The distributions are represented by the five quartiles with the best ones marked in bold.

	HILS	KGLS	SISR	HGS	HGSRR1	HGSRR2
Q_0 : Minimum	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
Q_1 : 25 th Percentile	0.1799	0.2845	0.0735	0.0000	0.0020	0.0080
Q_2 : Median	0.5304	0.5474	0.1734	0.0491	0.0561	0.0693
Q_3 : 75 th Percentile	1.0031	0.7671	0.2577	0.1851	0.1770	0.1454
Q_4 : Maximum*	2.1595	1.2366	0.4508	0.4416	0.3934	0.3564

* The maximum excluding outliers.

Table 7 Best known solutions improved by HGSRR

Dataset	Instance	Size	BKS	Improved BKS
Uchoa et al. (2017)	X-n384-k52	383	65938	65928
Uchoa et al. (2017)	X-n641-k35	640	63684	63682

3.7 Best known solutions experiment

In a final experiment we run the HGSRR for an extended time limit of 16 hours in an attempt to improve the best known solutions, which can be found on the CVRPLIB (2021) website. The experiment is conducted on the 48 instances in the X dataset for which a proven optimal solution has not yet been found. The parameter configuration remains the same as in the previous experiments, except for the γ^E controlling the scope of the elite education. Because larger γ^E resulted in better final gap in the Benchmark Test, the parameter value is further increased to 1,000,000 given the prolonged runtime. The experiment resulted in two new improved solutions, as listed in Table 7. An overview of the currently used solutions

4 Conclusions

The Capacitated Vehicle Routing Problem is a central problem in combinatorial optimization. Due to its combination of industrial relevance and scientific interest, the CVRP has been subject to intense research efforts for more than sixty years. Still, there is strong competition regarding efficient solution algorithms, and substantial improvements are being made.

Among the most competitive heuristics for the CVRP of today are Hybrid Genetic Search (HGS) by Vidal (2020), and Slack Induction by String Removals (SISR) by Christiaens and Vanden Berghe (2020). HGS may be characterized as a memetic algorithm where a population of solutions is evolved, new solutions are created through recombination, and the solutions are improved by local search. In contrast, SISR is a trajectory-based metaheuristic where a single solution is iteratively improved through large neighborhood search (LNS) with ruin and recreate (R&R) operators and acceptance is governed by simulated annealing.

To meet our goal of investigating combinations of HGS and LNS for the CVRP, we formulated three research questions. To answer them, we developed a novel metaheuristic and conducted a sequence of six computational experiments. Our proposed metaheuristic is called Hybrid Genetic Search with Ruin and Recreate (HGSRR). It combines HGS and SISR in two different ways: First, the R&R based LNS of SISR complements the local search operators of HGS in the improvement of solutions in each generation of the population, also called education. Second, SISR is used to create an elite solution in the initial population.

Our experimental results show that devoting a substantial part of the computing time in the education phase (around 35%) to the R&R large neighborhood search of SISR yields the best results and improves both the average gap and the final gap, compared to the original HGS and SISR methods. Further, our experiments clearly indicate that adding SISR changes the relative importance of the local search operators employed by HGS. In fact, the results show that removing the M3 neighborhood operator (i.e., relocation of a sequence of two customers) from the ten original operators of HGS provides the best results of the 15 configurations investigated. SISR may even replace six of the ten HGS operators and still exhibit performance improvement relative to the original HGS.

Our results also show that it pays off to devote between 5% and 40% of the total computational effort to create an elite solution for the initial population, both in terms of convergence and the final gap. However, different configurations yield the best results for these two performance metrics. Further, we observe that injecting an elite solution in the initial population of HGS does not require additional diversity control mechanisms.

The results of the Benchmark Test on the 100 X-instances by Uchoa et al. (2017) show that HGSRR is highly competitive. We investigated two configurations of HGSRR: HGSRR1 that focuses on average gap performance, and HGSRR2 that is the best configuration for the final gap metric. Both outcompete four state-of-the-art metaheuristics regarding final gap, including HGS and SISR, on all but the smallest X-instances. With a third configuration and a 16 hour time limit, HGSRR found new best known solutions to two X instances with 383 and 640 customers, respectively.

Hybrid genetic search and R&R based large neighborhood search are both among the best solution methods for vehicle routing and many other combinatorial optimization problems. Our investigations show that it is possible to design combinations of the two that are competitive and in fact improve performance relative to each of them. Our results should motivate further work in this direction.

Acknowledgements We would like to thank Professor Thibaut Vidal for sharing the detailed results from his paper (Vidal 2020). Further, we would like to acknowledge the support for this research from the Research Council of Norway, through the Axiom project with research Grant Number 263031.

Funding Open access funding provided by SINTEF.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

Appendix A: Parameters

The HGSRR metaheuristic is controlled by a set of several different parameters, as defined throughout Section 2. Because the HGSRR combines two metaheuristics, the parameters are divided into two tables. Table 8 shows all the parameters used in the genetic algorithm based on the HGS by Vidal (2020), and Table 9 presents

Table 8 Overview of the parameters used to control the genetic algorithm in HGSRR. The same parameters and values are used in the HGS by Vidal (2020)

Parameter	Value	Description
μ	25	Minimum population size
μ^I	100	Number of initial individuals
λ	40	Generation size
Γ	20	Neighborhood granularity
ξ^{REF}	0.2	Target proportion of feasible individuals
N^C	5	Number of individuals in the diversity contribution
N^E	4	Number of elite individuals
N^{IT}	20,000	Number of iterations without improvement before restart

Table 9 Overview of the parameters used to control the R&R-based search in the HGSRR. The first five parameters and values are taken from the SISR metaheuristic proposed by Christiaens and Vanden Berghe (2020)

Parameter	Value	Description
\bar{c}	10	Average number of customers removed during ruin
L^{MAX}	10	Maximum length of removed string
α	0.01	Used to determine number of preserved customers in Split String procedure
β	0.01	Blink probability
\mathcal{T}_f	1	Final temperature
\mathcal{T}_0	10	Start temperature in the genetic cycle
\mathcal{T}_0^E	50	Start temperature in the elite education
γ	1	Factor determining number of R&R moves in the genetic cycle
γ^E	10,000/100,000	Factor determining number of R&R moves in the elite education

the parameters used to control the R&R search based on the SISR metaheuristic by Christiaens and Vanden Berghe (2020). The first four parameters in Table 9 are used to control the R&R operators themselves, and for sake of completeness, they are included in the Table despite not being defined in Section 2. For information about the first four parameters, we refer to the original paper.

All the parameters and values are taken from the respective papers, except for the four last parameters in Table 9. Parameter \mathcal{T}_0 takes another value than in Christiaens and Vanden Berghe (2020), while \mathcal{T}_0^E , γ , and γ^E are introduced in this paper.

Appendix B: Benchmark test results

This appendix presents all the results from the Benchmark Test presented in Section 3.6. The results are reported as the final solution value, denoted z^F , and the average solution value, denoted z^{AVG} . For definitions of the values, see Section 3.1.

Recall that for all the problem instances, ten independent runs were performed with each of the metaheuristics. Therefore, for each metaheuristic and problem instance, the value from the best run is reported, in addition to the average value from the ten runs. The best z^F and the average z^F from the ten runs can be found in Table 10, Table 11, and Table 12. Similarly, the best z^{AVG} and the average z^{AVG} from the ten runs can be found in Table 13, Table 14, and Table 15.

Table 10 Comparison of the best and the average z^F (1 of 3)

Instance	HILS(2013)			KGLS(2018)			SISR(2020)			HGS(2020)			HGSRR1(2021)			HGSRR2(2021)			BKS
	Avg	Gap	Best	Avg	Gap	Best	Avg	Gap	Best	Avg	Gap	Best	Avg	Gap	Best	Avg	Gap	Best	
X-nl01-k25	27591.0	0.00	27591	27631.9	0.15	27595	27593.3	0.01	27591	27591.0	0.00	27591	27591.0	0.00	27591	27591.0	0.00	27591	27591
X-nl06-k14	26391.1	0.11	26381	26413.2	0.19	26375	26380.9	0.07	26368	26381.4	0.07	26364	26368.8	0.03	26362	26368.8	0.03	26362	26362
X-nl10-k13	14971.0	0.00	14971	14971.0	0.00	14971	14972.1	0.01	14971	14971.0	0.00	14971	14971.0	0.00	14971	14971.0	0.00	14971	14971
X-nl15-k10	12747.0	0.00	12747	12747.1	0.00	12747	12747.0	0.00	12747	12747.0	0.00	12747	12747.0	0.00	12747	12747.0	0.00	12747	12747
X-nl20-k6	13333.7	0.01	13332	13332.0	0.00	13332	13332.0	0.00	13332	13332.0	0.00	13332	13332.0	0.00	13332	13332.0	0.00	13332	13332
X-nl25-k30	55846.5	0.55	55713	55740.8	0.36	55670	55559.8	0.04	55539	55539.0	0.00	55539	55539.0	0.00	55539	55539.0	0.00	55539	55539
X-nl29-k18	28972.1	0.11	28954	28971.6	0.11	28954	28948.9	0.03	28940	28940.0	0.00	28940	28940.0	0.00	28940	28940.0	0.00	28940	28940
X-nl34-k13	10947.5	0.29	10929	10940.5	0.22	10930	10937.7	0.20	10918	10916.0	0.00	10916	10916.0	0.00	10916	10916.0	0.00	10916	10916
X-nl39-k10	13591.2	0.01	13590	13590.0	0.00	13590	13590.4	0.00	13590	13590.0	0.00	13590	13590.0	0.00	13590	13590.0	0.00	13590	13590
X-nl43-k7	15735.7	0.23	15723	15730.6	0.19	15726	15727.8	0.18	15700	15700.0	0.00	15700	15700.0	0.00	15700	15700.0	0.00	15700	15700
X-nl48-k46	43448.0	0.00	43448	43588.3	0.32	43507	43464.1	0.04	43448	43448.0	0.00	43448	43448.0	0.00	43448	43448.0	0.00	43448	43448
X-nl53-k22	21452.3	1.09	21225	21386.0	0.78	21375	21228.6	0.04	21225	21225.0	0.02	21225	21225.0	0.02	21225	21225.0	0.02	21225	21220
X-nl57-k13	16876.0	0.00	16876	16877.5	0.01	16876	16878.2	0.01	16876	16876.0	0.00	16876	16876.0	0.00	16876	16876.0	0.00	16876	16876
X-nl62-k11	14152.4	0.10	14138	14147.0	0.06	14147	14159.0	0.15	14138	14138.0	0.00	14138	14138.0	0.00	14138	14138.0	0.00	14138	14138
X-nl67-k10	20603.7	0.23	20557	20586.9	0.15	20557	20558.6	0.01	20557	20557.0	0.00	20557	20557.0	0.00	20557	20557.0	0.00	20557	20557
X-nl72-k51	45665.3	0.13	45607	45802.8	0.43	45763	45622.6	0.03	45607	45607.0	0.00	45607	45607.0	0.00	45607	45607.0	0.00	45607	45607
X-nl76-k26	48218.5	0.85	48140	47991.6	0.38	47958	47823.7	0.02	47812	47812.0	0.00	47812	47812.0	0.00	47812	47812.0	0.00	47812	47812
X-nl81-k23	25572.1	0.01	25569	25602.3	0.13	25594	25575.1	0.02	25569	25569.0	0.00	25569	25569.0	0.00	25569	25569.0	0.00	25569	25569
X-nl86-k15	24170.7	0.11	24147	24178.3	0.14	24156	24166.2	0.09	24151	24145.0	0.00	24145	24145.0	0.00	24145	24145.0	0.00	24145	24145
X-nl90-k8	17108.0	0.75	17029	17033.5	0.32	17001	16982.8	0.02	16980	16980.0	0.00	16980	16980.0	0.00	16980	16980.0	0.00	16980	16980
X-nl95-k51	44305.0	0.18	44225	44427.2	0.46	44396	44292.0	0.15	44241	44225.0	0.00	44225	44225.0	0.00	44225	44225.0	0.00	44225	44225

Table 10 continued

Instance	HILS(2013)			KGLS(2018)			SISR(2020)			HGS(2020)			HGSRR(2021)			HGSRR2(2021)			BKS
	Avg	Gap	Best	Avg	Gap	Best	Avg	Gap	Best	Avg	Gap	Best	Avg	Gap	Best	Avg	Gap	Best	
X-n200-k36	58784.0	0.35	58617	58828.0	0.43	58756	58635.6	0.10	58587	58578.0	0.00	58578	58590.3	0.02	58578	58597.4	0.03	58578	58578
X-n204-k19	19617.6	0.27	19565	19621.0	0.29	19581	19653.2	0.45	19565	19565.0	0.00	19565	19565.0	0.00	19565	19565.0	0.00	19565	19565
X-n209-k16	30739.0	0.27	30702	30709.7	0.18	30685	30661.7	0.02	30656	30656.0	0.00	30656	30656.0	0.00	30656	30656.0	0.00	30656	30656
X-n214-k11	11077.2	2.04	10917	10944.3	0.81	10913	10894.4	0.35	10874	10860.5	0.04	10856	10864.4	0.08	10856	10866.9	0.10	10856	10856
X-n219-k73	117595.0	0.00	117595	117689.1	0.08	117651	117623.7	0.02	117596	117596.1	0.00	117595	117596.1	0.00	117595	117598.3	0.00	117595	117595
X-n223-k34	40549.8	0.28	40490	40714.4	0.69	40686	40535.5	0.24	40504	40437.0	0.00	40437	40438.6	0.00	40437	40446.4	0.02	40437	40437
X-n228-k23	25803.7	0.24	25745	25836.8	0.37	25808	25814.3	0.28	25782	25742.8	0.00	25742	25746.3	0.02	25742	25746.8	0.02	25742	25742
X-n233-k16	19296.0	0.34	19276	19328.6	0.51	19268	19285.7	0.29	19232	19230.0	0.00	19230	19231.6	0.01	19230	19231.4	0.01	19230	19230
X-n237-k14	27068.8	0.10	27042	27095.9	0.20	27044	27081.1	0.14	27043	27042.0	0.00	27042	27042.0	0.00	27042	27048.2	0.02	27042	27042
X-n242-k48	82867.9	0.14	82809	83209.2	0.55	83136	82885.6	0.16	82805	82806.0	0.07	82771	82811.3	0.07	82751	82797.2	0.06	82778	82751
X-n247-k50	37502.3	0.61	37300	37388.4	0.31	37317	37379.6	0.28	37274	37277.1	0.01	37274	37274.4	0.00	37274	37274.1	0.00	37274	37274
X-n251-k28	38859.4	0.45	38798	38893.3	0.54	38847	38765.2	0.21	38687	38689.9	0.02	38684	38686.7	0.01	38684	38706.8	0.06	38684	38684
X-n256-k16	18880.8	0.22	18880	18891.6	0.28	18888	18887.3	0.26	18880	18839.6	0.00	18839	18880.0	0.22	18880	18880.0	0.22	18880	18839
X-n261-k13	26808.2	0.94	26692	26717.5	0.60	26671	26595.8	0.14	26558	26558.2	0.00	26558	26560.7	0.01	26558	26560.4	0.01	26558	26558

Table 11 Comparison of the best and the average z^F (2 of 3)

Instance	HILS(2013)			KGLS(2018)			SISR(2020)			HGS(2020)			HGSRR(2021)			HGSRR2(2021)			BKS
	Avg	Gap	Best	Avg	Gap	Best	Avg	Gap	Best	Avg	Gap	Best	Avg	Gap	Best	Avg	Gap	Best	
X-n266-k58	75611.4	0.18	75478	75954.6	0.63	75793	75609.2	0.17	75549	75564.7	0.11	75478	75601.5	0.16	75529	75579.3	0.13	75529	75478
X-n270-k35	35352.9	0.18	35324	35462.1	0.48	35447	35364.4	0.21	35325	35303.0	0.03	35303	35303.4	0.04	35303	35305.6	0.04	35291	35291
X-n275-k28	21262.4	0.08	21245	21299.4	0.26	21265	21250.5	0.03	21245	21245.0	0.00	21245	21245.0	0.00	21245	21245.0	0.00	21245	21245
X-n280-k17	33803.4	0.90	33725	33670.1	0.50	33598	33648.6	0.43	33545	33543.2	0.12	33506	33540.7	0.11	33506	33562.5	0.18	33505	33503
X-n284-k15	20415.9	0.99	20325	20360.0	0.72	20323	20287.6	0.36	20261	20245.5	0.15	20231	20241.6	0.13	20225	20246.1	0.15	20226	20215
X-n289-k60	95515.0	0.38	95401	95882.8	0.77	95770	95345.8	0.20	95245	95300.9	0.16	95242	95307.3	0.16	95240	95254.2	0.11	95205	95151
X-n294-k50	47262.0	0.21	47240	47454.1	0.62	47413	47251.9	0.19	47199	47184.1	0.05	47167	47205.6	0.09	47174	47226.3	0.14	47210	47161
X-n298-k31	34383.7	0.45	34318	34377.4	0.43	34359	34267.8	0.11	34234	34234.8	0.01	34231	34241.8	0.03	34231	34252.0	0.06	34231	34231
X-n303-k21	21900.7	0.76	21806	21903.4	0.77	21845	21772.9	0.17	21753	21748.5	0.06	21739	21756.0	0.09	21745	21748.7	0.06	21738	21736
X-n308-k13	26058.6	0.77	25989	26076.4	0.84	25999	26281.0	1.63	26224	25870.8	0.05	25862	25873.3	0.06	25862	25893.7	0.13	25870	25859
X-n313-k71	94290.3	0.26	94216	94763.8	0.77	94652	94155.7	0.12	94098	94112.2	0.07	94045	94090.5	0.05	94045	94108.9	0.07	94045	94043
X-n317-k53	78355.0	0.00	78355	78413.5	0.07	78391	78386.1	0.04	78361	78355.4	0.00	78355	78356.8	0.00	78355	78356.4	0.00	78355	78355
X-n322-k28	29996.5	0.54	29923	30038.0	0.68	30010	29892.5	0.20	29861	29848.7	0.05	29834	29850.0	0.05	29834	29854.9	0.07	29848	29834
X-n327-k20	27815.8	1.03	27767	27646.8	0.42	27613	27644.7	0.41	27611	27540.8	0.03	27532	27576.7	0.16	27560	27588.5	0.21	27572	27532
X-n331-k15	31227.4	0.40	31136	31200.1	0.32	31111	31124.5	0.07	31122	31103.0	0.00	31102	31103.2	0.00	31103	31105.5	0.01	31103	31102
X-n336-k84	139560.0	0.32	139351	140831.3	1.24	140716	139429.8	0.23	139272	139273.5	0.12	139205	139328.7	0.16	139224	139276.7	0.12	139177	139111
X-n344-k43	42307.5	0.61	42190	42350.5	0.71	42229	42122.7	0.17	42081	42075.6	0.06	42061	42073.9	0.06	42056	42078.8	0.07	42061	42050
X-n351-k40	26134.7	0.92	26050	26190.7	1.14	26150	25976.5	0.31	25965	25943.6	0.18	25924	25942.3	0.18	25936	25946.8	0.20	25933	25896
X-n359-k29	52089.2	1.13	51820	51901.3	0.77	51662	51549.8	0.09	51514	51620.0	0.22	51566	51574.8	0.14	51507	51529.6	0.05	51507	51505
X-n367-k17	22985.5	0.75	22956	22944.7	0.57	22867	22836.1	0.10	22821	22814.0	0.00	22814	22814.0	0.00	22814	22814.0	0.00	22814	22814
X-n376-k94	147713.4	0.00	147713	147854.1	0.10	147801	147763.5	0.03	147736	147714.5	0.00	147713	147718.3	0.00	147713	147733.6	0.01	147713	147713
X-n384-k52	66407.8	0.71	66351	66443.0	0.76	66363	66113.6	0.26	66046	66049.1	0.17	65997	66031.0	0.14	65985	66043.4	0.16	65974	65940

Table 11 continued

Instance	HILS(2013)		KGLS(2018)		SISR(2020)		HGS(2020)		HGSRR1(2021)		HGSRR2(2021)		BKS						
	Avg	Gap Best	Avg	Gap Best	Avg	Gap Best	Avg	Gap Best	Avg	Gap Best	Avg	Gap Best							
X-n393-k38	38515.7	0.67	38360	38466.4	0.54	38433	38384.5	0.33	38338	38260.0	0.00	38260	38292.0	0.08	38260	38260			
X-n401-k29	66729.5	0.86	66597	66501.9	0.51	66466	66239.5	0.12	66222	66252.5	0.14	66209	66213.7	0.08	66183	66187	66163		
X-n411-k19	19970.8	1.31	19834	19924.8	1.08	19782	19776.7	0.33	19757	19720.3	0.04	19716	19741.2	0.15	19734	19712	19712		
X-n420-k130	107838.0	0.04	107801	108295.3	0.46	108175	107853.4	0.05	107809	107839.8	0.04	107810	107837.1	0.04	107815	107810	107798		
X-n429-k61	65786.8	0.52	65689	65857.5	0.62	65795	65539.3	0.14	65494	65502.7	0.08	65484	65507.1	0.09	65456	65504.2	0.08	65458	65449
X-n439-k37	36448.5	0.16	36402	36483.8	0.26	36445	36457.7	0.18	36402	36395.5	0.01	36395	36400.9	0.03	36395	36403.2	0.03	36395	36391
X-n449-k29	56272.8	1.88	56154	55770.7	0.97	55675	55388.8	0.28	55296	55368.5	0.25	55306	55351.1	0.21	55300	55355.3	0.22	55285	55233
X-n459-k26	24479.3	1.41	24363	24251.0	0.46	24234	24228.3	0.37	24187	24163.8	0.10	24139	24166.4	0.11	24160	24168.7	0.12	24146	24139
X-n469-k138	222189.0	0.16	221939	223468.0	0.74	223086	222253.9	0.19	222090	222170.1	0.16	221916	222280.1	0.21	222107	222063.9	0.11	221916	221824
X-n480-k70	89857.0	0.46	89629	89986.3	0.60	89926	89515.1	0.07	89458	89524.4	0.08	89498	89542.0	0.10	89465	89526.0	0.09	89457	89449
X-n491-k59	67238.7	1.13	67098	67145.6	0.99	67034	66606.9	0.18	66502	66641.5	0.23	66569	66628.7	0.21	66592	66606.0	0.18	66560	66487
X-n502-k39	69380.4	0.22	69340	69333.9	0.16	69307	69271.4	0.07	69238	69239.5	0.02	69230	69256.7	0.04	69233	69250.8	0.04	69241	69226
X-n513-k21	24406.9	0.85	24316	24360.7	0.66	24293	24293.9	0.38	24237	24201.0	0.00	24201	24205.4	0.02	24201	24216.9	0.07	24201	24201

Table 12 Comparison of the best and the average z^F (3 of 3)

Instance	HILS(2013)			KGLS(2018)			SISR(2020)			HGS(2020)			HGSRRI(2021)			HGSRR2(2021)			BKS			
	Avg	Best	Gap	Avg	Best	Gap	Avg	Best	Gap	Avg	Best	Gap	Avg	Best	Gap	Avg	Best	Gap	Avg	Best	Gap	
X-n524-k153	155176.6	0.38	154656	155699.6	0.72	155422	154894.6	0.20	154758	154747.6	0.10	154646	154648.8	0.04	154646	154640.9	0.03	154607	154593			
X-n536-k96	95713.0	0.89	95663	95864.7	1.05	95781	95145.9	0.29	95071	95091.9	0.24	95040	95133.4	0.28	95000	95068.1	0.21	94986	94868			
X-n548-k50	86976.2	0.32	86813	86938.6	0.28	86901	86789.1	0.10	86710	86778.4	0.09	86710	86745.2	0.05	86701	86746.7	0.05	86701	86700			
X-n561-k42	43095.7	0.89	42918	43031.7	0.74	42989	42875.0	0.37	42799	42742.7	0.06	42726	42759.2	0.10	42743	42780.0	0.15	42750	42717			
X-n573-k30	51203.9	1.05	51102	50957.2	0.56	50849	50842.6	0.33	50777	50813.0	0.28	50757	50801.2	0.25	50776	50824.4	0.30	50732	50673			
X-n586-k159	190835.6	0.27	190695	191411.4	0.58	191260	190640.1	0.17	190454	190588.1	0.14	190470	190653.0	0.18	190532	190593.8	0.15	190417	190316			
X-n599-k92	109460.7	0.93	109119	109356.1	0.83	109125	108684.8	0.22	108598	108656.0	0.19	108605	108632.2	0.17	108546	108597.9	0.14	108522	108451			
X-n613-k62	60457.8	1.55	60333	60201.2	1.12	60111	59705.6	0.29	59609	59696.3	0.27	59636	59656.2	0.20	59593	59619.4	0.14	59580	59535			
X-n627-k43	63052.4	1.43	62928	62568.1	0.65	62486	62291.8	0.21	62221	62371.6	0.33	62238	62325.6	0.26	62254	62303.1	0.22	62270	62164			
X-n641-k35	64709.8	1.59	64591	64094.3	0.63	63952	63851.8	0.25	63802	63874.2	0.28	63782	63875.0	0.28	63774	63820.7	0.20	63750	63694			
X-n655-k131	106785.7	0.01	106780	106956.8	0.17	106936	106841.6	0.06	106808	106808.8	0.03	106785	106829.3	0.05	106813	106818.4	0.04	106798	106780			

Table 12 continued

Instance	HILS(2013)			KGLS(2018)			SISR(2020)			HGS(2020)			HGSRRI(2021)			HGSRR2(2021)			BKS
	Avg	Best	Gap	Avg	Best	Gap	Avg	Best	Gap	Avg	Best	Gap	Avg	Best	Gap	Avg	Best	Gap	
X-n670-k130	148272.8	147974	1.33	147654.2	0.90	147477	146961.5	0.43	146676	146777.7	0.30	146640	146855.1	0.36	146439	146979.7	0.44	146540	146332
X-n685-k75	68988.4	1.15	68843	68854.7	0.95	68628	68379.6	0.26	68271	68343.1	0.20	68288	68372.0	0.24	68301	68347.5	0.21	68299	68205
X-n701-k44	83159.4	1.51	82982	82513.5	0.72	82447	82053.9	0.16	82007	82237.3	0.38	82075	82137.0	0.26	82042	82037.2	0.14	81946	81923
X-n716-k35	44264.0	2.02	44058	43730.4	0.79	43627	43492.0	0.24	43449	43505.8	0.27	43459	43523.9	0.32	43482	43474.6	0.20	43429	43387
X-n733-k159	137014.7	0.61	136848	137299.3	0.81	137185	136445.2	0.19	136344	136426.9	0.17	136323	136377.1	0.14	136318	136358.7	0.12	136285	136190
X-n749-k98	78323.3	1.31	78213	78211.9	1.16	78109	77534.9	0.29	77399	77655.4	0.44	77563	77565.5	0.33	77489	77522.2	0.27	77418	77314
X-n766-k71	115858.3	1.23	115396	115186.0	0.64	115011	114836.0	0.33	114751	114764.5	0.27	114679	114787.9	0.29	114700	114809.3	0.31	114708	114454
X-n783-k48	73765.3	1.89	73512	73043.8	0.90	72974	72637.3	0.34	72544	72790.7	0.55	72704	72629.6	0.33	72528	72649.2	0.35	72560	72394
X-n801-k40	74141.6	1.14	73970	73590.5	0.39	73500	73412.0	0.15	73362	73500.4	0.27	73396	73433.2	0.17	73338	73420.2	0.16	73367	73305
X-n819-k171	159363.2	0.79	159261	159572.5	0.92	159396	158424.5	0.19	158344	158511.6	0.25	158391	158513.6	0.25	158405	158367.2	0.16	158249	158121
X-n837-k142	195053.8	0.68	194857	195135.0	0.72	194988	193946.6	0.11	193868	194231.3	0.26	194103	194109.5	0.19	194017	193961.4	0.12	193767	193737
X-n856-k95	89266.2	0.34	89082	89333.5	0.41	89218	89111.1	0.16	89042	89037.5	0.08	88986	89073.8	0.12	89008	89067.3	0.11	89033	88965

Table 12 continued

Instance	HILS(2013)			KGLS(2018)			SISR(2020)			HGS(2020)			HGRRR1(2021)			HGRRR2(2021)			BKS
	Avg	Gap	Best	Avg	Gap	Best	Avg	Gap	Best	Avg	Gap	Best	Avg	Gap	Best	Avg	Gap	Best	
X-n876-k59	100487.3	1.20	100418	100115.7	0.82	100048	99484.5	0.19	99405	99682.7	0.39	99596	99533.3	0.24	99487	99490.9	0.19	99421	99299
X-n895-k37	55023.1	2.16	54856	54306.0	0.83	54240	54072.3	0.39	53982	54070.6	0.39	54023	54071.9	0.39	54011	54090.2	0.43	53982	53860
X-n916-k207	331158.4	0.60	330920	331111.0	0.59	331006	329584.3	0.12	329418	329852.0	0.20	329572	329760.3	0.18	329582	329503.7	0.10	329420	329179
X-n936-k151	135052.1	1.75	134732	133831.4	0.83	133713	133497.1	0.58	133190	133369.9	0.49	133121	133238.4	0.39	133064	133198.0	0.36	133102	132725
X-n957-k87	85979.8	0.60	85864	85746.6	0.33	85656	85559.8	0.11	85493	85550.1	0.10	85506	85583.3	0.14	85543	85561.2	0.11	85518	85465
X-n979-k58	120569.7	1.33	120142	119600.1	0.52	119559	119108.2	0.10	119065	119247.5	0.22	119180	119197.5	0.18	119056	119139.1	0.13	119055	118987
X-n1001-k43	74158.5	2.49	73749	72998.9	0.88	72882	72533.1	0.24	72414	72748.8	0.54	72678	72551.3	0.27	72411	72516.3	0.22	72414	72359
Min	0.0000		0.0000		0.0000		0.0000		0.0000		0.0000		0.0000		0.0000		0.0000		
Gap																			
Avg	0.6570			0.5329			0.1937		0.1084			0.1034					0.0962		
Gap																			
Max	2.4869			1.2366			1.6319		0.5480			0.3934					0.4426		
Gap																			
Nb BKS			18			6			21		44				41				41

Table 13 Comparison of the best and the average z^{AVG} (1 of 3)

Instance	HILS(2013)			KGLS(2018)			SISR(2020)			HGS(2020)			HGSRRL(2021)			HGSRR2(2021)			BKS	
	Avg	Gap	Best	Avg	Gap	Best	Avg	Gap	Best	Avg	Gap	Best	Avg	Gap	Best	Avg	Gap	Best		
X-n101-k25	27591.2	0.00	27591.0	27645.7	0.20	27634.5	27614.3	0.08	27609.0	27591.0	0.00	27591.0	27591.4	0.00	27591.0	27591.4	0.00	27591.0	27591.0	27591.0
X-n106-k14	26395.6	0.13	26387.9	26417.6	0.21	26376.8	26438.5	0.29	26431.6	26389.2	0.10	26374.9	26372.9	0.04	26364.9	26373.7	0.04	26363.7	26362	26362
X-n110-k13	14971.0	0.00	14971.0	14971.5	0.00	14971.0	14982.2	0.07	14976.9	14971.0	0.00	14971.0	14971.1	0.00	14971.0	14971.2	0.00	14971.0	14971.0	14971.0
X-n115-k10	12747.1	0.00	12747.0	12748.7	0.01	12747.0	12747.3	0.00	12747.1	12747.0	0.00	12747.0	12747.0	0.00	12747.0	12747.0	0.00	12747.0	12747.0	12747
X-n120-k6	13337.1	0.04	13332.0	13332.8	0.01	13332.0	13340.7	0.07	13337.9	13332.1	0.00	13332.0	13332.0	0.00	13332.0	13332.0	0.00	13332.0	13332.0	13332
X-n125-k30	55903.3	0.66	55794.0	55755.1	0.39	55670.7	55652.5	0.20	55631.7	55540.1	0.00	55539.4	55544.0	0.01	55540.1	55552.0	0.02	55548.8	55539	55539
X-n129-k18	28982.7	0.15	28956.0	28974.9	0.12	28956.1	28976.7	0.13	28966.5	28940.4	0.00	28940.1	28940.5	0.00	28940.0	28941.1	0.00	28940.3	28940	28940
X-n134-k13	10954.2	0.35	10930.8	10942.0	0.24	10931.1	11013.6	0.89	11005.8	10917.0	0.01	10916.5	10918.0	0.02	10916.0	10921.6	0.05	10918.1	10916	10916
X-n139-k10	13592.8	0.02	13590.3	13594.9	0.04	13590.2	13607.3	0.13	13601.6	13590.0	0.00	13590.0	13590.3	0.00	13590.0	13590.9	0.01	13590.3	13590	13590
X-n143-k7	15745.0	0.29	15723.0	15740.0	0.25	15729.5	15733.1	0.21	15711.4	15700.3	0.00	15700.0	15702.0	0.01	15700.6	15702.7	0.02	15700.7	15700	15700
X-n148-k46	43464.2	0.04	43450.4	43593.9	0.34	43515.0	43559.3	0.26	43537.5	43448.4	0.00	43448.0	43449.8	0.00	43448.0	43456.3	0.02	43449.9	43448	43448

Table 13 continued

Instance	HILLS(2013)			KGLS(2018)			SISR(2020)			HGS(2020)			HGSRRL(2021)			HGSRR2(2021)			BKS
	Avg	Gap	Best	Avg	Gap	Best	Avg	Gap	Best	Avg	Gap	Best	Avg	Gap	Best	Avg	Gap	Best	
X-n153-k22	21456.4	1.11	21228.3	21392.7	0.81	21376.8	21293.9	0.35	21286.8	21226.1	0.03	21225.3	21230.0	0.05	21225.2	21230.3	0.05	21228.8	21220
X-n157-k13	16876.1	0.00	16876.0	16881.0	0.03	16876.1	16952.0	0.45	16946.0	16876.0	0.00	16876.0	16876.1	0.00	16876.0	16878.6	0.02	16878.0	16876
X-n162-k11	14153.0	0.11	14138.8	14148.0	0.07	14147.0	14181.0	0.30	14163.3	14138.0	0.00	14138.0	14138.4	0.00	14138.0	14141.4	0.02	14139.5	14138
X-n167-k10	20609.2	0.25	20564.5	20588.0	0.15	20557.8	20612.7	0.27	20608.4	20557.9	0.00	20557.0	20558.0	0.01	20557.3	20559.6	0.01	20558.0	20557
X-n172-k51	45666.3	0.13	45607.0	45810.5	0.45	45767.5	45766.4	0.35	45739.4	45607.2	0.00	45607.0	45609.5	0.01	45607.2	45613.0	0.01	45609.7	45607
X-n176-k26	48255.0	0.93	48184.3	48036.1	0.47	48008.4	47916.8	0.22	47898.9	47815.3	0.01	47813.3	47825.6	0.03	47815.8	47836.1	0.05	47817.0	47812
X-n181-k23	25573.4	0.02	25569.5	25607.9	0.15	25601.1	25701.6	0.52	25697.0	25570.3	0.01	25569.1	25570.7	0.01	25569.1	25576.9	0.03	25575.1	25569
X-n186-k15	24181.7	0.15	24158.3	24193.4	0.20	24166.1	24218.3	0.30	24201.5	24145.2	0.00	24145.0	24146.1	0.00	24145.1	24149.5	0.02	24147.0	24145
X-n190-k8	17122.1	0.84	17059.3	17046.7	0.39	17010.5	17110.7	0.77	17103.8	16992.0	0.07	16989.0	16984.9	0.03	16980.2	16989.8	0.06	16986.9	16980
X-n195-k51	44306.8	0.18	44225.9	44440.0	0.49	44412.8	44421.5	0.44	44377.1	44231.3	0.01	44225.1	44229.1	0.01	44225.7	44250.4	0.06	44232.4	44225
X-n200-k36	58944.8	0.63	58631.8	58988.7	0.70	58918.3	58874.4	0.51	58830.9	58589.8	0.02	58580.7	58603.4	0.04	58580.4	58619.3	0.07	58588.5	58578
X-n204-k19	19623.2	0.30	19574.6	19628.2	0.32	19582.0	19744.1	0.92	19684.8	19565.4	0.00	19565.1	19568.4	0.02	19567.2	19582.5	0.09	19570.4	19565
X-n209-k16	30785.7	0.42	30726.9	30716.5	0.20	30686.0	30774.0	0.38	30766.4	30661.7	0.02	30658.9	30659.3	0.01	30656.1	30662.3	0.02	30659.1	30656

Table 13 continued

Instance	HILS(2013)			KGLS(2018)			SISR(2020)			HGS(2020)			HGSRR1(2021)			HGSRR2(2021)			BKS
	Avg	Gap	Best	Avg	Gap	Best	Avg	Gap	Best	Avg	Gap	Best	Avg	Gap	Best	Avg	Gap	Best	
X-n214-k11	11088.2	2.14	10938.9	10958.5	0.94	10914.1	11026.7	1.57	11005.7	10867.2	0.10	10860.4	10868.9	0.12	10860.7	10884.6	0.26	10879.0	10856
X-n219-k73	117595.0	0.00	117595.0	117702.5	0.09	117681.8	117764.8	0.14	117750.6	117599.5	0.00	117596.8	117599.5	0.00	117595.0	117610.8	0.01	117604.8	117595
X-n223-k34	40566.0	0.32	40508.5	40724.4	0.71	40690.6	40723.6	0.71	40694.9	40444.1	0.02	40439.8	40450.5	0.03	40444.1	40484.0	0.12	40455.9	40437
X-n228-k23	25818.3	0.30	25757.5	25840.0	0.38	25808.9	25943.8	0.78	25918.4	25744.1	0.01	25742.5	25754.2	0.05	25743.6	25771.2	0.11	25760.7	25742
X-n233-k16	19312.8	0.43	19281.7	19335.8	0.55	19293.8	19375.7	0.76	19343.7	19232.5	0.01	19230.6	19239.0	0.05	19230.0	19248.9	0.10	19237.3	19230
X-n237-k14	27085.6	0.16	27052.7	27102.0	0.22	27051.1	27199.7	0.58	27162.4	27044.1	0.01	27042.2	27052.2	0.04	27045.0	27066.5	0.09	27048.9	27042
X-n242-k48	82966.9	0.26	82862.4	83230.0	0.58	83168.2	83136.0	0.47	83079.4	82843.4	0.11	82798.4	82849.5	0.12	82753.6	82832.0	0.10	82790.8	82751
X-n247-k50	37546.7	0.73	37382.6	37417.8	0.39	37354.4	37473.2	0.53	37400.8	37301.7	0.07	37279.0	37291.4	0.05	37276.0	37317.2	0.12	37288.1	37274
X-n251-k28	38873.7	0.49	38819.1	38904.4	0.57	38851.1	39012.9	0.85	38951.5	38717.5	0.09	38693.8	38708.3	0.06	38687.2	38754.3	0.18	38714.1	38684
X-n256-k16	18882.2	0.23	18880.2	18901.9	0.33	18890.3	18961.4	0.65	18951.7	18849.5	0.06	18839.7	18880.5	0.22	18880.1	18889.7	0.27	18885.2	18839
X-n261-k13	26865.4	1.16	26752.5	26755.5	0.74	26693.2	26762.2	0.77	26729.6	26579.2	0.08	26566.4	26584.7	0.10	26569.5	26601.4	0.16	26590.5	26558

Table 14 Comparison of the best and the average z^{AVG} (2 of 3)

Instance	HILLS(2013)		KGLS(2018)		SISR(2020)		HGS(2020)		HGSR1(2021)		HGSR2(2021)		BKS						
	Avg	Gap Best	Avg	Gap Best	Avg	Gap Best	Avg	Gap Best	Avg	Gap Best	Avg	Gap Best							
X-n266-k58	75695.8	0.29	75520.0	75983.4	0.67	75809.5	75909.1	0.57	75859.2	75639.1	0.21	75567.4	75619.5	0.19	75572.5	75478			
X-n270-k35	35371.1	0.23	35347.3	35470.0	0.51	35447.8	35595.4	0.86	35566.5	35305.6	0.04	35304.3	35314.7	0.07	35306.4	35326.0	0.10	35313.7	35291
X-n275-k28	21268.3	0.11	21246.5	21311.5	0.31	21268.8	21489.8	1.15	21473.1	21246.0	0.00	21245.3	21246.2	0.01	21245.1	21263.3	0.09	21257.7	21245
X-n280-k17	33853.6	1.05	33741.7	33737.8	0.70	33653.2	33788.2	0.85	33714.5	33567.1	0.19	33540.4	33559.0	0.17	33531.9	33598.1	0.28	33578.7	33503
X-n284-k15	20441.6	1.12	20360.2	20384.1	0.84	20333.8	20550.2	1.66	20542.5	20268.5	0.26	20251.2	20258.8	0.22	20249.6	20280.5	0.32	20261.1	20215
X-n289-k60	95650.7	0.53	95483.9	95925.4	0.81	95832.2	95800.1	0.68	95729.6	95369.2	0.23	95307.5	95354.8	0.21	95295.9	95326.8	0.18	95248.4	95151
X-n294-k50	47288.1	0.27	47252.9	47464.3	0.64	47414.6	47538.9	0.80	47504.8	47199.2	0.08	47176.2	47220.3	0.13	47196.7	47257.1	0.20	47235.0	47161
X-n298-k31	34436.8	0.60	34382.3	34387.4	0.46	34366.9	34494.0	0.77	34468.2	34243.2	0.04	34232.3	34248.6	0.05	34234.1	34268.6	0.11	34240.6	34231
X-n303-k21	21924.1	0.87	21850.2	21916.2	0.83	21854.2	21989.7	1.17	21980.8	21757.4	0.10	21752.5	21770.2	0.16	21758.7	21781.0	0.21	21767.7	21736
X-n308-k13	26123.5	1.02	26014.5	26108.9	0.97	26046.0	26397.5	2.08	26345.4	25887.3	0.11	25872.4	25898.6	0.15	25883.2	25980.9	0.47	25923.0	25859
X-n313-k71	94374.9	0.35	94240.4	94799.3	0.80	94681.1	94651.3	0.65	94619.7	94162.4	0.13	94078.7	94122.5	0.08	94060.2	94185.4	0.15	94102.9	94043
X-n317-k53	78381.1	0.03	78355.8	78434.9	0.10	78419.0	78750.3	0.50	78729.7	78363.4	0.01	78357.6	78365.2	0.01	78358.3	78395.4	0.05	78384.8	78355
X-n322-k28	30027.2	0.65	29960.1	30053.8	0.74	30017.1	30162.9	1.10	30130.2	29858.9	0.08	29841.6	29878.0	0.15	29858.3	29887.6	0.18	29875.8	29834
X-n327-k20	27867.8	1.22	27798.0	27657.4	0.46	27626.0	27856.5	1.18	27827.1	27561.3	0.11	27549.7	27596.5	0.23	27579.0	27622.4	0.33	27599.1	27532
X-n331-k15	31271.5	0.54	31156.9	31240.7	0.45	31128.2	31343.4	0.78	31335.8	31112.7	0.03	31106.8	31111.2	0.03	31105.5	31126.2	0.08	31119.6	31102
X-n336-k84	139682.9	0.41	139387.0	140951.0	1.32	140857.1	139950.2	0.60	139815.0	139446.3	0.24	139345.2	139419.8	0.22	139238.7	139335.4	0.16	139241.6	139111
X-n344-k43	42361.2	0.74	42270.7	42368.4	0.76	42259.3	42519.7	1.12	42486.5	42103.6	0.13	42081.1	42094.2	0.11	42067.1	42118.9	0.16	42094.6	42050
X-n351-k40	26161.9	1.03	26065.5	26201.4	1.18	26153.1	26413.9	2.00	26396.0	25961.1	0.25	25944.5	25950.8	0.21	25941.5	25997.2	0.39	25980.2	25896
X-n359-k29	52145.5	1.24	51946.3	51923.7	0.81	51689.2	51951.0	0.87	51916.5	51699.8	0.38	51643.5	51633.3	0.25	51557.3	51563.1	0.11	51533.9	51505

Table 14 continued

Instance	HILS(2013)		KGLS(2018)		SISR(2020)		HGS(2020)		HGSRR(2021)		HGSRR2(2021)		BKS						
	Avg	Gap Best	Avg	Gap Best	Avg	Gap Best	Avg	Gap Best	Avg	Gap Best	Avg	Gap Best							
X-n367-k17	23032.4	0.96	22999.6	23034.0	0.96	22898.1	23054.3	1.05	23037.4	22817.0	0.01	22815.8	22824.6	0.05	22816.1	22853.0	0.17	22846.2	22814
X-n376-k94	147714.1	0.00	147713.2	147882.4	0.11	147826.6	148200.3	0.33	148168.0	147731.6	0.01	147721.3	147732.1	0.01	147723.0	147776.8	0.04	147764.8	147713
X-n384-k52	66483.3	0.82	66400.2	66452.5	0.78	66369.1	66607.1	1.01	66544.1	66098.8	0.24	66062.1	66067.9	0.19	66022.1	66091.8	0.23	66007.8	65940
X-n393-k38	38557.7	0.78	38454.2	38483.9	0.59	38458.2	38815.9	1.45	38774.8	38270.0	0.03	38263.5	38293.1	0.09	38267.6	38348.5	0.23	38315.2	38260
X-n401-k29	66849.7	1.04	66695.0	66522.0	0.54	66501.5	66665.3	0.76	66630.6	66291.1	0.19	66258.4	66244.4	0.12	66209.9	66276.2	0.17	66242.8	66163
X-n411-k19	20019.0	1.56	19938.8	19970.2	1.31	19803.8	20025.9	1.59	20004.3	19731.6	0.10	19723.0	19770.8	0.30	19758.6	19806.1	0.48	19785.0	19712
X-n420-k130	107909.3	0.10	107859.6	108317.4	0.48	108202.7	108601.0	0.74	108547.7	107877.0	0.07	107836.6	107854.8	0.05	107819.2	107908.0	0.10	107874.4	107798
X-n429-k61	65878.3	0.66	65750.1	65878.5	0.66	65815.7	66143.4	1.06	66113.2	65529.1	0.12	65509.5	65534.7	0.13	65499.7	65566.2	0.18	65502.4	65449
X-n439-k37	36476.5	0.23	36435.1	36507.0	0.32	36458.1	36832.4	1.21	36780.7	36400.1	0.02	36395.2	36409.2	0.05	36398.3	36449.6	0.16	36431.2	36391
X-n449-k29	56337.6	2.00	56257.2	55812.9	1.05	55727.6	55962.2	1.32	55883.6	55460.9	0.41	55381.6	55397.2	0.30	55353.7	55420.6	0.34	55342.6	55233
X-n459-k26	24524.9	1.60	24476.8	24309.5	0.71	24245.7	24699.7	2.32	24647.2	24180.2	0.17	24165.4	24189.0	0.21	24175.5	24252.8	0.47	24243.5	24139
X-n469-k138	222358.9	0.24	222047.8	223634.8	0.82	223318.4	223080.8	0.57	222919.1	222436.6	0.28	222195.2	222329.2	0.23	222124.7	222164.3	0.15	222029.7	221824
X-n480-k70	89953.0	0.56	89764.3	90005.4	0.62	89944.3	90367.3	1.03	90306.6	89581.9	0.15	89530.0	89587.2	0.15	89475.2	89609.3	0.18	89550.2	89449
X-n491-k59	67332.5	1.27	67146.4	67182.1	1.05	67049.4	67369.9	1.33	67290.2	66717.7	0.35	66611.3	66671.2	0.28	66619.9	66705.5	0.33	66652.3	66487
X-n502-k39	69396.0	0.25	69366.4	69347.9	0.18	69321.4	69821.6	0.86	69790.4	69255.2	0.04	69236.4	69269.2	0.06	69249.9	69314.1	0.13	69298.6	69226
X-n513-k21	24463.5	1.08	24338.2	24391.3	0.79	24313.5	24539.9	1.40	24503.5	24210.6	0.04	24203.2	24229.7	0.12	24211.6	24293.5	0.38	24274.7	24201

Table 15 Comparison of the best and the average z^{AVG} (3 of 3)

Instance	HILS(2013)		KGLS(2018)		SISR(2020)		HGS(2020)		HGSRR1(2021)		HGSRR2(2021)		BKS						
	Avg	Gap Best	Avg	Gap Best	Avg	Gap Best	Avg	Gap Best	Avg	Gap Best	Avg	Gap Best							
X-n524-k153	155489.5	0.58	155028.1	1.55810.8	0.79	155609.8	155103.7	0.33	155003.3	154830.6	0.15	154714.4	154733.4	0.09	154693.7	154824.6	0.15	154752.1	154593
X-n536-k96	95800.5	0.98	95732.5	95885.6	1.07	95790.0	96360.3	1.57	96322.7	95162.0	0.31	95099.8	95210.8	0.36	95101.5	95227.6	0.38	95138.1	94868
X-n548-k50	87038.8	0.39	86862.5	86977.9	0.32	86933.0	87555.5	0.99	87509.1	86826.6	0.15	86789.1	86777.7	0.09	86732.6	86817.8	0.14	86787.6	86700
X-n561-k42	43173.0	1.07	43062.8	43063.3	0.81	43040.8	43443.1	1.70	43402.5	42766.2	0.12	42748.0	42801.8	0.20	42770.0	42890.4	0.41	42861.3	42717
X-n573-k30	51231.1	1.10	51127.9	51041.1	0.73	50890.4	51253.1	1.14	51216.4	50861.8	0.37	50821.4	50845.1	0.34	50809.1	50935.8	0.52	50847.9	50673
X-n586-k159	191083.4	0.40	190779.6	191509.5	0.63	191370.1	191757.9	0.76	191625.2	190741.7	0.22	190627.1	190735.8	0.22	190650.8	190780.7	0.24	190673.0	190316
X-n599-k92	109540.9	1.00	109358.5	109401.3	0.88	109158.4	109723.7	1.17	109660.6	108752.9	0.28	108686.7	108674.9	0.21	108581.9	108700.3	0.23	108647.9	108451
X-n613-k62	60505.3	1.63	60340.5	60239.5	1.18	60141.1	60623.1	1.83	60550.3	59761.4	0.38	59707.5	59693.6	0.27	59606.8	59770.8	0.40	59723.5	59535
X-n627-k43	63111.4	1.52	63022.1	62609.8	0.72	62553.2	63240.2	1.73	63183.8	62487.9	0.52	62390.6	62364.3	0.32	62320.3	62419.3	0.41	62388.9	62164
X-n641-k35	64768.3	1.69	64608.6	64143.3	0.71	64013.2	64719.5	1.61	64669.9	63985.1	0.46	63894.9	63900.6	0.32	63801.5	63941.2	0.39	63912.5	63694
X-n655-k131	106808.0	0.03	106791.1	106965.0	0.17	106944.9	107901.8	1.05	107881.6	106827.5	0.04	106812.0	106854.6	0.07	106841.1	106922.6	0.13	106909.6	106780
X-n670-k130	149206.5	1.96	148919.2	147699.5	0.93	147648.5	147569.1	0.85	147315.8	146904.6	0.39	146705.6	147041.5	0.48	146659.7	147308.6	0.67	147116.9	146332
X-n685-k75	69116.4	1.34	69018.4	68928.7	1.06	68827.5	69379.2	1.72	69269.4	68431.7	0.33	68388.7	68431.5	0.33	68372.9	68522.4	0.47	68433.3	68205

Table 15 continued

Instance	HLS(2013)			KGLS(2018)			SISR(2020)			HGS(2020)			HGSRR1(2021)			HGSRR2(2021)			BKS
	Avg	Gap	Best	Avg	Gap	Best	Avg	Gap	Best	Avg	Gap	Best	Avg	Gap	Best	Avg	Gap	Best	
X-n701-k44	83264.1	1.64	83083.9	82587.2	0.81	82482.6	83128.6	1.47	83071.1	82408.0	0.59	82245.2	82179.1	0.31	82064.9	82189.4	0.33	82100.9	81923
X-n716-k35	44390.7	2.31	44217.0	43812.9	0.98	43716.5	44411.3	2.36	44377.9	43575.8	0.44	43514.1	43568.4	0.42	43529.7	43674.9	0.66	43638.6	43387
X-n733-k159	137171.9	0.72	137060.6	137376.0	0.87	137255.0	138000.7	1.33	137916.0	136503.0	0.23	136412.0	136439.2	0.18	136388.7	136572.1	0.28	136517.4	136190
X-n749-k98	78389.8	1.39	78322.7	78260.8	1.22	78140.0	79088.3	2.29	79005.9	77805.4	0.64	77698.9	77611.8	0.39	77537.2	77762.7	0.58	77600.1	77314
X-n766-k71	116168.7	1.50	115679.2	115440.2	0.86	115135.8	115919.6	1.28	115842.9	114878.0	0.37	114773.2	114880.8	0.37	114780.2	115028.3	0.50	114913.8	114454
X-n783-k48	73915.6	2.10	73691.5	73105.3	0.98	73045.5	73776.8	1.91	73692.8	72986.8	0.82	72897.1	72707.8	0.43	72566.3	72880.4	0.67	72777.5	72394
X-n801-k40	74287.1	1.34	74216.9	73688.6	0.52	73638.1	74549.5	1.70	74512.8	73637.4	0.45	73494.6	73473.4	0.23	73395.0	73544.7	0.33	73504.3	73305
X-n819-k171	159515.2	0.88	159398.1	159664.2	0.98	159523.6	160385.4	1.43	160327.1	158647.5	0.33	158528.6	158589.5	0.30	158498.2	158619.0	0.31	158504.6	158121
X-n837-k142	195170.1	0.74	195039.9	195230.8	0.77	195069.2	195867.8	1.10	195806.6	194538.0	0.41	194404.9	194173.9	0.23	194071.5	194199.6	0.24	193994.7	193737
X-n856-k95	89340.5	0.42	89189.8	89363.3	0.45	89268.9	90602.0	1.84	90552.9	89068.5	0.12	89008.6	89108.1	0.16	89042.9	89225.7	0.29	89172.6	88965
X-n876-k59	100625.8	1.34	100564.2	100160.4	0.87	100105.2	100966.7	1.68	100887.0	99862.2	0.57	99772.8	99598.4	0.30	99534.3	99820.9	0.53	99753.3	99299

Table 15 continued

Instance	HILS(2013)		KGLS(2018)		SISR(2020)		HGS(2020)		HGSRR1(2021)		HGSRR2(2021)		BKS						
	Avg	Gap Best	Avg	Gap Best	Avg	Gap Best	Avg	Gap Best	Avg	Gap Best	Avg	Gap Best							
X-n895-k37	55117.0	2.33	54870.7	54377.6	0.96	54323.4	55218.3	2.52	55141.6	54174.7	0.58	54097.9	54140.4	0.52	54090.5	54351.3	0.91	54255.2	53860
X-n916-k207	331297.4	0.64	330970.3	331397.0	0.67	331265.6	331912.2	0.83	331808.6	330584.5	0.43	330353.4	329845.2	0.20	329663.4	329831.5	0.20	329761.0	329179
X-n936-k151	136754.6	3.04	136392.6	134103.7	1.04	133974.8	134427.7	1.28	134164.0	133479.0	0.57	133173.3	133393.2	0.50	133165.7	133540.0	0.61	133439.8	132725
X-n957-k87	86049.0	0.68	85933.5	85837.8	0.44	85769.5	87116.2	1.93	87086.6	85614.2	0.17	85579.3	85627.7	0.19	85578.5	85761.1	0.35	85697.9	85465
X-n979-k58	120855.8	1.57	120324.3	119687.6	0.59	119652.5	120786.5	1.51	120723.1	119582.6	0.50	119489.2	119270.5	0.24	119142.9	119511.2	0.44	119427.3	118987
X-n1001-k43	74321.7	2.71	73856.8	73113.0	1.04	72984.1	74094.9	2.40	74041.9	72986.4	0.87	72837.1	72621.9	0.36	72474.1	72863.6	0.70	72759.9	72359
Min Gap	0.0000		0.0000		0.0000		0.0000		0.0000		0.0000		0.0000		0.0000		0.0000		0.0000
Avg Gap	0.7746		0.6011		0.9870		0.1754		0.1480		0.1754		0.1480		0.1754		0.2221		0.2221
Max Gap	3.0360		1.3227		2.5219		0.8670		0.5207		0.8670		0.5207		0.8670		0.9122		0.9122

Appendix C: Implementation comparison

The Benchmark Test presented in Section 3.6 compares several metaheuristics implemented in different programming languages, using different compilers, and implemented by different programmers. Consequently, it is uncertain whether the relative difference in performance comes from methodological improvements, or simply from details in the implementation environment. Therefore, to mitigate this uncertainty, we compare our implementation of HGS in Rust against the original HGS in C++.

The convergence profiles of the two implementations are presented in Fig. 13, alongside HGSRR1 and HGSRR2 for reference. Table 16 shows the metrics of the two implementations, in addition to the corresponding p-values from the statistical tests described in Section 3.1. The results show that the two implementations have very similar convergence profiles, although the Rust implementation performs consistently and significantly worse. As a result, the relative improvement in performance coming from the additional components in HGSRR compared to HGS may be greater than seen in the Benchmark Test presented in Section 3.6. On the other hand, if the Rust implementation had performed significantly better than the original implementation, further tests would have been necessary to determine what part of the increase in performance, if any, that comes from the methodological improvement.

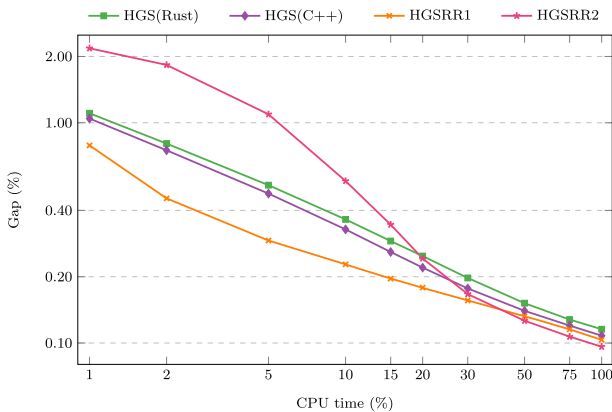


Fig. 13 Convergence profiles for the different metaheuristics in the Benchmark Test are shown on a logarithmic scale

Table 16 Metrics for the two implementations of HGS in a Benchmark Test

Instances	Metric	HGS(C++)	HGS(Rust)	H_0	H_1
Small	g^F	0.0231	0.0286	0.009879	0.995060
	g^{AVG}	0.0497	0.0581	0.000028	0.999986
Short Routes	g^F	0.1029	0.1148	0.000604	0.999698
	g^{AVG}	0.1585	0.1810	0.000000	1.000000
Large	g^F	0.1937	0.2023	0.011685	0.994158
	g^{AVG}	0.3011	0.3234	0.000000	1.000000
Long Routes	g^F	0.1301	0.1322	0.392317	0.803842
	g^{AVG}	0.2199	0.2301	0.009803	0.995099
All	g^F	0.1084	0.1155	0.000531	0.999734
	g^{AVG}	0.1754	0.1908	0.000000	1.000000

The values in bold are not statistically worse than the best performing implementation on the respective subset, in addition to the p-values of the rejected hypotheses

References

- Arnold, F., Sörensen, K.: Knowledge-guided local search for the vehicle routing problem. *Computers & Operations Research* **105**, 32–46 (2019)
- Arnold, F., Gendreau, M., Sörensen, K.: Efficiently solving very large-scale routing problems. *Computers & Operations Research* **107**, 32–42 (2019)
- Baxter, J.: Local optima avoidance in depot location. *Journal of the Operational Research Society* **32**(9), 815–819 (1981)
- Berthold, T.: Measuring the impact of primal heuristics. *Oper. Res. Lett.* **41**(6), 611–614 (2013)
- Christiaens, J., Vanden Berghe, G.: Slack Induction by String Removals for Vehicle Routing Problems. *Transp. Sci.* **54**(2), 417–433 (2020)
- CVRPLIB: Capacitated vehicle routing problem library. Retrieved from <http://vrp.atd-lab.inf.puc-rio.br/index.php/en/>. Accessed 8. July 2021 (2021)
- Dantzig, G.B., Ramser, J.H.: The Truck Dispatching Problem. *Manage. Sci.* **6**(1), 80–91 (1959)
- Eksioglu, B., Vural, A.V., Reisman, A.: The vehicle routing problem: A taxonomic review. *Computers & Industrial Engineering* **57**(4), 1472–1483 (2009)
- Glover, F.: Future paths for integer programming and links to artificial intelligence. *Computers & Operations Research* **13**(5), 533–549 (1986)
- Glover, F.: Tabu search-part i. *ORSA J. Comput.* **1**(3), 190–206 (1989)
- Kirkpatrick, S., Gelatt, C.D., Vecchi, M.P.: Optimization by Simulated Annealing. *Science* **220**(4598), 671–680 (1983)
- Laporte, G.: Fifty years of vehicle routing. *Transp. Sci.* **43**(4), 408–416 (2009)
- Laporte, G., Ropke, S., Vidal, T.: Chapter 4: Heuristics for the vehicle routing problem. In: *Vehicle Routing: Problems, Methods, and Applications*, pp. 87–116. SIAM, Second Edition (2014)
- Lenstra, J.K., Kan, A.R.: Complexity of vehicle routing and scheduling problems. *Networks* **11**(2), 221–227 (1981)
- Oliver, I., Smith, D., Holland, J.R.: Study of permutation crossover operators on the traveling salesman problem. In: *Genetic algorithms and their applications: proceedings of the second International Conference on Genetic Algorithms: July 28-31, 1987 at the Massachusetts Institute of Technology, Cambridge, MA, Hillsdale, NJ: L. Erlbaum Associates, 1987* (1987)
- PassMark Software: CPU benchmarks. Retrieved from <https://www.cpubenchmark.net/>. Accessed 13. April 2021 (2021)
- Pecin, D., Pessoa, A., Poggi, M., Uchoa, E.: Improved branch-cut-and-price for capacitated vehicle routing. *Math. Program. Comput.* **9**(1), 61–100 (2017)

- Pessoa, A., Sadykov, R., Uchoa, E., Vanderbeck, F.: A generic exact solver for vehicle routing and related problems. *Math. Program.* **183**(1), 483–523 (2020)
- Prins, C.: A simple and effective evolutionary algorithm for the vehicle routing problem. *Computers & Operations Research* **31**(12), 1985–2002 (2004)
- Subramanian, A., Uchoa, E., Ochi, L.S.: A hybrid algorithm for a class of vehicle routing problems. *Computers & Operations Research* **40**(10), 2519–2531 (2013)
- Toth, P., Vigo, D.: *Vehicle Routing: Problems, Methods, and Applications*. Society for Industrial and Applied Mathematics (2014)
- Uchoa, E., Pecin, D., Pessoa, A., Poggi, M., Vidal, T., Subramanian, A.: New benchmark instances for the Capacitated Vehicle Routing Problem. *Eur. J. Oper. Res.* **257**(3), 845–858 (2017)
- Vidal, T.: Split algorithm in $O(n)$ for the capacitated vehicle routing problem. *Computers & Operations Research* **69**, 40–47 (2016)
- Vidal, T.: Hybrid Genetic Search for the CVRP: Open-Source Implementation and SWAP* Neighborhood. arXiv preprint [arXiv:2012.10384](https://arxiv.org/abs/2012.10384) (2020)
- Vidal, T., Crainic, T.G., Gendreau, M., Lahrichi, N., Rei, W.: A hybrid Genetic Algorithm for Multidepot and Periodic Vehicle Routing Problems. *Oper. Res.* **60**(3), 611–624 (2012)
- Vidal, T., Crainic, T.G., Gendreau, M., Prins, C.: Heuristics for multi-attribute vehicle routing problems: A survey and synthesis. *Eur. J. Oper. Res.* **231**(1), 1–21 (2013)
- Voudouris, C., Tsang, E.P.: Guided Local Search. In: *Handbook of Metaheuristics*, Springer, pp 185–218 (2003)
- Wilcoxon, F.: Individual comparisons by ranking methods. *biometrics bulletin* **1**, 6: 80–83. URL <http://www.jstor.org/stable/3001968> (1945)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.