Submitted to *INFORMS Journal on Computing*
manuscript (Please, provide the manuscript number!)

A Hybrid Genetic Algorithm with Multi-population for Capacitated Location Routing

Pengfei He
School of Automation and Key Laboratory of Measurement and Control of Complex Systems of Engineering, Ministry of Education, Southeast University, Nanjing 210096, China.
pengfeihe606@gmail.com

Jin-Kao Hao* (Corresponding author)
Department of Computer Science, LERIA, University of Angers, 2 Boulevard Lavoisier, 49045 Angers, France
jin-kao.hao@univ-angers.fr

Qinghua Wu* (Corresponding author)
School of Management, Huazhong University of Science and Technology, No. 1037, Luoyu Road, Wuhan, China.
qinghuawu1005@gmail.com

The capacitated location-routing problem involves determining the depots from a set of candidate capacitated depot locations and finding the required routes for a fleet of vehicles starting from and ending at the selected depots to serve a set of customers such that the solution minimizes a cost function that includes the cost of opening the selected depots, the fixed utilization cost per vehicle used, and the total cost (distance) of the routes. This paper presents a hybrid genetic algorithm with multi-population, which includes an effective multi-depot edge assembly crossover to generate promising offspring from the perspective of both depot location and route edge assembly, a neighborhood-based local search to optimize the routes of each offspring solution, and a diversification-oriented mutation. Of particular interest is the multi-population scheme which organizes the population into multiple subpopulations according to the depot configurations being explored. Extensive experiments on four sets of 281 benchmark instances from the literature show that the algorithm performs remarkably well. Additional experiments are presented to gain insight into the role of the key elements of the algorithm.

Key words: Location-routing; Multi-population based search; Multi-depot edge assembly crossover; Neighborhood search; Depot configurations.

1. Introduction

Let $\mathcal{I}=\{1,\cdots,m\}$ be a set of depot locations (or depots), where each depot $i\in\mathcal{I}$ is associated with a positive capacity w_i and an opening cost o_i. Each depot has an unlimited fleet of vehicles. Each vehicle has a limited capacity Q and a fixed utilization cost F. Let $\mathcal{J}=\{1,\cdots,n\}$ be a set of customers, where each customer has a positive demand d_j. The capacitated location-routing problem (CLRP) is defined on a weighted and directed graph $\mathcal{G}=(\mathcal{V},\mathcal{A})$ with vertex set $\mathcal{V}=\mathcal{I}\cup\mathcal{J}$ and arc set $\mathcal{A}=\{(i,j)|i\in\mathcal{I},j\in\mathcal{J}\}\cup\{(i,j)|i,j\in\mathcal{J},i\neq j\}\cup\{(i,j)|i\in\mathcal{J},j\in\mathcal{I}\}$. $\mathcal{A}$ is associated with a matrix $\mathcal{C}=(c_{ij})$ where c_{ij} is a non-negative value representing the distance on the arc (i,j). $\mathcal{C}$ is said to be symmetric if $c_{ij}=c_{ji}$, for any $(i,j)\in\mathcal{A}$ and asymmetric otherwise. Let $\mathcal{E}$ be an edge set and $\mathcal{E}=\mathcal{A}$ if $\mathcal{G}$ is undirected. The CLRP problem involves determining the depots to open and the routes for a fleet of vehicles starting from and ending at these depots to visit all customers under the following constraints: (i) vehicle and depot capacities are respected; (ii) each vehicle ends at the depot from which it originated; (iii) each customer is visited exactly once. The objective of the CLRP is to minimize the total cost, including the cost of opening the chosen depots, the fixed utilization cost per vehicle used, and the total cost (distance) of the routes.

The CLRP covers a variety of problems that arise in single and two-echelon distribution networks in urban logistics (Prodhon and Prins 2014). On the one hand, the CLRP reduces to the multi-depot vehicle routing problem (MDVRP) (Cordeau et al. 1997) once the opening depots are fixed. On the other hand, locations in the CLRP correspond to satellites in the two-echelon vehicle routing problem (2E-VRP) (Perboli et al. 2011). The relationship between these three problems has been analyzed in Schneider and Löffler (2019) and Voigt et al. (2022). Moreover, the CLRP represents a class of well-studied vehicle routing problems (VRPs) in which decisions about depots and routes are associated. Many variations of the

CLRP arise in freight distribution and urban logistics with the addition of specific constraints that model practical scenarios (Drexl and Schneider 2015, Mara et al. 2021). See Baldacci et al. (2011) for a mathematical formulation of the CLRP.

Given the computational challenge and practical relevance of the CLRP (Nagy and Salhi 2007), much effort has been devoted to the development of efficient solution methods to better solve the problem. The three most representative exact algorithms (Baldacci et al. 2011, Contardo et al. 2014a, Liguori et al. 2023), based on the branch-and-cut-and-price framework, are able to optimally solve instances with up to 200 customers and 10 depots at the cost of high computational times (e.g., up to 176 hours for instances with up to 150/199 customers and 14 depots (Contardo et al. 2014a)). To handle larger instances, a number of heuristic algorithms have been proposed that aim to find high-quality solutions in a reasonable amount of time.

In this work, we propose an effective heuristic algorithm called hybrid genetic algorithm with multi-population (HGAMP) to advance the state of the art for better solving large CLRP instances by providing an original framework to effectively implement the integrated approach. Thanks to this algorithm, we report 103 new upper bounds on the CLRP benchmark instances in the literature.

We summarize the main contributions as follows.

- The proposed algorithm uses an innovative multi-population scheme where each subpopulation contains a set of high-quality (or elite) solutions that share the same depot configuration being explored. This multi-population scheme allows the algorithm to conveniently and simultaneously manage multiple promising depot configurations and multiple solutions associated with these depot configurations, allowing for a better exploration of the search space. Moreover, the multi-population scheme is general and can be applied to other related two-level decision problems.

- The proposed algorithm integrates an effective multi-depot edge assembly crossover, which enables the generation of promising offspring solutions with new depot configurations by recombining route edges associated with existing depot configurations. Combined with a feasibility-restoring procedure and a diversification-oriented mutation, this crossover proves to be highly useful for solving the CLRP. Moreover, the crossover can be applied to solve other related routing problems with little or no modification, as we show by applying the same crossover to solve the MDVRP.
- The improved best results (updated upper bounds) for 103 instances discovered by the proposed algorithm are useful for future research on the CLRP. In addition, the publicly available source codes of the algorithm can contribute to the practical solution of related real-life applications that can be formulated as the CLRP.

The rest of the paper is organized as follows. Section 2 reviews the related studies in the literature. Section 3 describes the proposed HGAMP algorithm. Section 4 evaluates the performance of HGAMP. Section 5 presents an analysis to gain insight into the roles of the main algorithmic components. The last section summarizes the contributions and provides concluding remarks.

2. Literature review

This section reviews the representative heuristic algorithms for the CLRP. For a comprehensive presentation of the existing algorithms, the reader is referred to dedicated surveys (Prodhon and Prins 2014, Schneider and Drexl 2017).

We organize the overview according to two solution approaches: the *hierarchical approach* and the *integrated approach*. The *hierarchical approach* treats the CLRP in two sequential steps, where the first step identifies promising depot configurations and the second step solves the resulting MDVRP problems. The *integrated approach* treats the location decision and the

Table 1 Representative heuristic algorithms for the CLRP.

References	Matheuristics/Metaheuristics	Approach	Strategy		Capacity		Instances
			Divided	Simultaneous	Routes	Depots	
Tuzun and Burke (1999)	Two phase tabu search	Hierarchical	✓		✓	×	T
Prins et al. (2006b)	GRASP + path relinking	Hierarchical	✓		✓	✓	T, P, B
Prins et al. (2006a)	Memetic algorithm	Integrated		✓	✓	✓	T, P, B
Prins et al. (2007)	Lagrangian relaxation and TS	Hierarchical	✓		✓	✓	T, P, B
Duhamel et al. (2010)	GRASP+Evolutionary LS	Integrated		✓	✓	✓	T, P, B
Vincent et al. (2010)	Simulated annealing	Integrated	✓		✓	✓	T, P, B
Hemmelmayr et al. (2012)	ALNS	Integrated	✓		✓	✓	T, P, B
Ting and Chen (2013)	Ant colony optimization	Hierarchical	✓		✓	✓	T, P, B
Escobar et al. (2013)	Two phase hybrid heuristic	Hierarchical	✓		✓	✓	T, P, B
Escobar et al. (2014)	Granular tabu search	Integrated	✓		✓	✓	T, P, B
Contardo et al. (2014b)	GRASP + ILP	Integrated	✓		✓	✓	T, P, B
Lopes et al. (2016)	Hybrid genetic algorithm	Integrated		✓	✓	✓	T, P, B
Quintero-Araujo et al. (2017)	Biased randomized	Hierarchical	✓		✓	✓	P, B
Schneider and Löffler (2019)	Tree-based search algorithm	Hierarchical	✓		✓	✓	T, P, B, S
Accorsi and Vigo (2020)	AVXS	Integrated	✓		✓	×	T
Akpunar and Akpinar (2021)	Hybrid ALNS	Integrated	✓		✓	✓	T, P, B
Arnold and Sörensen (2021)	Progressive filtering	Hierarchical	✓		✓	✓	T, P, B, S
Voigt et al. (2022)	HALNS	Integrated	✓		✓	✓	T, P, B
Sobhanan et al. (2024)	GA with neural cost predictor	Hierarchical		✓	✓	✓	T, B
This paper	Hybrid genetic algorithm	Mixed		✓	✓	✓	T, P, B, S

route optimization together. Table 1 summarizes the most representative CLRP heuristics. For each heuristic, we indicate the algorithmic framework (Matheuristics/Metaheuristics), the solution approach used (Hierarchical or Integrated), the strategy for generating depot configurations and routes, the presence of capacity constraints for routes and depots, and the benchmark sets tested (see Section 4.1 for these sets).

2.1. Hierarchical approach

The *hierarchical approach* decomposes the CLRP into two subproblems, the facility location problem (FLP) and the MDVRP, which are solved separately. This approach has the advantage of allowing a direct application of many effective routing algorithms to solve the MDVRP with a fixed depot configuration. However, it is a challenge to identify the best or most promising depot configurations, although some advanced strategies such as the progressive filtering heuristic (Arnold and Sörensen 2021) have been proposed.

Tuzun and Burke (1999) introduced a pioneering hierarchical algorithm. This algorithm uses a two-phase tabu search to make decisions: the first to determine the depot configuration and the second to optimize routes for the chosen depot configuration. Experiments on 36 instances showed significant performance improvements over reference methods.

Schneider and Löffler (2019) proposed a tree-based search algorithm that systematically explores depot configurations in a tree-like fashion. The algorithm alternates between a

location phase and a routing phase. First, it uses minimum spanning trees to estimate the approximated cost. Then, in the routing phase, it uses a dedicated tabu search algorithm to improve the MDVRP solution. The algorithm matched or improved on the vast majority of previous best-known solutions for the instances of the three classical sets, and demonstrated the ability to efficiently solve newly generated large-scale instances.

Arnold and Sörensen (2021) developed a progressive filtering heuristic. The algorithm begins by empirically estimating an upper bound on the depots and employs a heuristic construction procedure to significantly reduce the number of promising configurations. Then, a MDVRP heuristic is applied to evaluate each of these promising depot configurations. Experimental results showed that the algorithm outperformed existing heuristics, especially on the largest benchmark instances.

Sobhanan et al. (2024) proposed a deep learning based approach called genetic algorithm with neural cost predictor to solve hierarchical vehicle routing problems, including the CLRP. A pretrained graph neural network is used to learn the objective values of the open-source HGS-CVRP package for solving the CVRP. Experiments have shown that the algorithm can find good results quickly.

2.2. Integrated approach

The *integrated approach* considers the location decision and the routing problem simultaneously. Unlike the *hierarchical approach*, the integrated approach typically manipulates solutions associated with different depot configurations and improves the routes by reassigning customers and applying local optimization.

Prins et al. (2006a) introduced a memetic algorithm with population management. The algorithm employs a two-part chromosome encoding to represent each solution and uses the classical one-point crossover to generate offspring solutions combined with a customer

reassignment procedure. The algorithm focuses on route recombination of the parent solutions represented as two giant tours and customer reassignment. This pioneering algorithm produced interesting results on the benchmark instances tested.

Lopes et al. (2016) presented an evolutionary algorithm, which uses a route copy crossover to generate offspring solutions, allowing the discovery of new depot configurations during the crossover, and applies a local search procedure to reduce the routing cost. The algorithm reported competitive results on the benchmark instances tested.

Voigt et al. (2022) proposed a hybrid adaptive large neighborhood search algorithm (HALNS) that uses a population of solutions generated via an adaptive large neighborhood search algorithm (ALNS), similar to (Hemmelmayr et al. 2012). In particular, it uses ALNS, instead of a crossover operator, to combine two solutions, while preserving selected parts of those solutions. The algorithm explores depot configurations by removing, opening, and swapping depots during the destroy/repair process. The algorithm performed very well by finding new best solutions for three benchmark instances.

Our work is motivated by the following observations. First, the literature review shows that promising depot configurations play a critical role in both the hierarchical and integrated approaches. However, although several methods have been proposed for this purpose (Lopes et al. 2016, Schneider and Löffler 2019, Arnold and Sörensen 2021, Voigt et al. 2022), there is no proven method that can effectively recombine different depot configurations and assemble route edges simultaneously. Second, in terms of solution methodology, hybrid genetic algorithm (HGA) (also called memetic algorithm (Neri et al. 2012)) has shown excellent performance in solving various vehicle routing problems including multi-depot and periodic VRPs (Vidal et al. 2012), capacitated vehicle routing problem (CVRP) (Nagata and Bräysy 2009), and split delivery vehicle routing problem (SDVRP) (He and Hao 2023). The

HGA approach has also been applied to the CLRP (Prins et al. 2006a, Lopes et al. 2016) with competitive results. However, the crossover used in Prins et al. (2006a) only assembles route edges without considering depot configurations from parent solutions. The route copy crossover used in Lopes et al. (2016) focuses on transferring routes to offspring and overlooks the importance of constructing promising depot configurations from existing configurations.

In this work, we present an effective HGA with multi-population within the integrated approach for better solving the CLRP. The algorithm is characterized by its multi-population scheme, which organizes the population into subpopulations where each subpopulation is composed of a set of high-quality solutions using the *same* depot configuration. Although the idea of subpopulation is a well-known concept in population-based algorithms (Cochran et al. 2003, Toledo et al. 2013), this is the first time that the multi-population scheme is used to explicitly manage different depot configurations for solving the CLRP. In addition, the HGAMP algorithm uses a powerful multi-depot edge assembly crossover (mdEAX), which generalizes the crossovers in (Nagata and Bräysy 2009, He and Hao 2023), to generate new offspring solutions by mixing different depot configurations and assembling route edges simultaneously. The crossover, combined with a feasibility recovery procedure and a dedicated mutation, helps the algorithm to effectively explore the search space. Finally, the proposed algorithm integrates a mixed strategy to generate good initial depot configurations and applies classical routing heuristics to optimize the routes of offspring solutions.

3. Hybrid genetic algorithm with multi-population for the CLRP

This section first introduces the general approach of the HGAMP algorithm, and then presents its main components.

3.1. General HGAMP procedure

The proposed HGAMP algorithm follows the general hybrid genetic search or memetic search framework (Neri et al. 2012), which combines population-based search and neighborhood-

Algorithm 1: Main framework of the HAGMP algorithm

Input: Instance I;

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

1 **begin**
2 $\mathcal{D} = \{\mathcal{D}_1, \mathcal{D}_2, \cdots, \mathcal{D}_\gamma\} \leftarrow CRH(I)$; /* Search promising depot configurations, Section 3.2.1 */
3 $\mathcal{P} = \{\mathcal{P}_1, \mathcal{P}_2, \cdots, \mathcal{P}_\gamma, \mathcal{P}_{\gamma+1}\} \leftarrow PopInitial(\mathcal{D}, I)$ /* Initialize population $\mathcal{P}$, Section 3.2.2 */
4 $\varphi^* \leftarrow \arg\min\{f(\varphi) | \forall \varphi \in \mathcal{P}\}$; /* Record the best solution found so far */
5 **while** *Stopping condition is not met* **do**
6 $\{\varphi_A, \varphi_B\} \leftarrow ParentSelection(\mathcal{P})$; /* Select two parent solutions, Section 3.3 */
7 $\{\varphi_O^1, \varphi_O^2, \cdots, \varphi_O^\beta\} \leftarrow mdEAX(\varphi_A, \varphi_B)$; /* Generate offspring solutions, Section 3.3 */
8 **for** $i = 1$ *to* β **do**
9 $\varphi_O^i \leftarrow Repair(\varphi_O^i)$; /* Restore feasibility, Section 3.4 */
10 $\varphi_O^i \leftarrow Mutation(\varphi_O^i)$; /* Generate mutation, Section 3.5.1 */
11 $\varphi_O^i \leftarrow LocalSearch(\varphi_O^i)$; /* Improve the offspring solution, Section 3.5.2 */
12 **if** $f(\varphi_O^i) < f(\varphi^*)$ **then**
13 $\varphi^* \leftarrow \varphi_O^i$;
14 **end**
15 $\{\mathcal{P}, \mathcal{D}\} \leftarrow ManagingPop(\mathcal{P}, \mathcal{D}, \varphi_O^i, \varphi^*)$; /* Manage the population, Section 3.6 */
16 **end**
17 **end**
18 **return** φ^*;
19 **end**

based local optimization. Such a hybrid algorithm benefits from the complementary search strategies of the two combined methods and is expected to achieve a balance between exploitation and exploration of the given search space.

HGAMP, whose pseudo-code is illustrated in Algorithm 1, starts with an initial population of solutions, which is generated in two steps (lines 2 and 3). The first step uses a mixed strategy to find γ good and diverse initial depot configurations $\{\mathcal{D}_1, \mathcal{D}_2, \cdots, \mathcal{D}_\gamma\}$ (Section 3.2.1), while the second step uses each identified depot configuration $\mathcal{D}_i$ $(i = 1, \ldots, \gamma)$ to build a subpopulation $\mathcal{P}_i$ of solutions (Section 3.2.2). These γ subpopulations plus an additional

$\mathcal{P}_{\gamma+1}$ form the population $\mathcal{P}$. Then the algorithm evolves the population $\mathcal{P}$ over a number of generations using crossover, repair, mutation, local search, and population management (lines 5-17). In each generation, two solutions φ_A and φ_B are selected from two subpopulations (φ_A and φ_B necessarily have two different depot configurations) and serve as parent solutions for the mdEAX crossover. The crossover recombines route edges of φ_A and φ_B and the underlying depot configurations to generate β offspring solutions $\{\varphi_O^1, \varphi_O^2, \cdots, \varphi_O^\beta\}$ (Section 3.3). If an offspring solution violates capacity constraints, it is repaired to restore feasibility before being mutated and optimized by local search (Sections 3.4 and 3.5). Each offspring solution is then assigned to the subpopulation corresponding to its depot configuration (Section 3.6). If a depot configuration associated with an offspring solution differs from the γ promising depot configurations in the population, the solution is inserted into the subpopulation $\mathcal{P}_{\gamma+1}$. HGAMP stops and returns the best recorded solution φ^* when the predefined stopping condition is met (e.g., a maximum cutoff time or a maximum number of generations).

3.2. Initial population generation

This section illustrates the two-step process for generating the initial population with its subpopulations: a mixed strategy for the identification of initial depot configurations and the method for the generation of the corresponding subpopulations of solutions.

3.2.1. Mixed strategy for the identification of initial depot configurations. To identify a set of good and diverse initial depot configurations, we adopt a mixed strategy that combines the progressive filtering of Arnold and Sörensen (2021) presented in Section 2 and a new coverage ratio heuristic (CRH) introduced in this paper. Specifically, we use the progressive filtering to obtain M ($0 < M \leq 20000$) depot configurations, which are evaluated by Clarke-Wright's regret algorithm (Arnold and Sörensen 2021) to retain at most 100 promising depot configurations. We then merge these depot configurations with those from our CRH to obtain the set $\mathcal{D}$ of depot configurations.

CRH is composed of a preliminary filter and a secondary filter. The preliminary filter identifies a set of depot configurations that strike a balance between estimated travel costs and geographic dispersion of the candidate depots. To evaluate the costs without constructing complete solutions, the most common way is to use the minimum spanning tree (MST) to estimate the costs (Schneider and Löffler 2019). For this purpose, we use Prim's algorithm to construct a MST from each depot $i \in \mathcal{I}$ as the root node with respect to its capacity w_i and save the served customers in set $\mathcal{S}_i$. The estimated cost u_i of depot i is the sum of the open cost o_i, the fixed cost of the vehicles needed to service the customers in $\mathcal{S}_i$, and the total travel cost of the MST. The set $\mathcal{S}_i$ can be thought of as the coverage area for depot i. Then a new depot configuration is built from an empty set by adding greedily candidate depots based on the estimated cost of each candidate depot and the overlap coverage ratio between the candidate depot and the selected depots. From the set $\mathcal{D}$ of depot configurations from the preliminary filter, the secondary filter identifies a reduced set of most promising initial depot configurations. We use each depot configuration $\mathcal{D}_i$ in $\mathcal{D}$ to generate a limited number N_t (a parameter, fixed to 10) of solutions, which are improved by the local search procedure of Section 3.5.2. The average cost of these N_t improved solutions is used to assess the attractiveness of $\mathcal{D}_i$. Based on the attractiveness values, the top γ depot configurations are retained in $\mathcal{D}$ and used to initialize the subpopulations of the population. More details about CRH are given in the online supplement (He et al. 2025).

3.2.2. Subpopulation generation. From the γ promising depot configurations $\mathcal{D} = \{\mathcal{D}_1, \mathcal{D}_2, \ldots, \mathcal{D}_\gamma\}$, a population $\mathcal{P}$ composed of $\gamma + 1$ subpopulations is built. For each depot configuration $\mathcal{D}_i$, a subpopulation $\mathcal{P}_i$ is created whose solutions are all based on this depot configuration. An additional subpopulation $\mathcal{P}_{\gamma+1}$ is also built to promote and maintain diversity, containing solutions that are not related to any of the γ depot configurations.

To build the subpopulation $\mathcal{P}_i$ for a given depot configuration $\mathcal{D}_i$, a random greedy heuristic (RGH) is used to construct each solution, which is then improved by local search to increase its quality. First, RGH randomly selects a depot from the given $\mathcal{D}_i$ and uses the cheapest customer to start a route. Second, it greedily inserts other unvisited customers into the route based on the nearest neighbors rule (Section 3.5.2). When the capacity of the vehicle is reached, RGH starts a new route. When the maximum capacity of the current depot is exceeded, RGH selects a new depot from $\mathcal{D}_i$. RGH continues this process until all customers are covered, resulting in a feasible initial solution.

Each initial solution is then improved by local search (Section 3.5.2) and added to subpopulation $\mathcal{P}_i$. To ensure population diversity, we apply the technique in Vidal et al. (2012) to obtain the final $\mathcal{P}_i$. That is, we expand $\mathcal{P}_i$ until $\mathcal{P}_i$ reaches the maximum limit of $\mu+\lambda$ (μ and λ are two parameters, see Section 3.6). Then we apply a distance-and-quality updating strategy (described in Section 3.6) to retain μ solutions in $\mathcal{P}_i$.

The last subpopulation $\mathcal{P}_{\gamma+1}$ consists of solutions without fixed depot configurations. To create a solution of $\mathcal{P}_{\gamma+1}$, we start with all candidate depots i sorted in a list $\mathcal{L}$ with their rough costs u_i (see Section 3.2.1). We then use a random greedy method to iteratively add depots from $\mathcal{L}$ to the depot configurations until the total capacity of the selected depots covers the total demand of the customers. Unlike CRH, this method does not take into account geographic dispersion of the candidate depots.

3.3. Multi-depot edge assembly crossover

A carefully designed crossover operator is a driving force of a successful hybrid genetic algorithm. To be effective, the crossover must be designed to ensure the transmission of useful building blocks (or solution patterns) from parents to offspring (Hao 2012). In addition, the design of the crossover should ideally ensure the production of diversified offspring solutions.

From the perspective of exploration and exploitation, such a crossover aims to play the role of strategic diversification with the long-term goal of enhancing intensification.

Before applying the crossover operator, two parent solutions are selected in two steps. First, two subpopulations are selected using the binary tournament strategy based on their average objective values. Second, within each selected subpopulation, another binary tournament selection is applied to obtain a parent based on its objective value. As a result, the two parent solutions are necessarily based on two different depot configurations, and the mdEAX crossover aims to generate offspring solutions by simultaneously considering the routes of the parent solutions and their underlying depot configurations.

Similar to the crossovers for the CVRP (Nagata and Bräysy 2009) and the SDVRP (He and Hao 2023), mdEAX is based on the general idea of the popular EAX crossover for the traveling salesman problem (Nagata and Kobayashi 2013). However, unlike these studies, mdEAX is designed to be able to consider routing solutions with different depot configurations, which is not the case in the previous studies. Furthermore, mdEAX also differs from the crossovers for the CLRP used in Prins et al. (2006a), Lopes et al. (2016), which focus only on routes while overlooking the construction of new depot configurations.

Given two parent solutions φ_A and φ_B, we build two graphs $\mathcal{G}_\mathcal{A} = (\mathcal{V}_\mathcal{A}, \mathcal{E}_\mathcal{A})$ and $\mathcal{G}_\mathcal{B} = (\mathcal{V}_\mathcal{B}, \mathcal{E}_\mathcal{B})$ where $\mathcal{V}_\mathcal{A}$ and $\mathcal{V}_\mathcal{B}$ are the sets of vertices representing the depots and customers visited by φ_A and φ_B, respectively, and $\mathcal{E}_\mathcal{A}$ and $\mathcal{E}_\mathcal{B}$ are the sets of edges traversed by φ_A and φ_B, respectively. Since each customer is visited exactly once, each customer vertex in $\mathcal{G}_\mathcal{A}$ and $\mathcal{G}_\mathcal{B}$ has a degree of two. However, the vertices associated with the depots of φ_A and φ_B may have different degrees because the parents use two different depot configurations. Following (He and Hao 2023, He et al. 2023), we extend $\mathcal{E}_\mathcal{A}$ and $\mathcal{E}_\mathcal{B}$ by adding dummy loops so that each vertex in $\mathcal{G}_\mathcal{A}$ has the same degree as in $\mathcal{G}_\mathcal{B}$, resulting in the extended edge sets $\mathcal{E}'_\mathcal{A}$ and

$\mathcal{E}'_{\mathcal{B}}$. Then, the joint graph $\mathcal{G}_{\mathcal{AB}} = (\mathcal{V}_{\mathcal{A}} \cup \mathcal{V}_{\mathcal{B}}, (\mathcal{E}'_{\mathcal{A}} \cup \mathcal{E}'_{\mathcal{B}}) \backslash (\mathcal{E}'_{\mathcal{A}} \cap \mathcal{E}'_{\mathcal{B}}))$ is created and used by the mdEAX crossover.

Specifically, from the parent solutions φ_A and φ_B, the mdEAX crossover generates β offspring solutions (β is a prefixed threshold value) by the following steps.

1. Add dummy loops. Dummy loops are introduced to extend graphs $\mathcal{G}_{\mathcal{A}}$ and $\mathcal{G}_{\mathcal{B}}$ to ensure that the *degree difference* becomes 0 for each depot belonging to $\mathcal{V}_{\mathcal{A}} \cup \mathcal{V}_{\mathcal{B}}$, leading to two extended graphs $\mathcal{G}'_{\mathcal{A}} = (\mathcal{V}_{\mathcal{A}}, \mathcal{E}'_{\mathcal{A}})$ and $\mathcal{G}'_{\mathcal{B}} = (\mathcal{V}_{\mathcal{B}}, \mathcal{E}'_{\mathcal{B}})$.

2. Build the joint graph $\mathcal{G}_{\mathcal{AB}} = (\mathcal{V}_{\mathcal{A}} \cup \mathcal{V}_{\mathcal{B}}, (\mathcal{E}'_{\mathcal{A}} \cup \mathcal{E}'_{\mathcal{B}}) \backslash (\mathcal{E}'_{\mathcal{A}} \cap \mathcal{E}'_{\mathcal{B}}))$ from $\mathcal{G}'_{\mathcal{A}}$ and $\mathcal{G}'_{\mathcal{B}}$.

3. Construct *AB-cycles* from $\mathcal{G}_{\mathcal{AB}}$. The edges in $\mathcal{G}_{\mathcal{AB}}$ are partitioned into disjoint *AB-cycles* as follows. An AB-cycle is a cycle such that its edges alternate between edges of $\mathcal{E}'_{\mathcal{A}}$ and edges of $\mathcal{E}'_{\mathcal{B}}$. Because each vertex in $\mathcal{G}_{\mathcal{AB}}$ is connected to an even number of edges, and half of these edges come from $\mathcal{E}'_{\mathcal{A}}$, while the other half come from $\mathcal{E}'_{\mathcal{B}}$, the edges in $\mathcal{G}_{\mathcal{AB}}$ can be partitioned into *AB-cycles* accurately.

4. Generate *E-sets*. The *AB-cycles* are grouped into *E-sets*. If the number of *E-sets* exceeds the threshold β, some *E-sets* are randomly merged to retain only β *E-sets*. By combining *AB-cycles* that share common vertices into larger *E-set*, we ensure a sufficient diversity of the routes in the offspring solutions (Nagata and Bräysy 2009).

5. Construct intermediate solutions. A random solution φ (φ_A or φ_B, say φ_A) is selected first. For each *E-set* $\mathcal{E}_s$, we create an intermediate solution $\varphi' = (\mathcal{E}_{\mathcal{A}} \backslash (\mathcal{E}_s \cap \mathcal{E}_{\mathcal{A}})) \cup (\mathcal{E}_s \cap \mathcal{E}_{\mathcal{B}})$, where the added dummy loops are ignored.

6. Split mega tours. An intermediate solution may contain routes that visit more than one depot (called mega tours). For these tours, we remove the extra depots by taking into account the capacity of the opening depots. Specifically, for each mega tour with depots $i_1, i_2, \ldots, i_k$ ($k > 1$), we calculate the residual capacity (which can be negative) of each depot

i_l $(l = 1, 2, \ldots, k)$ and then keep only the depot with the largest residual capacity. For the case of depots with unlimited capacity, we remove the extra depots, which lead to the largest reduce of the travel cost.

7. Eliminate subtours. If an intermediate solution contains some isolated tours (called subtours), the 2-opt* operator (Potvin and Rousseau 1995) is applied to link these isolated subtours to other tours.

Because the complexity of splitting mega tours is limited to $\mathcal{O}(m)$, the time complexity of mdEAX is $\mathcal{O}((n+m) \times \alpha)$, as other steps of mdEAX are the same as gEAX in He and Hao (2023). Here, α is the number of nearest neighbors (see Section 3.5.2). Furthermore, the space complexity is bounded by $\mathcal{O}(|\mathcal{E}'_{\mathcal{A}}|)$, assuming that $|\mathcal{E}'_{\mathcal{A}}| \geq |\mathcal{E}'_{\mathcal{B}}|$.

The mdEAX crossover not only extracts backbone information from parent solutions and transmits it to offspring solutions, but also effectively promotes diversity. In addition, it can generate new promising depot configurations when building new solutions. As such, it plays a key role in our integrated approach, which simultaneously considers depot decision and route optimization, and avoids the difficulty of the hierarchical approach, which requires the prior determination of promising depot configurations.

3.4. Restore the feasibility of offspring individuals

The mdEAX crossover ignores capacity constraints, as a result, some offspring solutions may be infeasible. To address this issue, we adopt a greedy procedure that uses a generalized cost function to eliminate all violations of both depot and vehicle capacities.

Let φ be an offspring solution, let T_φ denote the cumulative depot capacity for the given depot configuration $\mathcal{D}_\varphi$ and T denote the total demand of customers. Then $T_\varphi < T$ or $T_\varphi \geq T$. In the first case, we need to select additional depots to expand $\mathcal{D}_\varphi$. A customer is removed from the route that violates the depot capacity and inserted into a new route with a new open

depot with respect to travel costs, while ignoring the fixed depot and vehicle costs to improve feasibility. This process stops when $T_\varphi \geq T$. Then, we restore the capacity constraints for both vehicles and depots. Four well-known inter-route neighborhood operators (Relocate and Swap with single customers and 2-Opt* in Section 3.5.2) are used to explore the neighbor solutions based on Eq. (1), where $f(\varphi)$ denotes the objective function value of solution φ, $f_l(\varphi)$ represents the cumulative overcapacity of depots, $f_r(\varphi)$ is the cumulative overcapacity of routes, and p_l and p_r are penalty factors for overcapacity of depots and routes, respectively. To avoid excessive iterations, we use a tabu list to prevent undoing a performed move. If no feasible move can be found while the solution is still infeasible, we multiply the penalty factors p_l and p_r by 10. This process continues until the solution becomes feasible. We set p_l and p_r to 1 initially.

$$f_g(\varphi) = f(\varphi) + p_l \times f_l(\varphi) + p_r \times f_r(\varphi) \tag{1}$$

3.5. Mutation and local search

3.5.1. Mutation. When using mdEAX to generate offspring solutions, one faces the challenging problem that the edges of the offspring solutions are almost exclusively from their parents, leaving little room for new edges in the solutions. To introduce diversity into the offspring solutions, we apply a mutation operator, based on an idea presented in Shaw (1998), to modify each offspring solution with probability ζ.

At the beginning, a random customer j is chosen to initialize an empty list $\mathcal{Q}$. The similarity between customer j and every other customer $(\mathcal{J} \backslash \mathcal{Q})$ is calculated based on their distance in the distance matrix $\mathcal{C}$ (small distance means high similarity). The customers are then sorted in ascending order, starting with the customer with the highest similarity. Next, a customer is selected and added to $\mathcal{Q}$ using the roulette-wheel selection, and this process is repeated until $\xi \times n$ customers (ξ is the mutation strength) are selected and added to $\mathcal{Q}$

(such that $|\mathcal{Q}| = \xi \times n$). Then the customers in $\mathcal{Q}$ are removed from the current solution and added back to the solution as follows. For each customer $j \in \mathcal{Q}$, a vertex i from its α-nearest neighborhood (Section 3.5.2) in $\mathcal{V} \backslash \mathcal{Q}$ is selected and customer j is inserted after vertex i, subject to the capacity constraint and the objective value. The procedure ends when all customers in $\mathcal{Q}$ are reinserted into the solution. The worst-case time complexity of the mutation is bounded by $\mathcal{O}(\xi \times n \times \alpha)$.

3.5.2. Local search. Local search ensures the key role of search intensification to obtain high-quality solutions. HGAMP adopts ten commonly used neighborhood (or move) operators from (Duhamel et al. 2010, Schneider and Löffler 2019) and explores them with the variable neighborhood descent method.

Let vertex $v \in \mathcal{V}$ be the α-nearest neighbor of customer $u \in \mathcal{J}$, where α ($\alpha < |\mathcal{I}| + |\mathcal{J}|$) is the granularity threshold that restricts the search to nearby vertices. Let $r(u)$ and $r(v)$ denote the two routes that visit vertices u and v, respectively. Additionally, let x and y represent the successors of u in $r(u)$ and v in $r(v)$, respectively.

If $v = y$ and $v \in \mathcal{I}$, then $r(v)$ is an empty route. Let x^+ denote the successor of x. Let u_b and v_b be the first customer visited by routes $r(u)$ and $r(v)$, respectively, and let u_l and v_l be the last customer visited by routes $r(u)$ and $r(v)$, respectively. Let (u, x) be the substring from customer u to x and (v, y) be the substring from customer v to y. The ten move operators used are defined as follows.

- M1: Customer u is removed from $r(u)$ and inserted into $r(v)$ after vertex v.
- M2: Two consecutive customers u and x are removed from $r(u)$ and inserted into $r(v)$ after vertex v.
- M3: Two consecutive customers u and x are removed from $r(u)$ and (x, u) is placed before vertex v.

- M4: Interchange the position of customer x and the position of customer v.
- M5: Interchange two consecutive customers (x, x^+) and the position of customer v.
- M6: Interchange two consecutive customers (x, x^+) and two consecutive customers (v, y).
- M7: Interchange two consecutive customers (x^+, x) and two consecutive customers (v, y).
- M8: This is the 2-opt operator, which replaces (u, x) and (v, y) by (u, v) and (x, y) if $r(u) = r(v)$.
- M9: Interchange substring (x, u_l) and substring (v, v_l) if $r(u) \neq r(v)$. This is the 2-opt* operator with respect to different or same depots.
- M10: Interchange substring (x, u_l) and substring (v_b, v) if $r(u) \neq r(v)$. This is the 2-opt* operator with respect to different or same depots.

Due to the nearest neighbor rule, the time complexity of the move operators is bounded by $\mathcal{O}((n+m) \times \alpha)$.

3.6. Population management

Population management aims to maintain a healthy diversity of the population $\mathcal{P}$ throughout the search process. This involves two key tasks: 1) updating the depot configurations when a better one is discovered, and 2) managing and updating each subpopulation.

The set of depot configurations $\mathcal{D}$ is updated each time the global best solution φ^* is replaced by the current solution φ whose corresponding depot configuration $\mathcal{D}_\varphi$ is not in set $\mathcal{D}$ ($\mathcal{D}_\varphi \notin \mathcal{D}$). The worst subpopulation $\mathcal{P}_w$ is identified among the first γ subpopulations and then discarded with its corresponding individuals. The new depot configuration $\mathcal{D}_\varphi$ is then used to re-initialize the discarded subpopulation.

Furthermore, an advanced distance-and-quality updating rule (Vidal et al. 2012, He and Hao 2023) is used to manage each subpopulation. Each new offspring solution is inserted

into the corresponding subpopulation based on its depot configuration, and clones are not permitted. If the depot configuration of an offspring solution is different from the γ promising depot configurations, the solution is inserted into $\mathcal{P}_{\gamma+1}$. For each subpopulation, when the number of solutions reaches the maximum size $\mu + \lambda$ (λ is the generation size), λ solutions are removed based on a biased fitness. Specifically, we first compute the Hamming distance between all pairs of solutions using the method of Prins et al. (2006b) and use this distance to measure the contribution of each solution to the diversity of the subpopulation, where the Hamming distance between two solutions is computed from the differences in terms of routes, edges, and depots of the two solutions. Then, the biased fitness of each solution is calculated with the weighted average of the objective value (fitness) rank and the diversity rank in the subpopulation. The solution associated with the worst biased fitness is removed and the biased fitness of the remaining solutions is updated accordingly. We repeat this removal process until there are no more than μ solutions left in the subpopulation.

If the best solution found so far φ^* cannot be improved for η successive invocations of the local search (η is a parameter called the population rebuilding threshold), HGAMP restarts by generating a completely new population. Finally, a large number of subpopulations slows down the convergence of the algorithm. So, when the algorithm reaches half the total time budget, we reduce the number of subpopulations by half, keeping only the most promising subpopulations with respect to the average objective values.

3.7. Discussion

HGAMP enhances the canonical hybrid genetic algorithm framework with three key techniques: the multi-population scheme, the mdEAX crossover, and the coverage ratio heuristic CRH, distinguishing itself from existing HGAs for various VRPs (Prins et al. 2006a, Nagata and Bräysy 2009, Vidal et al. 2012, He and Hao 2023).

First, compared to the simple method of organizing the solutions of the population into a single pool, the multi-population scheme systematically manages a set of elite solutions associated with each promising depot configuration. As such, this scheme provides the algorithm with an enhanced ability to consistently explore a number of candidate solutions of interest and the underlying depot configurations. This allows the algorithm to achieve better coverage of the search space and favors the discovery of high-quality final solutions.

Second, the mdEAX crossover provides a powerful means to recombine parent solutions with different depot configurations. By inheriting meaningful route and depot building blocks from elite parents, mdEAX is able to generate promising offspring solutions that may rely on new depot configurations. Thus, the mdEAX crossover contributes to a meaningful search diversification while ensuring a certain degree of search intensification.

Third, to identify initial depot configurations of interest, CRH uses both cost efficiency and geographic distribution to strike a balance between these two factors, unlike previous methods that divide the given area into an increasing number of isometric regions (Arnold and Sörensen 2021). By considering both factors related to the depots, CRH helps to find interesting depot configurations.

Finally, beyond the studied CLRP, the general approach to simultaneously explore unknown depot configurations and new route solutions using the multi-population scheme and the mdEAX crossover is general. This approach can be advantageously applied to solve other related LRPs such as the 2E-VRP (Perboli et al. 2011) and the location arc routing problem (Lopes et al. 2014). In this context, we show in the online supplement (He et al. 2025) how we apply HGAMP to the challenging MDVRP with very little change.

4. Experimental Evaluation and Comparisons

The purpose of this section is to evaluate the performance of HGAMP through experiments and to compare its results with those of state-of-the-art algorithms.

4.1. Benchmark instances

In the literature, there are three classical benchmark sets containing 79 small and medium instances from Tuzun and Burke (1999), Prins et al. (2006b), and Barreto et al. (2007). These sets, denoted as $\mathbb{B}, \mathbb{P}, \mathbb{T}$, have been widely tested in the past, and their results are difficult to further improve upon. In addition, Schneider and Löffler (2019) introduced a rich and challenging set (denoted as $\mathbb{S}$) of 202 instances with different characteristics and sizes. The instances used in our experimentation and our solution files are available from https://github.com/pengfeihe-angers/CLRP.git.

- Set $\mathbb{B}$: This set contains 13 instances (7 with a proven optimal value) with 21 to 150 customers and 5 to 14 depots, and in most instances a depot capacity is imposed.
- Set $\mathbb{P}$: This set contains 30 instances (20 with a proven optimal value) with 20 to 200 customers and 5 to 10 capacitated depots.
- Set $\mathbb{T}$: This set contains 36 instances (6 with a proven optimal value) with 100 to 200 customers and 10 to 20 uncapacitated depots.
- Set $\mathbb{S}$: This set is composed of 202 instances (64 with a proven optimal value) with varying characteristics, such as the number of customers, depots, vehicle capacity, and depot costs. The instances have 100 to 600 customers and 5 to 30 capacitated depots. The instances are named in the format of $n-m-tz$, where n represents the number of customers, m is the number of depots, t indicates the geographic distribution, and z denotes the vehicle capacity and depot costs.

4.2. Experimental protocol and reference algorithms

4.2.1. Parameter setting. The HGAMP algorithm has six parameters: the minimum size of each subpopulation μ, the generation size λ, the granularity threshold of nearest neighbors α, the mutation probability ζ, the mutation strength ξ, and the population rebuilding

threshold η. The automatic parameter tuning package irace (López-Ibáñez et al. 2016) was used to calibrate these parameters. The tuning procedure was performed on 10 instances with 100 to 300 customers. The tuning budget was set to be 2000 runs. Table 2 shows the candidate values for the parameters and the final values suggested by irace. Finally, to get a good compromise between search performance and computational effort, we empirically set $\gamma = 15$, which corresponds to $\gamma + 1$ subpopulations (in Section 5, we examine the influence of γ). These parameter values represent HGAMP's default settings, which are used in our experiments (unless otherwise noted).

Table 2 Parameter tuning results.

Parameter	Section	Description	Candidate values	Final values
μ	3.6	minimal size of each subpopulation	$\{10, 20, 30, 40, 50\}$	30
λ	3.6	generation size	$\{10, 20, 30, 40, 50\}$	30
α	3.5.2	granularity threshold	$\{5, 10, 15, 20, 25, 30\}$	20
ζ	3.5.1	mutation probability	$\{0, 0.05, 0.1, 0.15, 0.2, 0.25, 0.3\}$	0.15
ξ	3.5.1	mutation strength	$\{0.05, 0.1, 0.15, 0.2, 0.25\}$	0.25
η	3.6	population rebuilding threshold	$\{30000, 50000, 70000, 90000\}$	70000

4.2.2. Reference algorithms. We adopt the following best CLRP heuristic algorithms as well as the best known solutions BKS (best upper bounds) reported in the literature as the references for comparative experiments.

- BKS. This indicates the best known solutions (upper bounds) that are summarized from the state-of-the-art heuristic algorithms (Contardo et al. 2014b, Lopes et al. 2016, Schneider and Löffler 2019, Arnold and Sörensen 2021, Voigt et al. 2022), and the exact approaches (Baldacci et al. 2011, Contardo et al. 2014a, Liguori et al. 2023).
- AVXS (Accorsi and Vigo 2020). The algorithm was coded in C++ and executed on a computer with a 3.7 GHz Intel i7-8700k CPU with 32 GB of RAM. One notices that the algorithm only reported results on set $\mathbb{T}$. The algorithm was executed 10 times on each tested instance with distinct random seeds.

- HALNS (Voigt et al. 2022). The hybrid adaptive large neighborhood search was implemented in C++, running on a computer with a 3.8 GHz AMD Ryzen 9 3900X CPU with 32 GB of RAM. The algorithm reported results on the three classical sets $\mathbb{B}$, $\mathbb{P}$ and $\mathbb{T}$. The algorithm was executed 10 times on each tested instance with distinct random seeds.

- PF (Arnold and Sörensen 2021). This algorithm was coded in Java and the experiments were conducted on a computer with a 3.6 GHz Intel i7 4790 with 8 GB of RAM. The algorithm reported results on the four sets of benchmark instances on a single thread. The algorithm was executed one single time on each tested instance.

- TBSA$_{basic}$ (Schneider and Löffler 2019). The algorithm, implemented in C++ and run on a computer with a 2.6 GHz Xeon E5-2670 CPU with 32 GB of RAM, reported results on the four sets of benchmark instances. The algorithm was executed 5 times on each tested instance with distinct random seeds. TBSA$_{basic}$ has a good trade-off between solution quality and run time.

- TBSA$_{quality}$ (Schneider and Löffler 2019). This is the TBSA$_{basic}$ algorithm that was run for an increased granular threshold and increased number of iterations in the routing phase. TBSA$_{quality}$ only reported results on the three classical sets $\mathbb{B}$, $\mathbb{P}$ and $\mathbb{T}$.

- GANCP^{+} (Sobhanan et al. 2024). This is the latest algorithm coded in Julia and run on a computer with a 2.5 GHz Inter Core i9-11900H CPU and an NVIDIA GeForce RTX 3080 GPU with 8 GB. The algorithm was tested on $\mathbb{B}$ and $\mathbb{T}$ and only reported the best result for each tested instance out of 5 independent runs.

4.2.3. Experimental setting and stopping criterion. The HGAMP algorithm was implemented in C++ and compiled using the g++ compiler with the -O3 option. All experiments were performed on a 2.5 GHz Intel Xeon E-2670 processor with 2 GB of RAM running Linux on a single thread. The algorithm terminates after a maximum of 300,000 iterations (default

termination condition), where one iteration means that one offspring solution is constructed and subsequently improved by the local search. We report the best and average solution quality over 20 runs. We note that our default termination condition leads to runtimes similar to those of TBSA$_{quality}$ (Schneider and Löffler 2019) and HALNS (Voigt et al. 2022).

4.3. Computational results and comparisons

In the following subsections, we present HGAMP's results on the benchmark instances and compare them to the reference algorithms.

4.3.1. Performance analysis on the three classical sets $\mathbb{B}$, $\mathbb{P}$ and $\mathbb{T}$. Table 3 summarizes the results obtained by the HGAMP algorithm on the three classical sets, compared to the reference algorithms. For each benchmark set, we show the number of instances where our algorithm (HGAMP) has a better (Win), equal (Tie) or worse (Loss) result compared to each reference, in terms of the best and average results. Wilcoxon signed-rank p-values are also included. The '-' symbol indicates that the result is unavailable. Tables 2 and 3 in the online supplement (He et al. 2025) provide detailed results for these 79 instances. In addition, Figure 1 illustrates the performance comparison over the sets $\mathbb{B}$, $\mathbb{P}$ and $\mathbb{T}$, where we show the runtimes of the compared methods scaled with the Passmark scores (https://www.cpubenchmark.net/singleThread.html), and the gaps of the methods to BKS for each benchmark set. In Figure 1, we exclude the latest algorithm GANCP$^+$ because it is dominated by the other algorithms according to the results of Table 3.

For the 13 instances of set $\mathbb{B}$, HGAMP achieves the same performance as BKS, HALNS, and TBSA$_{quality}$ in terms of the best results, while reporting better results than GANCP$^+$. Moreover, our algorithm has a slight advantage over the reference algorithms in terms of the average results. Because the instances in this set are rather small and state-of-the-art algorithms produce identical results, this set may not be challenging enough to evaluate the

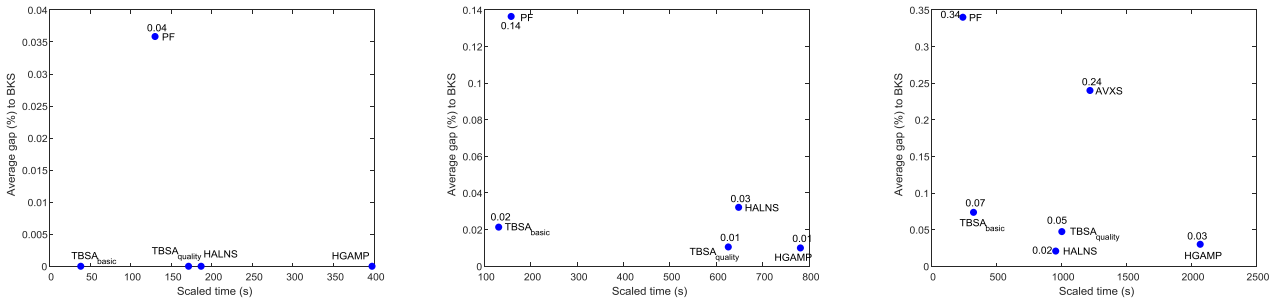
performance of new algorithms. Figure 1(a) confirms that all the methods have the same performance except for PF. However, our algorithm requires a longer scaled runtime. The reason is that the population contains a relatively large number of solutions ($\geq 30*16=480$), which makes the algorithm take longer to converge.

For the 30 instances of set $\mathbb{P}$, HGAMP improves upon one BKS with an improvement gap of 0.03% (shown in bold, see the online supplement (He et al. 2025) for detailed information), matches 24 BKS values, and misses 5 BKS values. Compared to the reference algorithms, HGAMP achieves, in terms of the best values, results similar to HALNS, TBSA$_{basic}$, and TBSA$_{quality}$ (without statistically significant differences), while reporting better results than PF. For average results, HGAMP performs similarly compared to TBSA$_{basic}$, but worse than HALNS and TBSA$_{quality}$. From Figure 1(b), we see that TBSA$_{basic}$ is already competitive with the other algorithms, while TBSA$_{quality}$ (under the relaxed runtime condition) performs even better compared to its competitors, although the performance differences are rather small.

For the 36 instances of set $\mathbb{T}$, HGAMP improves upon the BKS in two cases with an improvement gap of 0.01% (shown in bold, see the online supplement (He et al. 2025) for detailed information) and matches 23 BKS values, while missing 11 BKS values. It significantly outperforms AVXS, PF, GANCP^{+}, and TBSA$_{basic}$ in terms of best results (p value < 0.05), while performing slightly better than TBSA$_{quality}$ and slightly worse than HALNS (p value > 0.05). For average results, HGAMP performs significantly better than AVXS and marginally better than TBSA$_{basic}$, while being outperformed by HALNS and TBSA$_{quality}$. Furthermore, as shown in Figure 1(c), HALNS dominates the other algorithms for shorter running times, and the differences in the average gap to BKS for most of the compared algorithms remain marginal. HGAMP requires a longer scaled runtime due to the use of a relatively large population.

Table 3 Summary of the comparative results between HGAMP and the reference algorithms on the three classical benchmark sets $\mathbb{B}$, $\mathbb{P}$ and $\mathbb{T}$.

Instances(#)	Pair algorithms	Best				Avg.			
		#Wins	#Ties	#Losses	p-value	#Wins	#Ties	#Losses	p-value
$\mathbb{B}$(13)	HGAMP vs. BKS	0	13	0	>0.05	-	-	-	-
	HGAMP vs. PF	2	11	0	>0.05	-	-	-	-
	HGAMP vs. GANCP$^+$	6	7	0	**<0.05**	-	-	-	-
	HGAMP vs. HALNS	0	13	0	>0.05	1	11	1	>0.05
	HGAMP vs. TBSA$_{basic}$	2	11	0	>0.05	4	8	1	>0.05
	HGAMP vs. TBSA$_{quality}$	0	13	0	>0.05	3	9	1	>0.05
$\mathbb{P}$(30)	HGAMP vs. BKS	**1**	24	5	**<0.05**	-	-	-	-
	HGAMP vs. PF	12	16	2	**<0.05**	-	-	-	-
	HGAMP vs. HALNS	3	24	4	>0.05	4	15	11	>0.05
	HGAMP vs. TBSA$_{basic}$	7	20	3	>0.05	8	14	8	>0.05
	HGAMP vs. TBSA$_{quality}$	4	22	4	>0.05	3	15	12	**<0.05**
$\mathbb{T}$(36)	HGAMP vs. BKS	**2**	23	11	**<0.05**	-	-	-	-
	HGAMP vs. AVXS	17	13	6	**<0.05**	25	2	9	**<0.05**
	HGAMP vs. PF	25	9	2	**$\ll$0.05**	-	-	-	-
	HGAMP vs. GANCP$^+$	31	4	1	**$\ll$0.05**	-	-	-	-
	HGAMP vs. HALNS	5	23	8	>0.05	6	4	26	**$\ll$0.05**
	HGAMP vs. TBSA$_{basic}$	17	15	4	**<0.05**	21	2	13	>0.05
	HGAMP vs. TBSA$_{quality}$	14	16	6	>0.05	14	2	20	**<0.05**



(a) Set $\mathbb{B}$ (b) Set $\mathbb{P}$ (c) Set $\mathbb{T}$

Figure 1 Performance comparison on the three classical sets with the average gap to the BKS for each method.

4.3.2. Performance analysis on the rich and challenging instance set $\mathbb{S}$. We now evaluate the HGAMP algorithm on the 202 challenging instances of set $\mathbb{S}$. We only compare our results to the BKS, TBSA$_{basic}$, and PF because the other reference algorithms do not report results on this set.

Table 4 summarizes the results and highlights the high performance of our algorithm, and Tables 4 and 5 in the online supplement (He et al. 2025) show the detailed results for each instance. HGAMP achieves an improvement over the BKS by achieving 100 new upper bounds out of the 202 instances of set $\mathbb{S}$ (49.5%), while matching the BKS for 25

other instances and missing the remaining BKS. Specifically, according to the detailed results in the online supplement, for the 62 instances with 100 to 200 customers, our HGAMP algorithm improves 5 best-known results (improvement from 0.002% to 0.07%). The results become more significant for the 140 large instances with 300 to 600 customers, updating 95 best upper bounds (improvement from 0.003% (400-25-4b) to 1.45%, with an exceptional improvement of 4.61% for 600-30-3e). Compared to the highly effective TBSA$_{basic}$, HGAMP competes very favorably, achieving 144 better values in terms of the best and average results, and losing 42 best and 56 average results, respectively. Compared to PF, HGAMP shows a strong dominance with 185 superior best results and only 11 worse results. For the 140 largest instances with 300 to 600 customers, HGAMP outperforms TBSA$_{basic}$ and PF on 112 and 129 instances, respectively, with improvements of up to 9.72% and 4.61%, respectively. The differences between HGAMP and its competitors are statistically significant based on the Wilcoxon signed-rank tests (p-values $\ll 0.05$), except for the average result between HGAMP and TBSA$_{basic}$. Overall, HGAMP performs very well on $\mathbb{S}$ set compared to the best heuristic algorithms.

Table 4 Summary of the comparative results between HGAMP and the reference algorithms on the rich and challenging benchmark set $\mathbb{S}$ (202 instances).

Pair algorithms	Best				Avg.			
	#Wins	#Ties	#Losses	p-value	#Wins	#Ties	#Losses	p-value
HGAMP vs. BKS	**100**	25	77	$\ll$**0.05**	-	-	-	-
HGAMP vs. TBSA$_{basic}$	144	16	42	$\ll$**0.05**	144	2	56	$\ll$**0.05**
HGAMP vs. PF	185	6	11	$\ll$**0.05**	-	-	-	-

Figure 2 shows that HGAMP has a significant performance advantage over PF, although it has a higher scaled runtime. We also see that HGAMP significantly dominates TBSA$_{basic}$ in performance, with a slightly longer scaled runtime. A closer inspection of Tables 4 and 5 in the online supplement (He et al. 2025) reveals that HGAMP requires more times to solve small instances, e.g., 1070.08 seconds on instance 100-5-1c, while TBSA$_{basic}$ quickly

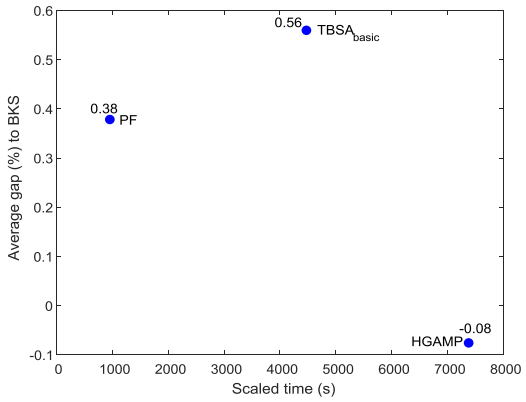


Figure 2 **Performance comparison on the rich and challenging set $\mathbb{S}$ with the average gap to the BKS for each method.**

finds good enough solutions in 81.24 seconds. This difference is due to the relatively large number of solutions in the HGAMP population, which causes the algorithm to take longer to converge. Given a sufficient number of iterations, such as 300,000, HGAMP performs well on both small and large instances, as shown in Tables 4 and 5 in the online supplement, while TBSA$_{basic}$ takes a considerable amount of time to solve large instances. From this figure, we conclude that our algorithm has the desirable potential to find better results if given more time.

5. Analysis of algorithmic components to understand their role

We now study the role of three critical algorithmic components: the initial depot configurations, the multi-population scheme, and the mdEAX crossover operator. The experiments are based on the 202 challenging instances of set $\mathbb{S}$.

5.1. Influence of initial depot configurations

HGAMP uses a mixed strategy combining CRH of Section 3.2.1 and the progressive filtering heuristic of Arnold and Sörensen (2021) to identify a set of promising depot configurations for the generation of the initial population. To show the benefit of this mixed strategy, we compare HGAMP with two variants HGAMP$_{CRH}$, where only CRH is used, and HGAMP$_{AS}$, where only the progressive filtering heuristic is used.

Table 5 Comparative results between HGAMP and two variants using different initialization methods (set $\mathbb{S}$).

Pair algorithms	Best				Avg.			
	#Wins	#Ties	#Losses	p-value	#Wins	#Ties	#Losses	p-value
HGAMP vs. HGAMP$_{CRH}$	116	33	53	$\ll$**0.05**	132	12	56	$\ll$**0.05**
HGAMP vs. HGAMP$_{AS}$	92	32	78	>0.05	110	10	81	<0.05

Table 5 summarizes the results of HGAMP and the two variants. HGAMP significantly outperforms HGAMP$_{CRH}$ on both the best and average values. Specifically, HGAMP wins 116 instances and matches 33 instances for the best values. This indicates that the hybrid strategy for creating the depot configurations for the initial population solutions is very beneficial to the performance of the algorithm. In addition, compared to HGAMP$_{AS}$, HGAMP significantly performs better, winning 92 and 110 instances in terms of the best and average values, respectively ($p < 0.05$). In summary, it is important for the HGAMP algorithm to start its search with high-quality initial depot configurations, and the progressive filtering and CRH together help to achieve this goal thanks to their complementary methods for identifying interesting depot configurations.

5.2. Influence of the number of subpopulations

To evaluate the multi-population scheme, we vary the γ parameter (which indicates γ+1 subpopulations): $\gamma = 0$ (corresponding to a single population where all solutions are included), $\gamma = 5$, and 10, and we denote these HGAMP variants by HGAMP$_0$, HGAMP$_5$, and HGAMP$_{10}$, respectively. Recall that HGAMP uses the default value $\gamma = 15$. The results of the comparison are summarized in Table 6.

Table 6 Summary of comparative results of HGAMP with the default $\gamma = 15$ and $\gamma = 0, 5, 10$ (set $\mathbb{S}$).

Pair algorithms	Best				Avg.			
	#Wins	#Ties	#Losses	p-value	#Wins	#Ties	#Losses	p-value
HGAMP vs HGAMP$_0$	154	31	17	$\ll$**0.05**	189	4	8	$\ll$**0.05**
HGAMP vs HGAMP$_5$	118	32	52	$\ll$**0.05**	165	8	29	$\ll$**0.05**
HGAMP vs HGAMP$_{10}$	96	34	72	<0.05	136	9	57	$\ll$**0.05**

We see that the HGAMP algorithm with $\gamma = 15$ significantly outperforms HGAMP$_0$, HGAMP$_5$ and HGAMP$_{10}$ in terms of both best and average values. These results also indicate

that a larger number of subpopulations leads to an improvement in performance. However, we also observed that as the number of subpopulations increases beyond 15, the performance improvement gradually becomes marginal. In conclusion, the multi-population scheme shows a significant advantage over the classical single-population approach, and by varying the number of subpopulations, one can expect different performance improvements.

5.3. Influence of the crossover

To evaluate the effectiveness of the mdEAX crossover, we create two HGAMP variants. The first variant, HGAMP$_d$, replaces mdEAX with the *ChangeDepot* operator. This operator removes a depot from the current solution and then greedily adds the unvisited customers back into the solution. Specifically, we select a solution φ from $\mathcal{P}$, and then choose a depot $i \in \mathcal{D}_\varphi$ based on the minimum capacity utility T_i/w_i, where $T_i = \sum_{j \in \mathcal{S}_i} d_j$ and $\mathcal{S}_i$ is the set of customers covered by depot i in solution φ. After deleting the depot i and the customers $j \in \mathcal{S}_i$, the removed customers are greedily reinserted back into the solution φ with respect to the objective. The second variant, HGAMP$_c$, replaces the mdEAX crossover with *ChangeDepot* and the mutation operator of Section 3.5.1, and randomly selects one of these two operators to generate new solutions in each generation. For this experiment, we again use HGAMP, which uses the hybrid strategy to create the initial depot configurations. The results of this comparison are summarized in Table 7.

Table 7 Summary of comparative results of HGAMP with two variants using different crossovers (set $\mathbb{S}$).

Pair algorithms	Best				Avg.			
	#Wins	#Ties	#Losses	p-value	#Wins	#Ties	#Losses	p-value
HGAMP vs HGAMP$_d$	192	10	0	$\ll$**0.05**	200	1	1	$\ll$**0.05**
HGAMP vs HGAMP$_c$	186	15	1	$\ll$**0.05**	198	2	2	$\ll$**0.05**

Table 7 clearly shows that HGAMP significantly outperforms HGAMP$_d$ and HGAMP$_c$ in terms of both best and average values. One notes that HGAMP$_c$ shows a slightly better performance than HGAMP$_d$, but its results are still far behind those of HGAMP with the

mdEAX crossover. This experiment demonstrates the critical role of the mdEAX crossover in the performance of the HGAMP algorithm.

6. Conclusions

In this paper, we propose a hybrid genetic algorithm with multi-population (HGAMP) for the capacitated location-routing problem. This approach maintains multiple subpopulations where each subpopulation consists of a set of high-quality solutions that share the same depot configuration. The algorithm uses a multi-depot edge assembly crossover (mdEAX) to explore candidate solutions of interest with new emerging depot configurations and an effective neighborhood-based local optimization to perform route optimization. The algorithm additionally applies a coverage ratio heuristic for initial depot configuration generation and a special mutation to enhance diversity.

Computational experiments on four sets of 281 commonly used benchmark instances show that the proposed algorithm is able to find 103 new best results (improved upper bounds) and match the best-known results for 85 other instances. These new results can be valuable for future research on the problem, especially for evaluating the performance of new algorithms on instances with different characteristics, such as customer distribution density and depot costs. In addition, we investigate the role and rationale of the multi-population scheme and the mdEAX operator. Since the CLRP is a relevant model for a number of real-world problems, our algorithm, whose code is publicly available, can be used to better solve some of these practical applications.

The proposed framework using subpopulations and multi-depot edge assembly crossover is general. In addition to the studied CLRP, it can be helpful for solving related problems where both location and routing decisions need to be made, such as periodic location routing, two-echelon vehicle routing, and location arc routing. We provided such an example in the online

supplement to show how the HGAMP algorithm can be applied to the popular MDVRP with little change to the algorithm. It would be interesting to investigate the usefulness of the main ideas of this work for other location routing problems by effectively deciding locations and route solutions simultaneously.

Acknowledgments

We would like to thank the editors and reviewers for their helpful comments, which helped us to significantly improve the paper. We also thank Dr. F. Arnold and Prof. K. Sörensen for kindly sharing their source code of the PF algorithm. This work was partially supported by the National Natural Science Foundation of China under Grant No. 62403123 and 72122006.

References

Accorsi L, Vigo D (2020) A hybrid metaheuristic for single truck and trailer routing problems. *Transportation Science* 54(5):1351–1371.

Akpunar ÖŞ, Akpinar Ş (2021) A hybrid adaptive large neighbourhood search algorithm for the capacitated location routing problem. *Expert Systems with Applications* 168:114304.

Arnold F, Sörensen K (2021) A progressive filtering heuristic for the location-routing problem and variants. *Computers & Operations Research* 129:105166.

Baldacci R, Mingozzi A, Wolfler Calvo R (2011) An exact method for the capacitated location-routing problem. *Operations Research* 59(5):1284–1296.

Barreto S, Ferreira C, Paixao J, Santos BS (2007) Using clustering analysis in a capacitated location-routing problem. *European Journal of Operational Research* 179(3):968–977.

Cochran JK, Horng S, Fowler JW (2003) A multi-population genetic algorithm to solve multi-objective scheduling problems for parallel machines. *Computers & Operations Research* 30(7):1087–1102.

Contardo C, Cordeau JF, Gendron B (2014a) An exact algorithm based on cut-and-column generation for the capacitated location-routing problem. *INFORMS Journal on Computing* 26(1):88–102.

Contardo C, Cordeau JF, Gendron B (2014b) A GRASP+ILP-based metaheuristic for the capacitated location-routing problem. *Journal of Heuristics* 20(1):1–38.

Cordeau JF, Gendreau M, Laporte G (1997) A tabu search heuristic for periodic and multi-depot vehicle routing problems. *Networks: An International Journal* 30(2):105–119.

Drexl M, Schneider M (2015) A survey of variants and extensions of the location-routing problem. *European Journal of Operational Research* 241(2):283–308.

Duhamel C, Lacomme P, Prins C, Prodhon C (2010) A GRASP×ELS approach for the capacitated location-routing problem. *Computers & Operations Research* 37(11):1912–1923.

Escobar JW, Linfati R, Baldoquin MG, Toth P (2014) A granular variable tabu neighborhood search for the capacitated location-routing problem. *Transportation Research Part B: Methodological* 67:344–356.

Escobar JW, Linfati R, Toth P (2013) A two-phase hybrid heuristic algorithm for the capacitated location-routing problem. *Computers & Operations Research* 40(1):70–79.

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

He P, Hao JK (2023) General edge assembly crossover-driven memetic search for split delivery vehicle routing. *Transportation Science* 57(2):482–511.

He P, Hao JK, Wu Q (2023) A hybrid genetic algorithm for undirected traveling salesman problems with profits. *Networks: An International Journal* 82(3):189–221.

He P, Hao JK, Wu Q (2025) A hybrid genetic algorithm with multi-population for capacitated location routing. URL http://dx.doi.org/10.1287/ijoc.2023.0416.cd, https://github.com/INFORMSJoC/2023.0416.

Hemmelmayr VC, Cordeau JF, Crainic TG (2012) An adaptive large neighborhood search heuristic for two-echelon vehicle routing problems arising in city logistics. *Computers & Operations Research* 39(12):3215–3228.

Liguori P, Mahjoub AR, Marques G, Uchoa E, Sadykov R (2023) Nonrobust strong knapsack cuts for capacitated location-routing and related problems. *Operations Research* 71(5):1577–1595.

Lopes RB, Ferreira C, Santos BS (2016) A simple and effective evolutionary algorithm for the capacitated location–routing problem. *Computers & Operations Research* 70:155–162.

Lopes RB, Plastria F, Ferreira C, Santos BS (2014) Location-arc routing problem: Heuristic approaches and test instances. *Computers & Operations Research* 43:309–317.

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

Mara STW, Kuo R, Asih AMS (2021) Location-routing problem: a classification of recent research. *International Transactions in Operational Research* 28(6):2941–2983.

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

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

Nagy G, Salhi S (2007) Location-routing: Issues, models and methods. *European Journal of Operational Research* 177(2):649–672.

Neri F, Cotta C, Moscato P, eds. (2012) *Handbook of Memetic Algorithms*, volume 379 of *Studies in Computational Intelligence* (Springer).

Perboli G, Tadei R, Vigo D (2011) The two-echelon capacitated vehicle routing problem: Models and math-based heuristics. *Transportation Science* 45(3):364–380.

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

Prins C, Prodhon C, Calvo RW (2006a) A memetic algorithm with population management (ma|pm) for the capacitated location-routing problem. Gottlieb J, Raidl GR, eds., *Evolutionary Computation in Combinatorial Optimization*, 183–194 (Berlin, Heidelberg: Springer Berlin Heidelberg).

Prins C, Prodhon C, Calvo RW (2006b) Solving the capacitated location-routing problem by a grasp complemented by a learning process and a path relinking. *4OR* 4(3):221–238.

Prins C, Prodhon C, Ruiz A, Soriano P, Wolfler Calvo R (2007) Solving the capacitated location-routing problem by a cooperative lagrangean relaxation-granular tabu search heuristic. *Transportation Science* 41(4):470–483.

Prodhon C, Prins C (2014) A survey of recent research on location-routing problems. *European Journal of Operational Research* 238(1):1–17.

Quintero-Araujo CL, Caballero-Villalobos JP, Juan AA, Montoya-Torres JR (2017) A biased-randomized metaheuristic for the capacitated location routing problem. *International Transactions in Operational Research* 24(5):1079–1098.

Schneider M, Drexl M (2017) A survey of the standard location-routing problem. *Annals of Operations Research* 259(1):389–414.

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

Shaw P (1998) Using constraint programming and local search methods to solve vehicle routing problems. Maher M, Puget JF, eds., *Principles and Practice of Constraint Programming — CP98*, 417–431 (Berlin, Heidelberg: Springer Berlin Heidelberg).

Sobhanan A, Park J, Park J, Kwon C (2024) Genetic algorithms with neural cost predictor for solving hierarchical vehicle routing problems. *Transportation Science* https://doi.org/10.1287/trsc.2023.0369.

Ting CJ, Chen CH (2013) A multiple ant colony optimization algorithm for the capacitated location routing problem. *International Journal of Production Economics* 141(1):34–44.

Toledo CFM, de Oliveira RRR, França PM (2013) A hybrid multi-population genetic algorithm applied to solve the multi-level capacitated lot sizing problem with backlogging. *Computers & Operations Research* 40(4):910–919.

Tuzun D, Burke LI (1999) A two-phase tabu search approach to the location routing problem. *European Journal of Operational Research* 116(1):87–99.

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

Vincent FY, Lin SW, Lee W, Ting CJ (2010) A simulated annealing heuristic for the capacitated location routing problem. *Computers & Industrial Engineering* 58(2):288–299.

Voigt S, Frank M, Fontaine P, Kuhn H (2022) Hybrid adaptive large neighborhood search for vehicle routing problems with depot location decisions. *Computers & Operations Research* 146:105856.

Submitted to *INFORMS Journal on Computing*
manuscript (Please, provide the manuscript number!)

Online Supplement for "A Hybrid Genetic Algorithm with Multi-population for Capacitated Location Routing"

Pengfei He
School of Automation and Key Laboratory of Measurement and Control of Complex Systems of Engineering, Ministry of Education, Southeast University, Nanjing 210096, China.
pengfeihe606@gmail.com

Jin-Kao Hao* (Corresponding author)
Department of Computer Science, LERIA, University of Angers, 2 Boulevard Lavoisier, 49045 Angers, France
jin-kao.hao@univ-angers.fr

Qinghua Wu* (Corresponding author)
School of Management, Huazhong University of Science and Technology, No. 1037, Luoyu Road, Wuhan, China.
qinghuawu1005@gmail.com

This online supplement provides nomenclature, coverage ratio heuristic and detailed computational results of the HGAMP algorithm and its main reference algorithms on benchmark instances.

Key words: Location-routing; Multi-population based search; Multi-depot edge assembly crossover; Neighborhood search; Depot configurations.

1. Nomenclature

Table 1 summarizes the symbols used in the paper.

Table 1 Nomenclature

Parameter	Description
$\mathcal{I}$	The set of depots
$\mathcal{J}$	The set of customers
$\mathcal{V}$	The set of vertices and $\mathcal{V}=\mathcal{I}\cup\mathcal{J}$
$\mathcal{E}$	The set of edges in graph $\mathcal{G}$
$\mathcal{D}$	The set of depot configurations
$\mathcal{D}_i$	The ith depot configuration
$\mathcal{D}_\varphi$	The depot configuration associated with solution φ
$\mathcal{P}$	The population
$\mathcal{P}_i$	The ith subpopulation composed of solutions associated to the same depot configuration $\mathcal{D}_i$
γ	The number of top promising depot configurations to initialize the population
$\mathcal{P}_{\gamma+1}$	The last subpopulation whose solutions use depot configurations that are different from those of $\mathcal{D}$
u_i	The rough cost of depot i, which is determined as the sum of the open cost o_i, the fixed utilization cost per vehicle used and the travel cost
$\mathcal{L}$	A list used to save depots and sorted by the rough costs of the depots
μ	The minimal size of each subpopulation
λ	The generation size of each subpopulation
α	The granularity threshold
φ^*	The best solution found so far
φ_O	An offspring solution
β	The maximum number of offspring solutions generated by an application of the mdEAX crossover

2. Coverage ratio heuristic

The CRH uses two filters to find good initial depot configurations. The preliminary filter identifies a set of depot configurations by considering estimated travel costs and geographic dispersion of the candidate depots, from which the secondary filter retains a reduced set of most promising configurations.

The preliminary filter uses minimum spanning trees (MST) to estimate the travel costs of the candidate depots without constructing complete solutions (Schneider and Löffler 2019). Specifically, Prim's algorithm is used to build a MST from each depot $i \in \mathcal{I}$ (as the root node) with respect to the capacity w_i. The customers covered by the MST are saved in set $\mathcal{S}_i$, which can be thought of as the coverage area for depot i. The rough cost u_i of depot i is determined as the sum of the open cost o_i, the fixed cost of the vehicles needed to service the customers in $\mathcal{S}_i$, and the total travel cost of the MST.

To build a new depot configuration $\mathcal{D}^{'}$, a random depot is used to initialize $\mathcal{D}^{'}$. Let $\hat{\mathcal{S}} = \cup_{j\in\mathcal{D}'}\mathcal{S}_j$, which is the set of customers served by the currently selected depots. For a candidate depot i, the overlap coverage ratio r_i of depot i with respect to $\hat{\mathcal{S}}$ is then calculated as the proportion of customers that overlap between $\hat{\mathcal{S}}$ and $\mathcal{S}_i$ ($r_i = |\hat{\mathcal{S}} \cap \mathcal{S}_i|/|\hat{\mathcal{S}} \cup \mathcal{S}_i|$). Intuitively, balancing rough costs and geographic dispersion is essential for identifying promising depot configurations that minimize depot costs and avoid repeatedly covering the same customers. Ideally, the ratio should be as small as possible to avoid that the same customers are covered by several selected depots. Each candidate depot i such that $r_i < r_b$ is added to a list $\mathcal{L}$, sorted based on their rough costs, where $r_b = (r_{max} - r_{min}) \times ((w_i + T_c)/T) + r_{min}$, with r_{min} and r_{max} being the minimum and maximum overlap coverage ratio, respectively, T being the total demand of all customers, and T_c being the cumulative total capacity of the selected depots increases. To extend the current depot configuration $\mathcal{D}^{'}$, depots from $\mathcal{L}$ are greedily added. The process stops when the total capacity T_c of the selected depots in $\mathcal{D}^{'}$ reaches T and $\mathcal{D}^{'}$ is added to the set of promising depot configurations $\mathcal{D}$. The preliminary filter imposes a maximum limit of depot configurations (D_{max}) to terminate the search. Without loss of generality, all parameters are independent of the instance size. We set $r_{min} = 0.1$, $r_{max} = 0.6$, $D_{max} = 1000$. Based on our preliminary tests, the running time of the preliminary filter is negligible.

The preliminary filter generates a large set $\mathcal{D}$ of depot configurations ($|\mathcal{D}| = D_{max} = 1000$). The secondary filter is used to identify a reduced set of most promising initial depot configurations for the algorithm. Each candidate configuration $\mathcal{D}_i$ is used to generate a limited number N_t (a parameter, fixed to 10) of solutions, which are improved by the local search procedure of Section 3.5.2. We then calculate the average cost of these N_t improved solutions, which is used to assess the attractiveness of the depot configuration. Based on

the attractiveness values, the first γ top depot configurations are retained in $\mathcal{D}$ and used to initialize the subpopulations of the population.

3. Computational results

3.1. Results of the CLRP

This section presents the detailed computational results of the proposed HGAMP algorithm for the three classic benchmark sets $\mathbb{B}$, $\mathbb{P}$, $\mathbb{T}$, and the more recent set $\mathbb{S}$, under the experimental condition given in Section 4.2, together with the results of the reference algorithms PF (Arnold and Sörensen 2021), GANCP$^+$ (Sobhanan et al. 2024) (only on sets $\mathbb{B}$ and $\mathbb{T}$), HALNS (Voigt et al. 2022), TBSA$_{basic}$ (Schneider and Löffler 2019), and TBSA$_{quality}$ (Schneider and Löffler 2019) (only on sets $\mathbb{B}$, $\mathbb{P}$ and $\mathbb{T}$). Like Voigt et al. (2022), Schneider and Löffler (2019), we report the following statistics, especially including the best and average results in Tables 2-5. Column Instances indicates the name of each instance; column BKS shows the best-known results (best upper bounds) summarized from the literature, including both exact algorithms (Liguori et al. 2023, Contardo et al. 2014) and heuristics (Arnold and Sörensen 2021, Voigt et al. 2022, Schneider and Löffler 2019); Best and Avg. are the best and average results over multiple independent runs (except for PF, which was run only one time per instance); Time indicates the average runtime in seconds of each algorithm.

Tables 2 and 3 show the results of the compared algorithms on sets $\mathbb{B}$, $\mathbb{P}$, and $\mathbb{T}$. $\delta_1(\%)$ in the last column is the gap of the best result of HGAMP to the BKS, which is calculated as $\delta_1 = 100 \times (f_{best} - BKS)/BKS$, where f_{best} is the best objective value of HGAMP.

Tables 4 and 5 show the results on set $\mathbb{S}$. In these tables, columns δ_1, δ_2 and δ_3 indicate the gap of our best result to the BKS, TBSA$_{basic}$ and PF, respectively, which are calculated as $\delta_1 = 100 \times (f_{best} - BKS)/BKS$, $\delta_2 = 100 \times (f_{best} - f^t_{best})/f^t_{best}$ and $\delta_3 = 100 \times (f_{best} - f^p_{best})/f^p_{best}$, where f_{best} is the best objective value of HGAMP, f^t_{best} and f^p_{best} are the best objective values of TBSA$_{basic}$ and PF, respectively.

In all tables, the *Average* row is the average value of a performance indicator over the instances of a benchmark set. Improved BKS values (new upper bounds) are indicated by negative $\delta_1(\%)$ values highlighted in bold. Except for set $\mathbb{B}$ in Table 2, for which most algorithms achieve the same value for each instance, the dark gray color indicates that the corresponding algorithm obtains the best result among the compared algorithms on the corresponding instance; the medium gray color indicates the second best results, and so on. Unavailable results are indicated by the '-' symbol.

Comparing the results of HGAMP with the BKS and the results of the reference algorithms shows that HGAMP performs very well on the four benchmark sets, especially on the most difficult instances of set $\mathbb{S}$. Remarkably, HGAMP finds improved best results (new upper bounds) for one instance of set $\mathbb{P}$, two instances of $\mathbb{T}$, and 100 instances of $\mathbb{S}$.

3.2. Results of applying the HGAMP algorithm to the MDVRP

To demonstrate the generality of the proposed HGAMP algorithm, we apply the algorithm to the multi-depot vehicle routing problem (MDVRP) (Cordeau et al. 1997), which can be considered as a special case of the CLRP when the opening cost of each depot is zero. All depots are assumed to have unlimited capacity and each depot $i \in \mathcal{I}$ has a fleet of size k_i with the same capacity. There is no fixed utilization cost for the vehicles. The objective of the MDVRP is to select a subset of vehicles for each depot and to construct routes of the vehicles, while respecting the capacity and minimizing the total travel cost.

To solve the MDVRP, it is sufficient to make very small changes to the HGAMP algorithm. In particular, CRH is discarded because there is no fixed cost of opening depots. Instead, we use a simple greedy procedure to generate each initial solution. Customers are placed on routes from different depots one at a time in such a way that the travel cost increases the least. When all customers are inserted into routes, a solution is obtained and the depots

associated with routes in the solution form the depot configuration. The initial solution is then inserted into a subpopulation with respect to its depot configuration. Once the number of depot configurations is greater than γ, the population management of Section 3.6 is triggered to update the depot configurations. The other components of the HGAMP algorithm, including the mdEAX crossover, the repair procedure, the mutation operator, the local search procedure, and the population management, remain unchanged.

The MDVRP has been extensively studied in the literature and many state-of-the-art heuristics have been proposed, including HGSADC (Vidal et al. 2012), AVXS (Accorsi and Vigo 2020), HALNS (Voigt et al. 2022), and the very recent GANCP$^+$ (Sobhanan et al. 2024). Following Accorsi and Vigo (2020), we test our HGAMP algorithm on the set of 11 instances with 50-360 customers and 4-9 depots (https://neo.lcc.uma.es/vrp/vrp-instances/multiple-depot-vrp-instances/). Note that among the MDVRP instances, only these 11 instances do not impose a duration constraint on vehicle routes, as in the CLRP, while the route duration constraint is imposed on the other instances. We run HGAMP 10 times to solve each instance with the parameter settings shown in Section 4.2.1. Note that the compared algorithms except GANCP$^+$ can all achieve the best-known solutions (BKS) for these instances. So the comparison focuses mainly on the average values.

The comparative results are shown in Table 6, where $\delta = 100 \times (f_{best} - BKS)/BKS$ for GANCP$^+$ indicates the gap of the best objective value f_{best} of GANCP$^+$ to the BKS, and $\delta_a = 100 \times (f_{avg} - BKS)/BKS$ for the other algorithms indicates the gap of the averaged best objective value f_{avg} of each of these algorithms to the BKS. The results of the reference algorithms are taken from the corresponding papers, and the runtimes are scaled by the Passmark scores (https://www.cpubenchmark.net/singleThread.html), except for GANCP$^+$, which uses a GPU.

From Table 6, we see that our HGAMP algorithm achieves the BKS for all 11 instances, as do HGSADC, AVXS, and HALNS, while GANCP$^+$ is dominated by the other algorithms. For 9 instances, HGAMP achieves the BKS for each of its runs, versus zero instance for GANCP$^+$, 8 instances for HGSADC, 9 instances for AVXS and HALNS. HGAMP has the same global average gap (0.03%) to the BKS as HGSADC, which is slightly worse than the gap of AVXS (0.02%) and the gap of HALNS (0.01%). For 7 instances, HGAMP finds its best results quickly (2 to 60 seconds), while for the remaining 4 instances it takes more than 100 seconds (with an exceptionally long time of 5698 seconds for p21), leading to a global runtime higher than the reference algorithms. In summary, although our algorithm is not specifically designed to solve the MDVRP, it remains competitive with dedicated and state-of-the-art MDVRP algorithms in terms of the best and average results.

References

Accorsi L, Vigo D (2020) A hybrid metaheuristic for single truck and trailer routing problems. *Transportation Science* 54(5):1351–1371.

Arnold F, Sörensen K (2021) A progressive filtering heuristic for the location-routing problem and variants. *Computers & Operations Research* 129:105166.

Contardo C, Cordeau JF, Gendron B (2014) An exact algorithm based on cut-and-column generation for the capacitated location-routing problem. *INFORMS Journal on Computing* 26(1):88–102.

Cordeau JF, Gendreau M, Laporte G (1997) A tabu search heuristic for periodic and multi-depot vehicle routing problems. *Networks: An International Journal* 30(2):105–119.

Liguori P, Mahjoub AR, Marques G, Uchoa E, Sadykov R (2023) Nonrobust strong knapsack cuts for capacitated location-routing and related problems. *Operations Research* 71(5):1577–1595.

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

Sobhanan A, Park J, Park J, Kwon C (2024) Genetic algorithms with neural cost predictor for solving hierarchical vehicle routing problems. *Transportation Science* https://doi.org/10.1287/trsc.2023.0369.

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

Voigt S, Frank M, Fontaine P, Kuhn H (2022) Hybrid adaptive large neighborhood search for vehicle routing problems with depot location decisions. *Computers & Operations Research* 146:105856.

Table 2 Results for the CLRP on benchmark sets $\mathbb{B}$ and $\mathbb{P}$.

Set	Instances	BKS	PF		GANCP^{+}		HALNS			TBSA$_{basic}$			TBSA$_{quality}$			HGAMP			
			Best	Time	Best	Time	Best	Avg.	Time	Best	Avg.	Time	Best	Avg.	Time	Best	Avg.	Time	δ_1(%)
$\mathbb{B}$	Christofides69-50x5	565.60	565.60	52.00	592.90	2.90	565.60	565.60	54.96	565.60	568.50	10.89	565.60	568.50	60.01	565.60	565.60	18.21	0.00
	Christofides69-75x10	848.90	848.90	69.00	855.80	3.41	848.85	848.85	120.92	848.90	848.90	52.64	848.90	848.90	177.60	848.90	848.85	734.75	0.00
	Christofides69-100x10	833.40	836.70	107.00	839.70	3.13	833.43	833.95	196.76	833.40	834.10	90.32	833.40	834.10	325.82	833.40	833.43	1005.19	0.00
	Daskin95-88x8	355.80	355.80	158.00	355.80	3.18	355.78	355.78	91.74	355.80	355.80	27.71	355.80	355.80	185.91	355.80	355.78	656.02	0.00
	Daskin95-150x10	43919.90	43919.90	274.00	47303.60	7.35	43919.90	43920.60	410.93	43923.50	43981.50	190.85	43919.90	43919.90	772.52	43919.90	43920.08	1617.88	0.00
	Gaskell67-21x5	424.90	424.90	19.00	429.60	3.37	424.90	424.90	24.90	424.90	424.90	2.30	424.90	424.90	12.74	424.90	424.90	5.16	0.00
	Gaskell67-22x5	585.10	585.10	40.00	585.10	3.01	585.10	585.10	23.67	585.10	585.10	2.04	585.10	585.10	11.19	585.10	585.11	5.77	0.00
	Gaskell67-29x5	512.10	512.10	41.00	512.10	2.57	512.10	512.10	29.05	512.10	512.10	3.47	512.10	512.10	20.21	512.10	512.10	7.94	0.00
	Gaskell67-32x5	562.20	562.20	55.00	562.20	3.02	562.22	562.22	34.49	562.20	562.20	4.60	562.20	562.20	23.89	562.20	562.22	237.85	0.00
	Gaskell67-32x5	504.30	504.30	40.00	504.30	2.70	504.33	504.33	32.52	504.30	504.30	3.48	504.30	504.30	20.12	504.30	504.33	282.48	0.00
	Gaskell67-36x5	460.40	460.40	51.00	460.40	2.54	460.37	460.37	35.88	460.40	460.40	4.47	460.40	460.40	28.78	460.40	460.37	298.69	0.00
	Min92-27x5	3062.00	3062.00	40.00	3062.00	2.85	3062.02	3062.02	28.55	3062.00	3062.00	3.14	3062.00	3062.00	17.58	3062.00	3062.02	215.20	0.00
	Min92-134x8	5709.00	5713.00	175.00	6112.00	4.95	5709.00	5709.00	240.03	5709.00	5711.00	106.22	5709.00	5709.00	572.18	5709.00	5709.00	80.57	0.00
	Average	4487.97	4488.53	86.23	4782.73	3.46	4487.97	4488.06	101.88	4488.25	4493.14	38.63	4487.97	4488.25	171.43	4487.97	4487.98	397.36	0.00
$\mathbb{P}$	20-5-1a	54793	54793	20.00	-	-	54793	54793.00	25.61	54793	54793.00	2.97	54793	54793.00	15.84	54793	54793.00	3.30	0.00
	20-5-1b	39104	39104	26.00	-	-	39104	39104.00	21.30	39104	39104.00	2.04	39104	39104.00	11.46	39104	39104.00	2.56	0.00
	20-5-2a	48908	48908	16.00	-	-	48908	48908.00	23.10	48908	48908.00	2.82	48908	48908.00	15.53	48908	48908.00	6.40	0.00
	20-5-2b	37542	37542	26.00	-	-	37542	37542.00	22.61	37542	37542.00	2.06	37542	37542.00	11.41	37542	37542.00	7.75	0.00
	50-5-1	90111	90111	43.00	-	-	90111	90111.00	61.47	90111	90111.00	13.37	90111	90111.00	71.22	90111	90111.00	19.92	0.00
	50-5-1b	63242	63242	62.00	-	-	63242	63242.00	54.47	63242	63242.00	10.54	63242	63242.00	63.66	63242	63242.00	15.83	0.00
	50-5-2	88298	88298	46.00	-	-	88298	88298.00	59.08	88298	88298.00	14.93	88298	88298.00	82.22	88298	88298.00	16.95	0.00
	50-5-2b	67308	67308	58.00	-	-	67308	67308.00	58.93	67308	67308.00	14.46	67308	67308.00	81.49	67308	67308.00	30.01	0.00
	50-5-2BIS	84055	84055	70.00	-	-	84055	84055.00	61.26	84055	84055.00	15.26	84055	84055.00	87.05	84055	84055.00	9.68	0.00
	50-5-2bBIS	51822	51822	56.00	-	-	51822	51822.00	52.94	51822	51822.00	12.25	51822	51822.00	72.64	51822	51822.00	12.38	0.00
	50-5-3	86203	86203	49.00	-	-	86203	86203.00	66.52	86203	86203.00	14.11	86203	86203.00	77.14	86203	86203.00	15.46	0.00
	50-5-3b	61830	61830	62.00	-	-	61830	61830.00	54.99	61830	61830.00	9.74	61830	61830.00	58.35	61830	61830.00	15.65	0.00
	100-5-1	274814	275505	106.00	-	-	275079	275120.60	385.09	274814	275270.80	91.14	274814	274814.00	524.03	275079	275174.80	955.11	0.10
	100-5-1b	213568	213971	95.00	-	-	213568	213588.60	279.77	213886	213925.20	71.21	213568	213598.60	441.35	213568	213742.85	1050.66	0.00
	100-5-2	193671	193671	92.00	-	-	193671	193671.00	210.24	193671	193762.00	69.50	193671	193671.00	383.33	193671	193671.00	117.49	0.00
	100-5-2b	157095	157110	82.00	-	-	157095	157095.00	186.10	157110	157151.20	49.83	157095	157142.80	319.71	157095	157101.35	506.15	0.00
	100-5-3	200079	200237	73.00	-	-	200079	200079.00	247.28	200079	200165.80	91.25	200079	200079.00	460.56	200079	200199.20	722.65	0.00
	100-5-3b	152441	152441	75.00	-	-	152441	152441.00	138.93	152441	152441.00	48.53	152441	152441.00	267.60	152441	152441.00	180.75	0.00
	100-10-1	287661	288395	138.00	-	-	287692	287870.00	394.79	287716	287806.60	100.21	287668	287735.80	486.79	287723	288424.00	1307.08	0.02
	100-10-1b	230989	232342	109.00	-	-	230989	230989.00	222.90	230989	231551.80	62.57	230989	230991.60	318.66	230989	231808.00	1377.41	0.00
	100-10-2	243590	243590	89.00	-	-	243590	243611.00	254.71	243590	243636.60	86.55	243590	243590.00	400.73	243590	243605.75	562.96	0.00
	100-10-2b	203988	203988	91.00	-	-	203988	203988.00	146.75	203988	203988.00	52.46	203988	203988.00	284.61	203988	203988.00	66.50	0.00
	100-10-3	250882	252918	122.00	-	-	250882	250945.60	332.39	250882	250991.40	93.67	250882	250947.00	473.83	250882	251545.45	1238.02	0.00
	100-10-3b	203114	204567	109.00	-	-	203114	203404.60	207.38	203114	203114.80	56.69	203114	203114.00	323.77	203114	203916.00	1304.88	0.00
	200-10-1	474702	474937	228.00	-	-	476472	477174.20	1478.59	475236	476350.60	683.59	474920	475304.00	2522.07	475165	476362.30	2148.43	0.10
	200-10-1b	375177	376485	225.00	-	-	375346	375512.60	1031.60	375703	376378.40	322.93	375721	376096.20	1536.77	375353	376400.35	2390.80	0.05
	200-10-2	448077	448850	223.00	-	-	448721	449138.60	1283.05	448378	448813.20	581.68	448135	448181.60	2609.55	448978	449185.25	2234.70	0.20
	200-10-2b	373696	374266	217.00	-	-	373696	373706.20	951.11	373886	374113.60	332.53	373796	373958.60	1639.86	373696	373922.80	2498.86	0.00
	200-10-3	469433	470978	261.00	-	-	470422	471149.00	1197.06	469493	470082.20	675.33	469461	469476.20	3726.44	469433	470061.30	2308.35	0.00
	200-10-3b	362320	362841	269.00	-	-	362630	362744.40	1091.56	362609	362842.80	343.21	362599	362783.60	1407.39	362253	362506.61	2312.16	**-0.02**
	Average	196283.77	196676.93	104.60	-	-	196423.03	196514.81	353.39	196360.03	196520.07	130.91	196324.90	196370.97	625.84	196343.77	196575.73	781.29	0.01

Table 3 Results for the CLRP on the instances of set $\mathbb{T}$.

Instances	BKS	AVXS		PF		GANCP$^+$		HALNS			TBSA$_{basic}$			TBSA$_{quality}$			HGAMP		
		Best	Avg.	Time	Best	Time	Best	Time	Best	Avg.	Time	Best	Avg.	Time	Best	Avg.	Time	Best	Avg.	Time	δ_1(%)
P111112	1467.68	1467.68	1467.68	110.00	1468.29	92.00	1500.42	3.35	1467.68	1467.68	163.22	1467.68	1467.80	101.07	1467.68	1467.68	345.07	1467.68	1467.68	1306.81	0.00
P111122	1448.37	1448.37	1448.80	183.00	1448.37	95.00	1453.89	2.99	1448.37	1448.54	239.21	1449.20	1451.64	115.72	1448.37	1448.70	369.68	1448.37	1448.78	1399.71	0.00
P111212	1394.8	1394.80	1394.80	100.00	1394.8	88.00	1427.53	4.01	1394.80	1394.80	162.94	1394.80	1395.15	98.52	1394.80	1396.36	326.02	1394.80	1394.80	1253.60	0.00
P111222	1432.29	1432.29	1435.30	198.00	1432.29	83.00	1435.36	3.97	1432.29	1432.60	234.86	1432.29	1432.29	112.78	1432.29	1432.29	336.30	1432.29	1437.06	1342.26	0.00
P112112	1167.16	1167.16	1168.56	588.00	1167.16	117.00	1217.84	3.35	1167.16	1167.16	184.35	1167.16	1167.24	90.10	1167.16	1167.24	354.64	1167.16	1167.16	1194.34	0.00
P112122	1102.24	1102.24	1102.24	584.00	1102.24	119.00	1102.24	3.50	1102.24	1102.24	151.51	1102.24	1102.24	98.97	1102.24	1102.24	354.15	1102.24	1102.24	1427.75	0.00
P112212	791.66	791.66	791.74	124.00	791.66	103.00	791.66	3.21	791.66	791.66	168.88	791.66	791.96	76.13	791.66	791.66	324.80	791.66	793.56	1050.53	0.00
P112222	728.3	728.30	728.30	131.00	728.3	105.00	728.30	3.15	728.30	728.30	137.87	728.30	728.30	74.47	728.30	728.30	294.14	728.30	728.30	1178.08	0.00
P113112	1238.24	1238.49	1238.86	542.00	1238.67	105.00	1285.25	3.30	1238.49	1238.49	197.30	1238.49	1238.74	102.83	1238.49	1238.56	349.72	1238.49	1238.49	994.84	0.02
P113122	1245.3	1245.30	1245.30	438.00	1245.31	93.00	1255.51	3.32	1245.31	1245.46	150.63	1245.31	1245.78	106.57	1245.31	1245.31	390.48	1245.31	1246.02	1332.28	0.00
P113212	902.26	902.26	902.26	433.00	902.34	110.00	902.26	3.28	902.26	902.26	141.88	902.26	902.28	75.22	902.26	902.26	311.39	902.26	902.26	883.29	0.00
P113222	1018.29	1018.29	1018.29	499.00	1018.29	99.00	1051.20	3.62	1018.29	1020.09	154.53	1018.29	1021.89	83.64	1018.29	1019.19	331.00	1018.29	1021.89	1145.65	0.00
P121112	2237.73	2238.40	2252.50	863.00	2245.07	213.00	2307.14	3.34	2238.59	2245.18	856.79	2240.06	2244.74	600.63	2238.52	2246.83	1773.32	2239.39	2271.11	2960.89	0.07
P121122	2137.45	2140.66	2159.25	855.00	2174.72	206.00	2264.78	4.87	2137.45	2140.31	1011.30	2139.64	2149.93	741.00	2138.38	2145.87	1801.37	2137.45	2142.87	3402.25	0.00
P121212	2195.17	2195.17	2209.66	883.00	2204.52	193.00	2257.68	3.69	2195.17	2199.60	858.64	2202.19	2210.42	644.43	2202.67	2206.69	2110.47	2202.19	2207.80	3072.83	0.32
P121222	2214.86	2226.16	2246.75	981.00	2224.02	211.00	2276.02	3.77	2214.86	2216.20	1014.78	2220.92	2235.00	660.88	2214.86	2219.90	2181.95	2214.86	2227.72	3334.95	0.00
P122112	2070.43	2083.06	2097.55	939.00	2074.28	312.00	2096.32	3.25	2070.43	2072.01	861.55	2073.28	2075.75	606.76	2072.92	2076.19	1947.71	2070.43	2073.62	3174.26	0.00
P122122	1685.52	1700.02	1715.19	1039.00	1693.53	309.00	1712.32	4.34	1685.52	1685.69	1029.48	1685.65	1685.65	716.85	1685.65	1685.77	1974.23	1685.52	1695.18	3640.74	0.00
P122212	1449.62	1467.33	1470.81	936.00	1449.91	284.00	1469.70	3.08	1449.62	1449.77	1067.59	1452.00	1452.52	553.85	1449.93	1450.58	1703.98	1449.46	1457.29	2758.02	**-0.01**
P122222	1082.46	1082.68	1083.22	937.00	1083.16	203.00	1160.78	3.51	1082.59	1082.59	842.09	1082.46	1083.11	563.65	1082.46	1082.64	1600.30	1082.46	1082.47	2785.39	0.00
P123112	1942.23	1969.03	1974.08	965.00	1961.49	215.00	1974.40	4.36	1949.95	1952.82	1047.41	1943.48	1957.87	564.04	1943.10	1946.35	1746.15	1946.55	1960.90	2998.85	0.22
P123122	1910.08	1911.80	1920.20	1064.00	1921.03	220.00	2102.32	4.58	1910.08	1913.17	1005.32	1922.74	1926.76	592.04	1910.66	1915.65	1618.94	1911.60	1918.51	3389.58	0.08
P123212	1760.2	1767.10	1773.97	1081.00	1800.08	242.00	1778.96	3.33	1760.20	1761.14	1118.70	1761.80	1764.67	495.61	1763.20	1764.39	1565.78	1760.04	1761.53	2669.22	**-0.01**
P123222	1390.74	1390.74	1391.00	736.00	1391.99	184.00	1423.86	4.93	1390.74	1390.79	663.32	1391.57	1392.53	485.39	1391.46	1392.10	1484.46	1390.74	1390.81	2866.03	0.00
P131112	1892.17	1902.77	1912.61	704.00	1897.92	124.00	1944.22	3.42	1892.17	1893.86	481.69	1897.34	1904.22	302.52	1895.48	1900.59	1104.45	1892.17	1905.83	1897.68	0.00
P131122	1819.68	1824.23	1838.97	770.00	1820.32	126.00	1880.57	4.64	1819.68	1822.70	473.25	1819.68	1820.47	314.86	1819.68	1821.16	901.59	1819.68	1824.41	2299.97	0.00
P131212	1960.02	1964.72	1965.31	420.00	1960.02	132.00	2038.57	3.18	1964.34	1965.40	406.93	1960.02	1966.73	281.49	1960.02	1963.17	989.23	1965.12	1966.04	1945.58	0.26
P131222	1792.77	1792.77	1797.43	688.00	1792.77	120.00	1879.86	3.45	1792.77	1795.23	377.68	1795.84	1799.08	257.95	1795.85	1796.51	879.03	1792.77	1800.23	2220.89	0.00
P132112	1443.32	1443.32	1446.50	808.00	1448.81	172.00	1531.33	3.90	1443.32	1443.32	368.65	1443.32	1444.71	257.53	1443.32	1444.20	776.99	1443.32	1444.16	1698.19	0.00
P132122	1429.3	1441.59	1447.02	875.00	1444.77	209.00	1446.68	3.54	1429.30	1439.00	441.85	1429.42	1439.34	326.58	1429.52	1434.95	941.75	1431.20	1441.67	2258.12	0.13
P132212	1204.42	1205.62	1206.83	699.00	1204.64	166.00	1204.42	3.01	1204.42	1204.42	427.53	1204.42	1204.89	243.44	1204.62	1204.74	880.01	1204.42	1204.42	1680.55	0.00
P132222	924.68	931.15	931.98	751.00	931.55	140.00	998.24	3.05	924.68	924.68	484.31	924.95	925.38	265.51	925.05	925.77	738.18	924.89	929.63	1769.45	0.02
P133112	1694.18	1695.54	1704.01	705.00	1709.34	157.00	1757.34	3.97	1694.68	1698.60	530.37	1694.85	1704.36	250.61	1694.18	1697.10	828.29	1694.18	1694.18	1686.77	0.00
P133122	1392.01	1397.16	1400.36	840.00	1400.5	159.00	1408.83	4.01	1392.01	1393.74	473.67	1392.18	1401.11	294.79	1392.18	1399.46	882.56	1392.01	1394.83	1946.84	0.00
P133212	1197.95	1197.95	1198.07	420.00	1200.14	121.00	1217.50	3.08	1197.95	1197.95	478.82	1198.08	1198.34	295.43	1198.08	1198.60	877.64	1197.95	1199.50	1563.17	0.00
P133222	1151.37	1152.18	1153.10	725.00	1156.61	164.00	1214.22	3.99	1151.69	1151.78	385.31	1151.83	1155.65	242.53	1151.83	1153.63	771.61	1151.80	1153.77	1985.42	0.04
Average	1497.638	1501.61	1506.62	655.94	1503.58	160.86	1541.60	3.65	1498.03	1499.42	522.06	1499.04	1502.46	324.84	1498.35	1500.35	1004.65	1498.25	1502.69	2069.86	0.03

Table 4 Results for the CLRP on the instances of set $\mathbb{S}$.

Instances	BKS	TBSA$_{basic}$			PF		HGAMP					
		Best	Avg.	Time	Best	Time	Best	Avg.	Time	δ_1(%)	δ_2(%)	δ_3(%)
100-5-1c	134516	134516.00	134603.80	81.24	134687	112.00	134516	134516.00	1070.08	0.00	0.00	-0.13
100-5-1d	275749	275749.00	275793.40	52.71	276154	122.00	275749	275749.95	949.90	0.00	0.00	-0.15
100-5-1e	292301	292311.00	292400.40	68.68	292565	159.00	292311	293223.90	1104.12	0.00	0.00	-0.09
100-5-2c	83855	83989.00	84234.20	67.20	85051	81.00	83855	83982.30	1069.67	0.00	-0.16	-1.41
100-5-2d	242105	242266.00	242411.40	52.39	242739	101.00	242105	242105.00	1099.48	0.00	-0.07	-0.26
100-5-2e	253888	253888.00	254063.80	65.86	254085	169.00	254025	254025.00	2170.79	0.05	0.05	-0.02
100-5-3c	87555	87555.00	87606.40	54.64	87555	66.00	87555	87555.00	949.37	0.00	0.00	0.00
100-5-3d	226634	226783.00	226846.00	48.91	226920	74.00	226752	226771.90	891.66	0.05	-0.01	-0.07
100-5-3e	252603	252603.00	252661.00	78.68	252677	119.00	252603	252634.95	1435.91	0.00	0.00	-0.03
100-5-4a	255853	255853.00	255892.00	52.14	255869	142.00	255853	255860.50	1087.49	0.00	0.00	-0.01
100-5-4b	214425	214425.00	214425.00	29.79	214531	107.00	214425	214425.00	1067.61	0.00	0.00	-0.05
100-5-4c	98104	98129.00	98187.60	52.91	98199	137.00	98104	98109.70	1091.08	0.00	-0.03	-0.10
100-5-4d	250301	250315.00	250778.60	45.68	251380	131.00	250921	250999.90	1061.54	0.25	0.24	-0.18
100-5-4e	211113	211159.00	211217.60	58.30	211444	205.00	211113	211251.25	1651.25	0.00	-0.02	-0.16
100-10-1c	92629	92629.00	92683.80	98.15	92979	77.00	92629	92629.00	1284.57	0.00	0.00	-0.38
100-10-1d	363930	363930.00	364172.00	60.95	363930	102.00	363930	363937.35	1132.06	0.00	0.00	0.00
100-10-1e	344322	344322.00	344583.00	60.10	344897	115.00	344322	344833.55	1113.86	0.00	0.00	-0.17
100-10-2c	84717	84717.00	84744.20	98.74	84817	86.00	84717	84717.60	1169.20	0.00	0.00	-0.12
100-10-2d	343252	343252.00	343252.00	55.86	343252	97.00	343252	343252.00	1254.75	0.00	0.00	0.00
100-10-2e	332900	332900.00	333181.20	60.66	333778	123.00	333599	333755.75	1000.13	0.21	0.21	-0.05
100-10-3c	85369	85618.00	85711.20	97.36	85369	79.00	85369	85371.20	1129.23	0.00	-0.29	0.00
100-10-3d	329990	329990.00	330025.20	51.56	329990	118.00	329990	330022.30	1160.81	0.00	0.00	0.00
100-10-3e	318109	318156.00	318270.20	63.86	318226	137.00	318109	318245.75	1185.29	0.00	-0.01	-0.04
100-10-4a	253471	253892.00	254207.40	47.77	253471	142.00	253471	253504.60	1333.12	0.00	-0.17	0.00
100-10-4b	211354	211354.00	211358.20	39.03	211361	125.00	211354	211354.00	1339.73	0.00	0.00	0.00
100-10-4c	86215	86215.00	86219.00	74.05	87277	104.00	86215	86215.00	1320.30	0.00	0.00	-1.22
100-10-4d	328181	328251.00	328344.00	63.21	328420	131.00	328231	328395.30	1381.03	0.02	-0.01	-0.06
100-10-4e	308757	308866.00	309298.60	71.84	310134	169.00	309126	309863.85	1320.10	0.12	0.08	-0.33
200-10-1c	156029	156087.00	156771.40	589.75	157428	199.00	156029	156173.70	3218.40	0.00	-0.04	-0.89
200-10-1d	638068	638452.00	640099.00	417.52	638372	249.00	638356	639304.15	2589.80	0.05	-0.02	0.00
200-10-1e	599069	599463.00	599708.20	463.54	600954	330.00	599719	601175.20	2520.99	0.11	0.04	-0.21
200-10-2c	144046	144337.00	144374.40	476.29	144666	185.00	144337	144351.90	2755.80	0.20	0.00	-0.23
200-10-2d	663154	663814.00	664241.20	342.97	664234	220.00	663509	664088.55	2801.01	0.05	-0.05	-0.11
200-10-2e	618858	619037.00	619162.00	318.10	619262	249.00	618932	619535.25	2834.42	0.01	-0.02	-0.05
200-10-3c	184783	184885.00	185913.80	588.15	186112	247.00	185681	186004.60	2650.80	0.49	0.43	-0.23
200-10-3d	640289	640357.00	640423.40	384.69	641424	248.00	640487	641420.95	2542.37	0.03	0.02	-0.15
200-10-3e	604480	604617.00	605285.60	388.10	606919	252.00	606211	606466.55	2509.29	0.29	0.26	-0.12
200-10-4a	452430	452870.00	453126.60	315.54	453435	297.00	452752	453020.30	2996.73	0.07	-0.03	-0.15
200-10-4b	369821	369951.00	370228.20	215.17	369821	271.00	369580	369964.50	3254.36	**-0.07**	-0.10	-0.07
200-10-4c	144013	144407.00	144607.80	611.48	144940	231.00	144326	144502.10	3322.80	0.22	-0.06	-0.42
200-10-4d	617932	618590.00	619015.80	369.70	618795	279.00	618116	618829.70	3105.39	0.03	-0.08	-0.11
200-10-4e	562854	562854.00	563419.80	403.10	564383	318.00	562843	563916.30	2959.38	**0.00**	0.00	-0.27
200-15-1a	460430	461203.00	461780.80	589.02	462359	236.00	460601	461951.50	2728.23	0.04	-0.13	-0.38
200-15-1b	366359	367397.00	367767.80	274.24	367330	232.00	366779	367018.95	2781.47	0.11	-0.17	-0.15
200-15-1c	148141	148218.00	149190.20	692.08	150091	215.00	148830	149129.30	3693.22	0.47	0.41	-0.84
200-15-1d	813576	813941.00	814570.00	593.25	814072	248.00	813576	814856.20	2754.53	0.00	-0.04	-0.06
200-15-1e	708585	708837.00	719810.80	623.89	709259	326.00	708855	709730.05	2833.86	0.04	0.00	-0.06
200-15-2a	513512	513893.00	514058.60	626.51	514199	265.00	513722	514267.30	3046.47	0.04	-0.03	-0.09
200-15-2b	406839	406843.00	407128.20	271.17	407449	265.00	406685	406911.60	3045.91	**-0.04**	-0.04	-0.19
200-15-2c	134779	135051.00	135505.60	466.28	135633	230.00	134871	135343.75	3598.71	0.07	-0.13	-0.56
200-15-2d	811361	811722.00	812486.20	543.50	813280	249.00	812721	813168.50	2854.02	0.17	0.12	-0.07
200-15-2e	712524	712524.00	713160.40	569.33	737081	1242.00	712978	714622.20	3325.88	0.06	0.06	-3.27
200-15-3a	455351	455676.00	456095.00	557.51	456081	258.00	455368	455655.10	2871.05	0.00	-0.07	-0.16
200-15-3b	356887	357086.00	357832.40	217.93	357233	249.00	356887	357262.05	2932.98	0.00	-0.06	-0.10
200-15-3c	140765	141129.00	141557.80	536.33	141703	223.00	140765	141105.45	3740.99	0.00	-0.26	-0.66
200-15-3d	877543	877638.00	878404.00	501.71	878358	242.00	877940	878403.10	2550.67	0.05	0.03	-0.05
200-15-3e	816129	816377.00	817017.00	524.26	816579	295.00	816001	816424.20	2722.50	**-0.02**	-0.05	-0.07
200-15-4a	432672	433268.00	433677.60	580.28	434165	362.00	432913	433342.25	3260.14	0.06	-0.08	-0.29
200-15-4b	349269	349269.00	349794.00	274.35	350509	339.00	349088	349575.35	3223.68	**-0.05**	-0.05	-0.41
200-15-4c	143052	143772.00	144072.80	558.34	144536	256.00	143517	143742.70	3806.64	0.33	-0.18	-0.71
200-15-4d	826829	828144.00	828332.80	546.97	828711	296.00	827346	827945.80	2947.43	0.06	-0.10	-0.16
200-15-4e	700013	700202.00	701066.60	743.54	701825	394.00	700521	701677.70	3050.35	0.07	0.05	-0.19
300-15-1a	854503	856267.00	857193.80	2084.63	856306	379.00	856023	857758.85	4775.71	0.18	-0.03	-0.03
300-15-1b	622412	622412.00	624326.40	1386.62	623644	399.00	621894	623318.80	5066.06	**-0.08**	-0.08	-0.28
300-15-1c	364979	366770.00	368123.00	1613.18	366675	455.00	365206	365627.15	6459.06	0.06	-0.43	-0.40
300-15-1d	1338255	1339010.00	1339804.40	1556.58	1341270	381.00	1337930	1339096.84	4299.79	**-0.02**	-0.08	-0.25
300-15-1e	1217690	1217690.00	1222415.80	1949.12	1219197	553.00	1218900	1220710.50	4739.73	0.10	0.10	-0.02
300-15-2a	757931	759999.00	760278.40	2135.43	762089	367.00	760081	760846.35	5264.95	0.28	0.01	-0.26
300-15-2b	557525	557912.00	559174.80	1250.03	557525	366.00	556948	557636.00	4935.24	**-0.10**	-0.17	-0.10
300-15-2c	310061	311558.00	312637.40	1465.10	312374	457.00	310673	311034.15	6298.85	0.20	-0.28	-0.54
300-15-2d	1301210	1301863.00	1303445.20	1578.69	1305933	352.00	1301720	1302798.50	4937.30	0.04	-0.01	-0.32
300-15-2e	1272700	1272700.00	1276006.40	1853.13	1276145	575.00	1275250	1276915.50	4914.91	0.20	0.20	-0.07
300-15-3a	776531	778023.00	778382.00	1896.65	778590	354.00	776857	778157.15	5245.62	0.04	-0.15	-0.22
300-15-3b	593892	594073.00	595017.80	1064.88	593892	348.00	593743	594004.10	4857.43	**-0.03**	-0.06	-0.03
300-15-3c	340155	341712.00	341939.40	1434.19	342206	360.00	341278	342180.70	6185.67	0.33	-0.13	-0.27
300-15-3d	1355955	1358223.00	1359934.00	1495.27	1355955	359.00	1355890	1357945.00	4594.51	**0.00**	-0.17	0.00
300-15-3e	1286877	1286877.00	1288553.40	2208.86	1289607	604.00	1287440	1289470.00	4778.26	0.04	0.04	-0.17
300-15-4a	746407	747730.00	748983.80	1976.41	750135	439.00	748474	749853.85	5756.94	0.28	0.10	-0.22
300-15-4b	559877	559877.00	560310.20	1291.47	560352	472.00	558831	559250.60	5235.91	**-0.19**	-0.19	-0.27
300-15-4c	302390	304254.00	304632.60	1842.92	303984	543.00	302676	303670.10	6658.45	0.09	-0.52	-0.43
300-15-4d	1285714	1288091.00	1289787.60	1779.44	1289757	458.00	1287110	1288907.50	4785.17	0.11	-0.08	-0.21
300-15-4e	1173516	1173516.00	1175283.60	1939.34	1174438	621.00	1172200	1173887.00	5073.71	**-0.11**	-0.11	-0.19
300-20-1a	945545	1009840.00	1011300.80	1612.80	945545	513.00	944798	948040.05	5097.99	**-0.08**	-6.44	-0.08
300-20-1b	739604	739604.00	755196.80	1408.82	740915	514.00	739410	741928.95	5251.51	**-0.03**	-0.03	-0.20
300-20-1c	361735	364096.00	364819.40	1554.02	364303	514.00	362460	362798.05	6815.62	0.20	-0.45	-0.51
300-20-1d	1572968	1575390.00	1577056.80	1625.01	1579766	401.00	1574820	1578889.00	5238.64	0.12	-0.04	-0.31
300-20-1e	1320811	1391567.00	1392971.00	1601.83	1320924	575.00	1316880	1320908.50	4772.42	**-0.30**	-5.37	-0.31
300-20-2a	909306	909306.00	910784.20	2044.26	909376	553.00	910162	912638.75	5634.41	0.09	0.09	0.09
300-20-2b	695155	695524.00	696328.80	1437.34	695155	558.00	694525	695945.60	5731.38	**-0.09**	-0.14	-0.09
300-20-2c	298522	299425.00	300442.80	1563.93	309529	470.00	298790	299538.85	6632.28	0.09	-0.21	-3.47
300-20-2d	1569042	1569139.00	1570655.00	1899.70	1571166	369.00	1570170	1572547.50	5864.49	0.07	0.07	-0.06
300-20-2e	1284831	1386386.00	1388265.00	2240.11	1284831	655.00	1282040	1284757.50	4994.35	**-0.22**	-7.53	-0.22
300-20-3a	927452	929901.00	930748.60	1779.00	930992	469.00	928706	930337.05	4923.79	0.14	-0.13	-0.25
300-20-3b	750844	751307.00	751886.20	1424.87	750844	489.00	750633	752352.25	4979.51	**-0.03**	-0.09	-0.03
300-20-3c	304269	305771.00	307195.00	1176.54	307684	488.00	304875	305550.95	6277.44	0.20	-0.29	-0.91
300-20-3d	1539008	1539008.00	1541074.60	1657.35	1541385	373.00	1538930	1541202.50	4735.57	**-0.01**	-0.01	-0.16
300-20-3e	1265041	1289734.00	1290731.60	2090.99	1265041	649.00	1261660	1263625.50	4501.11	**-0.27**	-2.18	-0.27
300-20-4a	859446	859474.00	860141.40	1992.62	859446	574.00	858649	859878.35	5834.96	**-0.09**	-0.10	-0.09
300-20-4b	687930	687930.00	689431.60	1333.38	688604	559.00	687179	688281.30	5951.58	**-0.11**	-0.11	-0.21
300-20-4c	299206	300285.00	301125.00	1677.81	301424	642.00	299157	299538.75	7350.46	**-0.02**	-0.38	-0.75
300-20-4d	1540194	1540194.00	1542447.80	1751.09	1542127	653.00	1539900	1542697.50	5910.55	**-0.02**	-0.02	-0.14
300-20-4e	1330488	1344056.00	1345433.00	2624.15	1330488	805.00	1322150	1324786.50	5383.62	**-0.63**	-1.63	-0.63

Table 5 Results for the CLRP on the instances of set $\mathbb{S}$.

Instances	BKS	TBSA$_{basic}$			PF		HGAMP					
		Best	Avg.	Time	Best	Time	Best	Avg.	Time	$\delta_1(\%)$	$\delta_2(\%)$	$\delta_3(\%)$
400-20-1a	1140605	1140605.00	1142848.60	5459.61	1140975	733.00	1139820	1142918.00	7167.78	**-0.07**	-0.07	-0.10
400-20-1b	876157	880393.00	881162.40	2844.00	876157	725.00	876842	878678.10	7833.38	0.08	-0.40	0.08
400-20-1c	467755	467755.00	470279.60	3935.31	471031	813.00	466857	468164.25	10503.38	**-0.19**	-0.19	-0.89
400-20-1d	1956824	1956824.00	1960789.40	3780.24	1957929	805.00	1962600	1966491.50	8333.51	0.30	0.30	0.24
400-20-1e	1645475	1748962.00	1750819.00	4669.29	1645475	701.00	1641880	1645169.50	6697.34	**-0.22**	-6.12	-0.22
400-20-2a	1053445	1053445.00	1055068.00	5849.92	1055570	671.00	1054500	1056902.00	8246.21	0.10	0.10	-0.10
400-20-2b	828932	829494.00	830091.20	2956.62	828932	636.00	827077	828305.30	7770.90	**-0.22**	-0.29	-0.22
400-20-2c	394712	394712.00	396074.80	3975.54	395668	647.00	394544	395798.40	10390.25	**-0.04**	-0.04	-0.28
400-20-2d	1875072	1875072.00	1877613.80	4029.08	1878113	546.00	1875310	1877750.50	8306.77	0.01	0.01	-0.15
400-20-2e	1562339	1608600.00	1610261.00	6084.98	1562339	776.00	1560160	1562185.00	6864.83	**-0.14**	-3.01	-0.14
400-20-3a	1098989	1098989.00	1100874.20	4704.81	1100714	828.00	1098670	1100114.50	7984.85	**-0.03**	-0.03	-0.19
400-20-3b	847618	849555.00	850730.40	2730.97	847618	640.00	847311	849163.35	7438.39	**-0.04**	-0.26	-0.04
400-20-3c	391928	391928.00	393227.60	2898.16	393648	647.00	390047	391164.25	9916.37	**-0.48**	-0.48	-0.91
400-20-3d	1929284	1929284.00	1930892.00	3356.00	1936927	517.00	1927290	1929653.00	7305.67	**-0.10**	-0.10	-0.50
400-20-3e	1679271	1778315.00	1780375.60	5251.94	1679271	714.00	1656860	1660426.00	6895.84	**-1.33**	-6.83	-1.33
400-20-4a	1081452	1081452.00	1083352.60	5802.08	1083252	706.00	1080890	1083175.50	8611.75	**-0.05**	-0.05	-0.22
400-20-4b	842078	842078.00	845150.80	3260.79	843238	717.00	841464	843201.05	9066.75	**-0.07**	-0.07	-0.21
400-20-4c	351715	351715.00	352200.60	2670.50	357250	776.00	350273	351572.95	10776.72	**-0.41**	-0.41	-1.95
400-20-4d	1834809	1834809.00	1836465.00	4318.18	1845315	544.00	1835540	1838164.50	9222.91	0.04	0.04	-0.53
400-20-4e	1558411	1620575.00	1621855.20	4993.24	1558411	818.00	1558920	1560822.50	8342.83	0.03	-3.80	0.03
400-25-1a	1156187	1156187.00	1156765.40	5401.22	1156802	507.00	1153960	1157421.50	7789.11	**-0.19**	-0.19	-0.25
400-25-1b	889828	890566.00	892176.80	2878.20	889828	501.00	889193	890514.90	7762.85	**-0.07**	-0.15	-0.07
400-25-1c	395268	395268.00	396632.40	3579.13	412212	609.00	393581	394741.30	12532.13	**-0.43**	-0.43	-4.52
400-25-1d	2341499	2341499.00	2343936.80	3281.81	2350173	650.00	2336630	2337907.50	7790.62	**-0.21**	-0.21	-0.58
400-25-1e	1890676	2053366.00	2058418.00	4180.70	1890676	730.00	1863200	1865231.00	6578.80	**-1.45**	-9.26	-1.45
400-25-2a	1091595	1091595.00	1092140.80	7027.44	1102122	490.00	1090120	1091785.00	8624.70	**-0.14**	-0.14	-1.09
400-25-2b	869254	869254.00	869963.20	3607.85	875593	508.00	867716	869317.15	8010.93	**-0.18**	-0.18	-0.90
400-25-2c	360923	360923.00	361783.60	3497.24	362684	644.00	359606	360481.00	11148.83	**-0.36**	-0.36	-0.85
400-25-2d	2351903	2351903.00	2353659.20	3709.96	2356224	699.00	2353260	2355954.00	7520.32	0.06	0.06	-0.13
400-25-2e	1910645	1954300.00	1983843.80	4964.71	1910645	870.00	1902730	1905257.00	7902.29	**-0.41**	-2.64	-0.41
400-25-3a	1105783	1105783.00	1106974.60	5606.25	1122031	454.00	1106300	1109104.50	8249.31	0.05	0.05	-1.40
400-25-3b	862180	862180.00	865586.40	3395.72	871206	477.00	861981	863888.35	8212.16	**-0.02**	-0.02	-1.06
400-25-3c	393783	393783.00	394555.40	2998.27	399436	706.00	392264	393142.95	9778.04	**-0.39**	-0.39	-1.80
400-25-3d	2321358	2321358.00	2324138.80	4211.50	2340658	527.00	2323480	2327336.50	7727.94	0.09	0.09	-0.73
400-25-3e	1901148	1946952.00	1950642.40	4098.50	1901148	878.00	1890950	1894935.50	6965.56	**-0.54**	-2.88	-0.54
400-25-4a	1015654	1015654.00	1016764.20	5539.99	1016670	546.00	1014350	1016874.50	8495.83	**-0.13**	-0.13	-0.23
400-25-4b	801722	801722.00	803138.40	3117.51	803237	532.00	801135	802199.45	9084.68	**-0.07**	-0.07	-0.26
400-25-4c	380824	380824.00	381782.80	3195.05	382491	736.00	378419	379026.60	11428.08	**-0.63**	-0.63	-1.06
400-25-4d	2362571	2362571.00	2364869.20	4334.50	2367103	719.00	2360020	2362993.00	8911.98	**-0.11**	-0.11	-0.30
400-25-4e	1942603	1992633.00	1993388.80	4204.16	1942603	913.00	1935600	1939077.00	7784.53	**-0.36**	-2.86	-0.36
500-25-1a	1773409	1773409.00	1774912.80	10864.49	1798898	705.00	1771040	1772769.50	10235.44	**-0.13**	-0.13	-1.55
500-25-1b	1331827	1331827.00	1334444.80	5803.00	1349058	709.00	1327800	1331534.50	10432.13	**-0.30**	-0.30	-1.58
500-25-1c	671756	673495.00	675806.80	6000.60	671756	1193.00	670025	672347.05	16214.14	**-0.26**	-0.52	-0.26
500-25-1d	3322248	3325312.00	3328954.60	6627.34	3322248	833.00	3321740	3324804.00	10119.59	**-0.02**	-0.11	-0.02
500-25-1e	2692197	2971616.00	2975454.60	7693.69	2692197	1322.00	2682740	2690438.00	9314.32	**-0.35**	-9.72	-0.35
500-25-2a	1619689	1619689.00	1622446.60	11998.29	1620927	713.00	1620140	1623115.00	12081.47	0.03	0.03	-0.05
500-25-2b	1251667	1252748.00	1254125.00	5212.56	1251667	681.00	1250270	1251370.00	9586.59	**-0.11**	-0.20	-0.11
500-25-2c	574794	574794.00	576678.00	5694.01	576080	856.00	571155	572426.75	14167.70	**-0.63**	-0.63	-0.85
500-25-2d	3338585	3338585.00	3340077.20	8014.10	3345184	834.00	3335790	3338640.50	10594.70	**-0.08**	-0.08	-0.28
500-25-2e	2729944	2802823.00	2862227.40	11992.46	2729944	1337.00	2725350	2728780.50	10784.34	**-0.17**	-2.76	-0.17
500-25-3a	1725918	1725918.00	1728825.20	13855.28	1756884	723.00	1724540	1728482.00	10791.69	**-0.08**	-0.08	-1.84
500-25-3b	1305521	1305521.00	1306926.20	5421.25	1314903	693.00	1302820	1305386.00	11221.64	**-0.21**	-0.21	-0.92
500-25-3c	580688	581425.00	583190.80	5098.87	580688	817.00	578519	579999.50	14532.09	**-0.37**	-0.50	-0.37
500-25-3d	3248557	3248557.00	3250056.20	6323.21	3249805	760.00	3253900	3255339.00	8733.39	0.16	0.16	0.13
500-25-3e	2697267	2768174.00	2795997.00	11756.12	2697267	1056.00	2681360	2685903.50	10271.55	**-0.59**	-3.14	-0.59
500-25-4a	1655310	1655514.00	1658637.00	13964.09	1655310	714.00	1653680	1657127.50	12930.89	**-0.10**	-0.11	-0.10
500-25-4b	1260960	1263496.00	1267151.40	6661.01	1260960	720.00	1259400	1262157.00	11869.51	**-0.12**	-0.32	-0.12
500-25-4c	664089	664089.00	667522.20	7012.11	669088	1229.00	665017	667099.50	14369.64	0.14	0.14	-0.61
500-25-4d	3362588	3362588.00	3364590.00	8393.65	3370668	967.00	3367700	3372920.00	12718.64	0.15	0.15	-0.09
500-25-4e	2630867	2630867.00	2698984.60	11932.18	2647733	1164.00	2625360	2629760.00	11267.73	**-0.21**	-0.21	-0.84
500-30-1a	1984150	1984150.00	1991223.20	11301.10	1999892	924.00	1991640	2001398.00	11687.61	0.38	0.38	-0.41
500-30-1b	1537821	1538027.00	1540872.60	6369.41	1537821	881.00	1532670	1538876.50	11862.10	**-0.33**	-0.35	-0.33
500-30-1c	614433	614433.00	618107.40	6388.81	630143	841.00	611584	612916.30	15735.55	**-0.46**	-0.46	-2.95
500-30-1d	3742922	3742922.00	3745969.00	6002.25	3762156	1061.00	3740220	3746680.50	12247.33	**-0.07**	-0.07	-0.58
500-30-1e	3246339	3485608.00	3488967.80	7835.62	3246339	986.00	3236740	3242864.00	11263.33	**-0.30**	-7.14	-0.30
500-30-2a	1820346	1820346.00	1822744.20	15907.97	1820556	899.00	1821340	1824145.50	12927.40	0.05	0.05	0.04
500-30-2b	1452028	1452171.00	1461842.20	6908.29	1452028	971.00	1452140	1455534.50	12417.00	0.01	0.00	0.01
500-30-2c	649471	649471.00	652143.60	5547.39	650913	1009.00	647932	649707.65	15833.36	**-0.24**	-0.24	-0.46
500-30-2d	3815306	3815306.00	3816129.20	8542.96	3819650	1234.00	3815210	3818836.50	11682.06	0.00	0.00	-0.12
500-30-2e	3249618	3293153.00	3344410.20	9894.94	3249618	1086.00	3218410	3224217.00	11131.64	**-0.96**	-2.27	-0.96
500-30-3a	1782554	1782554.00	1784835.00	9978.88	1788808	876.00	1781220	1783841.50	11986.08	**-0.07**	-0.07	-0.42
500-30-3b	1422148	1422148.00	1424048.80	5205.87	1424533	802.00	1417870	1420747.50	10956.50	**-0.30**	-0.30	-0.47
500-30-3c	570866	570866.00	573130.60	5936.08	571418	1036.00	567988	569645.65	12774.72	**-0.50**	-0.50	-0.60
500-30-3d	3690995	3690995.00	3695470.40	8057.93	3710414	1382.00	3697540	3706084.50	11605.87	0.18	0.18	-0.35
500-30-3e	3059470	3171977.00	3285552.20	10411.43	3059470	1111.00	3131950	3136709.00	10202.69	2.37	-1.26	2.37
500-30-4a	1716476	1716476.00	1722659.80	12070.89	1725178	1214.00	1717720	1722760.00	13180.76	0.07	0.07	-0.43
500-30-4b	1398401	1398401.00	1461336.80	6314.01	1402136	1155.00	1399990	1407113.50	13323.34	0.11	0.11	-0.15
500-30-4c	562542	562731.00	565040.00	9257.42	562542	1071.00	558398	560183.50	13567.86	**-0.74**	-0.77	-0.74
500-30-4d	3708479	3708479.00	3711494.60	8340.83	3710414	1382.00	3711620	3723293.50	14431.39	0.08	0.08	0.03
500-30-4e	3059470	3194234.00	3217200.80	9918.44	3059470	1111.00	3057950	3086177.00	11804.11	**-0.05**	-4.27	-0.05
600-30-1a	2198674	2198674.00	2200768.80	20140.67	2208578	1381.00	2197860	2203132.00	15026.22	**-0.04**	-0.04	-0.49
600-30-1b	1691805	1691805.00	1697677.40	10673.49	1697630	1346.00	1689980	1695922.00	15707.17	**-0.11**	-0.11	-0.45
600-30-1c	746485	748714.00	752951.00	10822.16	746485	1140.00	742450	745721.40	19931.45	**-0.54**	-0.84	-0.54
600-30-1d	4213337	4213337.00	4217214.00	14803.88	4214905	1564.00	4219390	4229457.50	16852.33	0.14	0.14	0.11
600-30-1e	3570518	3737075.00	3843513.00	15500.11	3570518	1380.00	3552440	3558027.00	14743.36	**-0.51**	-4.94	-0.51
600-30-2a	2017760	2017760.00	2019232.80	23638.81	2023484	1105.00	2015580	2018587.50	14461.08	**-0.11**	-0.11	-0.39
600-30-2b	1601675	1602833.00	1604821.00	10778.67	1601675	1019.00	1601140	1602709.50	14258.39	**-0.03**	-0.11	-0.03
600-30-2c	634787	634787.00	638360.20	10718.97	646907	1368.00	633935	635262.60	16477.06	**-0.13**	-0.13	-2.01
600-30-2d	4163772	4163772.00	4167031.80	15024.23	4201095	1493.00	4160240	4164647.50	15083.02	**-0.08**	-0.08	-0.97
600-30-2e	3572959	3682117.00	3714272.40	15009.62	3572959	1672.00	3563860	3585562.50	14761.36	**-0.25**	-3.21	-0.25
600-30-3a	2082824	2082824.00	2086671.80	24897.75	2103960	1168.00	2087280	2092834.50	15086.41	**0.21**	0.21	-0.79
600-30-3b	1615623	1615623.00	1620896.60	10761.47	1623113	1247.00	1612210	1616431.00	15230.77	**-0.21**	-0.21	-0.67
600-30-3c	662569	662569.00	664845.60	9549.06	683582	1311.00	658376	660757.80	18725.56	**-0.63**	-0.63	-3.69
600-30-3d	4068474	4068474.00	4071472.60	15657.11	4126907	1724.00	4079540	4086996.00	14432.48	0.27	0.27	-1.15
600-30-3e	3488544	3496852.00	3505305.40	15179.65	3488544	1526.00	3327790	3335378.00	12475.50	**-4.61**	-4.83	-4.61
600-30-4a	1940218	1940218.00	1943972.00	25360.71	1942190	1325.00	1938780	1942920.00	15595.75	**-0.07**	-0.07	-0.18
600-30-4b	1555080	1560237.00	1561534.00	11437.18	1555080	1366.00	1554350	1559693.00	15512.32	**-0.05**	-0.38	-0.05
600-30-4c	707288	707288.00	708463.60	14535.48	714196	1808.00	705601	707494.15	17901.53	**-0.24**	-0.24	-1.20
600-30-4d	4150775	4150775.00	4155109.00	17602.99	4164022	2391.00	4183040	4191568.00	18024.60	0.78	0.78	0.46
600-30-4e	3543712	3622885.00	3629323.20	18513.18	3543712	1890.00	3514040	3520449.50	14803.24	**-0.84**	-3.00	-0.84
Average	1181502.41	1192764.53	1197257.61	4473.96	1185090.37	633.32	1179988.97	1182327.72	7377.83	**-0.08**	-1.07	-0.43

Table 6 Summary of comparative results on the MDVRP.

Instances	BKS	GANCP$^+$			HGSADC			AVXS			HALNS			HGAMP		
		Best	δ(%)	Time	Avg.	δ_a(%)	Time	Avg.	δ_a(%)	Time	Avg.	δ_a(%)	Time	Avg.	δ_a(%)	Time
p01	576.58	586.39	1.65	3.64	576.87	0.00	13.80	576.58	0.00	1.00	576.58	0.00	60.02	576.58	0.00	2.20
p02	473.53	495.38	4.61	3.61	473.53	0.00	12.60	473.53	0.00	1.00	473.53	0.00	54.47	473.53	0.00	4.59
p03	641.19	662.14	3.27	4.73	641.19	0.00	25.80	641.19	0.00	2.00	641.19	0.00	103.00	641.19	0.00	7.66
p04	1001.04	1013.37	1.23	2.16	1001.23	0.02	116.40	1002.54	0.15	17.80	1001.04	0.00	198.66	1001.04	0.00	50.99
p05	750.03	763.90	1.85	2.24	750.03	0.00	63.60	750.03	0.00	5.60	750.03	0.00	145.98	750.03	0.00	61.43
p06	876.50	885.92	1.07	3.11	876.50	0.00	68.40	876.50	0.00	7.50	876.62	0.01	206.59	876.50	0.00	100.42
p07	881.97	916.65	3.93	3.24	884.43	0.28	93.00	882.50	0.06	8.60	881.97	0.00	203.58	883.97	0.23	608.98
p12	1318.95	-	-	-	1318.95	0.00	31.20	1318.95	0.00	1.00	1318.95	0.00	100.22	1318.95	0.00	1.48
p15	2505.42	-	-	-	2505.42	0.00	115.20	2505.42	0.00	8.90	2505.42	0.00	334.53	2505.42	0.00	24.94
p18	3702.85	-	-	-	3702.85	0.00	271.20	3702.85	0.00	27.30	3702.85	0.00	1243.84	3702.85	0.00	206.65
p21	5474.84	-	-	-	5476.41	0.03	600.00	5475.93	0.00	77.40	5480.90	0.11	3310.95	5482.31	0.14	5698.65
Average	1654.81	-	-	-	1655.22	0.03	128.29	1655.09	0.02	14.37	1655.37	0.01	541.99	1655.67	0.03	615.27