Hybrid genetic algorithm for undirected traveling salesman problems with profits

Pengfei He [a], Jin-Kao Hao [a,*], Qinghua Wu [b]

[a] *LERIA, Université d'Angers, 2 Boulevard Lavoisier, 49045 Angers, France*

[b] *School of Management, Huazhong University of Science and Technology, No. 1037 Luoyu Road, Wuhan, China*

Networks 82(3): 189-221, 2023

`https://onlinelibrary.wiley.com/doi/10.1002/net.22167`

Abstract

The orienteering problem (OP) and prize-collecting traveling salesman problem (PCTSP) are two typical TSPs with profits, in which each vertex has a profit and the goal is to visit several vertices to optimize the collected profit and travel costs. The OP aims to collect the maximum profit without exceeding the given travel cost. The PCTSP seeks to minimize the travel costs while ensuring a minimum profit threshold. This study introduces a hybrid genetic algorithm that addresses both the OP and PCTSP under a unified framework. The algorithm combines an extended edge-assembly crossover operator to produce promising offspring solutions, and an effective local search to ameliorate each offspring solution. The algorithm is further enforced by diversification-oriented mutation and population-diversity management. Extensive experiments demonstrate that the method competes favorably with the best existing methods in terms of both the solution quality and computational efficiency. Additional experiments provide insights into the roles of the key components of the proposed method.

Keywords: Traveling salesman; Genetic algorithm; Orienteering problem; Prize-collecting TSP; Edge assembly crossover.

---

* Corresponding author.

Email addresses: `pengfeihe606@gmail.com` (Pengfei He), `jin-kao.hao@univ-angers.fr` (Jin-Kao Hao), `qinghuawu1005@gmail.com` (Qinghua Wu).

1 Introduction

In many real-life applications, such as the home fuel delivery problem [18], tourist trip design problem [45], and bike repairing problem in a bike-share system [39], not all available customers can be visited owing to the limited time budget or other resource constraints. Traveling salesman problems (TSPs) with profits are typically used to formulate these applications, where the time budget or resource limits can be modeled by a knapsack constraint or generalized covering constraints. Thus, TSPs with profits can be viewed as a combination of two classical combinatorial optimization problems, i.e., the TSP and knapsack problem. Given their relevance, TSPs with profits have received considerable attention in the past several decades.

Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be an undirected graph, where $\mathcal{V} = \{v_0, v_1, \cdots, v_n\}$ is the vertex set, v_0 is the depot, $\mathcal{N} = \{v_1, \cdots, v_n\}$ represents n vertices (customers), and $\mathcal{E}$ is the edge set. Let p_i be the nonnegative profit associated with each vertex $v_i \in \mathcal{V}$ ($p_0 = 0$). Let $\mathcal{C} = (c_{ij})$ be the nonnegative cost (distance) matrix associated with $\mathcal{E}$ satisfying the triangle inequality ($c_{ij} + c_{jk} > c_{ik}$ for $v_i, v_j, v_k \in \mathcal{V}$ and $v_i \neq v_j \neq v_k$). TSPs with profits seek to determine an elementary circuit (i.e., each vertex is visited at most once) starting and ending at the depot, and visit several customers to optimize the collected profit and travel costs.

According to how the profit and travel cost objectives are considered, three different TSPs with profits have been identified in the literature [14]. The first problem is the profitable tour problem (PTP), where the two objectives are combined into a single objective function that seeks to minimize travel costs minus collected profit [14]. The second problem is the orienteering problem (OP) [18,50], which aims to maximize the collected profit under the constraint that the travel costs do not exceed a given value c_{max}. The OP is also known as the selective traveling salesperson problem [27,16]. The third problem is the prize-collecting TSP (PCTSP) [1,5], which aims to minimize the travel costs under the constraint that the collected profit must reach a given minimum value p_{min}. In Appendix A, we provide a mathematical formulation of the OP and PCTSP for a precise description of these problems. As indicated in [14], these problems are $\mathcal{NP}$-hard, and thus, computationally challenging. According to [14], among these three problems, the OP and PCTSP are under a primal dual relationship and attract substantially more attention than the PTP. In this study, we follow this trend and focus on the effective solution of the OP and PCTSP.

As shown in the comprehensive review of [14], numerous studies have contributed to improving the state of the art in solving these difficult problems. Several exact algorithms were proposed in [1,27,28,16,15,4] to solve small- and medium-sized instances with up to 532 vertices optimally. Remarkably, the revisited branch-and-cut algorithm presented by Kobeaga et al. [26] could find optimal solutions for OP instances with up to 2152 vertices. However, several heuristic algorithms have been developed for TSPs with profits to deal with large-sized instances, the optimal solutions of which cannot be determined by exact algorithms. In Section 2, we review the most representative heuristic algorithms. However, to date, these problems have been studied separately with specific algorithms designed for each problem, without a general and unified approach. Moreover, compared to research on exact algorithms, studies on effective heuristic algorithms remain rare and there is clearly a need for methods that can solve large instances effectively and efficiently.

This work aims to advance the state-of-the-art in solving two TSPs with profits (OP and PCTSP) using effective heuristic algorithms. For this purpose, we introduce a unified approach for the OP and PCTSP under the hybrid genetic search framework. Hybrid genetic algorithms, which are also known as memetic algorithms, take advantage of population-based genetic frameworks and neighborhood-based local search frameworks [21]. Owing to the use of a population of solutions, a genetic algorithm offers the possibility of creating new solutions by the recombination of existing solutions via a crossover operator. Furthermore, by exploring a neighborhood, a local search algorithm offers an effective means of locating high-quality solutions around a seeding solution. By combining these two complementary methods, a hybrid genetic algorithm is expected to achieve performance that cannot be attained by applying each individual approach separately. Several highly effective hybrid genetic algorithms have been proposed to solve various routing problems [41,34,35,42,52,53,54].

We devise a dedicated technique for the OP and PCTSP to adapt the popular edge-assembly crossover that was initially designed for the TSP [33,36] and apply it to routing problems [34,35,23,22]. The proposed approach relies on an extended edge-assembly crossover operator and benefits from synergy with effective local search and dedicated diversification strategies, such as mutation and population-diversity management. Our experiments on well-known benchmark instances in the literature demonstrate that the proposed algorithm competes favorably with the best-performing methods. In particular, the algorithm can improve many current best bounds for both the OP and PCTSP.

The remainder of this paper is organized as follows. Section 2 provides a literature review of solution approaches for the two TSPs with profits. The proposed algorithm is described in Section 3. Section 4 presents the computational results and comparisons. Section 5 analyzes the main components of the algorithm. Section 6 presents the conclusions of the study.

2 Literature review

We provide a literature review of the studies on the two TSPs with profits, namely the OP and PCTSP according to [14] and [49,50].

Table 1 summarizes the existing heuristic algorithms for the OP. A comprehensive review of heuristics up to 2010 was provided in [50]. Our review focuses on more recent studies. In 2010, Silberholz and Golden [45] studied the generalized OP and presented an iterated local search, whereby routes were improved by 2-opt, whereas unrouted vertices were inserted into the route when the travel costs were less than c_{max}. In 2014, Campos et al. [7] introduced the GRASP algorithm that combined the general greedy randomized adaptive search procedure, path relinking, and local search with three neighborhoods. The experimental results indicated that the algorithm obtained high-quality solutions within a short running time. In 2015, Marinakis et al. [31] used the GRASP procedure to construct a population of solutions, which was developed by applying a simple 1-point crossover and local search. In 2016, Keshtkaran and Ziarati [24] developed another GRASP, in which new solutions were generated using a segment-removing strategy. The computational results demonstrated the competitiveness of the algorithm on two standard benchmark instances. In 2017, Ostrowski et al. [38] implemented a specific crossover in which the common vertices involved in two routes were considered to produce offspring solutions by changing the fragments of the two routes. In this algorithm, feasible and infeasible routes were allowed to cross over, while the fitness function was redefined with respect to the travel costs.

In 2018, Kobeaga et al. [25] proposed an evolutionary algorithm for the OP (EA4OP) that featured an interesting edge recombination operator to produce offspring individuals. This recombination operator inherits two main characteristics from the parent solutions with respect to the vertices and edges. All vertices that are common to both parents are maintained, whereas vertices that belong to only one parent are included with a probability, and all vertices that do not belong to any parent are excluded. The edges of the parents are inherited as far as possible to pass on a maximum amount of information and make the length of offspring solutions as short as possible. The experimental results indicated that EA4OP is highly effective and efficient. In 2019, Santini [44] presented the adaptive large neighborhood search algorithm (ALNS) including various destroy and repair methods. Experiments on four sets of benchmark instances revealed that the algorithm was competitive, producing several new best results.

In addition to these heuristic algorithms, we mention the recent revisited branch-and-cut (RB&C) exact algorithm [26], which could prove many optimal solutions and update numerous lower bounds for small- and medium-sized benchmark instances.

This review reveals that the two heuristic algorithms presented in [25,44] and the exact algorithm of [26] represent the current state of the art for solving the OP. These works hold the best known results for the four sets of benchmark instances that are commonly tested in the literature.

Table 1
Summary of the taxonomy of representative heuristic algorithms for the OP

Literature	Year	Framework
Tsiligirides [47]	1984	Stochastic algorithm
Golden et al. [18]	1987	Centre of gravity heuristic
Ramesh and Brown [43]	1991	Tabu search
Wang et al. [56]	1995	Artificial neural network
Chao et al. [8]	1996	Record-to-record
Gendreau et al. [17]	1998	Tabu search
Tasgetiren and Smith [46]	2000	Genetic algorithm
Liang et al. [29]	2006	Ant colony optimzation
Silberholz and Golden [45]	2010	Iterated local search
Campos et al. [7]	2014	GRASP with path relinking
Marinakis et al. [31]	2015	Memetic-GRASP
Keshtkaran and Ziarati [24]	2016	GRASP
Ostrowski et al. [38]	2017	Evolution-inspired local improvement algorithm
Kobeaga et al. [25]	2018	Evolutionary algorithm
Santini [44]	2019	Adaptive large neighborhood search

The PCTSP was originally defined by Balas [1], where a penalty for each unvisited vertex was considered in the objective function. Since then, considerable efforts have been devoted to mathematical models and solution algorithms. Bienstock et al. [5] presented an approximation algorithm based on Christofides' algorithm and Balas [2] analyzed several effective inequalities for this problem. Later, Balas [3] summarized the results of polyhedral considerations and applications.

In recent years, several algorithms have been presented for the PCTSP, achieving good results on medium-sized instances with up to 500 vertices. Gomes et al. [19] proposed a hybrid GRASP+VNS algorithm and demonstrated its competitiveness against previous methods. Chaves and Lorena [10] presented a hybrid metaheuristic algorithm based on a clustering search and compared the results of the algorithm with those obtained by CPLEX. Pedro and Saldanha [40] introduced a tabu search approach and presented new upper bounds for several PCTSP instances. Clímaco et al. [11] proposed a branch-and-cut (B&C) algorithm and an MIP-based heuristic to solve the PCTSP, which exhibited highly satisfactory performance for the tested instances. To summarize, these approaches consider different objectives arising in the real world, with the aim of minimizing the sum of the travel costs and penalties for the unrouted vertices. However, several studies [14] on the PCTSP did not consider the penalty terms of the unrouted vertices. In this case, the aim is to minimize the travel costs under the constraint that the collected prize must reach a given minimum value p_{min}. Following this objective, Bérubé et al. [4] proposed a B&C algorithm and reported results on medium-sized instances with up to 532 vertices. This B&C algorithm represents the current state of the art for solving the PCTSP problem.

The OP and PCTSP consider only one vehicle in their application. Various studies have also investigated multi-vehicle routing problems with profits [6,55,20], such as the team OP, where several vehicles are available to collect the profit. Several hybrid genetic algorithms relating to our work can be found in the literature for various routing problems, such as the TSP, vehicle routing problem (VRP), and their variants. For instance, Nagata and Kobayashi [36] presented a powerful genetic algorithm that relies on edge-assembly crossover for the TSP. Nagata and Bräysy [34] further applied edge-assembly crossover to the capacitated VRP and a local search procedure to ameliorate each offspring solution. However, the edge-assembly crossover in these studies only dealt with situations in which each vertex had the same degree in both parent solutions. Edge-assembly crossover cannot be directly applied to TSPs with profits, given that each vertex may be associated with different degrees in distinct solutions. Thus, we extend the edge-assembly crossover to address this difficulty. Another popular hybrid genetic algorithm [54,51] implements a crossover operator based on the giant tour and split algorithms. We also implement this crossover operator to evaluate the performance of the genetic algorithm. However, the OP also integrates the well-known 0-1 knapsack problem as a subproblem, which has been widely studied [32]. However, we are unaware of a competitive hybrid genetic algorithm given that dynamic programming is very effective, even for large instances.

3 Hybrid genetic algorithm for TSPs with profits

This section presents the hybrid genetic algorithm (HGA) designed for the two studied TSPs with profits; that is, the OP and PCTSP. This is a unified algorithm in the sense that, with slight adjustments, the same algorithm is used to solve both problems effectively.

HGA is outlined in Algorithm 1. Starting from an initial population $\mathcal{P}$ constructed by the initialization procedure (line 2), the algorithm evolves the population throughout numerous generations by applying the crossover operator, local search procedure, mutation operator, and population management (lines 4–15). In each generation, two solutions are selected as parents using the binary tournament strategy, which selects the best solution among two random solutions from $\mathcal{P}$ as a parent [52]. Of particular interest is the extended edge-assembly crossover operator (line 6), which creates β offspring solutions by assembling the edges of the parent solutions. Subsequently, each offspring solution is submitted to the local search procedure for quality improvement (line 8). Finally, each solution is diversified by a mutation operator (line 12) and managed by an advanced pool updating strategy (line 13). The algorithm stops and returns the best solution φ^* once a predefined stopping condition is met (e.g., a maximum cutoff time or maximum number of generations). The crossover, mutation, and advanced pool updating strategies are exactly same

Algorithm 1: Hybrid genetic algorithm for two TSPs with profits

Input: Instance I;
Output: The best found solution φ^*;
begin
  $\mathcal{P} \leftarrow \mathit{InitialPopulation}(I)$; `/* Initializing the population` $\mathcal{P}$`, Section 3.1 */`
  $\varphi^* \leftarrow \arg\min\{f(\varphi_i)|i = 1, 2, \cdots, |\mathcal{P}|\}$; `/* Updating the best solution` $\varphi*$ `found so far; */`
  while *Stopping condition is not met* **do**
    $\{\varphi_A, \varphi_B\} \leftarrow \mathit{SelectParent}(\mathcal{P})$;
    $\{\varphi_O^1, \varphi_O^2, \cdots, \varphi_O^\beta\} \leftarrow E^2AX(\varphi_A, \varphi_B)$; `/* Generating promising offspring solutions, Section 3.2 */`
    for $i = 1$ *to* β **do**
      $\varphi_O^i \leftarrow \mathit{LocalSearch}(\varphi_O^i)$; `/* Ameliorating the offspring solution, Section 3.3 */`
      if $f(\varphi_O^i) < f(\varphi^*)$ **then**
        $\varphi^* \leftarrow \varphi_O^i$;
      end
      $\varphi_O^i \leftarrow \mathit{Mutation}(\varphi_O^i)$; `/* Generating mutation, Section 3.4.1 */`
      $\mathcal{P} \leftarrow \mathit{UpdatingPop}(\mathcal{P}, \varphi_O^i)$; `/* Updating the population, Section 3.4.2 */`
    end
  end
  return φ^*;
end

when the algorithm is applied to the OP and PCTSP. However, the initialization and local search differ slightly because the two problems consider different objectives.

The remainder of this section is dedicated to a detailed presentation of the methods for population initialization, crossover, local search, mutation, and population management.

3.1 Population initialization

The initial population $\mathcal{P}$ is generated in two phases using a method inspired by the technique presented in [51]. Phase 1 generates a pool of $4 \times \lambda$ solutions, where each solution is created greedily (see below) and subsequently improved by the local search described in Section 3.3. Phase 2 uses the surviving strategy described in Section 3.4.2 to retain λ solutions in $\mathcal{P}$ with respect to the solution quality and their contribution to the diversity of the population.

In phase 1, each solution in the population is generated by a two-stage technique: in the first stage (S1), a greedy strategy is adopted to construct an initial solution, and in the second stage (S2), the local search procedure is applied to improve the initial solution further. Because the OP and PCTSP pursue different optimization objectives, two different greedy strategies are used for the two problems in S1. For the OP, the greedy construction works as follows: It starts with an empty route and initializes the route using depot v_0. It then extends the route by inserting a vertex one by one into the route. Initially, a vertex is randomly selected and inserted after v_0. Subsequently, an unrouted vertex v_j from the δ-nearest neighborhood of the newly added vertex v_i is selected and inserted after vertex v_i such that the insertion leads to the minimum increase in the travel costs. If there are no unrouted vertices in the δ-nearest neighborhood of the newly inserted vertex, a new unrouted vertex is randomly selected and inserted after a vertex in the partial solution, such that the insertion leads to the minimum travel costs. This process is repeated until all vertices are inserted into the solution, or the current travel costs exceed $1.5 \times c_{max}$. For the PCTSP, the greedy construction works similarly and differs only in the selection of the next vertex to be added, which aims to maximize the collected profit in the PCTSP. The construction stops when the collected profit reaches $1.5 \times p_{min}$. In S2, the local search procedure is applied to restore the feasibility of the solution and to improve the quality of each solution as far as possible. Given that phase 1 generates $4 \times \lambda$ solutions, phase 2 eliminates additional solutions using the surviving strategy described in Section 3.4.2 to preserve exactly λ (population size) solutions in $\mathcal{P}$.

For the OP, one notes that the initial solutions constructed in S1 are not necessarily feasible. Given that the feasibility of an initial solution can be easily recovered by removing several vertices in the subsequent local search procedure, we ensure that the initial population is composed of only feasible solutions. The rationale behind the use of the threshold $1.5 \times c_{max}$ during the solution construction is to obtain a diversified and high-quality population. Indeed, by setting $1.5 \times c_{max}$ in S1, more diversified initial solutions can be produced because more vertices can participate in the construction of the solution in S1. For the PCTSP, each initial solution is necessarily feasible because a profit of $1.5 \times p_{min}$ is collected. Like in the case of OP, having more vertices in initial solutions promotes a better diversity. Indeed, our experiments show that the performance of the algorithm will not change significantly if we adjust the value slightly. However, if the value becomes too small < 1.1, initial solutions will only include a limited number of cities, impacting negatively the population diversity. If the coefficient of 1.5 is replaced by an extremely large value > 2.0, we will obtain initial solutions with a high degree of similarity, which in turn significantly affects the quality of the solution. Finally, the coefficient remains constant as it is only used when constructing initial solutions.

3.2 Extended edge-assembly crossover

The HGA algorithm relies on an extended edge-assembly crossover, which is an adaptation of the edge-assembly crossover (EAX) designed for the TSP [33,36] to TSPs with profits. Critically, it is difficult to apply EAX directly to TSPs with profits because EAX assumes that all vertices are visited exactly once in the solution of the TSP.

Given a TSP instance defined on a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, a candidate TSP solution φ corresponds to a partial graph $\mathcal{G}_\varphi = (\mathcal{V}, \mathcal{E}_\varphi)$, where $\mathcal{E}_\varphi$ is the set of edges traversed by φ. Given a solution of the TSP, each vertex in $\mathcal{V}$ is visited exactly once, and thus, has the same degree of two in $\mathcal{G}_\varphi$. Given two parent TSP solutions and their associated partial graphs, EAX uses this property to reassemble the edges from the parents to produce offspring solutions.

However, the situation is different for TSPs with profits. Given two parent solutions, some vertices may be visited in one parent but not in the other parent. Consequently, a vertex may have two distinct degrees in the partial graphs of the parent solutions. This particularity makes it impossible to apply EAX to TSPs with profits. For the OP and PCTSP, we design an extended EAX (E^2AX), whose key concept is to add dummy edges (self-loops) to ensure that each vertex has the same degree in the graphs of the parent solutions.

Given an instance of the OP or PCTSP in graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, let φ be a solution that visits $|\varphi|$ vertices ($|\varphi| \leq n$) and let $\mathcal{G}_\varphi = (\mathcal{V}, \mathcal{E}_\varphi)$ be the corresponding partial graph, where $\mathcal{E}_\varphi \subset \mathcal{E}$ is the set of edges traversed by φ. There are two cases for each vertex in $\mathcal{G}_\varphi$: 1) the vertex is visited by φ and its degree is 2 in $\mathcal{G}_\varphi$; and 2) the vertex is not visited by φ and its degree is 0. In the example of Fig. 1, the red vertices are not visited by φ_A and their degree is 0 in $\mathcal{G}_\mathcal{A}$, whereas the visited vertices in $\mathcal{G}_\mathcal{A}$ have a degree of 2.

Let φ_A and φ_B be two candidate solutions for the OP or PCTSP, and let $\mathcal{G}_\mathcal{A} = (\mathcal{V}, \mathcal{E}_\mathcal{A})$ and $\mathcal{G}_\mathcal{B} = (\mathcal{V}, \mathcal{E}_\mathcal{B})$ be the corresponding partial graphs. We define the *degree difference* of vertex v in $\mathcal{G}_\mathcal{A}$ and $\mathcal{G}_\mathcal{B}$ as $\delta_v = |deg_\mathcal{A}(v) - deg_\mathcal{B}(v)|$, where $deg_\varphi(v)$ denotes the degree of vertex v in graph $\mathcal{G}_\varphi$. In the example of Fig. 1, the degree difference δ_v of a vertex v equals 0 if v is visited by both solutions or none of them; otherwise, $\delta_v = 2$. For each vertex v with $\delta_v = 2$, we can add a dummy loop (v, v) in $\mathcal{G}_\mathcal{A}$ or $\mathcal{G}_\mathcal{B}$ to make the degree difference 0 (see Fig. 1 (left-middle)).

Let $\mathcal{G}'_\mathcal{A} = (\mathcal{V}, \mathcal{E}'_\mathcal{A})$ and $\mathcal{G}'_\mathcal{B} = (\mathcal{V}, \mathcal{E}'_\mathcal{B})$ be graphs extended with dummy loops such that $\delta_v = 0$ for all vertices. Clearly, the extended graphs $\mathcal{G}'_\mathcal{A}$ and $\mathcal{G}'_\mathcal{B}$ satisfy the basic properties that are required by the EAX; that is, each vertex has the same degree in these graphs. As a result, we can now benefit from the edge assembly idea of the EAX operator to create offspring solutions for the

OP and PCTSP.

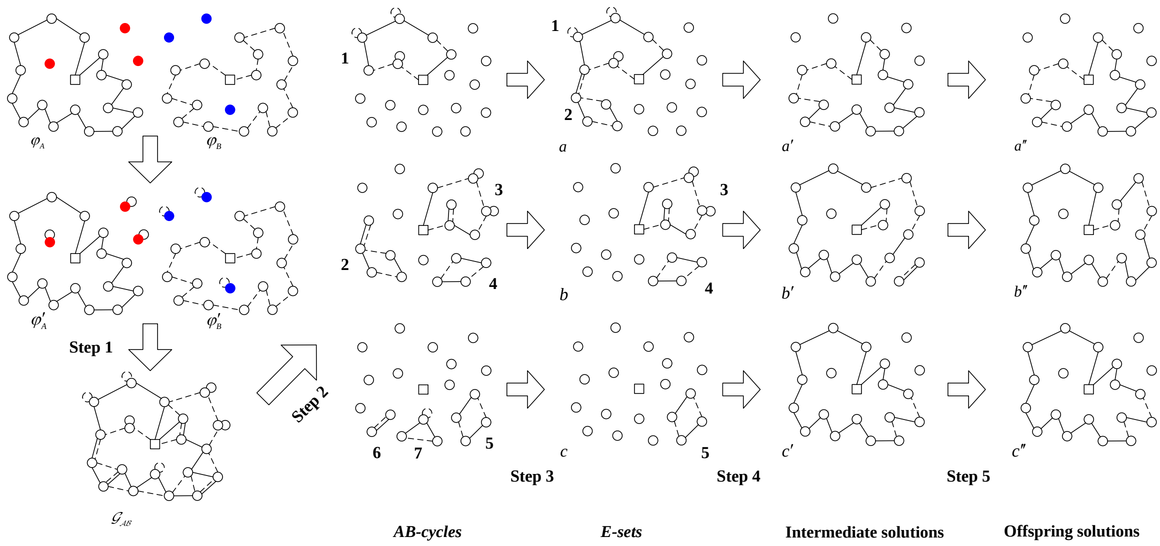


Fig. 1. Illustration of E^{2}AX.

Given two parent solutions φ_A and φ_B, the proposed E^{2}AX for the OP and PCTSP performs the following steps to generate β offspring solutions.

(1) *Generation of multigraph $\mathcal{G}_{\mathcal{AB}}$ with dummy loops.* Build partial graphs $\mathcal{G}_{\mathcal{A}} = (\mathcal{V}, \mathcal{E}_{\mathcal{A}})$ and $\mathcal{G}_{\mathcal{B}} = (\mathcal{V}, \mathcal{E}_{\mathcal{B}})$ for φ_A and φ_B. For each vertex v such that $\delta_v \neq 0$ in $\mathcal{G}_{\mathcal{A}}$ and $\mathcal{G}_{\mathcal{B}}$, add $\frac{|deg_{\mathcal{A}}(v) - deg_{\mathcal{B}}(v)|}{2}$ dummy self-loops in φ_A or φ_B to make $\delta_v = 0$. Build multigraph $\mathcal{G}_{\mathcal{AB}} = (\mathcal{V}, \mathcal{E}'_{\mathcal{A}} \cup \mathcal{E}'_{\mathcal{B}})$, where $\mathcal{E}'_{\mathcal{A}}$ and $\mathcal{E}'_{\mathcal{B}}$ are the edge sets extended with dummy loops.

(2) *Generation of AB-cycles from $\mathcal{G}_{\mathcal{AB}}$.* An *AB-cycle* is a closed path whose edges are alternatively obtained from the parents. From multigraph $\mathcal{G}_{\mathcal{AB}}$, build a set of *AB-cycles* as follows. Initialize an *AB-cycle* by a random vertex with one adjacent edge in $\mathcal{G}_{\mathcal{AB}}$. Then, add edges belonging to $\mathcal{E}'_{\mathcal{A}}$ and $\mathcal{E}'_{\mathcal{B}}$ alternatively until a cycle is obtained, which is an *AB-cycle*. Remove the edges of the *AB-cycle* from $\mathcal{G}_{\mathcal{AB}}$. Repeat the process to build the next *AB-cycle* until all edges in $\mathcal{G}_{\mathcal{AB}}$ are considered.

(3) *Generation of E-sets.* An *E-set* is a union of AB-cycles. Divide the set of *AB-cycles* randomly and uniformly into β subsets ($\beta = 3$ in this work), with each subset of *AB-cycles* defining an *E-set*.

(4) *Generation of intermediate solutions.* First, remove all dummy loops in the *E-sets*. Then, for each *E-set*, produce an intermediate solution from a random parent solution (say φ_A) by removing the edges of $\mathcal{E}_{\mathcal{A}}$ and adding the edges of $\mathcal{E}_{\mathcal{B}}$.

(5) *Elimination of isolated subtours.* For each intermediate solution containing subtours, merge the subtours with the main tour using the method presented in [36].

Fig. 1 provides an illustrative example of the recombination process with E^{2}AX applied to two parent solutions φ_A and φ_B. The second intermediate solution contains two small subtours that are merged with the main tour to form a single tour.

We now provide an analysis of the time complexity of E^{2}AX. Steps (1)–(4) must assemble $|\mathcal{E}'_{\mathcal{A}}| + |\mathcal{E}'_{\mathcal{B}}|$ edges to produce β offspring solutions, implying a time complexity of $\mathcal{O}(|\mathcal{E}'_{\mathcal{A}}| + |\mathcal{E}'_{\mathcal{B}}|)$. Given that a solution is necessarily an elementary tour, $n \geq |\mathcal{E}'_{\mathcal{A}}|$ and $n \geq |\mathcal{E}'_{\mathcal{B}}|$ hold. Thus, the times of steps (1)–(4) are bounded by $\mathcal{O}(n)$. In the final step, suppose that there are m isolated subtours including a maximum of e edges. The time complexity of this step is $\mathcal{O}(e \times \delta)$ [36], where δ is the number of closest vertices, as introduced in Section 3.3.1.

In practice, we have observed that when solving the OP, it is possible for two parent solutions with the same visited vertices (thus collecting the same profit, called them symmetric solutions) to be selected for crossover. In this case, the OP is essentially equivalent to the TSP and the parent solutions typically have different travel costs. As a result, E^{2}AX will degenerate to the standard EAX for the TSP. Moreover, we observe experimentally that the probability of selecting two parent solutions with the same profit is less than 0.2%, and thus these symmetric solutions have little impact on the performance of the algorithm. It is important to note that for the PCTSP, two solutions with the same objective value (i.e., the same traveling cost) cannot be selected as parent solutions, as clone solutions are not permitted in the population according to the population management strategy (see Section 3.4.2).

3.3 Offspring improvement

HGA employs a neighborhood-based local search to improve the offspring solutions generated by E^{2}AX. As discussed in [14], four neighborhood operators are typically used to transform a route for TSPs with profits: 1) adding an unrouted vertex, 2) removing a vertex from the route, 3) resequencing the route, and 4) replacing a routed vertex with an unrouted vertex. Note that the fourth operator is simply a combination of the first and second operators and our experiments also indicate that the fourth operator is of little help in improving the performance of our HGA. Consequently, our HGA adopts only the first three operators in its neighborhood exploration. Many TSP heuristics have been proposed to resequence a route. In our case, we adopt the 2-opt heuristic [12], which has been shown to be effective for the OP and PCTSP. We now explain the add and the remove operators.

3.3.1 Add operator

This operator is applied to add unrouted vertices into the route. For the OP, a heuristic that is commonly used in the literature [7,45] is adopted to perform vertex insertions. For each unrouted vertex v_i, its move gain $\Delta = \frac{p_i}{c_{i_p i}+c_{i i_n}-c_{i_p i_n}}$ is calculated, where v_{i_p} and v_{i_n} are the vertices before and after v_i, respectively. Then, the most favorable vertex with the largest move gain is selected and added to the route. The add operator is repeatedly applied until the constraint on the travel cost limit c_{max} is violated. For the PCTSP, the add operator is triggered to insert unrouted vertices when the collected profit is below the minimum profit threshold p_{min}. To collect more profits while maintaining the travel costs as low as possible, the vertex that leads to the minimum increase in the travel costs is selected and added to the route.

The worst time complexity of the add operator is $\mathcal{O}(|\varphi| \times (n - |\varphi|))$, where $|\varphi|$ is the number of vertices visited in the solution. This complexity can be reduced to $\mathcal{O}(\delta \times (n - |\varphi|))$ by considering only the δ-nearest vertices (δ is a parameter known as the granularity threshold) and using the streamlining techniques of [25].

3.3.2 Remove operator

This operator is applied to remove the visited vertices. For the OP, given a routed vertex v_i, the move gain of removing v_i is determined by $\Delta = \frac{p_i}{c_{i_p i}+c_{i i_n}-c_{i_p i_n}}$, where v_{i_p} and v_{i_n} are the vertices before and after v_i, respectively. If the solution is infeasible; that is, the travel costs are greater than c_{max}, the vertex associated with the smallest Δ is removed. The remove process stops when the solution becomes feasible. For the PCTSP, the operator attempts to reduce the travel costs as far as possible while maintaining the feasibility of the solution; that is, the collected profit is greater than the minimum profit threshold p_{min}. To achieve this, a vertex v_i associated with the maximum $\Delta = c_{i_p i} + c_{i i_n} - c_{i_p i_n}$ value is targeted and removed from the route, where v_{i_p} and v_{i_n} are the vertices before and behind v_i in the route, respectively. The process terminates when the removal of any vertex in the route reduces the collected profit to less than p_{min}. The time complexity of the remove operator is bounded by $\mathcal{O}(|\varphi|)$.

3.3.3 Application of move operators

It is important to decide the order in which the add and remove operators, as well as the 2-opt operator, are applied. Given that the OP and PCTSP pursue different objectives with different constraints, HGA applies a specific order for each problem. As indicated in Algorithms 2 and 3, for both problems, the 2-opt operator is first applied to reduce the travel costs as far as possible.

Then, for the OP, the remove operator is used to restore the feasibility when the travel costs exceed c_{max}, followed by the add operator to increase the profits and the 2-opt operator to reduce the travel costs as far as possible. For the PCTSP, the add operator is used to insert new vertices to satisfy the minimum profit constraint p_{min}, followed by the remove operator and 2-opt operator to reduce the travel costs as far as possible. Once the solution cannot be improved further, the local search phase terminates and returns the best solution.

Algorithm 2: The local search procedure for OP

Input: Solution φ;
Output: Local optima solution φ;

```
begin
    φ ← 2-opt (φ); /* Reducing the travel costs */
    φ ←Remove (φ); /* Restoring feasibility */
    var ← 0;
    while var ≠ f(φ) do
        var = f(φ);
        φ ←Add (φ); /* Adding vertices */
        φ ← 2-opt (φ);
    end
    return φ;
end
```

Algorithm 3: The local search procedure for PCTSP

Input: Solution φ;
Output: Local optima solution φ;

```
begin
    φ ← 2-opt (φ); /* Reducing the length */
    φ ←Add (φ); /* Restoring feasibility */
    var ← 0;
    while var ≠ f(φ) do
        var = f(φ);
        φ ←Remove (φ); /* Removing vertices */
        φ ← 2-opt (φ);
    end
    return φ;
end
```

3.4 Diversity preservation

Diversity is a key factor in population-based algorithms. HGA employs two different complementary strategies; that is, a specific mutation and dedicated population management, to preserve the population diversity effectively.

3.4.1 Mutation

An offspring solution created by E^2AX exclusively inherits the edges of its parents. That is, E^2AX cannot introduce vertices that are not visited by both parents into the offspring solutions. Furthermore, the local search can rarely introduce unrouted vertices into the solution, given that adding new vertices often increases the travel costs, which is undesirable. Consequently, the offspring solution may resemble the parents even after local optimization. To maintain sufficient diversity and avoid premature convergence, HGA applies a mutation with probability τ to modify each offspring solution by adding new vertices. The mutation removes some vertices from the solution and then greedily inserts unrouted vertices into the solution while respecting the corresponding constraints (i.e., the maximum travel costs c_{max} for the OP and minimum collected profit p_{min} for the PCTSP).

Given a solution φ, let $\mathcal{N}_\varphi$ and $\overline{\mathcal{N}}_\varphi$ be sets of routed and unrouted vertices in φ, respectively. The mutation process consists of two steps. First, l vertices (l is a parameter known as the mutation length) are selected and removed individually, and all of them are saved in set $\mathcal{T}$. Specifically, vertex v_i is selected for removal if its removal leads to the minimum move gain $\Delta = \frac{p_i}{c_{i_p i} + c_{i i_n} - c_{i_p i_n}}$, where v_{i_p} and v_{i_n} are the vertices before and behind v_i, respectively. Each removed vertex is forbidden from being reinserted into the route during the mutation. Second, vertex v_j is selected from $\overline{\mathcal{N}}_\varphi \backslash \mathcal{T}$ such that its insertion leads to the maximum increase in $\Delta = \frac{p_i}{c_{i_p i} + c_{i i_n} - c_{i_p i_n}}$ and is inserted into solution φ. For the OP, the insertion process stops when l unrouted vertices are inserted or if any of the possible insertions would render the solution infeasible (i.e., the travel costs would exceed c_{max}). For the PCTSP, the move gain for inserting an unrouted vertex is computed in the same manner as that for the OP. The insertion terminates when l vertices are inserted or the insertion makes the solution feasible (i.e., the collected profit reaches p_{min}). For the OP and PCTSP, the mutation operator aims to promote diversity by introducing as many unrouted vertices as possible. Cost-effective vertices can be considered as promising for improving the solution quality. Thus, the move gain of the PCTSP differs in the local search procedure. In Section 4.3, we demonstrate the importance of this mutation experimentally.

3.4.2 Population management

To maintain suitable population diversity $\mathcal{P}$, HGA adopts a variable population scheme similar to that used in [51], where clone solutions are not permitted in the population. From an initial population of λ solutions, the population is extended by the offspring solutions until the size reaches the upper limit $\mu + \lambda$, where μ is the generation size. When this occurs, the surviving selection is triggered to remove μ solutions with respect to the fitness and their contributions to the diversity of the population. Similar to Vidal [51], the distance between the two solutions is defined as the number of distinct edges. Let $|\mathcal{P}|$ denote the number of solutions to $\mathcal{P}$. Given a solution φ, the distances between φ and the other $|\mathcal{P}|-1$ solutions are computed and sorted from smallest to largest. Subsequently, the sum of the first $nbClost$ values ($nbClost$ is a parameter) is used as the diversity contribution of φ to $\mathcal{P}$, which is denoted by div_φ. Thus, each solution $\varphi \in \mathcal{P}$ is associated with a div_φ value. All these values are sorted from smallest to largest and each solution is associated with a rank rd_φ with respect to div_φ. Furthermore, we rank the solutions of $\mathcal{P}$ according to their objective values from worst to best, leading to another rank ro_φ for each solution φ. Finally, the biased fitness of solution φ is defined as $f(\varphi)_{biased} = ro_\varphi + (1 - \frac{nbElite}{|\mathcal{P}|}) \times rd_\varphi$, where $nbElite$ is a parameter. The biased fitness that we use aims to prevent premature convergence of the population by identifying and preserving the most promising and diversified solutions. The solution that is associated with the smallest biased fitness is removed from $\mathcal{P}$ and the biased fitness for each remaining solution in $\mathcal{P}$ is updated. The solution-removal process is repeated until there are λ solutions in $\mathcal{P}$. Following [51], we set $nbClost = 5$ and $nbElite = 4$.

If the best solution found thus far, $\varphi*$, cannot be improved over γ consecutive iterations (γ is a parameter known as the population rebuilding threshold and one iteration is the generation of one offspring solution followed by the local search), the algorithm restarts by generating a completely new population.

4 Computational results and comparisons

In this section, we evaluate the performance of the proposed algorithm on the OP and PCTSP. We present the benchmark instances, experimental protocol, reference algorithms, and comparisons with state-of-the-art methods.

4.1 Benchmark instances

For the OP, four sets of instances are used in the literature, all of which were introduced by Kobeaga et al. [25]. Each set includes 86 instances that are divided into two groups: medium-sized instances with up to 400 vertices and large-sized instances with up to 7,397 vertices. For the first three sets, the maximum travel cost $c_{max} = \lceil \alpha \cdot v(TSP) \rceil$, where $v(TSP)$ is the length of the shortest Hamiltonian route that visits all vertices and $\alpha = 0.5$. The profit of each vertex is generated using the three methods described by Fischetti et al. [15]. In the final set, α takes different values, whereas all vertices have the same profits as in the second set. Furthermore, Vansteenwegen and Gunawan [49] collected various OP benchmark instances, which are available online [1], including many small-sized instances. Because the four sets of 344 instances in

[1] https://www.mech.kuleuven.be/en/cib/op

[25] are representative, we ignore the small-sized instances mentioned in [49].

There are no unified instances for the PCTSP. We followed [4] and used the method in [15] to generate three sets of 240 instances with up to 7,397 vertices, where each set includes 80 instances and is further divided into two groups: medium-sized instances with up to 532 vertices and large-sized instances with up to 7,397 vertices. The profit of each vertex is the same as that in [4]. Furthermore, Vansteenwegen [48] stated that the most difficult instances are those where the selected number of vertices is slightly greater than half of the total number of vertices. Consequently, we set $p_{min}=\lfloor 0.5 \cdot \sum_{i\in\mathcal{N}} p_i \rfloor$.

These 344 instances for the OP and 240 instances for the PCTSP were used in our experiments to evaluate the performance of the proposed HGA. Indeed, both the OP and PCTSP benchmark instances were obtained from TSP benchmark instances and the prize for each vertex was generated in the same manner. The instances and best solutions that were obtained by HGA are available online [2].

4.2 Experimental protocol and reference algorithms

Parameter settings. HGA has six main parameters: the minimum population size λ and generation size μ, granularity threshold δ that is used in the local search, mutation probability τ, mutation length l, and population rebuilding threshold γ. The automatic parameter tuning package Irace [30] was used to identify suitable values for the parameters. The candidate and final values are listed in Table 2. These parameter values can be considered to constitute the default settings and were used consistently in our experiments.

Table 2
Parameter tuning results.

Parameter	Section	Description	Considered values	Final values	
				OP	PCTSP
λ	3.1 and 3.4.2	minimal size of population	$\{50, 100, 150, 200, 250\}$	100	100
μ	3.1 and 3.4.2	generation size	$\{25, 50, 75, 100, 125\}$	50	100
δ	3.3	granularity threshold	$\{5, 8, 10, 12, 15, 20\}$	10	12
τ	3.4.1	mutation probability	$\{0, 0.05, 0.1, 0.15, 0.2, 0.25, 0.3\}$	0.15	0.1
l	3.4.1	mutation length	$\{0.05, 0.1, 0.15, 0.2, 0.25\}$	0.25	0.25
γ	3.4.2	population rebuilding threshold	$\{5000, 10000, 20000, 30000, 50000, 80000\}$	30000	30000

Reference algorithms. According to the review in Section 2, we identified the best heuristic and exact algorithms for the OP and used them for our comparative study.

- BKS. This indicates the best known solutions (best lower bounds) that were compiled from all reference heuristic and exact approaches [26,44,25].
- RB&C [26]. This exact algorithm [26] was applied to solve the first three sets of instances and could obtain optimal solutions for many instances under a

[2] https://github.com/pengfeihe-angers/tsps-with-profits.git

time limit of 18,000 seconds.

- ALNS [44]. This algorithm was implemented in C++ and executed on an Intel Xeon E5 processor, running at 2.2 GHz under a time limit of 18,000 seconds or after 250,000 iterations without improvement. The algorithm was executed 10 times on each instance. It was tested on the four sets of instances.
- EA4OP [25]. This hybrid algorithm was implemented in C and executed on an Intel Xeon E5-2609 v3 1.90 GHz processor with 4 GB RAM. The algorithm terminates either when the first quartile of the population's fitness is the same as the best fitness or when the maximum running time exceeds 18,000 seconds. The algorithm was executed 10 times on each instance. It was tested on the four sets of instances.
- B&C [25]. This is the B&C algorithm that was presented in [15] and rerun in [25]. It stops when the maximum running time (18,000 seconds) is met or when the optimal solution is found. This algorithm was tested on the fourth set only.

For the PCTSP, only the B&C algorithm [4] was tested on medium-sized instances with up to 532 vertices. To obtain a reference algorithm for large-sized instances with up to 7397 vertices, we created an HGA variant (HGA-Giant), where we replaced E^2AX with a giant tour crossover, as described in Appendix B.1.

Experimental settings and stopping criterion. The HGA algorithm was implemented in C++ and compiled using the g++ compiler with the -O3 option [3]. All experiments were run on an Intel Xeon ES-2630 processor with 2.66 GHz and 6 GB RAM running Linux with a single thread. The algorithm was executed 20 times for each instance, with distinct random seeds. Following the literature, HGA terminated when it reached a time limit of 18,000 seconds or a maximum of 500,000 iterations (one iteration means the generation of one offspring solution followed by one local search run).

4.3 Computational results

To compare HGA and reference algorithms, two summarizing tables are presented for the OP and PCTSP, respectively. The Wilcoxon signed-rank test with a confidence level of 0.05 was applied to verify the statistically significant differences between HGA and each reference algorithm. A *p-value* lower than 0.05 indicates a significant difference.

[3] The code for the HGA algorithm will be available at https://github.com/pengfeihe-angers/tsps-with-profits.git

4.3.1 Comparative results on the OP

Because the two reference heuristic algorithms, namely ALNS [44] and EA4OP [25], did not provide their average values, we focus on the best objective values of the compared algorithms in Table 3. Detailed results for the four sets of 344 instances are presented in Tables B.1-B.8.

Regarding the BKS values, which represent the best values ever reported by all algorithms, HGA outperforms 67 BKS values (new lower bounds) out of 344 instances (19.5%) and matches 172 other BKS values (50%). Specifically, for both the medium- and large-sized instances from Set I, HGA exhibits a worse performance compared to the BKS values. For the medium-sized instances from Sets II, III, and IV, our HGA competes favorably with the BKS values, and the *p-values* (≥ 0.05) from the Wilcoxon signed-rank test reveals no significant statistical difference between the results of HGA and BKS values. For the large-sized instances from Sets II and III, although HGA yields several new bounds, the *p-values* (≥ 0.05) indicates that there are no significant differences between the compared results. Finally, for the large-sized instances from Set IV, our algorithm achieves remarkable performance compared to the BKS values, and the *p-values* (< 0.05) clearly indicates that the differences are statistically significant. Given that the BKS values are the best results compiled from all existing approaches, HGA can be considered to achieve a highly competitive performance.

HGA significantly outperforms the two best heuristic algorithms, namely ALNS and EA4OP (*p-value* $\ll 0.05$), except for ALNS on the first set. Furthermore, the two best exact algorithms, namely RB&C and B&C, obtain many optimal solutions for medium-sized instances within a reasonable running time, but their results and running times become unacceptable with the increase in instance size. As shown in Tables B.4, B.6 and B.8, HGA provides significant improvements for large-sized instances, particularly for instances with at least 2,000 vertices.

Tables B.1-B.8 present detailed results for all 344 OP instances. Although EA4OP exhibits a very short running time, its results are much worse than those of ALNS and HGA. Compared with ALNS, our HGA could obtain better results with a shorter running time. It is noticeable that exact algorithms require a very short time for medium-sized instances to obtain optimal solutions; however, the gap becomes unacceptable for large-sized instances. Thus, even if HGA can find high-quality solutions in a short time for small- and medium-sized instances, its main advantage is its capacity to solve large-sized OP instances.

The time limit of 5 hours is adopted from the literature to allow a fair comparison with the reference algorithms. In practice, our HGA algorithm needs

Table 3
The OP: summary of results between HGA and reference algorithms on the four sets of 344 instances in terms of the best objective values.

Instances	Pair algorithms	Medium-sized (45)				Large-sized (41)			
		#Wins	#Ties	#Losses	*p-value*	#Wins	#Ties	#Losses	*p-value*
Set I	HGA vs. BKS	0	35	10	2.00E-03	3	1	37	5.51E-06
	HGA vs. RB&C [26]	0	35	10	2.00E-03	5	1	35	4.62E-05
	HGA vs. EA4OP [25]	**12**	29	4	1.80E-02	**32**	2	7	6.70E-06
	HGA vs. ALNS [44]	3	35	7	4.59E-01	**20**	4	17	7.61E-02
Set II	HGA vs. BKS	0	43	2	5.00E-01	13	2	26	5.53E-01
	HGA vs. RB&C [26]	0	43	2	5.00E-01	13	2	26	7.64E-01
	HGA vs. EA4OP [25]	**31**	14	0	1.17E-06	**41**	0	0	2.42E-08
	HGA vs. ALNS [44]	**16**	29	0	4.35E-04	**40**	0	1	2.61E-08
Set III	HGA vs. BKS	0	43	2	5.00E-01	19	3	19	7.10E-02
	HGA vs. RB&C [26]	0	43	2	5.00E-01	19	3	19	6.24E-02
	HGA vs. EA4OP [25]	**28**	15	2	1.64E-05	**39**	0	2	5.26E-08
	HGA vs. ALNS [44]	**14**	29	2	1.13E-02	**38**	0	3	6.14E-08
Set IV	HGA vs. BKS	2	41	2	8.75E-01	**30**	4	7	1.54E-05
	HGA vs. B&C [25]	2	41	2	8.75E-01	**30**	4	7	4.15E-06
	HGA vs. EA4OP [25]	**27**	17	1	6.57E-05	**39**	0	2	7.81E-08
	HGA vs. ALNS [44]	**20**	24	1	1.01E-03	**39**	2	0	5.25E-08
Summary	HGA vs. BKS	2	162	16	-	65	10	89	-

much less time to reach its best results (as the TMB values showed in the tables). Compared to the exact algorithm for the OP, HGA generally attains the optimal values or better lower bounds within a much shorter time (often no more than half of the time of the exact algorithm), except for the large instances of Set I for which HGA finds (good) suboptimal solutions (gap from 0.29% to 5.11%) within 35% of the time needed by the exact algorithm. Finally, it is somewhat difficult to compare a heuristic (which aims to find the best possible solution as soon as possible) and an exact algorithm (which aims to find the optimal solution and prove its optimality).

4.3.2 Comparative results on the PCTSP

Table 4
The PCTSP: summary of best results between HGA and B&C [4] on the three sets of 138 medium-sized instances.

Instances	Pair algorithms	Medium-sized (46)			
		#Wins	#Ties	#Losses	*p-value*
Set I	HGA vs. B&C [4]	4	37	5	8.20E-01
Set II	HGA vs. B&C [4]	**7**	36	3	4.92E-01
Set III	HGA vs. B&C [4]	11	21	14	3.06E-01
	Summary	**22**	94	22	-

To demonstrate the effectiveness of our algorithm for the PCTSP, we compare HGA with the exact B&C algorithm [4] on the three sets of 138 medium-sized instances. Table 4 summarizes the comparative results and the detailed results of our algorithm on PCTSP are provided in Tables B.9-B.14. As indicated in Table 4, HGA competes well with B&C for the 138 medium-sized instances. Indeed, our HGA obtains 22 new bounds, although no statistically significant difference is observed between the compared results. It can also be observed from Table 4 that B&C fails to obtain optimal solutions for large-sized instances with more than 400 vertices, although it solves several medium-sized instances to optimality. However, the running time of B&C increases significantly with the size of the instance. Meanwhile, HGA finds high-quality solutions for large-sized instances within a short running time. In particular, HGA reaches new upper bounds for 120 out of the 240 instances (50%), matches the best solutions for 96 instances (40%), and misses the best known results for only 24 cases (10%). Additionally, HGA attains the optimal values or better upper bounds with only 20% of the time needed by the exact algorithm.

4.4 Discussion

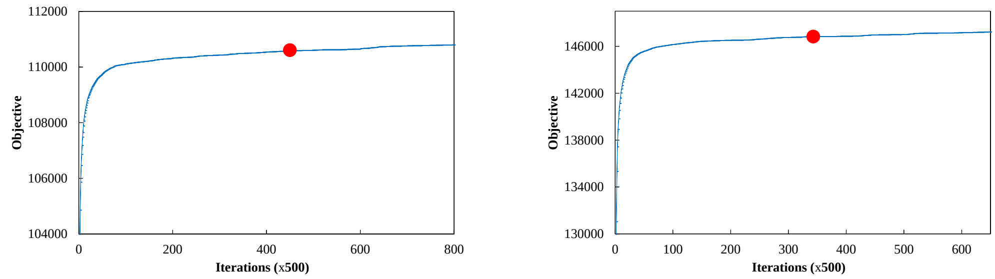
We now present several discussions related to the long-term behavior of HGA, generality of the hybrid algorithmic framework as well as the E^2AX operator, and instance features on the performance of the algorithms.

4.4.1 Long-term behavior of HGA

Our HGA algorithm uses a stopping condition defined by a time limit of 18,000 seconds or a maximum of 500,000 iterations of the neighborhood search. It is worth investigating whether the results of the algorithm could be further improved by prolonging the running time. To answer this question and to observe the behavior of HGA over time, Fig. 2 presents the running profiles on two representative instances (fl3795 in Set II and fl4461 in Set III of the OP instances). The running profiles are defined by the function $i \rightarrow f$, where i is the number of iterations and f is the achieved objective value at iteration i averaged over 20 runs. The red dots in Fig. 2 indicate the average objective values obtained at the end of the standard stopping conditions. It can be observed from Fig. 2 that the results of HGA on these two large-sized instances can be further improved when the running time is prolonged. Indeed, the best results for these two instances are 111086 and 148038, which are better than the best results reported in Tables B.3 and B.4 (11098 and 147641). This experiment confirms that HGA exhibits a highly desirable long-term search behavior, and it is expected to discover better solutions by taking advantage of prolonged stopping conditions.

4.4.2 Other applications of HGA and E^2AX operator

Although HGA along with the E^2AX operator is designed for solving TSPs with profits, its algorithmic framework and the idea of its crossover can be conveniently adapted to solve multi-vehicle problems such as the split delivery vehicle routing problem (SDVRP) [13] and team orienteering problem (TOP) [9] with some adjustments. For instance, in [22], the SDVRP was addressed by a memetic algorithm (SplitMA), which follows the same hybrid algorithmic framework and integrates a general edge assembly crossover (gEAX) as well



(a) fl3795-gen2 (b) fl4461-gen3

Fig. 2. Running profiles of the HGA algorithm on two representative instances.

as dedicated local search operators for the SDVRP. The SplitMA algorithm reported excellent results on the set of 162 popular benchmark instances, as shown in Table 5 (data extracted from [22]) where the BKS indicates the best known objective values ever reported in the literature. Specifically, for the SDVRP with limited fleet (SDVRP-LF), SplitMA finds 70 new upper bounds (43%), matches the BKS values for 75 other instances (46%) and only misses 17 BKS values (10%). SplitMA also significantly dominates its competitors including multistart iterated local search (SplitILS, 2015), iterative constructive and variable neighbourhood descent with diversification (iVNDiv, 2009), random granular tabu search (RGTS, 2014), scatter search (SS, 2008), and hybrid genetic algorithm (HGA, 2012). For the SDVRP with unlimited fleet (SDVRP-UF), SplitMA updates 73 BKS values (new upper bounds) and matches 81 other BKS values. Once again, SplitMA performs significantly better than the reference algorithms including tabu search with vocabulary building (TSVBA, 2010), forest-based tabu search (FBTS, 2015) memetic algorithm with population management (MAPM , 2007), and attribute based hill climber heuristic (ABHC, 2010). It is worth mentioning that the SplitMA algorithm was ranked second at the 12th DIMACS Implementation Challenge on Vehicle Routing - SDVRP Track.

Finally, the E^{2}AX operator does not consider edge directions when constructing *AB-cycles* and cannot be directly applied to directed cases. However, this crossover can be further extended to directed cases by considering the edge directions when building *AB-cycles*. We will explore this possibility in the future.

4.4.3 Influence of instance features on algorithm performance

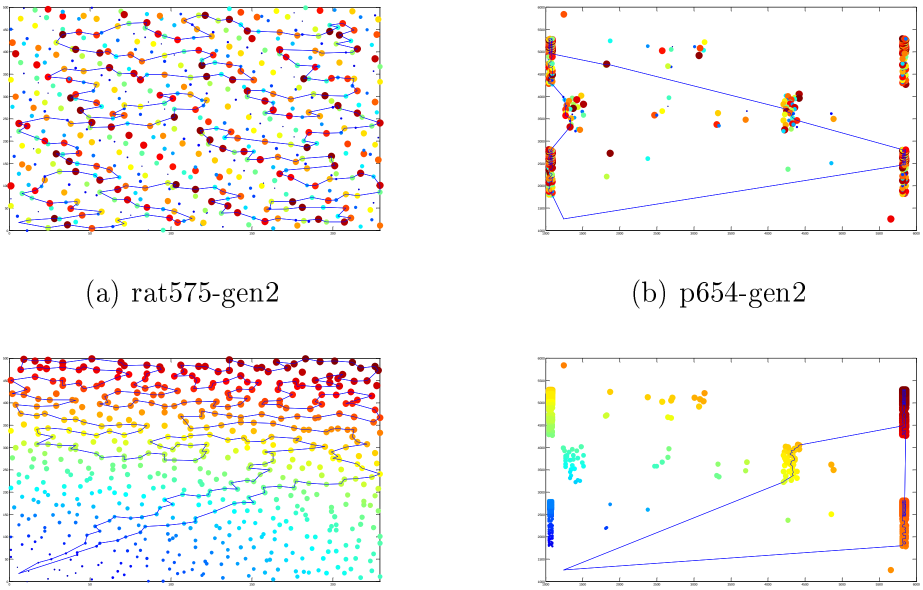
For the OP, we observe that the performance of the exact algorithms [4,25] is dependent on the size of the instances and the variability of the profits assigned to the vertices. Indeed, for the instances of Set I, all vertices have the same prize ($p_i = 1$), and the objective of the OP is reduced to cover as many vertices as possible. These instances are easily solved to optimality by the

Table 5
Summary of comparative results of SplitMA and reference algorithms for the split delivery vehicle routing problem with limited fleet (upper part) and unlimited fleet (lower part) in terms of the best objective values [22].

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

exact algorithms. For the instances in Sets II and III, all vertices are assigned a pseudo-random prize between 1 and 100, and larger prizes are assigned to vertices far from the depot. According to the results in Tables B.1-B.8, these instances are more challenging for the exact algorithms and many instances cannot be solved to optimality. Regarding our HGA, we observe that for large-sized instances and instances where the profits of the vertices are not uniformly distributed, HGA is more robust and powerful than the exact algorithms.

Indeed, additional experiments were conducted to investigate properties of instances that may affect the performance of HGA. Four representative instances with similar sizes, but different profit distributions (rat575-gen2 and rat575-gen3 with a uniform geographic distribution, p654-gen2 and p654-gen3 with a clustered distribution). The best results on these instances are illustrated in Fig. 3, with profits represented by circles. From Table B.4, we find that the time needed to hit the best result (TMB) of rat575-gen2 is significantly larger than that of p654-gen2, even if the running time of both instances is similar, indicating that HGA converges quickly when solving the latter (clustered) instance. From Fig. 3(a)-(b) and experimental logs, we observe that HGA spends more time carrying out the local search procedure to find local optima for instances with a uniform geographic distribution. This is particularly evident for Set III. For rat575-gen3 (random distribution) and p654-gen3 (clustered distribution), it can be observed that the running time and TMB for the clustered instance are considerably less than for the instance with a random distribution. As shown in Fig. 3(c) - (d) and experimental logs, HGA can easily find three clusters and attain local optima quickly when solving p654-gen3 (clustered distribution). Conversely, HGA has to spend a larger amount of time carrying out the local search procedure when solving rat575-gen3 (random distribution). Therefore, it is safe to conclude that HGA is more advantageous in solving instances with a clustered geographic distribution. Finally, we also observe that the size of instances and variability of profits also influence, to a certain extent, the performance of the algorithm.



(a) rat575-gen2 (b) p654-gen2

(c) rat575-gen3 (d) p654-gen3

Fig. 3. Graph structures of four representative instances.

5 Assessment of algorithmic components

In this section, we describe additional experiments that were conducted to study the benefits of the two key components of the proposed algorithm. The experiments were based on the instances of Sets II and III for the OP.

5.1 Significance of crossover

To assess the significance of E^2AX within HGA, we created an HGA variant (HGA-Giant), in which E^2AX was replaced with the giant tour crossover [6] (see Appendix B.1) and another HGA variant (HGA1), where E^2AX was disabled in HGA. We ran these algorithms under the same stopping condition as before. The comparative results are presented in Table 6 and Fig. 4.

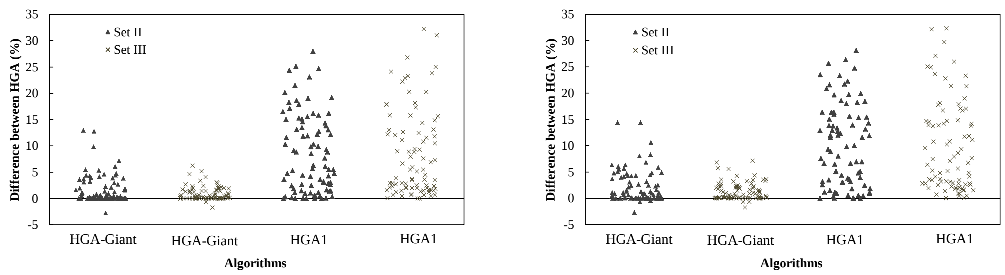
From these results, we observe that E^2AX played a strongly positive role in the good performance of HGA. Indeed, HGA dominated HGA-Giant by obtaining 108 better results and 61 equal results out of the 172 tested instances. HGA1 (without crossover) exhibited the worst performance compared with HGA and HGA-Giant, indicating that crossovers such as E^2AX and the giant tour are highly beneficial for the performance of the hybrid algorithm. In particular, for the PCTSP, HGA significantly outperformed HGA-Giant in terms of both the objective value and computational efficiency, as confirmed by the *p-values* $<$

Table 6
Comparative results between HGA and HGA-Giant (using the giant tour crossover).

Instances	Pair algorithms	OP							
		Best				*Avg.*			
		#Wins	#Ties	#Losses	*p-value*	#Wins	#Ties	#Losses	*p-value*
Set II (86)	HGA vs. HGA-Giant	**51**	34	1	4.25E-09	**65**	18	3	3.65E-11
	HGA vs. HGA1	**81**	5	0	5.36E-15	**83**	3	0	2.50E-15
Set III (86)	HGA vs. HGA-Giant	**57**	27	2	1.55E-10	**68**	15	3	4.84E-12
	HGA vs. HGA1	**84**	2	0	1.71E-15	**85**	1	0	1.17E-15
		PCTSP							
		Best				*Avg.*			
		#Wins	#Ties	#Losses	*p-value*	#Wins	#Ties	#Losses	*p-value*
Set I (80)	HGA vs. HGA-Giant	**65**	12	3	1.34E-11	**75**	0	5	2.43E-12
Set II (80)	HGA vs. HGA-Giant	**75**	3	2	5.70E-13	**77**	1	2	1.98E-13
Set III (80)	HGA vs. HGA-Giant	**73**	5	2	5.57E-13	**77**	0	3	2.24E-13

0.05. According to the detailed results for the PCTSP in Tables B.9-B.14, HGA required only half of the time required by HGA-Giant to find solutions of equal or better quality on medium-sized instances, and it could reach better solutions than HGA-Giant with a shorter running time on large-sized instances.

In summary, E^2AX positively contributes to the performance of HGA and outperforms the giant tour crossover.



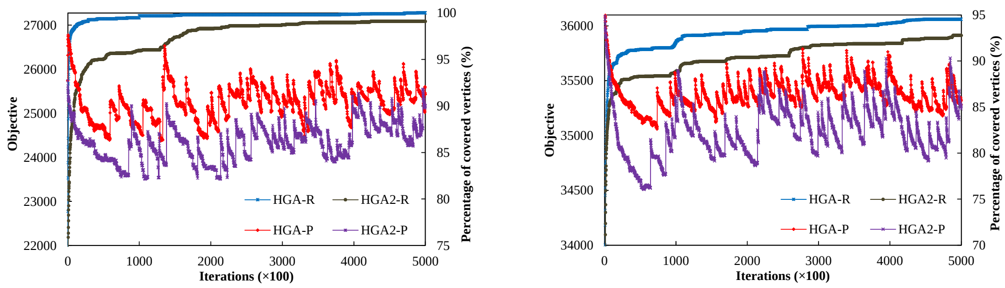
(a) Best (b) Avg.

Fig. 4. The differences between HGA and two variants for solving the instances of Sets II and III of the OP.

5.2 Benefits of mutation

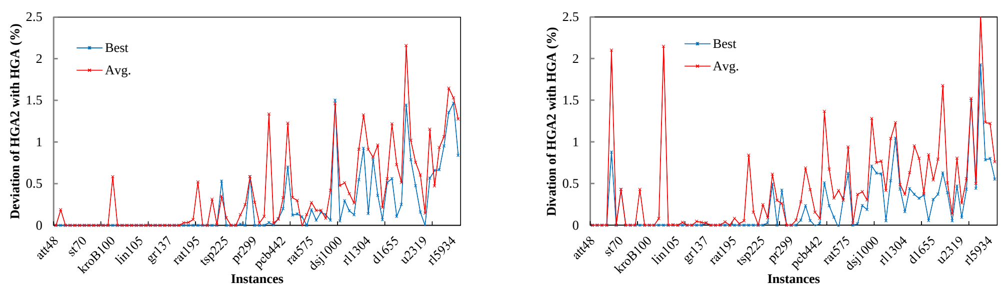
In HGA, the mutation operator is used to preserve the diversity of the population. To assess its usefulness, an HGA variant (HGA2) was created by disabling the mutation operator. We compared HGA and HGA2 in terms of the population diversity using the following diversity measure: Let $|\mathcal{P}|$ be the number of solutions in population $\mathcal{P}$. Let $\mathcal{N}_\varphi$ be the set of vertices visited by solution φ in $\mathcal{P}$. Let $\mathcal{H}$ be the set of vertices visited by all solutions in $\mathcal{P}$ and $\mathcal{H} = \cup_{i=1}^{|\mathcal{P}|} \mathcal{N}_{\varphi_i}$. Let ξ be the proportion of vertices covered by $\mathcal{P}$ and $\xi = \frac{|\mathcal{H}|}{n}$, $0 < \xi \leq 1$. We used the value of ξ to measure the population diversity. If $\xi \to 1$, $\mathcal{P}$ covers many vertices, offering good possibilities for the algorithm to explore larger search spaces and vice versa. We present the convergence charts of HGA and HGA2 together with the evolution of the population diversity based on two instances (rat783-gen3 and u1060-gen2). The results are presented in Fig. 5, where HGA-R and HGA2-R indicate the best results found, whereas HGA-P and HGA2-P are the current diversity values ξ of the population. It should be noted that HGA exhibited better convergence and dominated its counterpart in both instances. It is observed that HGA always maintained a higher value ξ along its evolution compared to HGA2, which indicates the contribution of the mutation to the diversity and performance of the HGA algorithm.

Finally, Fig. 6 depicts the comparative results of HGA and HGA2 in terms of both the best and average objective values on the 86 instances of Set II and 86 instances of Set III (the names of 15 instances are shown). The results are presented as the deviation in the percentage of the HGA2 results compared with the HGA results. For medium-sized instances, HGA and HGA2 obtained similar results. However, for instances with more than 200 vertices, HGA2 performed worse than HGA, and the difference became more significant as the size of the instances increased. These results confirm that the mutation operator plays a crucial role in HGA, especially for large-sized instances.



(a) rat783-gen3 (b) u1060-gen2

Fig. 5. Convergence charts of HGA and HGA2 for solving two representative instances.



(a) Set II (b) Set III

Fig. 6. Results of HGA2 (without mutation) in terms of deviation in percentage compared to the results of HGA (with mutation).

6 Conclusions

This study has presented a new hybrid genetic algorithm to address two TSPs with profits efficiently. We introduced several methodological contributions, including an extended edge-assembly crossover for producing promising solutions, an effective local search for solution refinement, and specific strategies for preserving the diversity of the population.

Extensive experiments were conducted on the OP and PCTSP. For the OP, four sets of 344 commonly used instances were tested, and 67 new lower bounds were discovered. The algorithm also matched the best known results for 172 other instances. For the PCTSP, the results on three sets of 240 instances exhibited high performance on large-sized instances, including 120 new best results that have never been reported in the literature. Additional experiments were conducted to obtain insight into the benefits of the proposed crossover and mutation. The new bounds reported in this study may be useful for future research on these issues.

The proposed algorithm can be further improved by investigating more powerful streamline techniques to increase the computational efficiency and to deal with larger problem instances. Moreover, this study confirms the merit of the general concept of assembling promising edges from elite parents, which may aid in the design of new crossovers for other routing problems such as multi-vehicle and directed cases.

Acknowledgments

We are grateful to the reviewers for their insightful and constructive comments, which helped us to significantly improve the paper. The authors also would like to thank the following colleagues: Dr. G. Kobeaga [25,26], Prof. M. Gendreau and Prof. J-Y. Potvin [4] for their kind helps. This work is partially supported by the National Natural Science Foundation Program of China (Grant No. 72122006). Support from the China Scholarship Council (CSC, No. 201906850087) for the first author is also acknowledged.

References

[1] E. Balas, The prize collecting traveling salesman problem, Networks: An International Journal 19 (6) (1989) 621–636.

[2] E. Balas, The prize collecting traveling salesman problem: Ii. polyhedral results, Networks: An International Journal 25 (4) (1995) 199–216.

[3] E. Balas, The prize collecting traveling salesman problem and its applications,

in: The traveling salesman problem and its variations, Springer, 2007, pp. 663–695.

[4] J.-F. Bérubé, M. Gendreau, J.-Y. Potvin, A branch-and-cut algorithm for the undirected prize collecting traveling salesman problem, Networks: An International Journal 54 (1) (2009) 56–67.

[5] D. Bienstock, M. X. Goemans, D. Simchi-Levi, D. Williamson, A note on the prize collecting traveling salesman problem, Mathematical Programming 59 (1) (1993) 413–420.

[6] H. Bouly, D.-C. Dang, A. Moukrim, A memetic algorithm for the team orienteering problem, 4OR 8 (1) (2010) 49–70.

[7] V. Campos, R. Martí, J. Sánchez-Oro, A. Duarte, Grasp with path relinking for the orienteering problem, Journal of the Operational Research Society 65 (12) (2014) 1800–1813.

[8] I.-M. Chao, B. L. Golden, E. A. Wasil, A fast and effective heuristic for the orienteering problem, European Journal of Operational Research 88 (3) (1996) 475–489.

[9] I.-M. Chao, B. L. Golden, E. A. Wasil, The team orienteering problem, European Journal of Operational Research 88 (3) (1996) 464–474.

[10] A. A. Chaves, L. A. N. Lorena, Hybrid metaheuristic for the prize collecting travelling salesman problem, in: European Conference on Evolutionary Computation in Combinatorial Optimization, Springer, 2008.

[11] G. Clímaco, L. Simonetti, I. Rosseti, A branch-and-cut and mip-based heuristics for the prize-collecting travelling salesman problem, RAIRO: Recherche Opérationnelle 55 (2021) 719.

[12] G. A. Croes, A method for solving traveling-salesman problems, Operations Research 6 (6) (1958) 791–812.

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

[14] D. Feillet, P. Dejax, M. Gendreau, Traveling salesman problems with profits, Transportation Science 39 (2) (2005) 188–205.

[15] M. Fischetti, J. J. S. Gonzalez, P. Toth, Solving the orienteering problem through branch-and-cut, INFORMS Journal on Computing 10 (2) (1998) 133–148.

[16] M. Gendreau, G. Laporte, F. Semet, A branch-and-cut algorithm for the undirected selective traveling salesman problem, Networks: An International Journal 32 (4) (1998) 263–273.

[17] M. Gendreau, G. Laporte, F. Semet, A tabu search heuristic for the undirected selective travelling salesman problem, European Journal of Operational Research 106 (2-3) (1998) 539–545.

[18] B. L. Golden, L. Levy, R. Vohra, The orienteering problem, Naval Research Logistics 34 (3) (1987) 307–318.

[19] L. Gomes, V. Diniz, C. A. Martinhon, An hybrid grasp+ vnd metaheuristic for the prize-collecting traveling salesman problem, XXXII Simpósio Brasileiro de Pesquisa Operacional (2000) 1657–1665.

[20] F. Hammami, M. Rekik, L. C. Coelho, A hybrid adaptive large neighborhood search heuristic for the team orienteering problem, Computers & Operations Research 123 (2020) 105034.

[21] J.-K. Hao, Memetic algorithms in discrete optimization, in: Handbook of memetic algorithms, Springer, 2012, pp. 73–94.

[22] P. He, J.-K. Hao, General edge assembly crossover-driven memetic search for split delivery vehicle routing, Transportation Science (2022) https://doi.org/10.1287/trsc.2022.1180.

[23] P. He, J.-K. Hao, Memetic search for the minmax multiple traveling salesman problem with single and multiple depots, European Journal of Operational Research 307 (3) (2023) 1055–1070.

[24] M. Keshtkaran, K. Ziarati, A novel grasp solution approach for the orienteering problem, Journal of Heuristics 22 (5) (2016) 699–726.

[25] G. Kobeaga, M. Merino, J. A. Lozano, An efficient evolutionary algorithm for the orienteering problem, Computers & Operations Research 90 (2018) 42–59.

[26] G. Kobeaga, M. Merino, J. A. Lozano, A revisited branch-and-cut algorithm for large-scale orienteering problems, arXiv preprint arXiv:2011.02743.

[27] G. Laporte, S. Martello, The selective travelling salesman problem, Discrete Applied Mathematics 26 (2-3) (1990) 193–207.

[28] A. C. Leifer, M. B. Rosenwein, Strong linear programming relaxations for the orienteering problem, European Journal of Operational Research 73 (3) (1994) 517–523.

[29] Y.-C. Liang, A. E. Smith, An ant colony approach to the orienteering problem, Journal of the Chinese Institute of Industrial Engineers 23 (5) (2006) 403–414.

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

[31] Y. Marinakis, M. Politis, M. Marinaki, N. Matsatsinis, A memetic-grasp algorithm for the solution of the orienteering problem, in: Modelling, computation and optimization in information systems and management sciences, Springer, 2015, pp. 105–116.

[32] S. Martello, D. Pisinger, P. Toth, New trends in exact algorithms for the 0–1 knapsack problem, European Journal of Operational Research 123 (2) (2000) 325–332.

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

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

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

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

[37] I. Oliver, D. Smith, J. R. Holland, Study of permutation crossover operators on the traveling salesman problem, in: Genetic algorithms and their applications: proceedings of the second International Conference on Genetic Algorithms: July 28-31, 1987 at the Massachusetts Institute of Technology, Cambridge, MA, Hillsdale, NJ: L. Erlhaum Associates, 1987., 1987.

[38] K. Ostrowski, J. Karbowska-Chilinska, J. Koszelew, P. Zabielski, Evolution-inspired local improvement algorithm solving orienteering problem, Annals of Operations Research 253 (1) (2017) 519–543.

[39] A. Paul, D. Freund, A. Ferber, D. B. Shmoys, D. P. Williamson, Budgeted prize-collecting traveling salesman and minimum spanning tree problems, Mathematics of Operations Research 45 (2) (2020) 576–590.

[40] O. Pedro, R. Saldanha, R. Camargo, A tabu search approach for the prize collecting traveling salesman problem, Electronic Notes in Discrete Mathematics 41 (2013) 261–268.

[41] J.-Y. Potvin, State-of-the art review-evolutionary algorithms for vehicle routing, INFORMS Journal on Computing 21 (4) (2009) 518–548.

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

[43] R. Ramesh, K. M. Brown, An efficient four-phase heuristic for the generalized orienteering problem, Computers & Operations Research 18 (2) (1991) 151–165.

[44] A. Santini, An adaptive large neighbourhood search algorithm for the orienteering problem, Expert Systems with Applications 123 (2019) 154–167.

[45] J. Silberholz, B. Golden, The effective application of a new approach to the generalized orienteering problem, Journal of Heuristics 16 (3) (2010) 393–415.

[46] M. F. Tasgetiren, A. E. Smith, A genetic algorithm for the orienteering problem, in: Proceedings of the 2000 Congress on Evolutionary Computation, vol. 2, IEEE, 2000.

[47] T. Tsiligirides, Heuristic methods applied to orienteering, Journal of the Operational Research Society 35 (9) (1984) 797–809.

[48] P. Vansteenwegen, Planning in tourism and public transportation, 4OR 7 (3) (2009) 293–296.

[49] P. Vansteenwegen, A. Gunawan, State-of-the-art solution techniques for op and top, in: Orienteering Problems, Springer, 2019, pp. 41–66.

[50] P. Vansteenwegen, W. Souffriau, D. Van Oudheusden, The orienteering problem: A survey, European Journal of Operational Research 209 (1) (2011) 1–10.

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

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

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

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

[55] T. Vidal, N. Maculan, L. S. Ochi, P. H. Vaz Penna, Large neighborhoods with implicit customer selection for vehicle routing problems with profits, Transportation Science 50 (2) (2016) 720–734.

[56] Q. Wang, X. Sun, B. L. Golden, J. Jia, Using artificial neural networks to solve the orienteering problem, Annals of Operations Research 61 (1) (1995) 111–120.

Appendix

A Mathematical model

In this section, we propose a possible mathematical model for the undirected OP and PCTSP following [14]. Let x_e be a binary variable and $x_e = 1$ if and only if the edge (e) is used in the solution. Let y_i be a binary variable and $y_i = 1$ if and only if vertex v_i is visited. For every vertex subset $\mathcal{S}$, let $\delta(S)$ be the set of edges with one end in $\mathcal{S}$ and the other end in $\mathcal{V}\backslash\mathcal{S}$.

$$maximize \sum_{v_i \in \mathcal{V}} p_i y_i \tag{A.1}$$

$$\sum_{e \in \mathcal{E}} c_e x_e \leq c_{max} \tag{A.2}$$

$$\sum_{e \in \delta(\{v_i\})} x_e = 2y_i \quad (v_i \in \mathcal{V}) \tag{A.3}$$

$$\sum_{e \in \mathcal{S}} x_e \leq |\mathcal{S}| - 1 \quad (\mathcal{S} \subset \mathcal{V} \backslash \{v_0\}, 3 \leq |\mathcal{S}| \leq n-2) \tag{A.4}$$

$$y_0 = 1 \tag{A.5}$$

$$x_e \in \{0,1\} \quad (e \in \mathcal{E}) \tag{A.6}$$

$$y_i \in \{0,1\} \quad (v_i \in \mathcal{V}) \tag{A.7}$$

We refer to constraints A.2 as knapsack constraints. Constraints A.3 are so-called assignment constraints, whereas constraints A.4 are used to eliminate subtours. The mathematical model for the PCTSP is

$$minimize \sum_{e \in \mathcal{E}} c_e x_e \tag{A.8}$$

subject to (A.3 - A.7) plus

$$\sum_{v_i \in \mathcal{V}} p_i y_i \geq p_{min} \tag{A.9}$$

We refer to constraints A.9 as generalized covering constraints.

B Computational results

This appendix presents the detailed computational results of the proposed HGA compared with the reference algorithms. The results of HGA are based on 20 independent runs per instance. For the OP, we compare our results with the four best algorithms in the literature: RB&C [26], B&C [25], EA4OP [25], and ALNS [44]. For the PCTSP, only one exact algorithm, namely B&C [4], is presented for a small number of instances. For the purpose of comparison, we implemented an HGA variant (HGA-Giant), in which we replaced the proposed E^2AX with a giant tour crossover, which was inspired by the giant tour crossover designed for the multi-route team OP [6]. We used HGA-Giant as the main reference algorithm for the PCTSP and ran the algorithm under the stopping condition presented in Section 4.2.

B.1 Giant tour crossover

Crossover operators based on the giant tour have been used to solve various routing problems [54], which rely on efficient split algorithms that are designed for specific constraints, such as capacity or time windows. The giant tour can also be applied to TSPs with profits with respect to the corresponding constraints. In this section, we introduce the giant tour crossover and an optimal split algorithm.

We consider the PCTSP as an example. Given a solution φ, let $\mathcal{N}_\varphi$ and $\overline{\mathcal{N}}_\varphi$ be sets of routed and unrouted vertices in φ, respectively. Furthermore, let φ_A and φ_B be two parent solutions. First, all routed vertices ($v_i \in \mathcal{N}_{\varphi_A}$) in solution φ_A are arranged into an array A. Second, all unrouted vertices ($v_i \in \overline{\mathcal{N}}_{\varphi_A}$) are arranged into A after the routed vertices in a sequential order. An array B is produced using solution φ_B in the same manner. Third, given two giant tours A and B, an ordered crossover [37] is used on a simple permutation-based representation. Subsequently, a new giant tour S is produced. Finally, a linear time-split algorithm with respect to the collecting prize optimally splits each giant tour by inserting a trip delimiter. Specifically, for each vertex in S, if the delimiter is inserted after the vertex, there are two tours, and $\mathcal{O}(1)$ time is required to compute the profits and travel costs. As there are n vertices in S, we can optimally split S in $\mathcal{O}(n)$ time. Following the split, a feasible offspring is returned.

B.2 Results

In the tables presented hereafter, the column *Instance* indicates the names of the instances and the column BKS presents the best known values summarized from the literature. For the exact algorithms B&C [25] and RB&C [26], LB and UB are the lower and upper bounds from the corresponding algorithm, respectively. *Gap* was calculated as $Gap = 100 \times (LB - UB)/UB$. A star * indicates a proven optimal solution. Time represents the running (ending) time of the corresponding algorithm. For the heuristic algorithms ALNS [44], EA4OP [25], and our HGA, *Best* is the best result over multiple runs (10 for ALNS and EA4OP, and 20 for HGA). Time is the average running (ending) time of the algorithm. Furthermore, for our HGA and its variant HGA-Giant, TMB is the average running time required for HGA or HGA-Giant to attain its overall best results. TMB was calculated based on the runs (over 20 runs) that hit the overall best result. Furthermore, two indicators are defined to illustrate the performance of HGA.

- $\delta_1 = 100 \times (BKS - f_{best})/f_{best}$.
- $\delta_2 = 100 \times (f_{best} - BKS)/BKS$.

δ_1 is the difference between the proposed HGA algorithm and the best known results of OP (a maximization problem), where f_{best} is the best objective value of HGA and BKS is the best known result summarized from the reference algorithms. δ_2 is the gap between HGA and the best known results of PCTSP (a minimization problem), where f_{best} is the best objective value of HGA and *BKS* is the best result among the B&C and HGA-Giant algorithms. In the tables, the *Average* row represents the average value of the instances of a benchmark set. Improved best results (new bounds) are indicated by the negative δ values highlighted in boldface.

Table B.1
Results for OP on medium-sized instances of Set I.

Instances	BKS		RB&C [26]				EA4OP [25]		ALNS [44]		HGA				
	LB	UB	LB	UB	*Gap*(%)	Time	*Best*	Time	*Best*	Time	*Best*	*Avg.*	Time	TMB	δ_1(%)
att48	31	31	31	31	*	0.03	31	0.25	31	6.77	31	31.00	0.85	0.84	0.00
gr48	31	31	31	31	*	0.02	31	0.13	31	9.99	31	31.00	0.04	0.01	0.00
hk48	30	30	30	30	*	0.01	30	0.24	30	7.20	30	30.00	2.51	2.51	0.00
eil51	29	29	29	29	*	0.01	29	0.24	29	9.51	29	28.85	11.92	7.16	0.00
berlin52	37	37	37	37	*	0.02	37	0.30	37	9.42	37	37.00	0.04	0.01	0.00
brazil58	46	46	46	46	*	0.07	46	1.00	46	9.13	46	45.30	44.65	6.38	0.00
st70	43	43	43	43	*	0.05	43	0.32	43	15.99	43	43.00	0.66	0.66	0.00
eil76	47	47	47	47	*	0.04	46	0.32	47	20.51	47	46.05	59.01	1.96	0.00
pr76	49	49	49	49	*	0.06	49	0.61	49	18.64	49	48.05	63.37	0.94	0.00
gr96	64	64	64	64	*	0.08	64	1.44	64	20.31	64	64.00	15.44	15.44	0.00
rat99	52	52	52	52	*	0.47	52	0.66	52	27.75	52	51.80	33.95	20.86	0.00
kroA100	56	56	56	56	*	0.41	55	0.34	56	34.75	56	56.00	9.92	9.92	0.00
kroB100	58	58	58	58	*	0.27	57	0.63	58	43.06	58	56.45	68.74	25.52	0.00
kroC100	56	56	56	56	*	0.25	56	0.48	56	34.32	56	56.00	14.68	14.68	0.00
kroD100	59	59	59	59	*	0.09	58	0.65	59	34.61	59	59.00	5.82	5.82	0.00
kroE100	57	57	57	57	*	5.53	57	0.50	57	32.26	57	56.35	60.60	27.06	0.00
rd100	61	61	61	61	*	0.12	61	0.74	61	29.49	61	60.90	40.18	33.52	0.00
eil101	64	64	64	64	*	0.06	64	0.79	64	31.73	64	64.00	7.20	7.20	0.00
lin105	66	66	66	66	*	0.48	66	1.42	66	32.11	66	66.00	0.45	0.44	0.00
pr107	54	54	54	54	*	0.08	54	0.93	54	78.46	54	54.00	0.11	0.01	0.00
gr120	75	75	75	75	*	0.28	74	1.20	75	29.58	75	75.00	28.58	28.58	0.00
pr124	75	75	75	75	*	0.35	75	1.11	75	49.64	75	75.00	0.86	0.86	0.00
bier127	103	103	103	103	*	0.38	103	1.18	103	40.84	103	103.00	5.05	5.05	0.00
pr136	71	71	71	71	*	1.75	71	0.96	71	29.97	71	70.95	40.26	35.01	0.00
gr137	81	81	81	81	*	0.24	78	3.44	81	59.21	81	81.00	7.44	7.44	0.00
pr144	77	77	77	77	*	1.46	77	2.61	77	87.82	77	76.50	74.23	46.61	0.00
kroA150	86	86	86	86	*	33.87	86	1.17	86	82.79	86	85.05	113.12	33.65	0.00
kroB150	87	87	87	87	*	2.21	86	1.00	87	61.64	86	86.00	146.01	36.24	1.16
pr152	77	77	77	77	*	1.29	77	3.64	77	91.38	77	76.45	90.19	30.72	0.00
u159	93	93	93	93	*	1.82	92	1.11	93	99.63	93	92.15	122.50	37.65	0.00
rat195	102	102	102	102	*	3.71	99	1.78	102	195.57	101	100.45	139.73	56.95	0.99
d198	123	123	123	123	*	5.28	123	6.68	123	65.57	123	122.70	118.46	60.17	0.00
kroA200	117	117	117	117	*	2.5	117	1.74	117	114.75	116	114.05	227.36	83.39	0.86
kroB200	119	119	119	119	*	9.91	119	1.66	119	86.58	118	117.70	211.44	81.31	0.85
gr202	145	145	145	145	*	2.71	145	6.89	145	187.56	145	144.60	157.66	77.48	0.00
ts225	124	124	124	126	1.59	18000.00	124	1.28	124	279.52	124	124.00	0.22	0.06	0.00
tsp225	129	129	129	129	*	4.31	127	2.29	128	198.47	128	126.05	223.06	102.75	0.78
pr226	126	126	126	126	*	107.69	126	6.61	126	181.94	126	125.20	168.44	16.25	0.00
gr229	176	176	176	176	*	0.32	176	8.81	173	108.27	175	174.30	324.03	84.10	0.57
gil262	158	158	158	158	*	0.35	156	2.84	158	240.02	155	153.50	323.80	125.41	1.94
pr264	132	132	132	132	*	3.92	132	5.62	132	314.29	132	132.00	2.44	2.35	0.00
a280	147	147	147	147	*	40.65	143	3.00	144	239.06	145	142.95	272.42	134.60	1.38
pr299	162	162	162	162	*	48.85	160	3.12	162	410.90	160	159.60	280.80	87.62	1.25
lin318	205	205	205	205	*	5.49	202	7.15	203	294.23	205	203.55	403.82	153.07	0.00
rd400	239	239	239	239	*	36.71	234	6.59	237	422.56	236	233.50	623.83	294.78	1.27
Average	89.31	89.31	89.31	89.36	-	407.20	88.62	2.12	89.07	99.51	88.96	88.44	101.02	40.07	-

Table B.2. Results for OP on large-sized instances of Set I.

Instances	BKS		RB&C [26]				EA4OP [25]		ALNS [44]		HGA				
	LB	UB	LB	UB	*Gap*(%)	Time	*Best*	Time	*Best*	Time	*Best*	*Avg.*	Time	TMB	δ_1(%)
fl417	228	231	228	231	1.3	18000.00	224	11.84	228	1056.07	227	224.20	372.44	162.00	0.44
gr431	350	350	350	350	*	29.05	349	32.84	347	533.55	347	345.35	743.86	251.44	0.86
pr439	313	313	313	313	*	414.00	310	9.92	307	1263.74	312	309.85	617.75	281.02	0.32
pcb442	251	251	251	251	*	7.21	244	6.93	249	1328.72	249	248.40	544.55	161.21	0.80
d493	320	320	320	320	*	13.37	315	19.10	317	1291.93	315	310.10	852.24	312.30	1.59
att532	363	363	363	363	*	312.50	347	23.14	359	1380.54	357	356.10	1131.88	499.40	1.68
ali535	425	426	425	426	0.23	18000.00	424	73.03	422	1846.10	419	416.40	1609.74	624.85	1.43
pa561	357	357	357	357	*	245.42	348	23.18	346	1605.42	344	340.65	1141.05	526.90	3.78
u574	354	354	354	354	*	24.00	344	17.93	347	1204.18	350	346.60	1106.06	536.17	1.14
rat575	322	322	322	322	*	42.82	309	13.76	317	3109.65	310	307.95	948.33	425.64	3.87
p654	344	396	342	396	13.64	18000.00	336	28.89	343	10866.70	343	340.70	1074.49	475.42	0.29
d657	386	386	386	386	*	92.48	377	23.24	380	3152.17	381	375.30	1334.10	731.54	1.31
gr666	503	503	503	503	*	400.56	497	109.54	486	660.30	495	491.85	1746.74	977.59	1.62
u724	439	439	439	439	*	188.61	429	27.77	434	4157.30	429	426.00	1591.42	755.99	2.33
rat783	438	438	438	438	*	514.68	422	34.59	428	2962.52	420	416.80	1247.89	637.76	4.29
dsj1000	656	656	656	656	*	3828.50	632	81.20	630	17284.30	638	631.90	2962.76	1594.68	2.82
pr1002	606	606	606	606	*	4483.81	572	45.92	581	18000.00	587	581.20	2395.05	960.38	3.24
u1060	660	660	660	660	*	16716.01	627	90.04	644	18000.00	647	641.65	2440.49	1076.03	2.01
vm1084	777	777	777	777	*	5012.60	770	56.29	765	18000.00	772	770.00	3806.86	2003.13	0.65
pcb1173	675	675	675	675	*	6819.83	633	60.65	652	18000.00	650	643.10	3283.19	1529.82	3.85
d1291	715	715	715	715	*	7916.85	646	434.87	699	18000.00	698	682.75	2494.85	1392.79	2.44
rl1304	802	802	802	802	*	6269.39	766	102.45	788	18000.00	789	782.35	3464.28	2231.75	1.65
rl1323	814	814	814	814	*	7740.17	782	89.68	785	14585.10	806	795.75	4187.98	2125.59	0.99
nrw1379	815	817	815	817	0.24	18000.00	771	106.97	790	18000.00	779	772.30	4687.57	2410.17	4.62
fl1400	1048	1084	1003	1084	7.47	18000.00	1043	518.25	1048	18000.00	1041	1036.85	5391.92	2387.03	0.67
u1432	754	764	754	764	1.31	18000.00	738	121.46	749	14573.50	739	732.90	5068.57	2481.63	2.03
fl1577	897	900	897	900	0.33	18000.00	880	286.47	748	18000.00	865	858.50	4425.22	2523.23	3.70
d1655	922	924	922	924	0.22	18000.00	846	757.70	890	18000.00	887	880.95	4708.59	2906.87	3.95
vm1748	1276	1282	1276	1282	0.47	18000.00	1246	178.50	1252	16959.80	1262	1253.65	7284.97	3624.73	1.11
u1817	983	983	983	983	*	11226.88	879	975.58	947	18000.00	939	928.25	5700.99	3433.93	4.69
rl1889	1226	1226	1226	1226	*	17010.43	1167	269.81	1156	18000.00	1208	1195.95	7920.36	4909.81	1.49
d2103	1200	1200	1200	1200	*	15855.62	1069	951.27	1171	18000.00	1198	1194.60	7188.64	4012.64	0.17
u2152	1151	1151	1151	1151	*	14703.25	1048	1350.23	1111	18000.00	1095	1082.00	8348.35	6032.38	5.11
u2319	1170	1171	1170	1171	0.09	18000.00	1167	423.26	1170	6088.42	1170	1170.00	8933.50	3082.05	0.00
pr2392	1316	1415	1316	1415	7	18000.00	1292	402.29	1294	18000.00	1333	1323.25	8904.12	5243.50	**-1.28**
pcb3038	1727	1730	1727	1730	0.17	18000.00	1572	681.94	1626	18000.00	1623	1604.35	14596.03	9637.53	6.41
fl3795	1965	2249	1965	2249	12.63	18000.00	1815	2994.90	1818	18000.00	1904	1900.85	18000.25	10663.92	3.20
fnl4461	2541	2570	2541	2570	1.13	18000.00	2350	2462.65	2342	18000.00	2443	2410.40	18000.28	14495.34	4.01
rl5915	3593	3786	3593	3786	5.1	18000.00	3358	5361.54	3328	18000.00	3668	3626.80	18000.45	15975.74	**-2.04**
rl5934	3632	3752	3632	3752	3.2	18000.00	3145	5382.25	3276	18000.00	3602	3561.37	18000.66	16239.41	0.83
pla7397	5289	5657	5289	5657	6.51	18000.00	5141	15981.78	5140	18000.00	5294	5263.80	18000.02	14906.27	**-0.09**
Average	1039.10	1068.66	1037.95	1068.66	-	10387.03	981.22	990.82	992.93	11802.68	1022.80	1014.19	5470.48	3542.43	-

Table B.3. Results for OP on medium-sized instances of Set II.

Instances	BKS		RB&C [26]				EA4OP [25]		ALNS [44]		HGA				
	LB	UB	LB	UB	*Gap*(%)	Time	*Best*	Time	*Best*	Time	*Best*	*Avg.*	Time	TMB	δ_1(%)
att48	1717	1717	1717	1717	*	0.04	1717	0.32	1717	6.77	1717	1717.00	0.62	0.62	0.00
gr48	1761	1761	1761	1761	*	1.32	1749	0.2	1761	7.87	1761	1761.00	1.01	1.01	0.00
hk48	1614	1614	1614	1614	*	0.10	1614	0.16	1614	7.19	1614	1614.00	0.03	0.01	0.00
eil51	1674	1674	1674	1674	*	0.96	1668	0.18	1674	10.13	1674	1674.00	0.68	0.68	0.00
berlin52	1897	1897	1897	1897	*	3.23	1897	0.35	1897	10.74	1897	1897.00	0.11	0.10	0.00
brazil58	2220	2220	2220	2220	*	0.46	2218	1.52	2220	12.32	2220	2220.00	8.77	8.77	0.00
st70	2286	2286	2286	2286	*	1.77	2285	0.31	2286	21.65	2286	2286.00	8.18	8.18	0.00
eil76	2550	2550	2550	2550	*	0.62	2550	0.43	2550	16.06	2550	2550.00	9.47	9.47	0.00
pr76	2708	2708	2708	2708	*	1.46	2708	0.48	2708	19.48	2708	2708.00	3.15	3.14	0.00
gr96	3396	3396	3396	3396	*	9.50	3394	1.44	3394	31.98	3396	3396.00	24.04	24.04	0.00
rat99	2944	2944	2944	2944	*	3.25	2944	0.49	2944	32.08	2944	2944.00	1.82	1.81	0.00
kroA100	3212	3212	3212	3212	*	0.70	3212	0.57	3212	32.85	3212	3212.00	4.74	4.74	0.00
kroB100	3241	3241	3241	3241	*	13.28	3238	0.52	3239	48.39	3241	3218.25	567.89	197.48	0.00
kroC100	2947	2947	2947	2947	*	2.22	2931	0.60	2947	39.27	2947	2947.00	10.97	10.97	0.00
kroD100	3307	3307	3307	3307	*	3.62	3307	0.65	3307	30.52	3307	3307.00	3.59	3.59	0.00
kroE100	3090	3090	3090	3090	*	11.31	3082	0.50	3090	39.57	3090	3090.00	1.79	1.79	0.00
rd100	3359	3359	3359	3359	*	0.36	3359	0.50	3359	30.80	3359	3359.00	1.46	1.46	0.00
eil101	3655	3655	3655	3655	*	4.15	3655	0.82	3655	26.19	3655	3655.00	12.71	12.70	0.00
lin105	3544	3544	3544	3544	*	2.51	3530	1.10	3544	36.22	3544	3544.00	15.90	15.90	0.00
pr107	2667	2667	2667	2667	*	0.20	2667	1.05	2667	69.67	2667	2667.00	0.07	0.01	0.00
gr120	4371	4371	4371	4371	*	6.57	4356	1.37	4371	40.41	4371	4371.00	7.86	7.86	0.00
pr124	3917	3917	3917	3917	*	1.07	3899	1.34	3917	55.25	3917	3917.00	3.80	3.80	0.00
bier127	5383	5383	5383	5383	*	0.96	5381	1.71	5366	23.01	5383	5383.00	23.60	23.60	0.00
pr136	4309	4309	4309	4309	*	1.25	4309	1.15	4309	35.63	4309	4309.00	5.39	5.38	0.00
gr137	4286	4286	4286	4286	*	10.65	4099	3.09	4286	639.80	4286	4286.00	2.33	2.33	0.00
pr144	4003	4003	4003	4003	*	32.23	3965	3.02	3969	100.20	4003	4003.00	7.49	7.49	0.00
kroA150	4918	4918	4918	4918	*	60.43	4902	1.26	4918	80.06	4918	4918.00	135.04	135.04	0.00
kroB150	4869	4869	4869	4869	*	16.94	4869	1.19	4869	61.96	4869	4869.00	82.07	82.05	0.00
pr152	4279	4279	4279	4279	*	1.85	4245	3.47	4279	67.41	4279	4279.00	165.93	165.93	0.00
u159	4960	4960	4960	4960	*	14.96	4941	1.44	4950	109.59	4960	4960.00	43.12	43.11	0.00
rat195	5791	5791	5791	5791	*	46.09	5703	1.55	5782	263.23	5791	5791.00	158.24	158.22	0.00
d198	6670	6670	6670	6670	*	298.24	6660	7.33	6661	88.47	6670	6669.35	798.66	273.99	0.00
kroA200	6547	6547	6547	6547	*	16.18	6534	1.71	6547	116.11	6547	6547.00	38.12	38.09	0.00
kroB200	6419	6419	6419	6419	*	20.62	6278	1.97	6413	189.98	6419	6387.70	721.43	396.77	0.00
gr202	7789	7789	7789	7789	*	139.90	7789	8.77	7719	188.27	7789	7789.00	92.11	92.11	0.00
ts225	6834	6834	6834	6834	*	95.22	6819	1.47	6782	394.00	6834	6812.30	619.32	286.64	0.00
tsp225	6987	6987	6987	6987	*	54.09	6936	1.87	6980	299.73	6987	6985.60	514.68	370.74	0.00
pr226	6662	6662	6662	6662	*	2894.81	6658	7.29	6662	201.68	6662	6662.00	3.39	3.39	0.00
gr229	9177	9177	9177	9177	*	16.67	9174	13.19	9177	1379.35	9177	9176.95	317.14	272.63	0.00
gil262	8321	8321	8321	8321	*	64.63	8175	3.47	8269	487.41	8321	8316.25	836.53	383.89	0.00
pr264	6654	6654	6654	6654	*	13.33	6173	5.94	6654	314.27	6654	6654.00	148.53	148.42	0.00
a280	8428	8428	8428	8428	*	519.95	8304	2.85	8404	215.31	8427	8405.70	1113.10	486.86	0.01
pr299	9182	9182	9182	9182	*	623.34	9112	3.23	9147	393.12	9182	9180.50	841.19	356.78	0.00
lin318	10923	10923	10923	10923	*	367.53	10866	8.29	10801	370.64	10923	10921.90	935.97	376.82	0.00
rd400	13652	13652	13652	13652	*	769.66	13442	6.80	13562	1174.91	13650	13648.90	1430.88	652.59	0.01
Average	4869.33	4869.33	4869.33	4869.33	-	136.63	4829.20	2.38	4857.31	173.77	4869.27	4866.88	216.06	112.91	-

Table B.4. Results for OP on large-sized instances of Set II.

Instances	BKS		RB&C [26]				EA4OP [25]		ALNS [44]		HGA				
	LB	UB	LB	UB	*Gap*(%)	Time	*Best*	Time	*Best*	Time	*Best*	*Avg.*	Time	TMB	δ_1(%)
fl417	11933	12294	11933	12387	3.67	18000.00	11787	16.73	11923	2144.94	11938	11934.10	1279.24	582.48	**-0.04**
gr431	18318	18318	18318	18318	*	2809.41	18287	51.38	18318	2740.82	18315	18313.20	1900.40	935.74	0.02
pr439	16171	16171	16171	16171	*	3765.86	16085	11.77	16128	629.44	16171	16170.05	1283.17	574.48	0.00
pcb442	14484	14484	14484	14484	*	13760.94	14273	6.83	14411	4410.74	14465	14452.45	1677.34	700.55	0.13
d493	16995	17007	16995	17007	0.07	18000.00	16729	17.15	16820	6231.42	16991	16972.20	1956.30	1016.13	0.02
att532	19635	19800	19635	19800	0.83	18000.00	19265	23.43	19465	1564.89	19658	19653.25	2193.88	765.79	**-0.12**
ali535	21954	21954	21954	21973	0.09	18000.00	21910	95.05	21761	1537.87	21953	21945.15	2257.58	1432.75	0.00
pa561	19576	19576	19576	19576	*	1961.95	18894	23.45	19092	790.31	19551	19497.32	2127.67	1438.76	0.13
u574	19351	19351	19351	19351	*	1026.82	18966	16.33	19028	5389.10	19351	19330.90	2063.79	1188.99	0.00
rat575	18251	18251	18251	18251	*	9616.70	17705	14.97	17984	2089.02	18235	18215.25	1986.76	1063.03	0.09
p654	17900	21566	17753	22248	20.2	18000.00	17821	42.82	17900	18000.00	17917	17876.60	1976.04	642.42	**-0.09**
d657	21503	21503	21503	21503	*	554.67	21162	22.90	21231	4161.44	21499	21490.30	2421.26	1280.89	0.02
gr666	26514	26569	26514	26569	0.21	18000.00	26336	136.48	25971	1024.22	26486	26455.60	2714.55	1322.13	0.11
u724	24223	24223	24223	24223	*	9829.42	23793	28.71	23878	5755.06	24198	24143.95	2832.77	1629.65	0.10
rat783	25474	25474	25474	25474	*	12246.90	24861	32.36	24987	6622.62	25353	25214.85	2873.49	1785.02	0.48
dsj1000	35835	35915	35835	35915	0.22	18000.00	34463	83.34	34641	18000.00	35524	35436.00	4857.53	3587.65	0.88
pr1002	33030	33092	33030	33092	0.19	18000.00	31746	46.19	32120	18000.00	33005	32949.80	4285.52	2802.95	0.08
u1060	36151	36291	36151	36291	0.39	18000.00	35110	77.78	35284	18000.00	36146	36027.30	4311.82	3023.31	0.01
vm1084	40777	40952	40777	40952	0.43	18000.00	40308	55.67	40240	18000.00	40774	40750.45	4942.20	2588.10	0.01
pcb1173	37035	37100	37035	37100	0.18	18000.00	35826	69.94	35946	18000.00	36874	36741.90	4912.86	2811.80	0.44
d1291	37778	37854	37778	37854	0.2	18000.00	35153	289.25	36815	18000.00	37564	37421.15	4538.07	2613.11	0.57
rl1304	42275	42359	42275	42359	0.2	18000.00	40561	97.68	40893	12853.40	42266	41963.65	5386.89	3415.69	0.02
rl1323	43377	43450	43377	43450	0.17	18000.00	41459	89.78	41210	18000.00	43375	43160.90	5669.84	3608.18	0.00
nrw1379	46676	46787	46676	46787	0.24	18000.00	45602	117.51	45576	18000.00	46529	46398.05	6638.98	4604.15	0.32
fl1400	56692	64298	54124	64298	15.82	18000.00	56258	794.15	56692	18000.00	56883	56832.60	7843.46	6002.60	**-0.34**
u1432	46946	47018	46946	47018	0.15	18000.00	44810	100.91	44982	18000.00	46617	46281.40	6328.82	4648.08	0.71
fl1577	45505	50154	45326	50154	9.63	18000.00	45505	334.28	41148	18000.00	47295	46971.35	6779.32	5151.11	**-3.78**
d1655	49319	53083	46158	53083	13.05	18000.00	47211	683.17	49319	18000.00	50239	50038.55	7663.78	5819.46	**-1.83**
vm1748	68042	68303	68042	68303	0.38	18000.00	66685	195.85	66636	18000.00	68090	67938.90	11054.04	7804.15	**-0.07**
u1817	54245	54554	54245	54554	0.57	18000.00	50366	734.39	51676	18000.00	53506	53280.05	8443.16	7270.04	1.38
rl1889	63308	64425	63308	64425	1.73	18000.00	60084	286.07	60928	18000.00	64036	63690.90	9906.39	7719.85	**-1.14**
d2103	63426	63426	63426	63426	*	16593.51	57202	682.28	61636	18000.00	62977	62682.10	9309.68	7721.66	0.71
u2152	64649	64775	64649	64775	0.19	18000.00	60211	1164.38	61052	18000.00	63718	63324.30	10367.26	8976.34	1.46
u2319	80914	81139	80914	81139	0.28	18000.00	78102	447.06	77610	18000.00	80521	80272.70	13356.77	9988.59	0.49
pr2392	72843	78237	72843	78237	6.89	18000.00	71018	440.57	71851	18000.00	75272	74926.15	14470.13	12654.40	**-3.23**
pcb3038	97902	97995	97902	97995	0.09	18000.00	91842	820.37	91457	18000.00	95980	95276.50	18000.55	16152.89	2.00
fl3795	103397	142895	98998	142895	30.72	18000.00	103397	4788.96	102642	18000.00	110988	110604.45	18000.35	15572.33	**-6.84**
fnl4461	147109	150189	147109	150189	2.05	18000.00	140424	2618.15	135515	18000.00	145968	145165.15	18000.58	17089.11	0.78
rl5915	184424	197729	184424	197729	6.73	18000.00	176678	5512.40	173500	18000.00	193626	192284.80	18000.59	17431.83	**-4.75**
rl5934	187034	196805	187034	196805	4.96	18000.00	171649	5757.80	166368	18000.00	192968	191017.65	18000.56	17164.14	**-3.08**
pla7397	281977	297246	281977	297246	5.14	18000.00	272452	18000	266038	18000.00	287214	286523.55	18000.91	17905.70	**-1.82**
Average	56413.37	59088.10	56158.39	59107.46	-	15369.91	54195.02	1093.37	53918.83	12827.93	57074.05	56820.13	7088.20	5621.61	-

Table B.5. Results for OP on medium-sized instances of Set III.

Instances	BKS		RB&C [26]				EA4OP [25]		ALNS [44]		HGA				
	LB	UB	LB	UB	*Gap*(%)	Time	*Best*	Time	*Best*	Time	*Best*	*Avg.*	Time	TMB	δ_1(%)
att48	1049	1049	1049	1049	*	1.17	1049	0.26	1049	7.18	1049	1049.00	0.21	0.21	0.00
gr48	1480	1480	1480	1480	*	0.72	1480	0.13	1480	8.87	1480	1480.00	1.32	1.32	0.00
hk48	1764	1764	1764	1764	*	0.06	1764	0.22	1764	8.51	1764	1764.00	0.06	0.06	0.00
eil51	1399	1399	1399	1399	*	1.46	1398	0.22	1399	6.87	1399	1399.00	1.04	1.04	0.00
berlin52	1036	1036	1036	1036	*	4.61	1034	0.64	1036	12.84	1036	1036.00	43.83	43.83	0.00
brazil58	1702	1702	1702	1702	*	0.02	1702	0.71	1702	11.09	1702	1702.00	1.32	1.32	0.00
st70	2108	2108	2108	2108	*	0.49	2108	0.31	2108	9.65	2108	2108.00	36.52	36.51	0.00
eil76	2467	2467	2467	2467	*	2.96	2467	0.36	2467	20.48	2467	2467.00	3.09	3.09	0.00
pr76	2430	2430	2430	2430	*	1.07	2430	0.57	2430	20.43	2430	2430.00	0.82	0.82	0.00
gr96	3170	3170	3170	3170	*	5.66	3166	1.41	3166	15.22	3170	3170.00	1.16	1.16	0.00
rat99	2908	2908	2908	2908	*	3.01	2886	0.78	2908	42.03	2908	2907.80	202.43	123.29	0.00
kroA100	3211	3211	3211	3211	*	1.81	3180	0.38	3211	32.31	3155	3155.00	586.14	6.30	1.77
kroB100	2804	2804	2804	2804	*	0.35	2785	0.51	2804	35.83	2804	2804.00	2.13	2.13	0.00
kroC100	3155	3155	3155	3155	*	1.82	3155	0.44	3155	34.67	3155	3155.00	1.78	1.76	0.00
kroD100	3167	3167	3167	3167	*	0.70	3141	0.58	3167	31.08	3167	3167.00	2.88	2.88	0.00
kroE100	3049	3049	3049	3049	*	1.36	3049	0.47	3049	31.96	3049	3049.00	26.43	26.42	0.00
rd100	2926	2926	2926	2926	*	23.20	2923	0.48	2926	16.35	2926	2926.00	3.27	3.27	0.00
eil101	3345	3345	3345	3345	*	1.37	3345	0.56	3345	28.61	3345	3345.00	25.05	25.04	0.00
lin105	2986	2986	2986	2986	*	16.02	2973	2.09	2986	38.24	2986	2986.00	2.32	2.31	0.00
pr107	1877	1877	1877	1877	*	3297.37	1802	0.82	1877	65.16	1877	1874.45	61.01	26.07	0.00
gr120	3779	3779	3779	3779	*	2.65	3748	1.36	3777	37.94	3779	3779.00	36.22	36.22	0.00
pr124	3557	3557	3557	3557	*	4507.38	3455	0.88	3557	99.87	3557	3557.00	5.67	5.66	0.00
bier127	2365	2365	2365	2365	*	40.07	2361	2.62	2361	49.90	2365	2365.00	150.74	150.73	0.00
pr136	4390	4390	4390	4390	*	30.50	4390	1.13	4390	61.84	4390	4390.00	14.83	14.82	0.00
gr137	3954	3954	3954	3954	*	14.01	3954	1.88	3954	637.09	3954	3954.00	21.85	21.84	0.00
pr144	3745	3745	3745	3745	*	116.68	3700	2.41	3744	112.92	3745	3745.00	60.76	60.76	0.00
kroA150	5039	5039	5039	5039	*	46.43	5019	1.07	5037	104.23	5003	5003.00	782.62	24.47	0.72
kroB150	5314	5314	5314	5314	*	28.53	5314	1.04	5314	63.05	5314	5314.00	57.85	57.85	0.00
pr152	3905	3905	3905	3905	*	83.51	3902	3.62	3539	184.38	3905	3904.75	374.94	212.53	0.00
u159	5272	5272	5272	5272	*	8.59	5272	0.94	5272	94.27	5272	5272.00	4.17	4.17	0.00
rat195	6195	6195	6195	6195	*	33.56	6139	2.00	6188	188.56	6195	6195.00	198.01	198.01	0.00
d198	6320	6320	6320	6320	*	461.18	6290	7.14	6320	105.70	6320	6320.00	94.10	94.09	0.00
kroA200	6123	6123	6123	6123	*	92.41	6114	1.72	6118	232.20	6123	6123.00	90.58	90.58	0.00
kroB200	6266	6266	6266	6266	*	3.87	6213	1.77	6266	188.77	6266	6224.40	424.28	356.42	0.00
gr202	8616	8616	8616	8616	*	315.26	8605	10.45	8564	57.88	8616	8615.30	413.75	323.12	0.00
ts225	7575	7575	7575	7575	*	6.62	7575	1.14	7575	450.25	7575	7575.00	20.38	20.38	0.00
tsp225	7740	7740	7740	7740	*	38.61	7488	2.58	7514	188.53	7740	7713.50	925.06	496.93	0.00
pr226	6993	6993	6993	6993	*	1170.00	6908	8.01	6993	177.59	6993	6986.00	571.88	168.34	0.00
gr229	6328	6328	6328	6328	*	42.63	6297	11.65	6328	1298.80	6328	6301.90	837.83	277.42	0.00
gil262	9246	9246	9246	9246	*	83.29	9094	3.94	9210	649.54	9246	9245.70	787.73	500.85	0.00
pr264	8137	8137	8137	8137	*	186.59	8068	3.62	8137	357.80	8137	8095.95	780.66	229.32	0.00
a280	9774	9774	9774	9774	*	126.80	8684	3.22	8789	378.80	9774	9774.00	74.80	74.80	0.00
pr299	10343	10343	10343	10343	*	913.13	9959	3.95	10233	549.11	10343	10343.00	216.38	216.38	0.00
lin318	10368	10368	10368	10368	*	327.58	10273	6.33	10337	528.20	10368	10368.00	124.69	124.69	0.00
rd400	13223	13223	13223	13223	*	214.40	13088	7.74	13122	727.58	13223	13212.85	1431.43	720.01	0.00
Average	4724.44	4724.44	4724.44	4724.44	-	272.43	4661.04	2.31	4681.51	177.83	4722.40	4718.92	211.20	106.43	-

Table B.6. Results for OP on large-sized instances of Set III.

Instances	BKS		RB&C [26]				EA4OP [25]		ALNS [44]		HGA				
	LB	UB	LB	UB	*Gap*(%)	Time	*Best*	Time	*Best*	Time	*Best*	*Avg.*	Time	TMB	δ_1(%)
fl417	14220	14220	14219	14387	1.17	18000.00	14186	12.45	14220	1131.05	14214	14201.95	952.43	401.78	0.04
gr431	10911	10911	10911	10911	*	7814.17	10817	54.5	10907	2411.45	10911	10895.40	1586.08	863.30	0.00
pr439	15176	15296	15176	15331	1.01	18000.00	15097	10.96	15080	1328.74	15162	15161.35	1669.39	613.07	0.09
pcb442	14819	14819	14819	14819	*	11574.76	14522	6.58	14695	1192.19	14817	14800.60	1623.75	739.64	0.01
d493	25167	25188	25167	25195	0.11	18000.00	24981	19.18	24849	3829.32	25150	24814.15	1912.07	1221.13	0.07
att532	15498	15498	15498	15498	*	318.44	15342	22.75	15335	4533.36	15422	15402.35	2081.94	1337.68	0.49
ali535	9414	9472	9414	9472	0.61	18000.00	9328	94.09	9308	13313.50	9406	9402.40	2068.92	1061.64	0.09
pa561	14482	14482	14482	14482	*	2539.41	14034	21.35	14162	3020.53	14447	14419.70	2042.58	1212.60	0.24
u574	20064	20064	20064	20064	*	2693.59	19691	19.77	19841	1671.01	20061	20025.05	1873.35	1058.58	0.01
rat575	20109	20109	20109	20109	*	929.99	19879	18.03	19954	7175.13	20107	20068.95	1915.85	1179.53	0.01
p654	24492	24518	24492	24518	0.11	18000.00	24130	18.54	24427	7543.02	24492	24491.15	550.26	449.63	0.00
d657	24562	24562	24562	24562	*	8777.39	23772	21.89	23829	4600.87	24562	24554.30	2272.47	1491.07	0.00
gr666	17023	17048	17023	17060	0.22	18000.00	16902	143.87	16709	2734.75	17011	16994.40	2817.11	1577.29	0.07
u724	28348	28348	28348	28348	*	10332.54	27932	29.26	28033	12058.60	28339	28316.65	2634.43	1488.15	0.03
rat783	27566	27566	27566	27566	*	3812.98	26797	30.64	27306	16331.50	27442	27297.55	2552.72	1842.65	0.45
dsj1000	31434	31454	31434	31454	0.06	18000.00	30943	79.18	31040	15962.00	31423	31340.65	3780.24	2725.06	0.04
pr1002	39526	39526	39526	39526	*	13955.69	38762	47.30	38502	18000.00	39519	39458.30	4103.94	2631.37	0.02
u1060	37492	37569	37492	37569	0.2	18000.00	36570	75.88	36598	18000.00	37496	37392.40	3732.20	2808.55	**-0.01**
vm1084	37669	37669	37669	37669	*	8710.50	37508	54.21	37178	3286.89	37665	37597.50	4614.19	2735.14	0.01
pcb1173	41257	41257	41257	41257	*	15133.74	40069	66.16	40513	18000.00	40865	40731.50	4694.10	2957.94	0.96
d1291	41509	42153	41509	42153	1.53	18000.00	38132	299.87	39919	18000.00	41784	41667.40	4604.59	2792.38	**-0.66**
rl1304	41881	42075	41881	42075	0.46	18000.00	41214	81.11	41679	18000.00	41893	41848.70	5052.61	2756.05	**-0.03**
rl1323	47213	47384	47213	47384	0.36	18000.00	46641	93.53	45500	8544.44	47173	47057.05	5421.88	3513.72	0.08
nrw1379	42920	42975	42920	42975	0.13	18000.00	-	-	-	-	42838	42763.35	4714.91	2822.60	0.19
fl1400	57470	59491	54661	59491	8.12	18000.00	57226	599.81	57470	18000.00	57548	57360.85	7359.73	5660.92	**-0.14**
u1432	47778	47895	47778	47895	0.24	18000.00	46657	138.02	47242	18000.00	47742	47660.40	5448.09	3420.03	0.08
fl1577	45935	48809	45768	48809	6.23	18000.00	45692	295.62	45935	18000.00	46205	46159.65	5666.76	3601.55	**-0.58**
d1655	62048	62945	62048	62945	1.43	18000.00	58728	674.25	60956	18000.00	62598	62491.45	6934.37	4031.41	**-0.88**
vm1748	71885	72010	71885	72010	0.17	18000.00	70958	225.29	71244	18000.00	71911	71865.85	8209.77	4583.09	**-0.04**
u1817	63639	67670	63618	67670	5.99	18000.00	63639	1302.35	63016	18000.00	65061	64851.25	7319.32	5187.11	**-2.19**
rl1889	70065	71106	70065	71106	1.46	18000.00	68422	244.97	68096	18000.00	70704	70532.50	9670.58	6948.90	**-0.90**
d2103	82787	82973	82787	82973	0.22	18000.00	77333	1168.90	81081	18000.00	82789	82710.95	9355.54	5977.87	**0.00**
u2152	74007	78066	74007	78066	5.2	18000.00	73400	1619.61	72733	18000.00	75117	74822.30	8681.40	6364.49	**-1.48**
u2319	79351	81050	79343	81050	2.11	18000.00	78113	569.76	79130	18000.00	79611	79551.90	11286.34	9192.20	**-0.33**
pr2392	85409	90261	85409	90261	5.38	18000.00	84094	422.73	85084	18000.00	87200	86787.50	12106.64	9306.23	**-2.05**
pcb3038	106928	112006	106928	112006	4.53	18000.00	104667	917.39	105337	18000.00	108475	107827.45	15247.40	13511.11	**-1.43**
fl3795	97707	116792	89218	116792	23.61	18000.00	97707	3158.89	95580	18000.00	100319	99971.05	18000.24	16045.69	**-2.60**
fnl4461	146995	152562	146995	152562	3.65	18000.00	-	-	-	-	147641	146824.75	18000.37	17191.81	**-0.44**
rl5915	203695	217366	203695	217366	6.29	18000.00	199336	5593.23	201814	18000.00	211017	210311.10	18000.31	17191.88	**-3.47**
rl5934	212021	229405	212021	229405	7.58	18000.00	207385	5881.87	203667	18000.00	221855	219778.65	18000.61	17148.22	**-4.43**
pla7397	322285	334885	322285	334885	3.76	18000.00	320744	18000	312645	18000.00	325632	323903.85	18000.84	17367.57	**-1.03**
Average	60311.15	62669.63	60030.78	62675.02	-	14843.74	57470.51	1080.35	57451.64	12529.96	61064.00	60832.05	6501.59	5000.26	-

Table B.7
Results for OP on medium-sized instances of Set IV.

Instances	BKS	B&C [25]			EA4OP [25]		ALNS [44]		HGA				
		LB	*Gap*(%)	Time	*Best*	Time	*Best*	Time	*Best*	*Avg.*	Time	TMB	δ_1(%)
att48	1870	1870	0.00	106.00	1870	0.52	1870	8.99	1870	1870.00	185.07	1.58	0.00
gr48	2264	2264	0.00	22.40	2264	0.40	2264	3.82	2264	2264.00	279.14	1.82	0.00
hk48	2177	2177	0.00	0.20	2177	0.15	2177	7.76	2177	2177.00	271.23	1.46	0.00
eil51	2490	2490	0.00	82.10	2490	0.24	2489	6.65	2490	2490.00	327.17	3.69	0.00
berlin52	2089	2089	0.00	115.00	2085	0.48	2089	10.75	2089	2089.00	253.33	9.03	0.00
brazil58	2070	2070	0.00	132.00	2060	1.08	2070	10.71	2070	2070.00	308.02	1.27	0.00
st70	3316	3316	0.00	127.70	3314	0.42	3316	7.82	3316	3315.50	450.39	118.57	0.00
eil76	3646	3646	0.00	45.10	3646	0.52	3640	9.38	3646	3646.00	471.84	31.66	0.00
pr76	3361	3361	0.00	1047.70	3361	0.62	3358	10.78	3361	3361.00	438.33	18.70	0.00
gr96	4851	4851	0.00	212.30	4851	0.37	4851	6.68	4851	4851.00	505.50	3.54	0.00
rat99	3502	3502	0.00	16.00	3502	0.60	3502	31.01	3502	3501.60	551.80	160.40	0.00
kroA100	4999	4999	0.00	187.10	4999	0.36	4999	6.44	4999	4999.00	590.22	66.04	0.00
kroB100	2935	2935	0.00	34.40	2935	0.61	2935	31.84	2935	2904.90	554.57	61.39	0.00
kroC100	1962	1962	0.00	261.60	1955	0.46	1962	31.46	1962	1962.00	336.67	2.33	0.00
kroD100	1212	1212	0.00	11.80	1212	0.41	1212	14.33	1212	1212.00	149.85	0.02	0.00
kroE100	4635	4635	0.00	203.40	4616	0.69	4631	13.14	4635	4635.00	681.46	22.99	0.00
rd100	3815	3815	0.00	164.60	3808	0.75	3815	22.99	3815	3815.00	647.49	21.46	0.00
eil101	4308	4308	0.00	90.80	4306	0.83	4308	39.55	4308	4308.00	609.52	4.68	0.00
lin105	2455	2455	0.00	1020.60	2453	0.81	2455	33.74	2455	2455.00	416.77	3.38	0.00
pr107	2072	2072	0.00	159.00	2072	1.95	2072	10.20	2072	2072.00	227.42	0.58	0.00
gr120	5830	5830	0.00	236.70	5830	1.25	5830	18.10	5830	5830.00	677.32	37.24	0.00
pr124	2036	2036	0.00	163.80	1937	1.18	2036	48.00	2036	2036.00	329.39	0.47	0.00
bier127	5068	5068	0.00	278.40	5067	2.28	5053	42.94	5068	5068.00	614.18	81.69	0.00
pr136	2860	2860	0.00	6303.60	2820	0.74	2860	51.86	2860	2858.80	542.13	210.35	0.00
gr137	6523	6523	0.00	203.10	6516	2.52	6523	35.45	6523	6523.00	779.57	15.72	0.00
pr144	5641	5641	0.00	357.90	5639	4.53	5641	70.02	5641	5639.30	855.54	228.20	0.00
kroA150	6858	6858	0.00	415.90	6855	1.69	6855	42.88	6858	6858.00	816.73	11.97	0.00
kroB150	7023	7023	0.00	303.00	7020	1.16	7014	23.86	7023	7023.00	890.59	4.47	0.00
pr152	5823	5823	0.00	483.60	5820	5.21	5823	43.79	5261	5261.00	823.84	20.03	10.68
u159	3147	3147	0.00	1145.20	3147	0.92	3147	161.92	3147	3147.00	499.78	5.35	0.00
rat195	9753	9753	0.00	205.40	9750	1.69	9737	27.11	9753	9752.75	928.42	233.11	0.00
d198	4661	4661	0.00	492.70	4654	4.95	4658	122.01	4661	4661.00	786.39	151.19	0.00
kroA200	9892	9892	0.00	340.30	9892	2.73	9854	47.90	9892	9889.85	1034.46	363.36	0.00
kroB200	9849	9849	0.00	253.20	9842	1.62	9846	20.18	9849	9849.00	1084.16	153.14	0.00
gr202	1071	1071	0.00	376.10	995	1.47	1055	30.88	1071	1071.00	116.43	0.31	0.00
ts225	11002	11002	0.00	3524.60	11002	1.87	10954	61.48	11002	11002.00	1177.47	109.28	0.00
tsp225	10972	10972	0.00	706.70	10972	2.52	10920	76.87	10973[1]	10969.60	1128.39	516.11	**-0.01**
pr226	4893	4893	0.00	1183.10	4890	4.83	4893	313.81	4893	4893.00	602.26	35.66	0.00
gr229	11482	11482	0.00	563.10	11482	6.46	11397	29.97	11482	11475.85	1019.99	444.54	0.00
gil262	2031	2031	0.00	1770.50	2030	1.35	2031	93.73	2031	2031.00	469.39	2.76	0.00
pr264	10253	10253	0.00	277.50	10166	6.42	10179	180.21	10253	10158.10	1201.48	206.18	0.00
a280	12064	12064	0.00	351.80	12048	3.39	11955	217.26	12064	12049.80	1333.67	491.79	0.00
pr299	14986	14986	0.00	7771.90	14980	3.46	14959	48.86	14986	14982.05	1284.58	471.44	0.00
lin318	15132	15132	0.00	-	15119	7.91	14960	106.15	15146[1]	15144.10	1614.19	600.93	**-0.09**
rd400	20107	20107	0.00	5093.10	20101	9.61	20060	103.75	20102	20097.25	1829.40	812.23	0.02
Average	5755.24	5755.24	-	837.30	5745.56	2.09	5739.00	51.93	5742.98	5739.30	682.12	127.60	-

[1] One notices that HGA finds better feasible solutions than the optimal solutions reported by B&C [25].

Table B.8
Results for OP on large-sized instances of Set IV.

Instances	BKS	B&C [25]			EA4OP [25]		ALNS [44]		HGA				
		LB	*Gap*(%)	Time	*Best*	Time	*Best*	Time	*Best*	*Avg.*	Time	TMB	δ_1(%)
fl417	20496	20496	0.00	18000.00	20494	39.61	20496	165.27	20496	20493.30	1738.74	620.28	0.00
gr431	13976	13976	0.00	18000.00	13969	50.29	13807	794.43	13979[1]	13978.60	1475.64	471.77	**-0.02**
pr439	19613	19613	0.00	3936.10	19510	13.61	19453	765.03	19613	19599.80	1871.00	1175.60	0.00
pcb442	5869	5839	0.51	18000.00	5650	3.40	5869	1290.30	5888	5888.00	904.60	51.57	**-0.32**
d493	21740	21740	0.00	18000.00	21674	21.00	21578	785.63	21744[1]	21688.15	2240.91	1428.08	**-0.02**
att532	26728	26728	0.00	18000.00	26728	17.20	26684	461.68	26721	26714.95	2175.10	968.04	0.03
ali535	13520	13520	0.00	15739.60	13442	73.07	13350	2346.62	13520	13396.35	1601.36	933.42	0.00
pa561	27719	27712	0.03	18000.00	27719	24.14	27445	570.88	27729	27712.50	2845.50	1612.59	**-0.04**
u574	28823	28823	0.00	18000.00	28822	26.03	28815	76.97	28827[1]	28818.10	2337.93	1089.82	**-0.01**
rat575	28364	28364	0.00	18000.00	28334	24.68	28237	436.44	28357	28330.10	2764.61	1590.76	0.02
p654	31814	31814	0.00	18000.00	31717	123.82	31724	267.05	31798	31748.15	2747.90	1396.04	0.05
d657	32548	32548	0.00	13485.10	32534	33.00	32378	304.43	32546	32523.45	3151.48	1861.76	0.01
gr666	21013	21013	0.00	18000.00	20901	132.65	20762	18000.00	21077[1]	21069.40	2317.14	1205.05	**-0.30**
u724	34988	34988	0.00	18000.00	34921	40.93	34554	629.02	34987	34952.45	3959.01	2095.75	0.00
rat783	7829	7829	0.00	18000.00	7548	13.35	7713	8573.52	7832[1]	7772.45	1208.69	572.72	**-0.04**
dsj1000	27357	27357	0.00	18000.00	25352	48.13	26573	18000.00	27431[1]	27407.05	3693.85	2017.84	**-0.27**
pr1002	23527	23527	0.00	18000.00	22482	35.67	22832	11291.20	23590[1]	23463.60	2966.59	1898.84	**-0.27**
u1060	51775	51768	0.01	18000.00	51775	150.58	51593	2079.50	51849	51795.15	6074.41	3741.43	**-0.14**
vm1084	38678	38678	0.00	18000.00	38228	50.34	37970	7560.86	38700[1]	38695.65	4929.41	2658.18	**-0.06**
pcb1173	56010	55954	0.10	18000.00	56010	77.73	55618	6709.24	56018	55926.85	7185.71	5406.83	**-0.01**
d1291	4029	4029	0.00	2335.60	4024	45.07	4029	1707.97	4029	4029.00	1043.51	7.06	0.00
rl1304	57782	57782	0.00	18000.00	57545	112.18	57576	8261.13	58220[1]	58137.95	7787.85	5404.39	**-0.75**
rl1323	65664	65476	0.29	18000.00	65664	99.81	65166	905.12	65667	65617.10	7640.51	4957.15	0.00
nrw1379	69214	69119	0.14	18000.00	69214	152.00	69150	2234.08	69184	69156.80	7689.43	3984.67	0.04
fl1400	70511	70476	0.05	18000.00	70488	287.75	70511	3310.76	70530	70528.45	7588.49	2439.46	**-0.03**
u1432	54540	54540	0.00	18000.00	53550	127.79	52742	14148.70	54490	54218.35	8482.44	6081.59	0.09
fl1577	33754	22191	34.26	18000.00	33754	200.71	31118	18000.00	34613	33808.55	4872.68	2631.43	**-2.48**
d1655	33231	29920	9.96	18000.00	31880	371.31	33231	18000.00	34203	33968.00	4756.13	3447.54	**-2.84**
vm1748	82126	81778	0.42	18000.00	82126	265.55	81786	18000.00	82461	82399.45	12942.79	8541.65	**-0.41**
u1817	37457	31800	15.10	18000.00	36416	418.80	37457	18000.00	38576	38106.95	5505.37	4524.52	**-2.90**
rl1889	83875	71527	14.72	18000.00	83081	363.35	83875	15860.70	84827	84708.10	14813.47	10162.37	**-1.12**
d2103	37124	31045	16.37	18000.00	34192	465.36	37124	18000.00	37825	37399.55	5371.52	3715.71	**-1.85**
u2152	55397	48472	12.50	18000.00	54744	906.84	55397	18000.00	57972	57435.30	9685.72	8004.51	**-4.44**
u2319	110995	110995	0.00	18000.00	110960	438.26	110555	18000.00	111327[1]	111146.50	18000.42	15008.89	**-0.30**
pr2392	50944	45407	10.87	18000.00	50902	285.26	50944	18000.00	54000	53252.75	8483.59	7303.01	**-5.66**
pcb3038	101173	91831	9.23	18000.00	101173	800.13	99612	18000.00	104367	104010.45	18000.18	15932.87	**-3.06**
fl3795	80069	71328	10.92	18000.00	80069	4496.09	76916	18000.00	94492	92311.60	17333.96	16339.19	**-15.26**
fnl4461	85088	84098	1.16	18000.00	85088	1490.80	83032	18000.00	92721	91987.95	17380.33	14834.60	**-8.23**
rl5915	279430	279116	0.11	18000.00	279277	8438.60	279430	18000.00	281337	280645.75	18000.03	17349.20	**-0.68**
rl5934	137838	-	-	-	137838	4037.07	134787	18000.00	158854	157125.89	18000.55	17000.64	**-13.23**
pla7397	142399	106131	25.47	18000.00	142399	6667.36	136820	18000.00	154773	153488.55	18000.06	16700.21	**-7.99**
Average	52500.66	47789.00	-	17087.41	52195.10	767.54	51874.02	9256.99	55540.73	55255.05	7062.81	5296.76	-

[1] One notices that HGA finds better feasible solutions than the optimal solutions reported by B&C [25].

Table B.9
Results for PCTSP on medium-sized instances of Set I.

Instances	B&C [4]		HGA-Giant				HGA				
	UB	Time	*Best*	*Avg.*	Time	TMB	*Best*	*Avg.*	Time	TMB	δ_2(%)
st70	260*	0.85	260	273.30	1266.53	664.74	260	260.00	792.75	5.85	0.00
eil76	235*	0.94	220	230.00	734.33	377.61	213[1]	213.30	488.85	120.07	**-3.18**
pr76	41248*	2.39	41248	41248.30	2110.90	686.59	41248	41248.00	1262.36	13.95	0.00
gr96	20688*	38.05	20688	20688.00	2126.95	464.85	20697	20697.00	1477.65	18.70	0.04
rat99	581*	14.30	581	582.45	1682.12	642.50	581	582.00	968.70	295.95	0.00
kroA100	9184*	9.11	9184	9342.90	2045.73	440.68	9184	9184.00	1446.20	20.41	0.00
kroB100	9096*	5.72	9096	9184.60	2122.75	571.52	9096	9098.05	1199.66	274.31	0.00
kroC100	9457*	18.26	9457	9701.65	1934.45	665.54	9457	9457.00	1441.24	18.38	0.00
kroD100	8719*	6.30	8997	9434.85	2138.77	606.55	8719	8719.00	1430.61	37.08	0.00
kroE100	9097*	6.87	9097	9249.40	2148.35	770.65	9097	9097.00	1543.35	23.68	0.00
rd100	3168*	6.53	3210	3243.15	2056.05	624.37	3168	3168.00	1538.52	94.76	0.00
eil101	232*	4.36	248	257.90	1035.23	376.38	232	232.20	741.61	277.42	0.00
lin105	5920*	168.02	5954	6001.45	1986.56	432.48	5920	5920.00	1495.24	26.85	0.00
pr107	18311*	6.87	18311	18315.80	2159.51	1041.38	18311	19313.10	1506.69	525.56	0.00
pr124	22998*	13.30	22998	23183.20	2320.54	600.97	22998	22998.00	1417.57	18.90	0.00
bier127	26347*	4.49	26347	26752.15	2354.50	895.53	26347	26347.00	1361.66	33.11	0.00
ch130	2408*	8.64	2426	2499.30	2221.70	714.48	2408	2408.00	1347.44	43.04	0.00
pr136	46167*	71.88	47087	47363.85	2189.83	929.18	46167	46167.00	1591.33	233.84	0.00
gr137	29575*	10.62	29575	29593.70	2215.35	639.47	29575	29575.00	1667.00	5.24	0.00
pr144	27424*	84.26	28061	28077.55	2265.45	690.53	27424	27424.00	1767.58	44.92	0.00
ch150	2760*	22.95	2792	2913.70	2201.50	890.90	2760	2760.30	1603.15	455.81	0.00
kroA150	11496*	2137.76	11649	12051.80	2286.16	975.07	11496	11496.00	1699.46	84.79	0.00
kroB150	11357*	36.53	11452	11956.95	2124.50	875.65	11357	11357.00	1850.63	48.11	0.00
pr152	36333*	68.85	36606	36850.00	2270.58	824.73	36333	36333.00	1995.90	118.36	0.00
u159	18511*	570.01	18689	18902.80	2419.91	748.44	18511	18511.00	1764.78	26.31	0.00
rat195	1112*	156.26	1129	1143.85	2096.24	763.96	1112	1112.45	1543.24	741.92	0.00
d198	6913*	1366.88	6929	6948.10	2341.96	748.18	6913	6913.00	1991.81	38.38	0.00
kroA200	12372*	118.79	12898	13630.75	2434.66	1440.98	12372	12380.65	2186.29	974.14	0.00
kroB200	12338*	351.33	12747	13319.45	2402.80	1184.53	12338	12338.00	1777.11	54.89	0.00
gr202	13790*	328.51	13894	14072.85	2397.94	708.81	13790	13796.15	1708.05	585.02	0.00
ts225	57995	14400.00	57535	58461.45	2529.29	765.36	57535	57535.00	1998.88	6.50	0.00
tsp225	1721*	317.29	1822	1881.30	2272.71	870.35	1721	1721.75	1948.02	897.45	0.00
pr226	36720*	5429.52	36935	38738.25	2463.89	1079.71	36720	37151.00	1900.28	980.77	0.00
gr229	39875*	154.54	40822	42451.30	2518.79	916.35	39875	40144.75	1819.58	888.08	0.00
gil262	986*	165.14	1130	1186.50	2633.00	1285.05	991	994.55	2048.30	1002.03	0.51
pr264	22644*	532.23	22919	23268.90	2879.28	1740.95	22903	25727.60	2100.00	676.60	1.14
a280	1231*	303.65	1286	1326.65	2359.74	776.42	1252	1259.05	1922.74	709.23	1.71
pr299	23089	14400.00	23023	23426.00	2852.53	1176.44	22514	22522.80	2552.63	791.74	**-2.21**
lin318	15913*	2355.21	16418	16942.30	2652.22	948.33	15913	15913.00	2350.49	563.66	0.00
rd400	6284*	2110.73	6948	7317.45	3448.83	1987.45	6284	6316.15	2634.51	1678.07	0.00
fl417	5754	14400.00	5562	5624.00	3547.23	875.99	5449	5450.85	2771.75	994.36	**-2.03**
gr431	35222*	14285.70	35245	35747.50	3395.01	1295.71	35222	35224.60	2664.16	921.48	0.00
pr439	35297*	1483.28	36727	37401.65	3305.44	1200.81	35297	35350.65	2529.08	748.10	0.00
pcb442	22281	14400.00	23496	24007.00	3537.49	1244.83	22281	22301.10	3041.85	1485.59	0.00
d493	13582*	1943.26	14229	14448.15	3346.13	1587.33	13582	13600.95	3256.53	994.78	0.00
att532	8943*	10280.10	9289	9491.20	3787.90	1613.14	10433	10593.00	3270.04	1813.74	16.67
Average	16209.41	2230.44	16417.74	16711.59	2383.07	899.16	16218.61	16324.17	1813.38	443.74	-

[1] One notices that HGA finds better feasible solutions than the optimal solutions reported by B&C [4].

Table B.10
Results for PCTSP on large-sized instances of Set I.

Instances	HGA-Giant				HGA				
	Best	*Avg.*	Time	TMB	*Best*	*Avg.*	Time	TMB	δ_2(%)
ali535	47890	52262.80	4519.04	2758.00	42756	42984.10	3721.87	2148.39	**-10.72**
u574	15780	16551.30	4189.39	3013.66	14671	14700.85	3916.90	1981.10	**-7.03**
rat575	3270	3348.85	4099.71	1331.54	3023	3036.85	3545.65	1864.71	**-7.55**
p654	16552	17626.60	3941.21	1956.80	16173	16173.00	3259.72	804.99	**-2.29**
d657	21784	22598.00	4461.48	2880.53	20889	20904.65	3385.33	2084.09	**-4.11**
gr666	84119	86308.20	4358.81	2372.07	78410	80987.20	3552.02	1650.69	**-6.79**
u724	18543	19301.60	4991.53	4470.48	16692	16749.25	4003.05	1612.22	**-9.98**
rat783	4325	4411.10	5277.20	2495.65	3938	3979.55	4783.58	1768.90	**-8.95**
dsj1000	6997800	7075698.35	7245.65	5423.50	6940600	6956310.00	6896.05	4090.00	**-0.82**
pr1002	118568	122205.35	6048.96	5157.00	106138	108198.10	5336.69	3625.03	**-10.48**
u1060	100381	102363.10	7231.86	5815.26	88335	88665.00	5427.29	2880.01	**-12.00**
vm1084	76512	90154.95	7887.30	6953.26	65255	65256.90	5796.05	1498.21	**-14.71**
pcb1173	26934	27469.60	6878.13	5312.04	24916	24989.15	7144.23	3451.92	**-7.49**
d1291	24049	24600.05	7618.76	4238.72	23276	23380.85	6738.42	3042.44	**-3.21**
rl1304	114795	123217.65	9369.78	6994.19	100463	101120.10	8019.14	4149.27	**-12.48**
rl1323	132231	138641.90	9195.54	8503.08	107724	108437.25	8577.82	5991.34	**-18.53**
nrw1379	25518	25954.85	10504.95	6806.16	23831	23934.25	9366.99	3989.30	**-6.61**
fl1400	8084	8216.20	8644.44	4470.06	8336	8343.45	9068.82	4034.91	**3.12**
u1432	76281	78190.90	8709.27	4461.42	72688	72908.40	9032.67	3971.92	**-4.71**
fl1577	9941	10083.85	9384.07	4445.69	9728	9739.85	6949.05	4495.42	**-2.14**
d1655	29662	30511.45	11214.47	9003.36	28321	28730.45	8465.60	4938.04	**-4.52**
vm1748	112141	130574.70	16281.23	15758.16	82916	83133.20	10188.62	5078.04	**-26.06**
u1817	28613	29363.45	13185.74	8177.49	26490	26824.50	11088.22	6072.33	**-7.42**
rl1889	160227	167929.95	13974.32	11564.46	113498	114168.45	13340.98	7308.52	**-29.16**
d2103	36513	36972.85	11210.51	6603.01	34286	34287.90	13037.04	4268.58	**-6.10**
u2152	32478	33222.95	15191.68	10582.24	30649	30921.55	15537.73	7965.56	**-5.63**
u2319	118786	119444.20	15661.34	6877.61	116000	116000.00	16073.73	496.72	**-2.35**
pr2392	181451	185710.90	13677.79	5550.13	164029	164955.25	17622.28	10371.24	**-9.60**
pcb3038	68022	69834.65	18000.38	16520.10	62174	62818.70	18000.42	17591.35	**-8.60**
fl3795	13594	15375.20	18000.17	16308.70	12741	13404.40	18000.27	17055.30	**-6.27**
fnl4461	93107	94070.45	18000.19	17465.75	81399	81720.40	18000.59	17867.45	**-12.57**
rl5915	315805	323273.25	18000.65	13592.38	216241	218312.65	18001.11	17855.01	**-31.53**
rl5934	316957	323398.55	18000.26	10025.79	218703	222194.10	18000.77	17881.14	**-31.00**
pla7397	8837800	8896016.67	18000.63	15546.70	8296170	8328854.00	18001.35	17803.95	**-6.13**
Average	537309.21	544261.89	10381.07	7453.97	507395.85	509327.19	9761.18	6226.12	-

Table B.11
Results for PCTSP on medium-sized instances of Set II.

Instances	B&C [4]		HGA-Giant				HGA				
	UB	Time	*Best*	*Avg.*	Time	TMB	*Best*	*Avg.*	Time	TMB	δ_2(%)
st70	247*	1.31	247	254.60	1157.24	431.77	247	247.00	456.84	2.95	0.00
eil76	200*	6.44	202	206.20	574.44	244.88	200	200.00	290.26	49.88	0.00
pr76	38330*	22.13	38850	38977.00	2034.23	802.75	38330	38330.00	976.65	15.46	0.00
gr96	19380*	34.60	19380	19380.00	2215.10	55.86	19380	19380.00	1234.05	10.74	0.00
rat99	518*	61.50	526	535.60	1665.93	725.69	518	518.40	592.64	210.63	0.00
kroA100	8519*	29.70	8795	8975.55	2111.43	906.26	8519	8519.00	1190.93	95.89	0.00
kroB100	7794*	41.70	7821	8148.30	2333.69	977.55	7794	7794.00	1177.52	25.59	0.00
kroC100	9060*	41.16	9296	9421.35	2355.78	675.91	9060	9060.00	1417.05	10.46	0.00
kroD100	8267*	30.74	8459	8561.40	2377.51	786.14	8267	8267.00	1242.05	295.46	0.00
kroE100	7644*	17.90	8180	8663.35	2166.03	579.36	7644	7644.00	1239.45	19.25	0.00
rd100	2892*	22.29	2932	3009.95	2238.05	688.31	2892	2892.00	921.78	57.41	0.00
eil101	211*	9.11	221	230.40	1044.23	267.20	211	212.55	451.24	171.27	0.00
lin105	5614*	716.02	5802	5825.85	2069.88	648.43	5614	5622.15	1308.90	245.13	0.00
pr107	26372*	76.69	26485	26639.75	2371.05	1158.75	26372	26372.00	1433.83	111.59	0.00
pr124	23150*	162.39	23868	24103.75	2344.45	581.15	23150	23150.00	1267.54	26.94	0.00
bier127	24478*	37.55	24992	25129.05	2606.16	848.59	24478	24478.00	1276.40	30.21	0.00
ch130	2220*	83.66	2366	2435.55	2382.69	1007.95	2220	2220.00	1314.72	504.62	0.00
pr136	40241	14400.00	40636	41808.25	2156.29	1033.55	40023	40023.00	1403.00	202.39	**-0.54**
gr137	28242*	366.15	28242	28251.70	2292.34	753.94	28242	28242.00	1752.22	16.38	0.00
pr144	27073*	284.38	27449	28700.55	2697.84	1275.77	27073	27073.00	1450.82	29.21	0.00
ch150	2476*	541.81	2648	2740.70	2468.95	972.60	2476	2478.10	1360.78	235.18	0.00
kroA150	9968*	60.85	10715	11038.15	2540.79	974.39	9968	9968.00	1521.98	33.11	0.00
kroB150	10278*	469.95	10719	10939.35	2502.28	751.03	10278	10439.50	1657.29	72.20	0.00
pr152	34474*	249.75	34710	34912.80	2384.66	1117.21	34474	34474.30	1762.78	346.99	0.00
u159	17161*	763.28	18222	18597.75	2291.02	948.67	17161	17161.00	1617.72	63.57	0.00
rat195	988*	112.81	1031	1046.30	2280.34	916.64	990	994.25	1045.83	520.83	0.20
d198	6653*	2579.62	6676	6705.95	2458.00	729.38	6653	6653.00	1604.25	419.63	0.00
kroA200	11219*	2278.69	12027	12624.50	2626.21	1387.48	11219	11251.60	1898.65	850.61	0.00
kroB200	11250*	415.38	12799	13325.25	2749.84	1116.39	11250	11250.00	1811.84	180.79	0.00
gr202	12804*	753.52	13274	13391.90	2712.00	998.64	12804	12808.10	1600.15	635.79	0.00
ts225	53102*	907.77	54975	56442.90	2530.42	918.77	53102	53102.00	1583.29	229.16	0.00
tsp225	1585*	1803.93	1723	1782.85	2517.27	661.19	1585	1589.45	1463.25	842.61	0.00
pr226	36190*	8186.06	37088	38052.90	2426.58	885.13	36190	36775.45	1947.51	686.58	0.00
gr229	35856*	3478.96	36615	37188.75	2703.52	1180.40	35856	36020.80	1960.93	407.65	0.00
gil262	865*	165.21	1043	1095.30	2942.29	1356.09	865	865.55	1463.20	639.99	0.00
pr264	23660	14400.00	22790	23118.90	2758.50	1253.41	25080	25099.60	2121.82	634.38	10.05
a280	1143	14400.00	1217	1262.25	2643.18	1115.98	1131	1138.70	1363.13	556.05	**-1.05**
pr299	20613	14400.00	21636	22019.90	2677.80	1171.38	20534	20591.45	2247.42	1298.94	**-0.38**
lin318	14909*	3394.98	15404	16223.55	2748.88	1212.81	14909	14925.95	2345.94	953.60	0.00
rd400	5590*	3102.53	7082	7226.35	3394.91	1067.25	5590	5781.40	2517.79	1409.09	0.00
fl417	5971	14400.00	5466	5542.15	3552.72	1282.62	5354	5359.15	2366.97	858.14	**-2.05**
gr431	31725	14400.00	33331	33932.05	3792.43	2189.49	31725	31725.00	2453.09	835.34	0.00
pr439	33110	14400.00	34534	35038.95	3943.30	1696.64	33079	33086.10	2653.69	668.86	**-0.09**
pcb442	19165	14400.00	21878	22446.90	3990.55	1727.80	19162	19188.05	2909.42	1160.58	**-0.02**
d493	12835	14400.00	14240	14554.30	3811.67	1316.84	12687	12719.25	2915.55	1106.52	**-1.15**
att532	8231	14400.00	9068	9200.75	4335.16	1312.68	9792	9939.30	3210.29	1526.12	18.96
Average	15266.80	3811.10	15775.22	16080.64	2542.99	971.97	15307.57	15339.76	1604.40	419.65	-

Table B.12
Results for PCTSP on large-sized instances of Set II.

Instances	HGA-Giant				HGA				
	Best	*Avg.*	Time	TMB	*Best*	*Avg.*	Time	TMB	$\delta_2(\%)$
ali535	47600	50836.75	5481.66	2164.89	40838	41073.30	3416.66	2160.47	**-14.21**
u574	15790	16337.25	4802.40	1522.94	13660	13738.10	3456.81	2078.83	**-13.49**
rat575	3005	3073.95	4746.15	2022.61	2700	2712.00	2724.57	1257.38	**-10.15**
p654	16233	16492.95	4757.14	2166.05	15461	15469.85	3351.91	1702.63	**-4.76**
d657	21075	21497.50	5872.18	1932.54	18950	18979.90	3755.76	1820.76	**-10.08**
gr666	83808	86094.80	5538.10	2295.56	75702	76431.60	3826.03	1844.19	**-9.67**
u724	18070	18984.40	4889.32	1300.49	14924	15030.60	4356.68	2933.25	**-17.41**
rat783	4081	4150.45	5136.55	1833.39	3446	3513.55	3926.83	1968.81	**-15.56**
dsj1000	6658210	6698721.50	6578.54	5468.90	6428930	6473253.50	6368.08	4346.50	**-3.44**
pr1002	119676	122904.35	6145.16	1718.06	98795	100118.00	6100.34	3884.73	**-17.45**
u1060	97788	100108.30	7116.67	1771.21	81534	82636.45	6000.21	3785.21	**-16.62**
vm1084	69869	78092.75	9083.64	4339.40	63684	63744.74	6639.36	3508.90	**-8.85**
pcb1173	26350	27363.60	9777.27	3449.71	22982	23223.75	6765.56	3839.71	**-12.78**
d1291	23583	24179.10	8895.48	4675.84	22148	22327.05	5478.72	3323.13	**-6.08**
rl1304	113099	118157.60	8620.93	3407.41	95589	96046.30	6363.72	2790.89	**-15.48**
rl1323	121711	129922.50	9484.58	3661.33	102312	103271.80	6746.93	4499.24	**-15.94**
nrw1379	25049	25339.80	12061.47	5763.69	20805	21200.75	8473.57	4805.47	**-16.94**
fl1400	8064	8389.95	11851.43	8180.52	7732	7780.40	6437.19	3320.17	**-4.12**
u1432	73071	73931.15	9896.33	3153.81	58418	59232.45	7718.68	4745.64	**-20.05**
fl1577	9836	10023.90	9579.35	4623.14	9111	9174.95	8257.34	5592.10	**-7.37**
d1655	30854	31228.80	12543.25	5283.63	26257	26735.05	9320.19	7625.40	**-14.90**
vm1748	92731	102265.20	14105.31	8951.64	80034	80407.85	10982.57	6402.71	**-13.69**
u1817	29390	30224.75	12691.13	5159.59	24316	24679.65	9936.50	7717.81	**-17.26**
rl1889	141203	148699.15	16072.90	5702.52	110226	111197.95	12595.92	8933.69	**-21.94**
d2103	35451	36533.90	14148.48	6486.04	32935	32968.20	11825.39	5348.26	**-7.10**
u2152	33168	33655.30	15866.23	6010.47	27543	27942.90	10232.30	8366.51	**-16.96**
u2319	110241	112362.63	17030.77	6024.11	84351	85185.30	12510.91	9243.60	**-23.48**
pr2392	180615	183115.50	18000.32	7757.63	152816	156057.50	16755.98	16166.88	**-15.39**
pcb3038	67488	69128.40	18000.44	9210.16	55810	56828.40	18000.24	17887.34	**-17.30**
fl3795	14267	15973.30	18000.34	12775.48	11859	12732.70	18000.24	17682.58	**-16.88**
fnl4461	90189	91179.85	18000.37	12372.52	72006	73024.85	18000.60	17851.83	**-20.16**
rl5915	276743	290102.75	18000.48	16072.21	209848	212486.30	18001.09	17859.98	**-24.17**
rl5934	278156	293190.80	18000.33	16426.86	213500	215103.75	18000.86	17781.86	**-23.24**
pla7397	8641000	8641000.00	18000.20	13202.50	7376900	7466082.50	18001.12	17870.27	**-14.63**
Average	516984.24	520978.32	11140.44	5790.79	461062.41	465599.76	9186.14	7086.67	**-10.82**

Table B.13
Results for PCTSP on medium-sized instances of Set III.

Instances	B&C [4]		HGA-Giant				HGA				
	UB	Time	*Best*	*Avg.*	Time	TMB	*Best*	*Avg.*	Time	TMB	δ_2(%)
st70	308*	7.71	309	311.65	1113.25	375.41	308	308.65	525.64	228.30	0.00
eil76	204*	9.05	206	208.85	671.13	266.01	204	204.00	375.40	14.80	0.00
pr76	42200*	24.02	42955	43378.25	2157.69	917.24	42200	42200.00	1261.54	8.22	0.00
gr96	22491*	50.48	22340	22615.75	2297.87	999.25	22316[1]	22354.75	1407.77	413.90	**-0.11**
rat99	579*	55.05	600	611.05	1453.40	531.32	580	582.65	826.28	284.86	0.17
kroA100	8325*	17.47	8325	8444.70	2237.71	1008.02	8325	8325.00	1189.84	32.95	0.00
kroB100	8768*	42.17	8768	8829.25	2203.32	812.81	8768	8774.95	1222.64	403.58	0.00
kroC100	9283*	86.21	9410	9546.70	2222.38	923.35	9283	9363.90	1440.03	179.19	0.00
kroD100	8998*	57.52	8998	9063.60	2090.52	774.45	8998	8998.00	1278.01	24.81	0.00
kroE100	9313*	41.76	9398	9437.75	2273.53	722.27	9313	9313.00	1215.78	22.28	0.00
rd100	3377*	51.72	3377	3394.10	2010.36	813.90	3377	3419.00	1368.30	317.12	0.00
eil101	223*	17.55	230	232.05	1094.94	451.81	224	224.55	525.92	171.42	0.45
lin105	6547*	423.40	6667	6706.10	2172.05	845.92	6547	6547.00	1426.18	78.27	0.00
pr107	27198	14400.00	27198	27258.10	2400.69	1008.50	27184	27184.00	1097.56	24.53	**-0.05**
pr124	26375*	206.93	26785	27137.35	2512.35	1040.17	26375	26375.00	1184.75	10.29	0.00
bier127	42358*	4654.12	42930	43118.30	2501.51	868.81	42359	42360.40	1816.47	644.71	0.00
ch130	2305*	60.85	2338	2352.75	2408.65	632.41	2305	2312.55	1326.79	557.86	0.00
pr136	42179*	4564.78	43227	43905.45	2719.20	1166.28	42179	42188.10	1524.84	482.31	0.00
gr137	34023	14400.00	33714	34140.45	2598.89	842.07	33270	33403.95	1840.08	724.72	**-1.32**
pr144	30033	14400.00	30123	30402.00	2574.98	1030.72	29746	29746.00	1553.85	215.12	**-0.96**
ch150	2675*	132.36	2706	2740.70	2541.85	939.59	2675	2678.50	1328.33	134.24	0.00
kroA150	9409*	78.72	9750	9957.95	2601.10	1229.45	9409	9409.00	1412.88	189.20	0.00
kroB150	10392*	256.38	10763	10927.35	2428.56	617.70	10564	10564.00	1625.80	150.05	1.66
pr152	40937	14400.00	41488	41936.60	2224.23	1079.51	40599	40599.00	1511.65	27.00	**-0.83**
u159	17631*	328.30	17670	17803.75	2324.98	852.79	17631	17670.70	1705.66	616.38	0.00
rat195	999*	776.12	1086	1112.55	2285.45	1181.38	1013	1049.85	1257.77	458.17	1.40
d198	7388*	1974.04	7435	7472.80	2525.87	758.28	7388	7388.00	2067.25	329.20	0.00
kroA200	11987*	803.92	12293	12787.70	2760.52	847.09	12075	12104.85	1904.79	962.35	0.73
kroB200	10752*	1398.61	10888	11157.40	2858.80	1039.78	10752	10752.00	1729.95	378.08	0.00
gr202	14377*	5085.50	14806	14903.80	2780.78	928.50	14546	14558.75	1829.80	858.27	1.18
ts225	53414	14400.00	53325	53438.85	2402.22	702.62	53325	53325.00	1419.70	240.27	0.00
tsp225	1649	14400.00	1707	1749.30	2520.54	893.41	1649	1652.55	1416.66	569.65	0.00
pr226	39091	14400.00	39296	39669.05	2505.70	929.80	38874	38912.25	1946.43	875.65	**-0.56**
gr229	46791*	4004.14	47146	48360.10	3190.54	1254.60	46749[1]	47162.40	2337.96	672.78	**-0.09**
gil262	961*	387.02	1026	1057.85	2796.33	893.21	966	970.00	1462.00	580.02	0.52
pr264	23264	14400.00	23230	24015.65	3264.81	1951.50	23093	23108.45	1539.55	220.56	**-0.59**
a280	1084*	4324.35	1085	1094.90	2415.33	994.18	1087	1088.80	1230.21	572.81	0.28
pr299	20317*	5129.46	21324	21577.95	2922.90	1517.35	20317	20448.00	2148.44	1495.82	0.00
lin318	16401*	6867.39	17798	18404.75	3380.24	1080.94	16401	16402.15	2370.42	871.77	0.00
rd400	5700*	2602.41	6253	6558.75	3699.86	1367.74	5877	5895.65	2376.49	857.93	3.11
fl417	5740	14400.00	5553	5620.95	4028.67	2096.99	5368	5377.45	1103.21	357.33	**-3.33**
gr431	56484	14400.00	63656	65237.30	4405.86	1619.90	55817	57198.60	3233.26	1842.90	**-1.18**
pr439	35771	14400.00	37236	37795.45	3940.71	1968.45	35788	35814.35	2473.97	967.82	0.05
pcb442	19632	14400.00	21316	21976.10	3715.95	1426.35	19666	20028.25	2413.00	1819.01	0.17
d493	13480	14400.00	14137	14542.65	3513.02	1479.74	13507	13517.30	2902.48	995.09	0.20
att532	10258	14400.00	11953	12056.35	5744.27	2534.88	10315	10477.60	3437.85	1944.92	0.56
Average	17427.63	5350.42	17887.48	18153.28	2641.16	1048.18	17376.35	17442.15	1621.59	517.97	0.03

[1] One notices that HGA finds better feasible solutions than the optimal solutions reported by B&C [4].

Table B.14
Results for PCTSP on large-sized instances of Set III.

Instances	HGA-Giant				HGA				
	Best	*Avg.*	Time	TMB	*Best*	*Avg.*	Time	TMB	$\delta_2(\%)$
ali535	65661	66246.25	5730.36	2094.07	68710	69764.6	3977.863	2677.96	4.64
u574	16157	16397.70	4795.63	2385.67	13730	13822.35	3007.96	1315.915	**-15.02**
rat575	2917	2956.90	5530.21	2681.86	2658	2701.3	2660.836	1695.456	**-8.88**
p654	16085	16289.70	4901.00	2363.43	15853	15881.9	3031.258	1607.091	**-1.44**
d657	23671	24151.15	5338.83	1786.25	21380	21875.45	4052.302	2953.046	**-9.68**
gr666	95945	97270.30	7417.76	4435.51	92942	93215.35	4476.175	2486.513	**-3.13**
u724	16562	17093.85	5561.84	2368.25	14054	14093.9	3523.882	1816.394	**-15.14**
rat783	3681	3829.80	7990.99	3744.49	3503	3608	3702.949	1751.809	**-4.84**
dsj1000	6126650	6336921.60	6543.15	5799.36	6099840	6181059	5648.652	4855.394	**-0.44**
pr1002	111941	114649.10	11067.47	6027.71	92882	93979.3	5509.153	4359.783	**-17.03**
u1060	82282	86070.05	7319.40	2661.46	73035	73956.45	5231.196	3551.189	**-11.24**
vm1084	63126	66508.45	8700.17	4304.68	57945	58226.8	5199.252	3100.454	**-8.21**
pcb1173	24908	25581.65	13603.76	7447.96	22969	23934.6	6636.45	5696.322	**-7.78**
d1291	25119	25887.70	8792.17	4146.81	23049	23128.55	7140.085	3542.74	**-8.24**
rl1304	106362	115222.50	12498.09	6158.64	81770	82141.55	6307.316	4078.16	**-23.12**
rl1323	111527	117009.25	10944.81	4032.25	90419	90907.53	6129.159	4270.05	**-18.93**
nrw1379	22078	22430.05	18000.09	11323.12	21446	22760.75	7500.976	6377.507	**-2.86**
fl1400	7301	7421.05	18000.11	7555.35	6975	7007.15	5764.467	2576.688	**-4.47**
u1432	57658	59432.30	11420.94	5401.35	51171	51515.65	5986.278	3438.962	**-11.25**
fl1577	9801	10273.10	18000.14	8630.82	8967	8984.75	6507.709	3432.093	**-8.51**
d1655	31294	32498.55	14776.31	6702.76	27553	27646.1	8515.859	4979.303	**-11.95**
vm1748	103042	125572.90	18000.19	8477.87	67744	67934.45	8315.873	3773.694	**-34.26**
u1817	24718	25525.95	16373.74	6680.61	21427	21681.15	7770.203	5462.698	**-13.31**
rl1889	136123	143098.40	16635.66	7346.77	101257	101533	9976.649	6185.451	**-25.61**
d2103	30504	31208.05	18000.16	9900.11	29254	29261.45	8875.615	2896.752	**-4.10**
u2152	27756	28669.35	18000.13	8352.88	23677	24148.85	7911.774	5968.639	**-14.70**
u2319	95562	98274.15	18000.17	11252.63	86288	86996.95	11012.27	6800.371	**-9.70**
pr2392	152349	155399.85	18000.20	8465.84	151021	152901.5	13270.02	11648.99	**-0.87**
pcb3038	57885	59404.50	18000.18	9279.09	51095	54014.25	17992.01	17107.99	**-11.73**
fl3795	14806	15673.40	18000.43	7880.81	11850	11986.3	18002.67	14258.02	**-19.96**
fnl4461	78257	80511.60	18000.27	9824.28	77167	78417.1	18000.35	17879.18	**-1.39**
rl5915	263766	290963.05	18000.64	10836.60	177708	180451.5	18000.44	17735.67	**-32.63**
rl5934	284844	308088.90	18000.47	9457.43	183885	186586.2	18000.89	17772.43	**-35.44**
pla7397	6368310	6556739.50	18000.64	13159.04	6364860	6448766	18000.74	17789.99	**-0.05**
Average	431136.71	278919.72	12880.77	6557.82	418767.18	424261.46	8401.15	6348.31	-