European Journal of Operational Research 313 (2024) 44–68

Contents lists available at ScienceDirect

European Journal of Operational Research

journal homepage: www.elsevier.com/locate/ejor

Discrete Optimization

A revisited branch-and-cut algorithm for large-scale orienteering problems

Gorka Kobeaga[a], Jairo Rojas-Delgado[a], María Merino[a,b,*], Jose A. Lozano[a,c]

[a] Basque Center for Applied Mathematics - BCAM, Alameda Mazarredo 14, 48009 Bilbao, Bizkaia, Spain
[b] Department of Mathematics, University of the Basque Country UPV/EHU, Barrio Sarriena, s/n, 48940 Leioa, Bizkaia, Spain
[c] Department of Computer Science and Artificial Intelligence, University of the Basque Country UPV/EHU, Manuel Lardizabal Ibilbidea 1, 20018 Donostia, Gipuzkoa, Spain

ARTICLE INFO

Article history:
Received 9 September 2021
Accepted 27 July 2023
Available online 31 July 2023

Keywords:
Routing
Orienteering problem
Branch-and-cut
Large problems

ABSTRACT

The orienteering problem is a route optimization problem which consists of finding a simple cycle that maximizes the total collected profit subject to a maximum distance limitation. In the last few decades, the occurrence of this problem in real-life applications has boosted the development of many heuristic algorithms to solve it. However, during the same period, not much research has been devoted to the field of exact algorithms for the orienteering problem. The aim of this work is to develop an exact method which is able to obtain the optimum in a wider set of instances than with previous methods, or to improve the lower and upper bounds in its disability.

We propose a revisited version of the branch-and-cut algorithm for the orienteering problem which includes new contributions in the separation algorithms of inequalities stemming from the cycle problem, in the separation loop, in the variables pricing, and in the calculation of the lower and upper bounds of the problem. Our proposal is compared to three state-of-the-art algorithms on 258 benchmark instances with up to 7397 nodes. The computational experiments show the relevance of the designed components where 18 new optima, 76 new best-known solutions and 85 new upper-bound values were obtained.

1. Introduction

The Orienteering Problem (OP) is a well-known routing problem proposed in the 80s (Golden et al., 1987; Tsiligirides, 1984) and the first survey in OP was published by Vansteenwegen et al. (2011).

The OP goal can be defined as finding a cycle, within a weighted complete graph with associated vertex profits, so that the sum of the profits of the vertices in the cycle is maximized and its length is lower or equal to a given constant d_0. In addition to the length constraint, every feasible cycle solution must visit a given depot vertex. The OP can be seen as a combination of the Knapsack Problem (KP) and the Travelling Salesperson Problem (TSP).

The KP goal can be defined as finding the subset of items, within a larger set of items with associated weights and profits, so that the sum of the profits of the items in the subset is maximum and their total weight is lower than or equal to a given constant w_0. In the KP, the feasibility of a subset is checked in linear time. In the OP, however, the feasibility of a solution is checked by solving a TSP-decision problem. A subset of vertices is feasible if there exists a cycle (Hamiltonian in the subgraph obtained by the vertices) whose length does not exceed d_0, finding such a cycle is an NP-complete problem. This simple but non-trivial combination of two NP-hard problems makes the OP an interesting problem to study.

The OP is classified as one of the three generic problems in TSPs with profits (Feillet et al., 2005). The TSPs with profits have two opposite criteria: one that motivates the salesperson to travel and another that imposes a constraint in the route length, e.g., the route must have a minimum length or the route length must be not greater than a given value. The other two problems of TSPs with profits are the Profitable Tour Problem (PTP) (Dell'Amico et al., 1995) and the Price Collecting TSP (PCTSP) (Balas, 1989). In the PTP the goal is to maximize the difference between the total collected profit and the cost of the tour. Particularly, the PCTSP is closely related to the OP. In both problems, the solutions are simple cycles that contain a given depot vertex. The two prob-

* Corresponding author at: Basque Center for Applied Mathematics - BCAM, Spain.
E-mail addresses: maria.merino@ehu.eus (M. Merino), jlozano@bcamath.org (J.A. Lozano).

https://doi.org/10.1016/j.ejor.2023.07.034
0377-2217/

Table 1
Exact approaches for the OP.

Publication	Approach	Contributions	Benchmark
Laporte & Martello (1990)	B&B	KP bounds	Laporte & Martello (1990)
Ramesh et al. (1992)	B&B	Lagrangian relaxation	Ramesh et al. (1992)
Leifer & Rosenwein (1994)	B&C	Logical, Connectivity, Edge Cover	Tsiligirides (1984)
Fischetti et al. (1998)	B&C	Cycle Cover Path Inequalities Column Generation Primal Heuristics	Fischetti et al. (1998)
Gendreau et al. (1998)	B&C	Vertex Cover Alternative Obj Primal Heuristics	Gendreau et al. (1998)

lems differ in two aspects. First, the constraint of the problem in the PCTSP is limited to the minimum collected vertex profit rather than limited to the maximum length of the route as in the OP. Secondly, the objective function in PCTSP is to minimize the route length while in the OP it is to maximize the collected vertex profits. See Angelelli et al. (2014) for the study on the complexity and approximation algorithms for TSPs with profits.

The occurrence of the OP in many real-life applications, such as logistics and tourism, has boosted the emergence of many variants and algorithms to solve the problem over the last decades. Two surveys of the OP variants, approaches, and applications can be seen in Gunawan et al. (2016) and Vansteenwegen & Gunawan (2019). In this paper, we have focused on solving the classical OP through an exact algorithm that takes advantage of the similarities between the OP and other related problems, such as the KP or the TSP. In this direction, some exact algorithms have been proposed for the OP (Gendreau et al., 1998; Laporte & Martello, 1990; Leifer & Rosenwein, 1994; Ramesh et al., 1992). The most competitive approach thus far was proposed by Fischetti et al. (1998) two decades ago. To our knowledge, no exact algorithm for the classical OP has been published after this work. The first exact algorithm, a Branch-and-Bound (B&B) algorithm, was published in Laporte & Martello (1990) where bounds for the problem were provided based on the Knapsack relaxation of the OP. In Ramesh et al. (1992), new bounds for the B&B were obtained by Lagrangian relaxation. In Leifer & Rosenwein (1994) a Branch-and-Cut (B&C) algorithm was proposed, which included logical, connectivity, and cover cuts for the first time. In Gendreau et al. (1998) a B&C was proposed for a variant of the OP which considers multiple depot nodes. The B&C approach in Fischetti et al. (1998) outperformed the previous ones in middle-sized OP instances by considering column generation, new valid inequalities (cycle cover and path inequalities), problem-specific separation algorithms, and an efficient primal heuristic. See Table 1 for a summary of exact algorithms for the OP.

In the last two decades, authors dealing with exact approaches have focused on solving variants of the problem, such as Team OP (Bianchessi et al., 2018; Boussier et al., 2007; Dang et al., 2013; Keshtkaran et al., 2015; Poggi et al., 2010), Arc OP (Archetti et al., 2016), Team Arc OP (Archetti et al., 2014; Riera-Ledesma & Salazar-González, 2017) and Probabilistic OP (Angelelli et al., 2017).

Recent results for large-sized instances of the OP, presented in Kobeaga et al. (2018); Santini (2019) and Sun et al. (2022), have shown that the state-of-the-art B&C algorithm in Fischetti et al. (1998) does not obtain satisfactory results when the number of nodes is larger than 400. In half of the large-sized instances, the named B&C algorithm does not produce any output. In many of the other half of instances, the returned solution value is far from the values obtained by the heuristic algorithms. The motivation of this work is to develop a B&C algorithm which is able to improve the values of the best-known lower and upper bounds in the literature, and if possible, to obtain optimality certifications in a wider set of instances than the state-of-the-art B&C algorithm. By optimality certification we are referring to the fact that the algorithm is able to guarantee that the global optimum was found. In our view, there is room for improvement for B&C algorithms in the case of the OP, mainly if we consider that some of the successful techniques developed for the TSP, such as shrinking and efficient pricing (Applegate et al., 2007), have not yet been adapted for the OP. This paper is an attempt to combine the adaptation of some of those techniques with our OP specific contributions.

In this work, we have developed and adapted techniques to scale the B&C algorithm to large OP instances. Our main contributions are the following:

- Develop a joint separation algorithm for Subcycle Elimination Constraints and Connectivity Constraints using the shrinking techniques in Kobeaga et al. (2021) (Section 4.2.1), and blossom separation algorithms for Cycle Problems which generalize the heuristics in Padberg & Hong (1980) and Grötschel & Holland (1991) (Section 4.2.2).The combination of the previous separation algorithms with a three-level separation loop for the OP (Section 4.4), which treats the cuts according to their relevance, allows the algorithm to be speeded up providing a considerable improvement in the solution quality and running time.
- Design an efficient variable pricing procedure for the OP inspired by the one developed in Applegate et al. (2007) for the TSP. It enables repetitive calculations to be avoided and the exact calculation of the reduced cost of some variables to be skipped (Section 4.3).
- Devise a combination of two alternative primal heuristics, a greedy by Fischetti et al. (1998) in the separation loop and a metaheuristic based at the beginning of branch nodes by Kobeaga et al. (2018). The new combination boosts the quality of the obtained solutions in large-sized problems (Section 4.5).
- Formulate the computation of the global upper bound in the branching phase for the OP in a way which enables the upper bound obtained in the branching root node to be updated (Section 4.6).

The computational experiments presented in this paper show the importance of the proposed techniques and components for the B&C algorithm for the OP. We compare our B&C algorithm with three state-of-the-art algorithms, two heuristics (Kobeaga et al., 2018; Santini, 2019) and an exact (Fischetti et al., 1998), on 258 benchmark instances with up to 7397 nodes. In comparison to the literature, the revisited B&C obtains:

- 180 optimum values, from which 18 are new.
- 245 best-known solution values, from which 76 are new.
- 249 best-known upper-bound values, from which 85 are new.

The rest of the paper is organized as follows. In Section 2 the 0-1 Integer Linear model of the OP is introduced. In Section 3 we

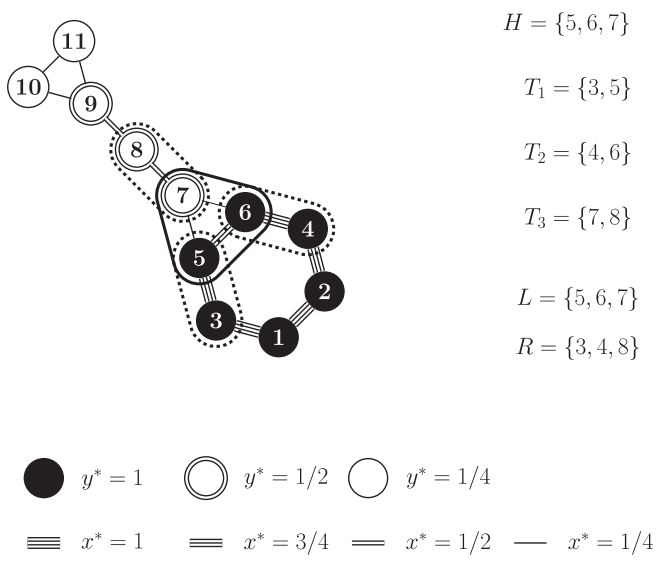

Fig. 1. Example of a comb tuple with $t = 3$ teeth with exactly two vertices each.

present the valid inequalities for the OP. In Section 4 we detail the proposed B&C algorithm for the problem. In Section 5 the results of the computational experiments are shown. The detailed experimental results can be found in the appendices.

2. OP modelling and polyhedral considerations

The OP can be defined by a 5-tuple $\langle G, d, s, o, d_0\rangle$, where $G = K_n = (V, E)$ is a complete graph with vertex set V and edge set E; $d = (d_e)$ where d_e is the positive distance value (time or weight) associated to each $e \in E$; $s = (s_v)$, where s_v is a positive value that represents the score (profit) of vertex $v \in V$; $o \in V$ is a vertex selected as the depot; and d_0 is a positive value that limits the cycle length.

Let us define the following sets:

$$(Q : W) := \{[u, v] \in E : u \in Q, v \in W\} \qquad Q, W \subseteq V \tag{1a}$$

$$\delta(Q) := (Q : V - Q) \qquad Q \subseteq V \tag{1b}$$

$$E(Q) := (Q : Q) \qquad Q \subseteq V \tag{1c}$$

$$V(T) := \{v \in V : T \cap (v : V) \neq \emptyset\} \qquad T \subseteq E \tag{1d}$$

where $(Q : W)$ are the edges connecting Q and W, $\delta(Q)$ is the set of edges in the coboundary of Q also known as the star-set of Q, $E(Q)$ is the set of edges between the vertices of Q, and $V(T)$ is the set of vertices incident with an edge set T. For simplicity, we sometimes denote $\{e\}$ and $\{v\}$ by e and v, respectively, e.g., $\delta(v)$ and $V(e)$.

We denote by $\mathbb{R}^V$, and $\mathbb{R}^E$, the space of real vectors whose components are indexed by elements of V, and E respectively. In the model of the OP, two types of decision variables are used, $y = (y_v) \in \mathbb{R}^V$ and $x = (x_e) \in \mathbb{R}^E$, associated with the nodes and edges of G, respectively, where:

$$y_v = \begin{cases} 1 & \text{if node } v \text{ is visited} \\ 0 & \text{otherwise} \end{cases}$$

$$x_e = \begin{cases} 1 & \text{if edge } e \text{ is traversed} \\ 0 & \text{otherwise} \end{cases}$$

For $(y, x) \in \mathbb{R}^{V\times E}$, $S \subseteq V$ and $T \subseteq E$, we define $y(S) = \sum_{v\in S} y_v$ and $x(T) = \sum_{e\in T} x_e$.

The OP goal is to determine a simple cycle that maximizes the sum of the scores of the visited vertices, such that it contains the depot node $o \in V$ and whose length is equal to or lower than the distance limitation, d_0. Then, the OP can be formulated as the following 0–1 Integer Linear model:

$$\max \quad \sum_{v\in V} s_v y_v \tag{2a}$$

$$\text{s.t.} \quad \sum_{e\in E} d_e x_e \leq d_0 \tag{2b}$$

$$x(\delta(v)) - 2y_v = 0, \qquad v \in V \tag{2c}$$

$$x(\delta(H)) - 2y_l - 2y_r \geq -2, \qquad l \in H \subseteq V, r \in V - H \tag{2d}$$

$$3 \leq |H| \leq |V| - 3$$
$$y_v - x_e \geq 0, \qquad v \in V, e \in \delta(v) \tag{2e}$$

$$0 \leq y_v \leq 1, \qquad v \in V \tag{2f}$$

$$0 \leq x_e \leq 1, \qquad e \in E \tag{2g}$$

$$y_o = 1 \tag{2h}$$

$$x_e \in \mathbb{Z} \qquad e \in E \tag{2i}$$

where the objective function (2a) is to maximize the total collected profit. The constraint (2b) limits the total cycle length. The degree Eq. (2c), together with the logical constraints (2e), the variable bound constraints (2f) and (2g) and the integrality constraints (2i), ensure that the visited vertices have exactly two incident edges and the unvisited vertices none. The Subcycle Elimination Constraints (SEC) (2d) ensure that only one connected cycle exists. Throughout the paper, we use the notation $\langle H, l, r\rangle$ for the SEC defined by the set $H \subseteq V$ and the vertices $l \in H$ and $r \notin H$. The constraints (2g) and (2i) impose that the edge variables are 0–1, consequently, considering these together with the Logical Constraints (2e) and the (2f), the vertex variables are also 0–1. The constraint (2h) defines the depot condition.

As mentioned in the introduction, the OP can be seen as a combination of the TSP-decision and the KP problems. Particularly, the OP is a Cycle Problem (CP) where the solutions, which are cycles, need to satisfy a certain length constraint. This relation with the two classical optimization problems is useful when identifying the valid inequalities and their respective separation algorithms for the OP. Let us show how the solution space of the OP is related to those well-known problems. The OP Polytope (P_{OP}) of the complete graph K_n is defined by:

$$P_{OP} := conv(\{(y, x) \in \mathbb{R}^{V\times E} : (y, x) \text{ satisfies (2b), (2c), (2d), (2e), (2f), (2g), (2h), (2i)}\}) \tag{3}$$

where $conv(S)$ is the convex hull of the set S, i.e., the smallest convex set that contains it.

The Knapsack Polytope (P_{KP}) (Balas, 1975) is a well-studied polytope closely related to the P_{OP}:

$$P_{KP} := conv(\{x \in \mathbb{R}^E : x \text{ satisfies (2b), (2g), (2i)}\}) \tag{4}$$

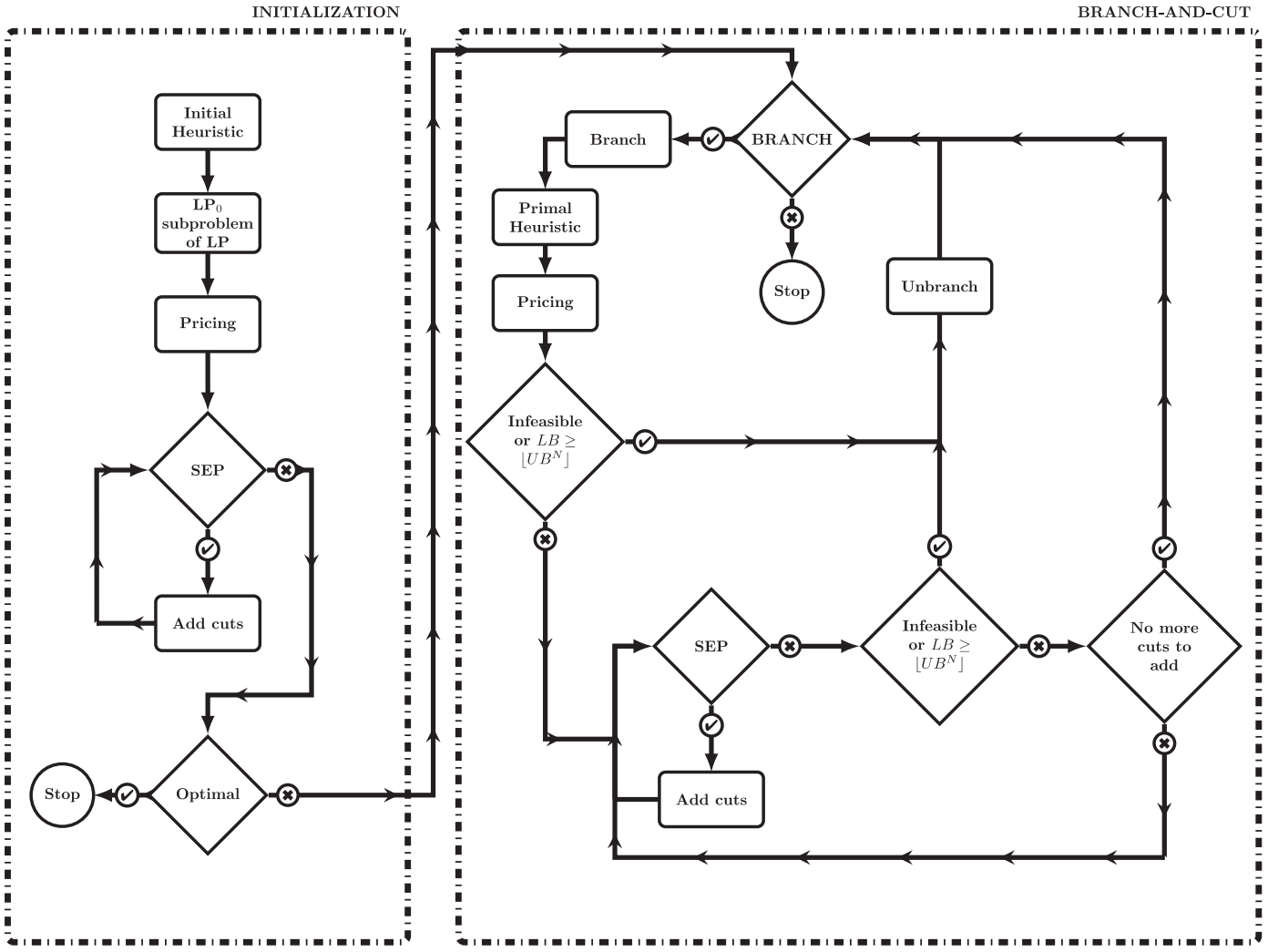

Fig. 2. Flowchart of the Branch-and-Cut algorithm considered in this work. BRANCH is an oracle which returns an unevaluated node in the branching tree. SEP refers to the separation algorithms. At each action box of the flowchart the subproblem LP_0 is updated and solved.

Since the solutions of the OP are cycles, the Cycle Polytope (P_{CP}), plays a crucial role when solving the OP with B&C. Based on Bauer (1997), the P_{CP} can be characterized as:

$$P_{CP} := conv(\{(y,x) \in \mathbb{R}^{V\times E} : (y,x) \text{ satisfies (2c), (2d), } x(E) \geq 3, \text{ (2e), (2f), (2g), (2i)}\}) \quad (5)$$

We have the following relationship:

$$P_{OP} \subseteq P_{CP} \cap (\mathbb{R}^V \times P_{KP}) \cap \{(y,x) \in \mathbb{R}^{V\times E} : y_o = 1\} \quad (6)$$

Consequently, the potential valid inequalities for the OP are those which are valid for P_{CP} and the P_{KP}. However, the P_{OP} and the intersected polytopes in the relationship (6) are not equal and alternative valid inequalities are needed to deal with the OP. An example of a point in $P_{CP} \cap (\mathbb{R}^V \times P_{KP}) \cap \{(y,x) : y_o = 1\}$ but not in P_{OP} is given in Fig. 2 of Fischetti et al. (1998).

3. Valid inequalities

In this section, we present valid inequalities for the *OP*. The straightforward inequalities, as motivated in Section 2, are based on the P_{KP} (Edge Cover inequalities) and P_{CP} (Comb inequalities) relaxations of the P_{OP} and they were mainly proposed in Fischetti et al. (1998) and Gendreau et al. (1998). Additional valid inequalities to those based on P_{KP} and P_{CP} have also been proposed in the literature: the Connectivity Constraints in Leifer & Rosenwein (1994), the Vertex Cover inequalities in Gendreau et al. (1998), and the Cycle Cover and the Path inequalities in Fischetti et al. (1998). The novelty of this section is an alternative representation of comb inequalities, which is then used for the efficient pricing in Section 4.3.

3.1. Connectivity constraints

The Connectivity Constraints (CC) are well-known inequalities for the OP (Gendreau et al., 1998; Leifer & Rosenwein, 1994), and are a particular case of the conditional cuts proposed in Fischetti et al. (1998). The CCs exploit the depot constraint (2h). Given a lower bound, LB, of the OP, let T be a subset of nodes such that $o \in T$, $|T| \geq 2$ and $\sum_{v\in T} s_v \leq LB$. The inequality defined by T

$$x(\delta(T)) \geq 2 \quad (7)$$

is valid for the OP. Since $x(\delta(T)) = x(\delta(V-T))$, the inequality can also be defined for $T \subseteq V$ such that $o \notin T$ and $\sum_{v\notin T} s_v \leq LB$. So, it is always possible to assume that $|T| \leq |V|/2$.

3.2. Comb inequalities

The comb inequalities were generalized from the TSP to cycle problems in Bauer (1997). A comb is a tuple $\langle H, \{T_1, \ldots, T_t\}, L, R\rangle$ of

three vertex subsets and a family $\mathcal{T} = \{T_1, \ldots, T_t\}$ of vertex subsets such that satisfies the following properties:

(i) $t \geq 3$ and an odd integer
(ii) $T_i \cap T_j = \emptyset$ for $1 \leq i < j \leq t$
(iii) $T_i \cap H \neq \emptyset$ and $T_i - H \neq \emptyset$ for $i = 1, \ldots, t$
(iv) $L = \{l_1, l_2, \ldots, l_t\}$, where $l_i \in T_i \cap H$ for all $1 \leq i \leq t$.
(v) $R = \{r_1, r_2, \ldots, r_t\}$, where $r_i \in T_i - H$ for all $1 \leq i \leq t$.

The set H is called the handle, the sets in $\mathcal{T}$ are called the teeth, the set R is called the *Root* set, and L is called the *Link* set. Then, the inequality

$$x(\delta(H)) + \sum_{j=1}^{t} x(\delta(T_j)) - 2y(R) - 2y(L) \geq 1 - t \tag{8}$$

is facet-defining for P_{CP}, as was shown in Bauer (1997), and therefore, a valid inequality for *OP*. When all the teeth consist of exactly two vertices, the comb inequalities are known as blossom inequalities.

Fig. 1 shows a violated comb inequality, where the vertex y^* and edge x^* values are detailed in the bottom legend. We represent a comb tuple with $t = 3$ teeth with exactly two vertices each, and the handle set H so that one of the vertices in each tooth T_i belongs to the intersection $T_i \cap H$, i.e., the Link set, while the other belongs to the difference $T_i - H$, i.e., the Root set.

3.3. Edge cover inequalities

The maximum length constraint (2b), which is a capacity constraint for the edge variables, defines a *KP* polytope, as explained in Section 2. For every feasible (y, x), the edge variable, x, belongs to P_{KP}. For the OP, the Edge Cover inequalities are the cover inequalities of the associated P_{KP} (Balas, 1975). These inequalities were first introduced for the OP in Leifer & Rosenwein (1994) and also used in Fischetti et al. (1998) and Gendreau et al. (1998). Let $F \subseteq E$ be a subset with $\sum_{e \in F} d_e > d_0$, then:

$$x(F) \leq |F| - 1 \tag{9}$$

defines an Edge Cover inequality for the OP. We assume that F is a minimal cover, i.e. for every $F_0 \subset F$, we have $\sum_{e \in F_0} d_e \leq d_0$.

3.4. Cycle cover inequalities

Every feasible cycle $F \subseteq E$ satisfies the equation $x(F) = y(V(F))$. Let $F \subseteq E$ be a subset that defines a cycle with $\sum_{e \in F} d_e > d_0$, then the inequality

$$x(F) \leq y(V(F)) - 1 \tag{10}$$

is valid for the OP. These cuts were used in Fischetti et al. (1998) and Gendreau et al. (1998).

3.5. Vertex cover inequalities

Let UB be an upper bound of the OP and $Q \subseteq V$ be a subset with $\sum_{v \in Q} s_v > UB$, then:

$$y(Q) \leq |Q| - 1 \tag{11}$$

defines a Vertex Cover inequality for the OP. We assume that S is a minimal cover. These inequalities were first used for the OP in Gendreau et al. (1998).

3.6. Path inequalities

The goal of these cuts is to exclude the paths that due to the length constraint (2b) cannot be part of a feasible solution. Let $P = \{[i_1, i_2], [i_2, i_3], \ldots, [i_{k-1}, i_k]\}$ be any simple path through $V(P) = \{i_1, \ldots, i_k\} \subseteq V - \{o\}$, and define the vertex set:

$$W(P) := \left\{ v \in V - V(P) : d_{o,i_o} + \sum_{e \in P} d_e + d_{i_k,v} + d_{v,o} \leq d_0 \right\} \tag{12}$$

Then the following Path inequality

$$x(P) - y(V(P)) + y_{i,o} + y_{i,k} - \sum_{v \in W(P)} x_{i_k,v} \leq 0 \tag{13}$$

is valid for the OP, see the explanation in Fischetti et al. (1998).

4. Branch-and-cut algorithm

In this section, we present the principal contributions of this paper. These contributions deal with the separation algorithms of inequalities stemming from the cycle problem (SECs and comb inequalities), the design of the separation loop, the pricing of variables for the column generation and the calculation of the lower and upper bounds of the problem. In Fig. 2 a flowchart representing a simplified B&C algorithm can be consulted.

4.1. Initialization

First of all, we obtain an initial heuristic solution. To that aim, we make use of the EA4OP metaheuristic in Kobeaga et al. (2018) considering a small size population. Next, we build the initial subproblem, LP_0. Given the computational requirements of considering all the variables and constraints that define the OP, an initial subproblem LP_0 is built. The LP_0 is initialized considering the following subset of constraints and variables:

(i) All the vertex variables.
(ii) Edges in the 10 nearest neighbourhood graph, that is, the graph created by keeping the edges of the 10 nearest neighbours of each node and removing the rest.
(iii) Maximum length constraint (2b), degree constraints (2c), and depot constraint (2h).
(iv) Variable bounds, (2f) and (2g).

Due to the large number of subtour elimination constraints and logical constraints, we consider the relaxation of constraints (2d) and (2e) as explained in Section 4.2, and the integrality constraints (2i) are also relaxed. Immediately after the initialization, the edge variables are priced, see Section 4.3. In the rest of the paper, we use the LP_0 symbol to refer to any subproblem of the OP, regardless of whether it is the initial one or not.

4.2. Separation algorithms

In this section, we present the heuristic and exact separation algorithms used to find the violated inequalities. Our contributions are concentrated in the separation algorithms for SECs, CCs and blossom inequalities. Hence, we only give details of these separation algorithms in the section. The details of separation algorithms for the rest of the inequalities (Logical Constraints, Edge Cover, Vertex Cover, Cycle Cover, and Path inequalities) can be found in Fischetti et al. (1998). Let (y^*, x^*) be a solution of a particular LP_0 problem and define $V^* = \{v \in V : y^*_v > 0\}$ and $E^* = \{e \in E : x^*_e > 0\}$. Then, $G^* = (V^*, E^*)$ is called the support graph associated with the solution (y^*, x^*).

4.2.1. SECs and CCs

Violated SECs (2d) and CCs (7) are found using a common separation algorithm. This is natural since, in both constraint families, the incidence vector of the arcs, x in the inequality can be written as the star-set value, $x(\delta(Q))$ of a subset Q of vertices. Since $\delta(Q)$

is the cut associated with Q, the separations of both inequalities are closely related to the minimum cut problem. In Kobeaga et al. (2021) it was shown that the shrinking techniques substantially speed up the SEC separation algorithms. However, as explained below, the shrinking might also have a negative impact on the finding of violated CCs. In this section, we study how to efficiently use the shrinking to speed up the joint separation algorithm by reducing the adverse effects for CCs.

A violated SEC and a violated CC can appear simultaneously when considering a solution (y^*, x^*) and a subset Q. In general, the CCs tend to be more violated and stable than the SECs in the sense that they remain active in subsequent updates of the LP_0. The reason for this is that the CCs do not depend on the value of the vertices while the SECs do. Therefore, we treat the CCs with a higher priority. Although SECs are part of the OP model, in order to control the size of the working LP_0, they are included only when required. This strategy is reasonable since there exist polynomial exact separation algorithms for SECs. In contrast, the separation problem for CCs is not known to be polynomial, and it can be modeled as follows:

$$\min \quad 2\sum_{v\in V^*} y^*_v z_v - 2\sum_{v\in V^*} x^*_{(v,u)} z_v z_u \tag{14a}$$

$$s.t: \quad \sum_{v\in S} s_v z_v \le LB \tag{14b}$$

$$z_o = 1 \tag{14c}$$

$$z_v \in \{0,1\} \qquad \forall v \in V \tag{14d}$$

where $z = (z_v)$ are binary variables whose values are $z_v = 1$ if the node v is selected and 0 otherwise. The problem (14) is a Quadratic Knapsack Problem (QKP) with a fixed variable. Consequently, there exists a violated CC for (y^*, x^*) if and only if the optimal solution of Problem (14) has a value lower than 2. Taking into consideration that repeatedly solving QKPs during the B&C is not viable, the CCs are not separated exactly, but in a heuristic manner take advantage of the SEC separation algorithm. The well-known approaches for the separation of SECs in the TSP, the connected component heuristic and Hong's approach can be extended to jointly separate the SECs and CCs:

Connected components heuristic. The straightforward heuristic to find violated SECs and CCs is to search for the connected components of G^* using the depth-first-search algorithm. When a connected component contains the depot vertex o and the sum of the vertices scores in the component is lower than LB, we record the associated CC of the component, otherwise, we record the associated SECs. On the other hand, if the S component does not contain the depot, we consider its complement in the support graph, $S' = G^* - S$. If S' contains only two vertices, we record the two associated logical constraints. Otherwise, since S' contains the depot, we record the associated CC or SECs following the above criteria.

Extended Hong's approach. There are two main strategies to exactly separate SEC inequalities in cycle problems, which are extensions of Hong's approach and the Padberg–Grötschel approach (also known as the Gomory–Hu tree-based approach) for the TSP (Kobeaga et al., 2021). In both approaches, the separation is carried out by solving a sequence of $|V^*| - 1$ (s,t)-minimum cut problems. On the one hand, in the extended Hong's approach, the vertex with a higher y^* value (the depot vertex o) is fixed to be the source, s, and the sink vertices, t, are chosen from the set $V^* - \{o\}$. On the other hand, the extended Padberg–Grötschel approach is based on the so-called Gomory–Hu tree (directed and rooted in o), which is constructed by solving $|V^*| - 1$ (s,t)-minimum cut problems.

As mentioned above, and as already proposed in the literature, the SEC separation strategies are leveraged to find violated CCs as well. Although the extended Padberg–Grötschel approach obtains a larger number of violated SECs, it is not appropriate to find violated CCs, since the obtained sets do not contain the depot vertex o. Contrarily, the extended Hong's approach for SECs can be easily adapted to additionally find violated CCs. It can be achieved, by solving at each step of the separation algorithm the (o,v)-minimum cut (useful to find violated SECs) and (v,o)-minimum cut (useful to find violated CCs) problems. For these reasons, we use the extended Hong's approach as the base strategy for the joint separation algorithm.

The running time of these SEC separation algorithms can be improved using the shrinking techniques for cycle problems, as was seen in Kobeaga et al. (2021). In that publication, three general shrinking rules (C1, C2, and C3) and three SEC specific shrinking rules (S1, S2, and S3) for cycle problems were presented. However, although the shrinking is a key strategy for efficiently separating the SECs, it might be unfavorable for the separation of CCs. The point is that when the vertices are contracted and grouped, the chance to obtain the subset of vertices with a score sum lower than LB decreases, consequently, some violated CCs might vanish. Note that, the mentioned shrinking techniques are safe for valid inequalities of the cycle polytope and CCs are not. Therefore, since CCs are important cuts for the OP, shrinking might have a negative impact on the performance of the overall B&C algorithm for the OP. One contribution in this paper is to propose strategies to minimize the possible disadvantages of the shrinking (which is important to speed up the separation) in the joint separation algorithm for SECs and CCs.

Following this, not all the shrinking strategies for cycle problems described in Kobeaga et al. (2021) are adequate for the OP problem. Particularly, we exclude the S2 shrinking rule (which leads to excessively aggressive shrinking strategies and hence to vanish violated CCs in some cases) and only consider the shrinking strategy S1 (or alternatively, the strategies C1C2) in the preprocess of the joint separation algorithm. Once entered in the separation algorithm, the shrinking rule S3, which contracts the sink and target of the solved minimum cut, contributes positively to separate both families of constraints since it enables a wider family of subset candidates to be obtained. Hence, the S3 rule is used in combination with the S1 (or alternatively, C1C2) shrinking strategy in the separation algorithm. After the S3 rule is applied, we search for new shrinkable sets using the selected shrinking strategy.

Classically, the candidate subsets for SECs and CCs are obtained by the minimum cut algorithm. However, considering the importance of CCs, we intensify the search for extra candidate subsets for CCs, which is made more efficient by taking advantage of the vertex clustering obtained by the shrinking. We propose new strategies based on the following lemma:

Lemma 4.1. *Let (y,x) be a vector that satisfies the degree constraints. If U and W are subsets of V such that $W \subseteq U$, the following inequality is satisfied:*

$$x(\delta(U - W)) \le x(\delta(U)) + x(\delta(W)) \tag{15}$$

Proof. It is straightforward since the following holds when $W \subseteq U$:

$$x(\delta(U)) + x(\delta(W)) = x(\delta(U - W)) + 2x(\delta(U) \cap \delta(W)) \quad \square$$

We use the following notation for shrinking. Let $\bar{G} = (\bar{V}, \bar{E})$ be the graph and $(\bar{x}, \bar{y})$ the vector obtained by applying a shrinking strategy to G^* and (y^*, x^*), respectively, and $\pi : \mathcal{P}(\bar{V}) \to \mathcal{P}(V)$ the unshrinking function. Let $\bar{S}$ be the subset obtained by the $(\bar{v}, \bar{o})$-minimum cut (where $\bar{o}$ is the contracted vertex that contains the

depot vertex o), so $o \in \pi(\bar{S})$, and suppose that $\bar{x}(\delta(\bar{S})) < 2$. Note that, $x(\delta(S)) = \bar{x}(\delta(\bar{S}))$, where $S = \pi(\bar{S})$. If $\sum_{v \in S} us_v \leq LB$, the subset S defines a violated CC. Otherwise, after each $(\bar{v}, \bar{o})$-minimum cut problem is solved, and in the case that $\bar{x}(\delta(\bar{S})) < 2$, we test the following strategies to find candidate subsets for CCs:

(i) First, when $|\pi(\bar{o})| > 2$, we check if $y_{\bar{o}} < 1$ and $\sum_{v \in \pi(\bar{o})} s_v \leq LB$. If this is the case, the subset $Q = \pi(\bar{o})$ defines a violated CC.

(ii) Then, we check if there exists $\bar{v} \in \bar{S} - \bar{o}$, such that $x(\delta(\bar{S})) + 2y_{\bar{v}} < 2$ and $\sum_{v \in \pi(\bar{S}-\bar{v})} s_v \leq LB$. If both inequalities are satisfied for $\bar{v}$, the subset $\pi(\bar{S} - \bar{v})$ defines a violated CC.

(iii) Finally, we sort the vertices in $\bar{S} - \bar{o}$ in non-decreasing order of $\bar{y}$, and check greedily for the greatest subset $S' = \{\bar{v}_1, \ldots, \bar{v}_k\}$ of $\bar{S}$ such that $\bar{x}(\delta(\bar{S})) + 2\sum_{v \in S'} y_{\bar{v}} < 2$. If $\sum_{\bar{v} \in \pi(\bar{S}-S')} s_v \leq LB$, the subset $\pi(\bar{S} - S')$ defines a violated CC.

4.2.2. Comb Inequalities (blossoms)

For the B&C presented in this work, we only use the blossom subfamily of comb inequalities. In this section, we present two heuristics to search for violated blossom inequalities in cycle problems, and in particular, for the OP. The heuristics are extensions of the Padberg & Hong (1980) and Grötschel & Holland (1991) separation algorithms, developed in the context of the TSP.

The key point of the heuristics for blossom inequalities is to identify a subset of candidate handles to restrict the search of violated blossoms. In the literature related to the OP, a heuristic to find handle candidates is detailed in Fischetti et al. (1998). In this heuristic, the search is guided by the greedy algorithm of Kruskal for the Minimum Spanning Tree. At each iteration of the Kruskal algorithm, a new edge is inserted into the tree, and the connected component containing the edge is chosen as a candidate handle. In this work, we consider two alternative approaches to find candidate handles: the Extended Padberg–Hong heuristic and the Extended Grötschel–Holland heuristic.

Extended Padberg–Hong heuristic (EPH). Padberg & Hong (1980) proposed a blossom separation heuristic for the TSP, which is known as the odd-component heuristic. In this heuristic for the TSP, the violated blossoms are found by restricting the set of candidate handles to the connected components of the fractional graph $G_o^* = (V_o^*, E_o^*)$, where $E_o^* = \{e \in E^* : 0 < x_e^* < 1\}$ and $V_o^* = V(E_o^*)$. We generalize this heuristic for the general cycle problems by applying the Padberg–Hong algorithm by levels. A level, λ, is defined by each different value of the set $\{y_v^*\}_v$. We call L the set of distinct levels. Note that, the number of levels, $|L|$, is bounded by $|V|$. Associated with a level we have the level graph $G_\lambda^* = (V_\lambda^*, E_\lambda^*)$, where $E_\lambda^* = \{e \in E^* : 0 < x_e^* < \lambda\}$ and $V_\lambda^* = V(E_\lambda^*)$. A faster heuristic to find the handle candidates can be designed by omitting some connected components of G_λ^*. At every level, λ, we discard the connected components, C_i^λ, such that $y_v \neq \lambda$ for all $v \in C_i^\lambda$. Now, we identify the connected component of vertices with $y_v = \lambda$. So, in total, we search for $|V^*|$ different connected components of, in the worst case, G_o^*.

Once we have identified an initial list of candidate handles, the next step is to find the associated teeth for these handles. Let H be a candidate handle, and define the set of teeth as $\mathcal{T}_H = \{e \in \delta(H) : x_e^* \geq \lambda\}$. Recall that the teeth of blossoms are edges. Not all the teeth families obtained using this strategy satisfy the comb (blossom) definition. If two teeth overlap in $v \notin H$, then these two teeth are removed from the family of teeth $\mathcal{T}_H$ and the handle is updated as $H = H \cup \{v\}$. If, eventually, the list of teeth $\mathcal{T}_H$ consists of an odd number of at least three disjoint teeth, $\langle H, \mathcal{T}_H, L, R\rangle$ forms a blossom inequality where $L_i = T_i^j \cap H$ and $R_i = T_i^j - H$. If there is just one tooth i.e., $\mathcal{T}_H = \{T\}$, we test if H defines a violated CC. In the case that it does not, then H alone defines a violated SEC.

In Fig. 3 we illustrate the EPH blossom heuristic for cycle problems. In Fig. 3(a)) the given support graph is presented, where there are three distinct levels, $L = \{1, 1/2, 1/4\}$. In Fig. 3(b)) the candidate handles are presented. Three candidate handles are obtained in level 1: $\{1, 2, 3\}$, $\{5, 6, 7\}$ and $\{10, 11, 12, 13, 14, 15, 16\}$. Two candidate handles are obtained in level 1/2: $\{10, 11, 12\}$ and $\{14, 15, 16\}$. There are no candidate handles obtained in level 1/4. Next, we check for violated cuts. The star-set of $\{10, 11, 12, 13, 14, 15, 16\}$ is formed by two non-overlapping edges, so it is excluded. The candidates $\{5, 6, 7\}$ and $\{10, 11, 12\}$ define violated blossoms, e.g., $\langle\{\{10, 11, 12\}, \{\{8, 10\}, \{9, 11\}, \{12, 13\}\}, L, R\rangle$ where $L = \{10, 11, 12\}$ and $R = \{8, 9, 13\}$ as shown in Fig. 3(c). The candidates $\{1, 2, 3\}$ and $\{14, 15, 16\}$ define violated SECs, e.g. $\langle\{1, 2, 3\}, 1, 4\rangle$ and $\langle\{14, 15, 16\}, 14, 1\rangle$, but first for $\{1, 2, 3\}$ it should be checked whether it defines a violated CC.

Extended Grötschel–Holland heuristic (EGH). Another fast heuristic for the TSP was proposed in Grötschel & Holland (1991) whose aim was to minimize the influence of small perturbations of x^* in the separation algorithm. We have adapted this heuristic for the OP using the strategy of levels mentioned above. In this approach, the handles are considered as the vertex sets of the connected components of the graph $G_{\lambda,\epsilon}^* = (V^*, E_{\lambda,\epsilon}^*)$ where

$$E_{\lambda,\epsilon}^* = \left\{e \in E_\lambda^* : \epsilon \leq x_e^* \leq (1-\epsilon)\lambda\right\}$$

for a small ϵ, $0 < \epsilon < 1$. Let H denote the vertex set of such a component, a candidate handle, and let $e_1, \ldots, e_t$ be the edges in the set

$$\mathcal{T}_H = \left\{e \in \delta(H) \subseteq E^* : x_e^* > (1-\epsilon)\lambda\right\}$$

in the non-increasing order of x_e^*. If t is even, then append to $\mathcal{T}_H$ the edge with the highest x_e^* in

$$\left\{e \in \delta(H) \subseteq E^* : x_e^* < \epsilon\right\}$$

If the edges intersect, the strategy outlined above is followed to obtain a handle H and a teeth family $\mathcal{T}_H$ that satisfies the blossom definition.

4.3. Column generation

During the B&C algorithm, only a subset of edges is included in the working LP_0. Therefore, we need to price the excluded edge variables and add them to the LP_0 with two goals in mind: 1) to guarantee that the working relaxation is an upper bound of the problem or branched subproblem and 2) to recover, if possible, a feasible LP_0 after feasibility breaking cuts have been added to the LP_0. Taking into account that usually only a small subset of variables is included in the LP_0, and that the excluded variables could participate in multiple cuts of the LP_0, the pricing phase could constitute a bottleneck of the B&C algorithm. In this section, we develop a technique, inspired by that used in Applegate et al. (2007), which enables us to avoid repetitive calculations and to skip the exact calculation of the reduced cost of some variables.

Let us call $\mathcal{L}^V$ the family of SECs (2d), CC (7), and comb (8) cuts. In these cuts, the edge variables with non-negative coefficients can be represented as the sum star-set of subsets of vertices. Complementarily, let us call $\mathcal{L}^E$ the family of Logical (2e), Edge Cover (9), Cycle Cover (10) and Path (13) cuts. Note that the Vertex Cover (11) inequalities do not contribute to the reduced cost of the edge variables. So, in the OP, the reduced cost of an edge variable, $e = [v, w]$, can be calculated by:

$$rc_e = -d_e\pi_{d_0} - \pi_v - \pi_w + rc_e^V + rc_e^E \tag{16}$$

where π_{d_0} is the dual variable of the maximum length constraint (2b), π_v and π_w are the dual variables of the degree constraints (2c) of v and w respectively, and rc_e^V and rc_e^E are the contributions made by the cuts in $\mathcal{L}^V$ and $\mathcal{L}^E$, respectively. We will see that the

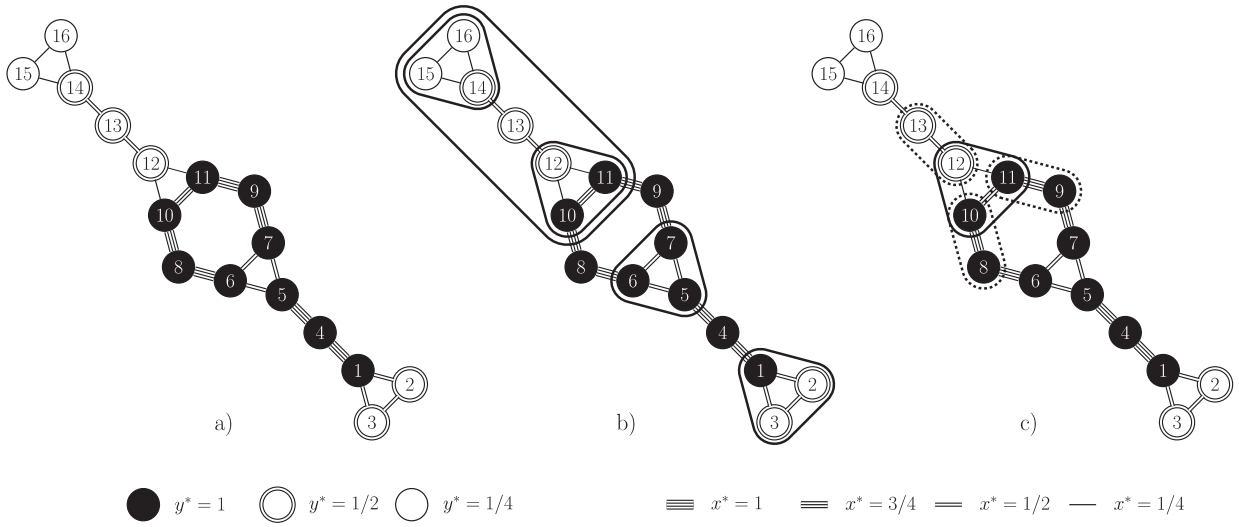

Fig. 3. Illustration for the Extended Padberg–Hong blossom heuristic. Figure a) represents the support graph, with the vertex and edge values detailed in the bottom legend. Figure b) shows all the handle candidates obtained by the heuristic. Figure c) a violated blossom found by the heuristic involving vertices with different y values.

rc_e^E values can be obtained in linear time in terms of $|V|$ and $|\mathcal{L}^E|$, and we will reproduce the pricing strategy used in Applegate et al. (2007) to calculate the rc_e^V values. It can be seen that the cost of the calculation of all the rc_e^E is $O(|\mathcal{L}^E||V|)$. To that aim, it is sufficient to check that the number of edges with a non-negative coefficient in each cut of $\mathcal{L}^E$ is bounded by $|V|$. In the case of Logical, Cycle Cover, and Path inequalities, it is derived from the definition of the valid inequality. For Edge Cover inequalities, this bound is obtained in Lemma 4.2.

Lemma 4.2. *Let $T \subseteq E$ denote a subset defining a violated edge cover inequality. If the degree Eq. (2c) are satisfied by $(y, x) \in \mathbb{R}^{V \times E}$ then $|T| \leq |V|$.*

Proof. When the degree constraints are satisfied by (y, x), as a consequence of the well-known equality $x(\delta(S)) = 2y(S) - 2x(E(S))$, the inequality $x(E(V(T))) \leq y(V(T))$ is always satisfied. Suppose that T violates the cover inequality (9) then

$$|T| - 1 < x(T) \leq x(E(V(T))) \leq y(V(T)) \leq |V| \tag{17}$$

□

Calculating all the rc_e^V values has a $O(|\mathcal{L}^V||V|^2)$ complexity when the cuts are stored externally as edge variable coefficient arrays. The strategy used in Applegate et al. (2007) speeds up the pricing by obtaining a fast lower bound of the reduced cost rc_e^V (TSP is a minimization problem) and excluding for exact pricing the edges that have a negative lower bound. In order to use this strategy for the OP, first, the edge variables of the cuts in $\mathcal{L}^V$ must be represented and stored as a family of subsets of vertices, as we have done in Section 3. Let $\mathcal{S} = \mathcal{F}_1 \cup \ldots \cup \mathcal{F}_r$ be the family of all the subsets involved in the cuts of $\mathcal{L}^V$ where $\mathcal{F}_i = \{H_i\} \cup \mathcal{T}_i$. For combs, H_i and $\mathcal{T}_i$ represent the handle and teeth set, respectively. For SECs and CCs we can assume that $\mathcal{T}_i = \emptyset$ and $H_i = \emptyset$, respectively.

Based on the representation of the cuts in $\mathcal{L}^V$ by means of subsets of vertices, the cuts are stored in an efficient data structure by pointing to the subsets involved in the cut. This way each subset is saved once at most for all the cuts. Moreover, it allows us to speed up the evaluation of rc_e^V values as follows. Since the OP is a maximization problem, during the pricing, we need to identify the edge with positive reduced cost. We aim to define upper bounds, $\hat{rc}_e$, of the reduced costs rc_e, to exclude for exactly pricing the edges that have a non-positive upper bound $\hat{rc}_e^V$. For each subset, $S \in \mathcal{S}$, let us call π_S the dual of the subset S defined as:

$$\pi_S = \sum_{j=1}^{r} \chi_j(S)\pi_j \tag{18}$$

where $\chi_j(S) = 1$ if $S \in \mathcal{F}_j$ and 0 otherwise, and π_j is the dual variable associated with the cut j. Then, the contribution of the cuts in $\mathcal{L}^V$ in the reduced cost of an edge e can be written as:

$$rc_e^V = \sum_{\substack{S \in \mathcal{F} \\ V(e) \cap S \neq \emptyset \\ V(e) - S \neq \emptyset}} \pi_S \tag{19}$$

where π_S is the dual of a subset S. Since, for the edge $e = [v, w]$, each S must contain either v or w, an upper bound, $\hat{rc}_e^V$, of rc_e^V can be obtained by:

$$\hat{rc}_e^V = \sum_{\substack{S \in \mathcal{F} \\ v \in S}} \pi_S + \sum_{\substack{S \in \mathcal{F} \\ w \in S}} \pi_S$$

which satisfies $rc_e^V \leq \hat{rc}_e^V$. Therefore, we have the desired upper bound:

$$\hat{rc}_e = -d_e\pi_{n+1} - \pi_v - \pi_w + rc_e^E + \hat{rc}_e^V \tag{20}$$

Note that, each edge appears at most twice in a comb inequality, so the calculation of all the $\hat{rc}_e^V$ has a $O(M|\mathcal{L}^V||V|)$ time complexity where M is the maximum number of subsets involved in a cut. Therefore, the calculation of all the $\hat{rc}_e$ has a $O(M|\mathcal{L}^V||V|)$ time complexity. In our B&C, the value of M is related to the number of teeth in the combs. To ensure a true linear time complexity procedure, one could limit the number of teeth in the combs. However, in practice, the number of teeth tends to be small and it can be assumed that $M << |V|$. We can exclude exactly pricing the edges that $\hat{rc}_e \leq 0$. For those edges that $\hat{rc}_e > 0$, the exact reduced cost, rc_e, can be calculated by using the upper bound value:

$$rc_e = \hat{rc}_e - 2 \sum_{\substack{S \in \mathcal{F} \\ V(e) \subseteq S}} \pi_S \tag{21}$$

The pricing loop is done in batches. In the first step, a fixed number of $\hat{rc}_e$ are calculated, the first batch of variables and those with positive values are preselected. In the next step, for those preselected variables, we calculate the exact reduced cost, rc_e, and add to the LP_0 the edges whose value is positive. Then, the LP_0 is updated. Next, we select the second batch of variables and we

repeat the procedure. When the pricing aims to obtain the upper bound of the branched subproblem, we exit the pricing loop when a whole round of evaluation is performed without introducing a variable to the LP_0. When the pricing aims to recover a feasible LP_0, we exit the pricing loop once a feasible LP_0 is obtained without the need to price all the excluded variables.

4.4. Separation loop

The separation loop to find the violated cuts is accomplished in three subloops. In the inner loop, we consider the separation of logical constraints (2e) and the connected components heuristic for SECs and CCs. In the middle loop, we consider the separations of cuts which are related to the cycle essence of the OP, i.e., SECs, CCs, blossoms, and Cycle Cover cuts. In the outer loop, we consider the rest of the cuts, i.e., the Edge Cover, Vertex Cover, and the Path inequalities. The separation loop is illustrated in Fig. 4.

At each subloop, the separation of the considered cuts is performed sequentially, instead of restarting from the beginning of the list. This is, we always carry out the next separation in the subloop list, regardless of whether or not we are coming from an interior subloop. This way, we give the same chance to all separations in a subloop and decrease the probability of bounding in the same separation algorithm in consecutive iterations of the subloop.

The separation algorithms of the inner loop are fast since both have a $O(|E^*|)$ time complexity. First, we carry out the connected components heuristic and then the separation of logical constraints. In the inner loop, intending to keep it as a fast loop, we price the edge variables only when the floor part of the objective value is equal to the lower bound of the OP, i.e., if $\lfloor s \cdot y^* \rfloor = LB$. When both separations fail and no new edges have been added, we find a feasible solution using the PB primal heuristic (see Section 4.5) and update the LB if needed. We add the associated CC of the heuristic solution if it is violated and then we price the variables. When a new CC cut or a priced edge has been added to the LP_0, the inner loop is repeated. Otherwise, we return to the middle loop.

The middle and outer loops only differ in the considered constraint families. In the middle loop, we consider the separation algorithms in the following order: the extended Padberg–Hong algorithm for blossom, the extended Grötschel-Holland algorithm for blossom, the joint SEC/CC separation algorithm, and Cycle Cover separation algorithm. In the outer loop, we consider the Edge Cover algorithm, the Vertex Cover algorithm, the Path algorithm. When we enter in any of the loops, the first step is to execute the lower level subloops. Then, we start with the first algorithm on the list. If no violated cuts are found we move on to the next algorithm. If violated cuts are found, we first add the cuts and optimize the LP_0. Then, we search for a feasible solution using a primal heuristic and update the LB if needed. We add the associated CC of the heuristic solution in case it is violated and then we price the variables. At this point, we move to the lower level loop and continue with the next separation in the list.

In the separation loop, after adding the violated cuts found in a separation algorithm, we check if any edge variable or constraint can be removed from the LP_0. We remove an edge variable from the LP_0 if, during a number of consecutive evaluations, its associated value, x_e^*, has been zero. We remove a constraint from the LP_0 if during a number of consecutive evaluations its slack has been higher than zero.

4.5. Primal heuristics and lower bounds

We use two primal heuristics to obtain feasible solutions from a fractional solution (y^*, x^*). In the first heuristic, we obtain a single solution, by using the x^* values related to edges, inspired by the heuristic proposed in Fischetti et al. (1998). In the second heuristic, first, we build a population of cycles and then evolve it using the EA4OP metaheuristic (Kobeaga et al., 2018). The cycles in the population are constructed by selecting first the subset of vertices in each cycle using the y^* values.

Path Building primal heuristic (PB). The PB heuristic was presented in Fischetti et al. (1998). First, the edges $e \in E^*$ are sorted in decreasing order of x_e^*, and the ties are randomly broken. The procedure starts with an empty path $T = \emptyset$. At each step we select an edge $e \in E^*$ whose x_e^* has the largest value from the set of edges which have not been considered yet. If the inclusion of e in T does not lead to a vertex with a degree larger than 2, then $T = T \cup \{e\}$ otherwise we exclude e and repeat the process. The path building heuristic finishes when the inclusion of e leads to T being a cycle or when there are no edges left to check. If the depot vertex is not in one of the paths in T, it is included as a single point path. If T consists of multiple paths, we extend it to a cycle by randomly connecting the extreme vertices (in the original paper the paths were joined using the nearest neighbor heuristic). Since this primal heuristic is fast, it is used in the separation loop.

Vertex picking primal heuristic (VP) with the EA4OP metaheuristic. In the VP heuristic, we first select a collection of vertices in V^* and then build a random cycle through the selected vertices. Each vertex v is selected according to a Bernouilli distribution with parameter y_v^*. By applying multiple times the VP strategy to obtain feasible solutions from (y^*, x^*), we build a small population. Once we have a population, we ensure that the solutions in the population are feasible and evolve it using the EA4OP metaheuristic proposed by Kobeaga et al. (2018). The EA4OP with VP heuristic is used to find feasible solutions after an edge is branched, as shown in Fig. 2.

We improve the route lengths of the solutions obtained by the PB and VP heuristics using the Lin-Kernighan heuristic for the TSP. Afterwards, we check if the resulting solutions satisfy the constraint (2b). If it does not, we apply the drop operator which consists in deleting vertices from the solution until the cycle satisfies the length constraint. Then we try to improve the solution by the k-d tree based vertex inclusion procedure as explained in Kobeaga et al. (2018).

4.6. Branching and upper bounds

The branching is carried out in a classical way following a depth-first-search, where the edges are branched first to 1 and then to 0. In order to select the edge variable to branch, we use the classical branching strategy: the edge e, with the fractional value closest to 0.5 is selected, i.e., the edge that minimizes $|x_e^* - 0.5|$.

The global upper bound and branch node upper bound are calculated just before pruning a branch. The branch node upper bound, UB^N, is used to verify the pruning, i.e, that $LB \geq \lfloor UB^N \rfloor$. The global upper bound is calculated with two aims: firstly, to use it in Vertex Cover separation, and secondly, to compute the optimality gap when the algorithm finishes due to time limitations. The global upper bound, UB^G of the OP, is obtained using the dual solution π^* of the solution (y^*, x^*) of the LP_0:

$$UB^G = \sum_{i=1}^{c} \pi_i^* b_i + rc_o^* + \sum_{\substack{v \in V - \{o\} \\ rc_v^* > 0}} rc_v^* + \sum_{\substack{e \in E \\ rc_e^* > 0}} rc_e^* \tag{22}$$

where the reduced costs rc_v^* and rc_e^* are calculated using the dual variables π^* and c is the number of constraints. The upper bound of a branch node, UB^N, can be calculated by subtracting the contributions of the branched edges to UB^G. Let $B_0, B_1 \subseteq E$ be the subset of edges branched to 0 and 1, respectively. Then, we obtain UB^N

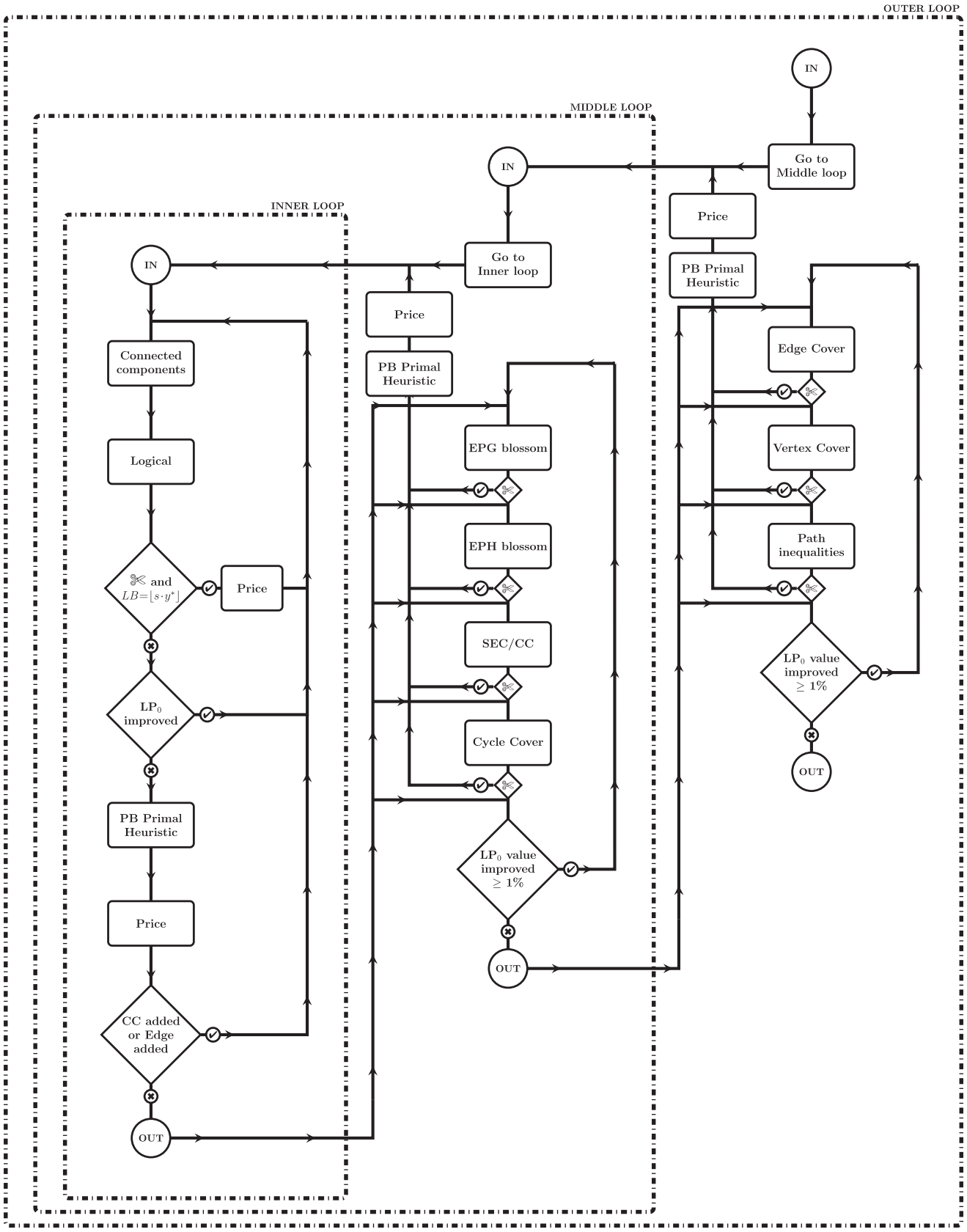

Fig. 4. Illustration of the separation loop. The symbol ✂ represents that some cuts have been added to the LP_0.

Table 2
Generations for instances based on TSPLIB.

Generation	Score for the ith node, $i \in [n]$	α	# medium $n \leq 400$	# large $n > 400$
Gen1	1	0.5	45	41
Gen2	$1 + (7141 \cdot (i-1) + 73) \mod 100$	0.5	45	41
Gen3	$1 + \lfloor 99 \cdot d_{o,i} / \max_{j \in [n]} d_{o,j} \rfloor$	0.5	45	41

Table 3
Different RB&C configurations. The first column represents the name of the configurations and each column on the right represents a different component of the algorithm that is included or not.

	SEC/CC					Blossom			Literature				Loop		B. HEUR		
	C1	C2	S1	S3	CC	EGH	EPH	FST	CL	VX	EE	PH	2	3	PB	VP	EA
Reference	✗	✗	√	√	√	√	√	✗	√	✗	√	√	✗	√	✗	√	√
-SRK	✗	✗	✗	✗	√	√	√	✗	√	✗	√	√	✗	√	✗	√	√
SRK=C1C2S3	√	√	✗	√	√	√	√	✗	√	✗	√	√	✗	√	✗	√	√
-CC STRATS	✗	✗	√	√	✗	√	√	✗	√	✗	√	√	✗	√	✗	√	√
-EPH	✗	✗	✗	✗	√	√	✗	✗	√	✗	√	√	✗	√	✗	√	√
-EGH	✗	✗	✗	✗	√	√	√	✗	√	✗	√	√	✗	√	✗	√	√
+FST	✗	✗	✗	✗	√	√	√	√	√	✗	√	√	✗	√	✗	√	√
EPH EGH=FST	✗	✗	√	√	√	✗	✗	√	√	✗	√	√	✗	√	✗	√	√
-CYCLE COVER	✗	✗	√	√	√	√	√	✗	✗	✗	√	√	✗	√	✗	√	√
-EDGE COVER	✗	✗	√	√	√	√	√	✗	√	✗	✗	√	✗	√	✗	√	√
-PATH	✗	✗	√	√	√	√	√	✗	√	√	√	√	✗	√	✗	√	√
+VERTEX COVER	✗	✗	√	√	√	√	√	✗	√	✗	√	✗	✗	√	✗	√	√
SEP=TWO	✗	✗	√	√	√	√	√	✗	√	✗	√	√	√	✗	✗	√	√
B. HEUR=PB	✗	✗	√	√	√	√	√	✗	√	✗	√	√	✗	√	√	✗	✗
B. HEUR=VP	✗	✗	√	√	√	√	√	✗	√	✗	√	√	✗	√	✗	√	✗

SEC/CC (separation algorithms in Section 4.2.1); Blossom (separation algorithms in Section 4.2.2); Literature (separation algorithms from the literature: CL-CYCLE, VX-VERTEX, EE-EDGE, PH-PATH); Loop (loop separation strategy in Section 4.4); B. HEUR (branch heuristic in Section 4.5).

by:

$$UB^N = \sum_{i=1}^{c} \pi_i^* b_i + rc_o^* + \sum_{\substack{v \in V-\{o\} \\ rc_v^* > 0}} rc_v^* + \sum_{\substack{e \in E \\ rc_e^* > 0}} rc_e^* - \sum_{\substack{e \in B_0 \\ rc_e^* > 0}} rc_e^* + \sum_{\substack{e \in B_1 \\ rc_e^* < 0}} rc_e^* \tag{23}$$

5. Computational experiments

In this section, we present the results of the computational experiments. Firstly, we evaluate the new designed components for the revisited B&C algorithm (RB&C); and secondly, we compare the performance of RB&C with state-of-the-art B&C and heuristic algorithms. The software used for the experiments is publicly available on github.com/gkobeaga/op-solver.

The experiments are carried out using well-known instances in the literature. These instances, which are based on the TSPLIB library, were first proposed in Fischetti et al. (1998) and then extended to larger problems in Kobeaga et al. (2018). The instances are split into two groups: medium-sized instances (up to 400 nodes) and large-sized instances (up to 7397 nodes). In total, we consider 258 benchmark instances. They are also classified into three generations (Gen1, Gen2 and Gen3) according to the definition of scores (Fischetti et al., 1998).

The definition of the scores is depicted in Table 2. For all of these three generations, the distance limitation is set as half of the TSP solution value. These instances are publicly available at github.com/bcamath-ds/OPLib.

In order to measure the performance of the algorithms, we compare the quality of the returned best solutions (LB) and the mean running time (in seconds) of the algorithms. In addition, in the case of the B&C algorithms, we also compare the obtained upper bounds (UB). All the experiments for the compared algorithms have been carried out using a 5-hour time limit.

5.1. Evaluation of components

In this section, we evaluate the designed components for the RB&C algorithm in Section 4. We have carried out experiments with several alternative configurations of the components.To that aim, a subset of 15 OP instances was selected: 5 TSP instances (pr76, att532, vm1084, rl1323 and vm1748, inspired by the subsets selected by Goldberg & Tsioutsiouliklis (2001)) with their respective score generations proposed by Fischetti et al. (1998). Then, for each instance and generation, we execute the different configurations 5 times.

In order to evaluate our contributions, we compare several configurations of the proposed RB&C algorithm. The problem of exploring the different alternative configurations can be framed within the hyper-parameter configuration framework, which is by itself an active area of research Waring et al. (2020). In our context, hyper-parameters can represent many concepts and behaviours. For example, the presence or absence of some components of the proposed algorithm can be considered a Boolean hyper-parameter. Similarly, some other *common parameters* such as the maximum number of cuts or the minimum violation of a cut can also be modelled as discrete hyper-parameters. In our work, we use manual hyper-parameter configuration as we are interested in how specific components affect the performance of the overall algorithm rather than finding hyper-parameter values for some specific problem instances.

In our experiments, we use fixed values for some *common parameters* for all the simulations of the RB&C algorithm that are detailed in appendix Section Appendix C - Table C.22. The values of such *common parameters* were chosen inspired by the parameters used in Applegate et al. (2007) and our preliminary experiments for the OP. In the case of the components proposed in Section 4, they are modelled as hyper-parameters that are tuned in the different ways presented in Table 3. Among the different proposed configurations, the reference configuration deserves special attention as it includes all the components proposed in Section 4. The

Table 4
Results of the REFERENCE and alternative configurations for RB&C. The values are the mean relative differences to the best overall achieved, in percentages. In italics, the values of the alternatives that are worse than those obtained by the REFERENCE configuration are shown.

Strategy	Gap								
	Gen1			Gen2			Gen3		
	LB	UB	Time	LB	UB	Time	LB	UB	Time
REFERENCE	0.05	0.00	262.06	0.05	0.04	23.11	0.02	0.01	44.02
- SRK	*0.13*	0.00	*532.37*	*0.10*	0.04	*25.86*	0.02	*0.02*	*134.74*
SRK=C1C2S3	0.02	0.00	88.32	*0.09*	0.04	*31.72*	0.01	0.01	*79.81*
- CC STRATS	0.02	0.00	115.91	0.04	0.01	21.85	0.01	0.01	*449.90*
- EPH	*0.09*	*0.15*	208.65	*0.12*	*0.15*	*33.64*	*0.10*	*0.22*	*199.79*
- EGH	0.02	0.00	*296.71*	0.04	0.04	*26.18*	*0.03*	0.01	*91.83*
+ FST	0.00	0.00	*345.32*	0.04	0.00	*26.43*	*0.04*	0.00	*66.54*
EPH EGH=FST	*0.09*	*0.13*	54.73	*0.10*	*0.08*	6.58	*0.12*	*0.13*	39.90
- EDGE COVER	*0.11*	0.00	137.73	*0.13*	0.04	*30.04*	*0.05*	0.01	35.50
- CYCLE COVER	*0.06*	0.00	124.79	0.02	0.04	*25.60*	*0.03*	0.01	*48.18*
- PATH	*0.08*	0.00	183.86	*0.10*	0.04	*32.00*	*0.03*	0.01	*69.01*
+ VERTEX COVER	0.05	0.00	61.10	0.03	0.04	22.33	*0.03*	0.01	*104.82*
SEP=TWO	0.05	0.00	*315.34*	*0.06*	0.04	17.05	*0.03*	0.01	*164.44*
B. HEUR=PB	*0.08*	0.00	179.14	*0.12*	0.01	2.37	*0.04*	0.01	*62.74*
B. HEUR=VP	0.02	0.00	222.46	*0.07*	0.04	7.17	0.01	0.01	*168.63*

Table 5
Comparison of the number of instances in which a feasible solution (#), an optimal (OPT), a best-known solution (LB) or a best upper bound value (UB) were obtained.

Size	Gen	#		OPT		LB		UB	
		FST	RB&C	FST	RB&C	FST	RB&C	FST	RB&C
Medium	Gen1	45	45	**45**	44	45	45	**45**	44
	Gen2	45	45	45	45	45	45	45	45
	Gen3	45	45	45	45	45	45	45	45
Large	Gen1	21	**41**	12	**24**	13	**39**	13	**40**
	Gen2	22	**41**	9	**10**	9	**36**	13	**38**
	Gen3	29	**41**	9	**12**	13	**35**	12	**37**
	All	207	**258**	165	**180**	170	**245**	173	**249**

other alternative configurations are obtained by adding, removing or replacing a single component of the reference configuration.

In Table 4 we summarize the mean relative difference to the best achieved LB and UB and the mean relative difference to the best-performing configuration in terms of running time. The results grouped by instances are presented in Appendix A. The corresponding results of the REFERENCE and alternative configurations on the root node (initialization phase) are shown in Table D.23 of Appendix D.

The results show that the alternative configurations decrease, most of the times, the performance of the REFERENCE configuration for the RB&C algorithm either in terms of solution quality, upper bound value, or running time. In general, we observe that the REFERENCE configuration is never dominated (in the multi-objective sense) by any other alternative configuration, in the sense that no other alternative configuration outperforms the REFERENCE configuration in all generations and evaluation metrics. Furthermore, we observe that the REFERENCE configuration outperforms many other alternative configurations, especially in the problem instances within Gen3, which are more difficult than the other problem instances in Gen1 and Gen2.

The experiments restate the importance of the shrinking techniques for the SEC/CC separation algorithm, as can be seen in the results for -SRK. It is not only worse not using the shrinking in terms of time, but indeed, the obtained LB values are also worse. In addition, the results suggest that the S1 shrinking technique, which is considered in REFERENCE, might be preferable to the C1C2 technique. Regarding the CC STRATS, the results for Gen3 suggest that not considering the strategies to find extra violated CCs might have a negative impact on the running time of the algorithm.

Next, looking at the separation algorithms for blossoms, the results show that the EPH heuristic is crucial in the RB&C, particularly, if we focus on the obtained LB and UB values. From the table, we can also extract that the EGH heuristic improves the running time of the B&C algorithm. Alternatively, although the FST blossom heuristic might improve the quality of the solutions, it reports worse running times.

With respect to the rest of the separation algorithms proposed in the literature for the OP, we include in REFERENCE all but Vertex Cover inequalities. This way, the RB&C uses the same families of cuts as in Fischetti et al. (1998), which enables us to evaluate the contributions in this paper in a better way. Finally, the experiments show that the VP primal heuristic plays an important role in obtaining better LB values, particularly for large problems, as can be seen in the detailed results in Appendix A. However, solving the VP primal heuristic in the branch node is more costly than PB primal heuristic, hence the running time of the RB&C is worsened in the smallest instances. Similarly, by using the EA4OP to improve the results by VP heuristic, the obtained LB values are improved in large problems at the expense of worsening the running time in the smallest instances.

5.2. Comparison with state-of-the-art algorithms

The proposed reference RB&C has been compared with the state-of-the-art B&C algorithm in Fischetti et al. (1998) (FST) and two state-of-the-art heuristics, Kobeaga et al. (2018) (EA4OP) and Santini (2019) (ALNS). Each configuration has been executed five times with a 5-hour execution time limit taking the best lower-bound values and upper-bound values for each configuration, i.e., the maximum for LB and the minimum for the UB of the five runs and the average of the running time. The detailed results can be found in Tables B.15, B.16, B.17, B.18, B.19, B.20, B.21 of Appendix B.

Table 6
Comparison of the number of obtained optimal solutions (OPT), number of best-known solutions (LB), number of best upper bounds (UB) and number of instances in which the considered B&C algorithm is faster than the competitor (Time), in the instances that FST does return a solution.

		OPT		LB		UB		Time	
	#	FST	RB&C	FST	RB&C	FST	RB&C	FST	RB&C
Gen1	66	1	**4**	0	**6**	2	**8**	15	**40**
Gen2	67	**1**	0	0	**11**	3	**9**	25	**27**
Gen3	74	1	**3**	1	**14**	4	**17**	23	**33**
All	207	3	**7**	1	**31**	9	**34**	63	**100**

Three notes before moving on to the discussion. First, the FST code reports the running times using one trailing digit while the rest of the algorithms report the times using two trailing digits. In order to make use of the reported times in the literature of the FST, we round the obtained times by the RB&C to one trailing digit when we compare it with the FST algorithm. Secondly, the FST returns a false optimum for pa561 in Gen1. We assume that this is a consequence of the rounding sensibility and we accept as valid the rest of the reported optima by FST. Thirdly, eight instances (rat99, rat195, tsp225, pa561, rat575, rat783, nrw1379, and fnl4461) of Gen3 have been excluded for the comparison of the RB&C with the EA4OP and the ALNS, due to an issue in the generation of scores of the instances used by those algorithms. Since the results of the current comparison are clear enough, we have discarded rerunning the experiments with the updated scores.

We compare the RB&C algorithm with the B&C by Fischetti et al. (1998). The results of the FST algorithm were updated using CPLEX12.5 in Kobeaga et al. (2018), which is the same version of CPLEX used for the experiments of RB&C. Moreover, the new experiments are run on the same machine with the same amount of reserved memory (4GB). In Table 5 we summarize, by size and generation, the number of instances returning a feasible solution, #, the obtained optimality certifications, OPT, the number of best-known solution (LB), and upper bound (UB) values.

In Table 5 it can be seen that the RB&C algorithm is able to obtain the best-known solutions value in all the medium-sized instances. Moving on to large-sized instances, the superiority of the RB&C algorithm compared to the FST approach becomes evident. While the FST algorithm fails to output a solution in almost half of the instances (mainly because of running out of memory), the RB&C algorithm returns a solution for every instance. Moreover, it obtains the best-known solution in significantly more instances than FST (245 against 170) and UB (249 against 173) values. Even more, it obtains more optimality certifications (180 against 165).

In Table 6 we compare the quality of the solutions and running times, restricted to those instances in which FST actually returns a solution. We mainly focus on the number of solutions (optimality certifications, best-known solutions and upper bounds) that are new in the literature, i.e., values not obtained by the rest of the algorithms. Thus, for the lower-bound values, we also take into account the results obtained by the EA4OP and ALNS heuristics. Additionally, we show the number of instances in which the considered B&C algorithms are faster than the competitor and the number of instances in which the solution was found in the root of the RB&C tree. When we restrict the considered instances to the instances where the FST obtains a feasible solution, the RB&C outperforms the results of the FST. While the FST obtains 1 new best-known solution (not obtained by any other algorithm) and 9 new UB values, the RB&C obtains 31 LB and 34 UB new values. In the same set of instances, the FST obtains 3 optimality certifications that the RB&C cannot obtain, while the RB&C obtains 7 optimality certifications that the FST cannot obtain. Moreover, it turns out that the RB&C is faster than the FST in 100 instances while the FST is faster than the RB&C in 63 instances. More details on the obtained results in the root of the tree can be found in Appendix D, Tables D.24, D.25 and D.26 for medium-sized instances and in Tables D.25 and D.27 for large-sized instances.

Table 7
Comparison in medium-sized instances against state-of-the-art algorithms in terms of quality, time and Pareto efficiency.

	Gen1			Gen2			Gen3		
	EA4OP	tie	RB&C	EA4OP	tie	RB&C	EA4OP	tie	RB&C
Quality	0	30	**15**	0	14	**31**	0	15	**27**
Time	15	0	**30**	**37**	0	8	**39**	0	3
Pareto	7	0	**30**	**10**	0	8	**13**	0	3
	ALNS	tie	RB&C	ALNS	tie	RB&C	ALNS	tie	RB&C
Quality	0	40	**5**	0	29	**16**	0	29	**13**
Time	1	0	**44**	4	0	**41**	8	0	**34**
Pareto	1	0	**44**	1	0	**41**	5	0	**34**
	FST	tie	RB&C	FST	tie	RB&C	FST	tie	RB&C
Quality	0	45	0	0	45	0	0	45	0
Time	14	6	**25**	17	2	**26**	18	1	**26**
Pareto	14	6	**25**	17	2	**26**	18	1	**26**

Table 8
Comparison in large-sized instances against state-of-the-art algorithms in terms of quality, time and Pareto efficiency.

	Gen1			Gen2			Gen3		
	EA4OP	tie	RB&C	EA4OP	tie	RB&C	EA4OP	tie	RB&C
Quality	1	0	**40**	5	0	**36**	3	0	**33**
Time	**39**	0	2	**40**	1	0	**35**	1	0
Pareto	1	0	**2**	**5**	0	1	**3**	0	1
	ALNS	tie	RB&C	ALNS	tie	RB&C	ALNS	tie	RB&C
Quality	2	2	**37**	4	1	**36**	4	0	**32**
Time	6	11	**24**	**13**	25	3	**13**	19	4
Pareto	4	0	**34**	5	0	**24**	4	0	**20**
	FST	tie	RB&C	FST	tie	RB&C	FST	tie	RB&C
Quality	0	13	**28**	0	9	**32**	3	11	**27**
Time	1	5	**35**	8	13	**20**	5	17	**19**
Pareto	1	1	**39**	8	0	**33**	7	2	**32**

Next, we compare the RB&C algorithm against state-of-the-art algorithms in terms of solution quality, running time, and Pareto efficiency. In Tables 7 and 8 the algorithms are compared pairwise and instance-by-instance for medium-sized and large-sized instances respectively. The aim is to measure the number of instances where an algorithm is simultaneously as least as fast as the opponent and obtains a better quality solution.

Table 7 shows that the RB&C algorithm is competitive in medium-sized instances. Compared to the ALNS heuristic and FST algorithm, it obtains better Pareto efficiency results in the three generations. Comparing it to EA4OP, the Pareto efficiency is lower because the heuristic is a faster algorithm. Nevertheless, the RB&C obtains much better quality solutions. Table 8 shows that RB&C is the best performing algorithm in large-sized instances. Particularly, it behaves better than the FST algorithm, obtaining the best quality and time solutions in most of the instances, hence obtaining better Pareto results. The ALNS algorithm is able to return some solutions with better quality or running time, however, overall, the RB&C performs better in large-sized instances. The EA4OP metaheuristic is faster than the B&C but, in general, obtains worse quality solutions. In particular, according to the results shown in the tables of Appendix B, the EA4OP is faster than the B&C in 64% of the medium-sized instances and faster in all the large-sized ones. The EA4OP obtains worse quality solutions in 56% of the medium-sized instances and 46% of the large-sized ones.

Table 9
New best-known optimum, lower bound and upper bound values.

	OPT	LB	UB
Gen1	12	25	28
Gen2	2	27	28
Gen3	4	24	29
All	18	76	85

Finally, in Table 9, we summarize the new best-known results obtained in the experiments. The RB&C algorithm obtains 18 new optimality certifications, 76 new best-known solution values and 85 new upper-bound values. When considering Gen1, in the case of the FST algorithm, the largest problem in which an optimality certification is obtained is vm1084, while in the case of the RB&C the largest problem in which an optimality certification is obtained is u2152 (plus another 7 optimality certifications in instances larger than vm1084). When considering Gen2, in the case of the FST algorithm, the largest problem in which an optimality certification is obtained is u724, while in the case of the RB&C the largest problem in which an optimality certification is obtained is d2013 (plus rat783).

6. Conclusions and future work

We have presented a revisited version of the B&C algorithm for the OP that brings multiple contributions together. We have proposed a joint separation algorithm for SECs and CCs, which efficiently uses the shrinking technique for cycle problems by reducing the adverse effects of the shrinking for CCs. We have developed two blossom heuristics for cycle problems which generalize the well-known approaches in the literature of the TSP. We have designed an efficient variable pricing procedure for the OP which enables us to avoid repetitive calculations and to skip the exact calculation of the reduced cost of some variables. We have proposed a separation loop for the OP that takes into consideration the different contributions and separation costs of the valid inequalities. We have used alternative primal heuristics, one of which is based on a metaheuristic, and a mechanism to update the global upper bound during the branching phase to tighten the lower and upper bounds for the cases when the algorithm finishes without an optimality certification.

The experiments have shown that the RB&C algorithm for the OP is a more efficient approach than the state-of-the-art B&C algorithm. The introduced algorithm has increased the number of solved problems, obtained better running times in more instances, succeeded in returning new optimality certifications, new best known solutions, and new upper-bound values for large problems. Additionally, it has been shown that the RB&C algorithm obtains better quality solutions than the state-of-the-art heuristics for the OP within the 5-hour running time limit.

Nevertheless, there are many research lines that remain open after this work. One of the most demanding aspects to improve in the presented approach is the implementation of advanced branching techniques. The use of more general cuts, such as combs and clique trees, and the development of their respective separation algorithms for cycle problems might help to improve the performance of the RB&C algorithm. All these future contributions might help to solve the remaining instances until optimality, but we can anticipate it will not be an easy challenge. Implementing the contributions in this paper to other cycle problems which are different from the OP will definitely help to comprehend their importance in the context of cycle problems with a more general view.

Acknowledgements

The authors are partially supported by the projects BERC 2022-2025 (Basque Government) and by SEV-2017-0718 (Spanish Ministry of Science, Innovation and Universities). The first and third authors are partially supported by the grant PID2019-104933GB-I00 funded by MCIN/AEI/10.13039/ 501100011033 (Spanish Ministry of Science and Innovation). The first author is also supported by the grant BES-2015-072036 (Spanish Ministry of Economy and Competitiveness) and project ELKARTEK (Basque Government). The third author is supported by IT-1494-22 (Basque Government) and GIU20/054 (University of the Basque Country). The fourth author is also supported by IT-1504-22 (Basque Government) and the grants PID2019-104966GB-I00 and PID2019-106453GA-I00 funded by MCIN/AEI/10.13039/501100011033 (Spanish Ministry of Science and Innovation). We gratefully acknowledge the authors of the TSP solver Concorde for making their code available to the public, since it has been the working basis of our implementations. We also thank Prof. J.J. Salazar-Gonzalez who provided us with the codes used in Fischetti et al. (1998).

Appendix A. Configuration of components: detailed results

In this section, we show the detailed results of the alternative RB&C configurations by instances and generations. Each configura-

Table A.10
pr76.

Strategy	Gen																	
	Gen1						Gen2						Gen3					
	LB		UB		Time		LB		UB		Time		LB		UB		Time	
	Best	Gap	Best	Gap	Mean	Gap	Best	Gap	Best	Gap	Mean	Gap	Best	Gap	Best	Gap	Mean	Gap
REFERENCE	49	0	49	0	0.04	123.66	2,708	0	2,708	0	1.13	90.94	2,430	0	2,430	0	1.03	39.55
- SRK	49	0	49	0	0.04	119.35	2,708	0	2,708	0	1.21	104.70	2,430	0	2,430	0	1.06	42.60
SRK=C1C2S3	49	0	49	0	0.04	111.83	2,708	0	2,708	0	1.38	133.98	2,430	0	2,430	0	0.90	21.84
- CC STRATS	49	0	49	0	0.04	124.73	2,708	0	2,708	0	1.13	90.57	2,430	0	2,430	0	1.04	39.74
- EPH BLOSSOM	49	0	49	0	0.03	64.52	2,708	0	2,708	0	1.44	143.58	2,430	0	2,430	0	0.74	0.00
- EGH BLOSSOM	49	0	49	0	0.02	0.00	2,708	0	2,708	0	1.22	106.29	2,430	0	2,430	0	0.97	30.56
+ FST BLOSSOM	49	0	49	0	0.09	398.92	2,708	0	2,708	0	1.32	123.29	2,430	0	2,430	0	0.85	14.47
- EDGE COVER	49	0	49	0	0.03	83.87	2,708	0	2,708	0	1.33	125.56	2,430	0	2,430	0	1.76	136.91
- CYCLE COVER	49	0	49	0	0.05	174.19	2,708	0	2,708	0	1.20	103.38	2,430	0	2,430	0	0.97	30.70
- PATH	49	0	49	0	0.04	116.13	2,708	0	2,708	0	1.39	135.40	2,430	0	2,430	0	0.79	7.02
+ VERTEX COVER	49	0	49	0	0.04	104.30	2,708	0	2,708	0	1.11	87.02	2,430	0	2,430	0	0.98	31.94
SEP: TWO SUBLOOPS	49	0	49	0	0.07	266.67	2,708	0	2,708	0	0.95	60.62	2,430	0	2,430	0	1.00	34.99
BRANCH HEUR=PB	49	0	49	0	0.05	175.27	2,708	0	2,708	0	0.59	0.00	2,430	0	2,430	0	1.41	90.47
BRANCH HEUR=VP - EA4OP	49	0	49	0	0.04	119.35	2,708	0	2,708	0	0.66	11.22	2,430	0	2,430	0	0.96	29.35

Table A.11
att532.

Strategy	Gen																	
	Gen1						Gen2						Gen3					
	LB		UB		Time		LB		UB		Time		LB		UB		Time	
	Best	Gap	Best	Gap	Mean	Gap	Best	Gap	Best	Gap	Mean	Gap	Best	Gap	Best	Gap	Mean	Gap
REFERENCE	363	0.00	363	0.00	359.51	1,031.58	19,633	0.06	19,801	0.01	18,000.00	0.00	15,498	0.00	15,498	0.00	166.80	29.99
- SRK	363	0.00	363	0.00	643.50	1,925.50	19,635	0.05	19,800	0.01	18,000.00	0.00	15,498	0.00	15,498	0.00	219.86	71.34
SRK=C1C2S3	363	0.00	363	0.00	120.89	280.53	19,634	0.05	19,800	0.01	18,000.00	0.00	15,498	0.00	15,498	0.00	284.01	121.34
- CC STRATS	363	0.00	363	0.00	118.09	271.70	19,633	0.06	19,802	0.02	18,000.00	0.00	15,498	0.00	15,498	0.00	2,696.49	2,001.40
- EPH BLOSSOM	363	0.00	363	0.00	31.77	0.00	19,643	0.01	19,801	0.01	18,000.00	0.00	15,498	0.00	15,498	0.00	316.47	146.63
- EGH BLOSSOM	363	0.00	363	0.00	420.83	1,224.61	19,634	0.05	19,801	0.01	18,000.00	0.00	15,498	0.00	15,498	0.00	252.53	96.80
+ FST BLOSSOM	363	0.00	363	0.00	423.05	1,231.61	19,644	0.00	19,801	0.01	18,000.00	0.00	15,498	0.00	15,498	0.00	210.91	64.36
- EDGE COVER	363	0.00	363	0.00	176.79	456.47	19,636	0.04	19,800	0.01	18,000.00	0.00	15,498	0.00	15,498	0.00	180.40	40.59
- CYCLE COVER	363	0.00	363	0.00	110.11	246.59	19,642	0.01	19,801	0.01	18,000.00	0.00	15,498	0.00	15,498	0.00	221.88	72.91
- PATH	363	0.00	363	0.00	252.18	693.77	19,629	0.08	19,801	0.01	18,000.00	0.00	15,498	0.00	15,498	0.00	212.04	65.25
+ VERTEX COVER	363	0.00	363	0.00	81.69	157.14	19,637	0.04	19,799	0.00	18,000.00	0.00	15,498	0.00	15,498	0.00	305.62	138.18
SEP: TWO SUBLOOPS	363	0.00	363	0.00	300.17	844.81	19,631	0.07	19,801	0.01	18,000.00	0.00	15,498	0.00	15,498	0.00	146.51	14.18
BRANCH HEUR=PB	363	0.00	363	0.00	190.93	500.97	19,611	0.17	19,800	0.01	18,000.00	0.00	15,498	0.00	15,498	0.00	194.63	51.68
BRANCH HEUR=VP - EA4OP	363	0.00	363	0.00	270.75	752.20	19,619	0.13	19,801	0.01	18,000.00	0.00	15,498	0.00	15,498	0.00	1,000.74	679.89

Table A.12
vm1084.

Strategy	Gen																	
	Gen1						Gen2						Gen3					
	LB		UB		Time		LB		UB		Time		LB		UB		Time	
	Best	Gap	Best	Gap	Mean	Gap	Best	Gap	Best	Gap	Mean	Gap	Best	Gap	Best	Gap	Mean	Gap
REFERENCE	777	0.00	777	0.00	5,378.5	144.79	40,770	0.02	40,954	0.02	18,000.0	0.00	37,669	0.00	37,669	0.00	4,735.9	150.57
- SRK	777	0.00	777	0.00	9,969.9	353.75	40,777	0.00	40,952	0.01	18,000.0	0.00	37,669	0.00	37,669	0.00	12,469.7	559.74
SRK=C1C2S3	777	0.00	777	0.00	2,731.9	24.34	40,765	0.03	40,953	0.02	18,000.0	0.00	37,669	0.00	37,669	0.00	6,725.9	255.85
- CC STRATS	777	0.00	777	0.00	4,937.8	124.73	40,772	0.01	40,953	0.02	18,000.0	0.00	37,669	0.00	37,669	0.00	5,828.3	208.36
- EPH BLOSSOM	777	0.00	777	0.00	13,669.6	522.14	40,777	0.00	41,006	0.15	18,000.0	0.00	37,665	0.01	37,758	0.24	18,000.0	852.33
- EGH BLOSSOM	777	0.00	777	0.00	7,073.6	221.94	40,773	0.01	40,948	0.00	18,000.0	0.00	37,669	0.00	37,669	0.00	8,161.1	331.78
+ FST BLOSSOM	777	0.00	777	0.00	2,197.2	0.00	40,775	0.00	40,946	0.00	18,000.0	0.00	37,669	0.00	37,669	0.00	6,688.2	253.86
- EDGE COVER	777	0.00	777	0.00	3,303.3	50.34	40,773	0.01	40,954	0.02	18,000.0	0.00	37,669	0.00	37,669	0.00	1,890.1	0.00
- CYCLE COVER	777	0.00	777	0.00	4,072.0	85.33	40,775	0.00	40,950	0.01	18,000.0	0.00	37,669	0.00	37,669	0.00	4,485.4	137.31
- PATH	777	0.00	777	0.00	4,103.7	86.77	40,775	0.00	40,952	0.01	18,000.0	0.00	37,669	0.00	37,669	0.00	7,045.8	272.77
+ VERTEX COVER	777	0.00	777	0.00	3,165.5	44.07	40,777	0.00	40,953	0.02	18,000.0	0.00	37,669	0.00	37,669	0.00	8,580.9	353.99
SEP: TWO SUBLOOPS	777	0.00	777	0.00	5,145.1	134.17	40,773	0.01	40,950	0.01	18,000.0	0.00	37,669	0.00	37,669	0.00	16,501.5	773.05
BRANCH HEUR=PB	777	0.00	777	0.00	2,596.3	18.16	40,767	0.02	40,955	0.02	18,000.0	0.00	37,669	0.00	37,669	0.00	5,133.0	171.57
BRANCH HEUR=VP - EA4OP	777	0.00	777	0.00	3,767.8	71.48	40,763	0.03	40,956	0.02	18,000.0	0.00	37,669	0.00	37,669	0.00	4,421.5	133.93

Table A.13
rl1323.

Strategy	Gen																	
	Gen1						Gen2						Gen3					
	LB		UB		Time		LB		UB		Time		LB		UB		Time	
	Best	Gap	Best	Gap	Mean	Gap	Best	Gap	Best	Gap	Mean	Gap	Best	Gap	Best	Gap	Mean	Gap
REFERENCE	814	0.00	814	0.00	3,565.7	10.26	43,377	0.00	43,454	0.18	18,000.0	24.62	47,162	0.11	47,373	0.00	18,000.0	0.00
- SRK	814	0.00	814	0.00	11,747.3	263.25	43,378	0.00	43,452	0.17	18,000.0	24.62	47,195	0.04	47,408	0.08	18,000.0	0.00
SRK=C1C2S3	814	0.00	814	0.00	4,039.8	24.92	43,378	0.00	43,457	0.18	18,000.0	24.62	47,212	0.01	47,382	0.02	18,000.0	0.00
- CC STRATS	814	0.00	814	0.00	5,121.9	58.38	43,378	0.00	43,378	0.00	17,145.6	18.71	47,213	0.00	47,386	0.03	18,000.0	0.00
- EPH BLOSSOM	814	0.00	819	0.61	18,000.0	456.60	43,371	0.02	43,543	0.38	18,000.0	24.62	47,075	0.30	47,698	0.69	18,000.0	0.00
- EGH BLOSSOM	814	0.00	814	0.00	4,431.3	37.02	43,377	0.00	43,455	0.18	18,000.0	24.62	47,190	0.05	47,394	0.05	18,000.0	0.00
+ FST BLOSSOM	814	0.00	814	0.00	6,341.4	96.09	43,378	0.00	43,378	0.00	15,722.3	8.85	47,200	0.03	47,371	0.00	18,000.0	0.00
- EDGE COVER	814	0.00	814	0.00	6,401.6	97.95	43,273	0.24	43,456	0.18	18,000.0	24.62	47,109	0.22	47,381	0.02	18,000.0	0.00
- CYCLE COVER	814	0.00	814	0.00	7,045.5	117.86	43,378	0.00	43,449	0.16	18,000.0	24.62	47,193	0.05	47,385	0.03	18,000.0	0.00
- PATH	814	0.00	814	0.00	3,965.2	22.61	43,378	0.00	43,446	0.16	18,000.0	24.62	47,201	0.03	47,379	0.02	18,000.0	0.00
+ VERTEX COVER	814	0.00	814	0.00	3,233.9	0.00	43,377	0.00	43,450	0.17	18,000.0	24.62	47,171	0.09	47,379	0.02	18,000.0	0.00
SEP: TWO SUBLOOPS	814	0.00	814	0.00	13,939.5	331.04	43,373	0.01	43,451	0.17	18,000.0	24.62	47,196	0.04	47,378	0.01	18,000.0	0.00
BRANCH HEUR=PB	814	0.00	814	0.00	9,743.9	201.30	43,378	0.00	43,378	0.00	16,153.4	11.84	47,215	0.00	47,387	0.03	18,000.0	0.00
BRANCH HEUR=VP - EA4OP	814	0.00	814	0.00	8,707.5	169.25	43,378	0.00	43,449	0.16	18,000.0	24.62	47,195	0.04	47,376	0.01	18,000.0	0.00

Table A.14
vm1748.

Strategy	Gen																	
	Gen1						Gen2						Gen3					
	LB		UB		Time		LB		UB		Time		LB		UB		Time	
	Best	Gap	Best	Gap	Mean	Gap	Best	Gap	Best	Gap	Mean	Gap	Best	Gap	Best	Gap	Mean	Gap
REFERENCE	1,276	0.23	1,282	0.00	18,000	0	68,013	0.16	68,305	0.01	18,000	0	71,903	0.01	72,018	0.02	18,000	0
- SRK	1,271	0.63	1,282	0.00	18,000	0	67,812	0.45	68,306	0.01	18,000	0	71,853	0.08	72,012	0.01	18,000	0
SRK=C1C2S3	1,278	0.08	1,282	0.00	18,000	0	67,863	0.38	68,306	0.01	18,000	0	71,887	0.03	72,010	0.01	18,000	0
- CC STRATS	1,278	0.08	1,282	0.00	18,000	0	68,016	0.15	68,304	0.01	18,000	0	71,894	0.02	72,012	0.01	18,000	0
- EPH BLOSSOM	1,273	0.47	1,284	0.16	18,000	0	67,735	0.57	68,460	0.23	18,000	0	71,755	0.21	72,118	0.16	18,000	0
- EGH BLOSSOM	1,278	0.08	1,282	0.00	18,000	0	68,029	0.14	68,311	0.02	18,000	0	71,854	0.08	72,016	0.02	18,000	0
+ FST BLOSSOM	1,279	0.00	1,282	0.00	18,000	0	67,986	0.20	68,300	0.00	18,000	0	71,773	0.19	72,003	0.00	18,000	0
- EDGE COVER	1,272	0.55	1,282	0.00	18,000	0	67,877	0.36	68,306	0.01	18,000	0	71,873	0.05	72,017	0.02	18,000	0
- CYCLE COVER	1,275	0.31	1,282	0.00	18,000	0	68,055	0.10	68,302	0.00	18,000	0	71,845	0.09	72,014	0.02	18,000	0
- PATH	1,274	0.39	1,282	0.00	18,000	0	67,831	0.43	68,309	0.01	18,000	0	71,808	0.14	72,013	0.01	18,000	0
+ VERTEX COVER	1,276	0.23	1,282	0.00	18,000	0	68,032	0.13	68,300	0.00	18,000	0	71,883	0.04	72,016	0.02	18,000	0
SEP: TWO SUBLOOPS	1,276	0.23	1,282	0.00	18,000	0	67,967	0.23	68,314	0.02	18,000	0	71,830	0.11	72,017	0.02	18,000	0
BRANCH HEUR=PB	1,274	0.39	1,282	0.00	18,000	0	67,830	0.43	68,300	0.00	18,000	0	71,779	0.18	72,017	0.02	18,000	0
BRANCH HEUR=VP - EA4OP	1,278	0.08	1,282	0.00	18,000	0	67,981	0.21	68,307	0.01	18,000	0	71,890	0.03	72,016	0.02	18,000	0

Table B.15
Generation 1, $n \leq 400$.

Instance	Best		FST				EA4OP			ALNS			RB&C				
	LB	UB	LB	GGap	UB	Time	LB	GGap	Time	LB	GGap	Time	LB	GGap	UB	OGap	Time
att48	31	31	**31**	*	**31**	**0.00**	**31**	*	0.25	**31**	*	6.77	**31**	*	**31**	*	**0.03**
gr48	31	31	**31**	*	**31**	**0.00**	**31**	*	0.13	**31**	*	9.99	**31**	*	**31**	*	**0.02**
hk48	30	30	**30**	*	**30**	**0.00**	**30**	*	0.24	**30**	*	7.20	**30**	*	**30**	*	**0.01**
eil51	29	29	**29**	*	**29**	**0.00**	**29**	*	0.24	**29**	*	9.51	**29**	*	**29**	*	**0.01**
berlin52	37	37	**37**	*	**37**	**0.00**	**37**	*	0.30	**37**	*	9.42	**37**	*	**37**	*	**0.02**
brazil58	46	46	**46**	*	**46**	**0.00**	**46**	*	1.00	**46**	*	9.13	**46**	*	**46**	*	0.07
st70	43	43	**43**	*	**43**	0.10	**43**	*	0.32	**43**	*	15.99	**43**	*	**43**	*	**0.05**
eil76	47	47	**47**	*	**47**	0.10	46	2.13	0.33	**47**	*	20.51	**47**	*	**47**	*	**0.04**
pr76	49	49	**49**	*	**49**	0.10	**49**	*	0.61	**49**	*	18.64	**49**	*	**49**	*	**0.06**
gr96	64	64	**64**	*	**64**	0.10	**64**	*	1.44	**64**	*	20.31	**64**	*	**64**	*	**0.08**
rat99	52	52	**52**	*	**52**	**0.40**	**52**	*	0.66	**52**	*	27.75	**52**	*	**52**	*	0.47
kroA100	56	56	**56**	*	**56**	**0.40**	55	1.79	0.34	**56**	*	34.75	**56**	*	**56**	*	**0.41**
kroB100	58	58	**58**	*	**58**	95.40	57	1.72	0.63	**58**	*	43.06	**58**	*	**58**	*	**0.27**
kroC100	56	56	**56**	*	**56**	0.40	**56**	*	0.48	**56**	*	34.32	**56**	*	**56**	*	**0.25**
kroD100	59	59	**59**	*	**59**	0.10	58	1.69	0.65	**59**	*	34.61	**59**	*	**59**	*	**0.09**
kroE100	57	57	**57**	*	**57**	159.20	**57**	*	0.50	**57**	*	32.26	**57**	*	**57**	*	**5.53**
rd100	61	61	**61**	*	**61**	0.20	**61**	*	0.74	**61**	*	29.49	**61**	*	**61**	*	**0.12**
eil101	64	64	**64**	*	**64**	0.10	**64**	*	0.79	**64**	*	31.73	**64**	*	**64**	*	**0.06**
lin105	66	66	**66**	*	**66**	**0.30**	**66**	*	1.42	**66**	*	32.11	**66**	*	**66**	*	0.48
pr107	54	54	**54**	*	**54**	0.30	**54**	*	0.93	**54**	*	78.46	**54**	*	**54**	*	**0.08**
gr120	75	75	**75**	*	**75**	**0.10**	74	1.33	1.20	**75**	*	29.58	**75**	*	**75**	*	0.28
pr124	75	75	**75**	*	**75**	**0.30**	**75**	*	1.11	**75**	*	49.64	**75**	*	**75**	*	0.35
bier127	103	103	**103**	*	**103**	**0.30**	**103**	*	1.18	**103**	*	40.84	**103**	*	**103**	*	0.38
pr136	71	71	**71**	*	**71**	**1.40**	**71**	*	0.96	**71**	*	29.97	**71**	*	**71**	*	1.75
gr137	81	81	**81**	*	**81**	1.50	78	3.70	3.44	**81**	*	59.21	**81**	*	**81**	*	**0.24**
pr144	77	77	**77**	*	**77**	**1.30**	**77**	*	2.61	**77**	*	87.82	**77**	*	**77**	*	1.46
kroA150	86	86	**86**	*	**86**	175.40	**86**	*	1.17	**86**	*	82.79	**86**	*	**86**	*	**33.87**
kroB150	87	87	**87**	*	**87**	**1.20**	86	1.15	1.00	**87**	*	61.64	**87**	*	**87**	*	2.21
pr152	77	77	**77**	*	**77**	1.40	**77**	*	3.64	**77**	*	91.38	**77**	*	**77**	*	**1.29**
u159	93	93	**93**	*	**93**	3.40	92	1.08	1.11	**93**	*	99.63	**93**	*	**93**	*	**1.82**
rat195	102	102	**102**	*	**102**	**2.60**	99	2.94	1.78	**102**	*	195.57	**102**	*	**102**	*	3.71
d198	123	123	**123**	*	**123**	**3.20**	**123**	*	6.68	**123**	*	65.57	**123**	*	**123**	*	5.28
kroA200	117	117	**117**	*	**117**	**1.20**	**117**	*	1.74	**117**	*	114.75	**117**	*	**117**	*	2.50
kroB200	119	119	**119**	*	**119**	14.10	**119**	*	1.67	**119**	*	86.58	**119**	*	**119**	*	**9.91**
gr202	145	145	**145**	*	**145**	12.70	**145**	*	6.89	**145**	*	187.56	**145**	*	**145**	*	**2.71**
ts225	124	124	**124**	*	**124**	**10,216.30**	**124**	*	1.28	**124**	*	279.52	**124**	*	126	1.59	18,000.00
tsp225	129	129	**129**	*	**129**	94.40	127	1.55	2.29	128	0.78	198.47	**129**	*	**129**	*	**4.31**
pr226	126	126	**126**	*	**126**	166.20	**126**	*	6.61	**126**	*	181.94	**126**	*	**126**	*	**107.69**
gr229	176	176	**176**	*	**176**	0.90	**176**	*	8.81	173	1.70	108.27	**176**	*	**176**	*	**0.32**
gil262	158	158	**158**	*	**158**	0.90	156	1.27	2.83	**158**	*	240.02	**158**	*	**158**	*	**0.35**
pr264	132	132	**132**	*	**132**	21.20	**132**	*	5.62	**132**	*	314.29	**132**	*	**132**	*	**3.92**
a280	147	147	**147**	*	**147**	**13.60**	143	2.72	3.00	144	2.04	239.06	**147**	*	**147**	*	40.65
pr299	162	162	**162**	*	**162**	111.50	160	1.23	3.12	**162**	*	410.90	**162**	*	**162**	*	**48.85**
lin318	205	205	**205**	*	**205**	22.40	202	1.46	7.15	203	0.98	294.23	**205**	*	**205**	*	**5.49**
rd400	239	239	**239**	*	**239**	37.40	234	2.09	6.59	237	0.84	422.56	**239**	*	**239**	*	**36.71**
average				*		**248.05**		0.62	2.12		0.14	99.51		*		0.04	407.20

Table B.16
Generation 1, $n > 400$.

Instance	Best		FST				EA4OP			ALNS			RB&C				
	LB	UB	LB	GGap	UB	Time	LB	GGap	Time	LB	GGap	Time	LB	GGap	UB	OGap	Time
fl417	228	230	**228**	*	**230**	**18,000.00**	224	1.75	11.84	**228**	*	1,056.07	**228**	*	231	1.30	**18,000.00**
gr431	350	350	**350**	*	**350**	139.90	349	0.29	32.84	347	0.86	533.55	**350**	*	**350**	*	**29.05**
pr439	313	313	**313**	*	**313**	833.30	310	0.96	9.92	307	1.92	1,263.74	**313**	*	**313**	*	**414.00**
pcb442	251	251	**251**	*	**251**	14.90	244	2.79	6.94	249	0.80	1,328.72	**251**	*	**251**	*	**7.21**
d493	320	320	**320**	*	**320**	347.30	315	1.56	19.10	317	0.94	1,291.93	**320**	*	**320**	*	**13.37**
att532	363	363	**363**	*	**363**	593.00	347	4.41	23.14	359	1.10	1,380.54	**363**	*	**363**	*	**312.50**
ali535	425	426	.	.	.	.	424	0.24	73.03	422	0.71	1,846.10	**425**	*	**426**	0.23	**18,000.00**
pa561	357	357	356	0.28	–	2,103.60	348	2.52	23.18	346	3.08	1,605.42	**357**	*	**357**	*	**245.42**
u574	354	354	**354**	*	**354**	61.40	344	2.82	17.93	347	1.98	1,204.18	**354**	*	**354**	*	**24.00**
rat575	322	322	**322**	*	**322**	59.50	309	4.04	13.76	317	1.55	3,109.65	**322**	*	**322**	*	**42.82**
p654	343	396	327	4.66	553	**18,000.00**	336	2.04	28.89	**343**	*	10,866.70	342	0.29	**396**	13.64	**18,000.00**
d657	386	386	**386**	*	**386**	715.70	377	2.33	23.24	380	1.55	3,152.17	**386**	*	**386**	*	**92.48**
gr666	503	503	**503**	*	**503**	634.20	497	1.19	109.54	486	3.38	660.30	**503**	*	**503**	*	**400.56**
u724	439	439	**439**	*	**439**	1,077.10	429	2.28	27.77	434	1.14	4,157.30	**439**	*	**439**	*	**188.61**
rat783	438	438	**438**	*	**438**	594.30	422	3.65	34.59	428	2.28	2,962.52	**438**	*	**438**	*	**514.68**
dsj1000	656	656	.	.	.	.	632	3.66	81.20	630	3.96	17,284.30	**656**	*	**656**	*	**3,828.50**
pr1002	606	606	604	0.33	608	18,000.00	572	5.61	45.92	581	4.13	18,000.00	**606**	*	**606**	*	**4,483.81**
u1060	660	660	.	.	.	.	627	5.00	90.04	644	2.42	18,000.00	**660**	*	**660**	*	**16,716.01**
vm1084	777	777	**777**	*	**777**	**4,927.40**	770	0.90	56.29	765	1.54	18,000.00	**777**	*	**777**	*	5,012.60
pcb1173	675	675	.	.	.	.	633	6.22	60.65	652	3.41	18,000.00	**675**	*	**675**	*	**6,819.83**
d1291	715	715	.	.	.	.	646	9.65	434.87	699	2.24	18,000.00	**715**	*	**715**	*	**7,916.85**
rl1304	802	802	.	.	.	.	766	4.49	102.45	788	1.75	18,000.00	**802**	*	**802**	*	**6,269.39**
rl1323	814	814	811	0.37	846	18,000.00	782	3.93	89.68	785	3.56	14,585.10	**814**	*	**814**	*	**7,740.17**
nrw1379	815	817	.	.	.	.	771	5.40	106.97	790	3.07	18,000.00	**815**	*	**817**	0.24	**18,000.00**
fl1400	1,048	1,084	909	13.26	1,230	**18,000.00**	1,043	0.48	518.25	**1,048**	*	18,000.00	1,003	4.29	**1,084**	7.47	**18,000.00**
u1432	754	764	.	.	.	.	738	2.12	121.46	749	0.66	14,573.50	**754**	*	**764**	1.31	**18,000.00**
fl1577	897	900	.	.	.	.	880	1.90	286.47	748	16.61	18,000.00	**897**	*	**900**	0.33	**18,000.00**
d1655	922	924	.	.	.	.	846	8.24	757.70	890	3.47	18,000.00	**922**	*	**924**	0.22	**18,000.00**
vm1748	1,276	1,282	873	31.58	.	**18,000.00**	1,246	2.35	178.50	1,252	1.88	16,959.80	**1,276**	*	**1,282**	0.47	**18,000.00**
u1817	983	983	.	.	.	.	879	10.58	975.58	947	3.66	18,000.00	**983**	*	**983**	*	**11,226.88**
rl1889	1,226	1,226	890	27.41	1,296	18,000.00	1,167	4.81	269.81	1,156	5.71	18,000.00	**1,226**	*	**1,226**	*	**17,010.43**
d2103	1,200	1,200	.	.	.	.	1,069	10.92	951.27	1,171	2.42	18,000.00	**1,200**	*	**1,200**	*	**15,855.62**
u2152	1,151	1,151	.	.	.	.	1,048	8.95	1,350.23	1,111	3.48	18,000.00	**1,151**	*	**1,151**	*	**14,703.25**
u2319	1,170	1,171	.	.	.	.	1,167	0.26	423.26	**1,170**	*	6,088.42	**1,170**	*	**1,171**	0.09	**18,000.00**
pr2392	1,316	1,415	1,140	13.37	.	**18,000.00**	1,292	1.82	402.29	1,294	1.67	18,000.00	**1,316**	*	**1,415**	7.00	**18,000.00**
pcb3038	1,727	1,730	.	.	.	.	1,572	8.98	681.94	1,626	5.85	18,000.00	**1,727**	*	**1,730**	0.17	**18,000.00**
fl3795	1,965	2,249	.	.	.	.	1,815	7.63	2,994.90	1,818	7.48	18,000.00	**1,965**	*	**2,249**	12.63	**18,000.00**
fnl4461	2,541	2,570	.	.	.	.	2,350	7.52	2,462.65	2,342	7.83	18,000.00	**2,541**	*	**2,570**	1.13	**18,000.00**
rl5915	3,593	3,786	.	.	.	.	3,358	6.54	5,361.54	3,328	7.38	18,000.00	**3,593**	*	**3,786**	5.10	**18,000.00**
rl5934	3,632	3,752	.	.	.	.	3,145	13.41	5,382.25	3,276	9.80	18,000.00	**3,632**	*	**3,752**	3.20	**18,000.00**
pla7397	5,289	5,657	.	.	.	.	5,141	2.80	15,981.78	5,140	2.82	18,000.00	**5,289**	*	**5,657**	6.51	**18,000.00**
average				4.35		**7,433.41**		4.32	990.82		3.12	11,802.68		0.11		1.49	10,387.02

tion has been executed five times with a 5-hour execution time limit. We show the obtained results of the configuration in terms of lower-bound values, LB, upper-bound values, UB, and time (in seconds) performance, Time. For the LB and UB, the obtained best value for each configuration (the maximum for LB and the minimum for the UB) is presented in the Best column. Regarding the Time, the Mean column shows the meantime of the five executions. The Gap column represents the relative distance to best-known value (higher Best value in the case of LB, and lower Best in the case of UB and Mean in the case of Time, respectively). (Tables A.10–A.14)

Appendix B. Comparison with state-of-the-art algorithms: detailed results

In this appendix, we detail the experimental results for the four algorithms (FST B&C, EA4OP, ALNS and RB&C). Table B.15 shows the results for medium-sized instances of generation 1, Table B.16 for large-sized instances of generation 1, Table B.17 for medium-sized instances of generation 2, Table B.18 for large-sized instances of generation 2, Table B.19 for medium-sized instances of generation 3 and Table B.20 for large-sized instances of generation 3.

In the Best column, we show the global best-known lower and upper-bound values. For each algorithm, we detail the best LB, the goodness gap GGap, the best UB, and the meantime (in seconds). The GGap represents the relative distance between the algorithm's best LB and the global best-known LB. For the RB&C algorithm we also detail the optimality gap OGap which represents the relative distance between the obtained LB and UB by RB&C.

For each algorithm, generation and size, we have calculated the average gap and running time over the instances where a feasible solution was obtained by the algorithm. In those instances where the time limit was reached, a running time of 5 hours has been used. These averages are shown in the last row of the tables. The symbols in the tables mean the following:

*: best-known solution achieved
–: not comparable result
.: the code finished unexpectedly

Finally, we present the extended lower bound and upper bound results for all problem instances. Table B.21 reports the minimum, average and maximum values of the LBs and UBs in the five runnings of RB&C for the large-sized instances ($n > 400$). It can be observed that the relative differences are very small in general. Moreover, the medium-sized instances ($n \leq 400$) are not reported because all the five executions for every instance are equal in terms of LBs and UBs and correspond to the optimal solutions for all the instances (except ts225 instance in generation 1).

Table B.17
Generation 2, $n \leq 400$.

Instance	Best		FST				EA4OP			ALNS			RB&C				
	LB	UB	LB	GGap	UB	Time	LB	GGap	Time	LB	GGap	Time	LB	GGap	UB	OGap	Time
att48	1,717	1,717	**1,717**	*	**1,717**	**0.00**	**1,717**	*	0.32	**1,717**	*	6.77	**1,717**	*	**1,717**	*	**0.04**
gr48	1,761	1,761	**1,761**	*	**1,761**	**0.20**	1,749	0.68	0.20	**1,761**	*	7.87	**1,761**	*	**1,761**	*	1.32
hk48	1,614	1,614	**1,614**	*	**1,614**	**0.10**	**1,614**	*	0.15	**1,614**	*	7.19	**1,614**	*	**1,614**	*	**0.10**
eil51	1,674	1,674	**1,674**	*	**1,674**	**0.40**	1,668	0.36	0.18	**1,674**	*	10.13	**1,674**	*	**1,674**	*	0.96
berlin52	1,897	1,897	**1,897**	*	**1,897**	93.40	**1,897**	*	0.35	**1,897**	*	10.74	**1,897**	*	**1,897**	*	**3.23**
brazil58	2,220	2,220	**2,220**	*	**2,220**	**0.10**	2,218	0.09	1.52	**2,220**	*	12.32	**2,220**	*	**2,220**	*	0.46
st70	2,286	2,286	**2,286**	*	**2,286**	19.40	2,285	0.04	0.31	**2,286**	*	21.65	**2,286**	*	**2,286**	*	**1.77**
eil76	2,550	2,550	**2,550**	*	**2,550**	**0.10**	**2,550**	*	0.43	**2,550**	*	16.06	**2,550**	*	**2,550**	*	0.62
pr76	2,708	2,708	**2,708**	*	**2,708**	**0.40**	**2,708**	*	0.48	**2,708**	*	19.48	**2,708**	*	**2,708**	*	1.46
gr96	3,396	3,396	**3,396**	*	**3,396**	**1.70**	3,394	0.06	1.44	3,394	0.06	31.98	**3,396**	*	**3,396**	*	9.50
rat99	2,944	2,944	**2,944**	*	**2,944**	**0.90**	**2,944**	*	0.49	**2,944**	*	32.08	**2,944**	*	**2,944**	*	3.25
kroA100	3,212	3,212	**3,212**	*	**3,212**	0.90	**3,212**	*	0.57	**3,212**	*	32.85	**3,212**	*	**3,212**	*	**0.70**
kroB100	3,241	3,241	**3,241**	*	**3,241**	**6.70**	3,238	0.09	0.52	3,239	0.06	48.39	**3,241**	*	**3,241**	*	13.28
kroC100	2,947	2,947	**2,947**	*	**2,947**	85.60	2,931	0.54	0.60	**2,947**	*	39.27	**2,947**	*	**2,947**	*	**2.22**
kroD100	3,307	3,307	**3,307**	*	**3,307**	45.00	**3,307**	*	0.65	**3,307**	*	30.52	**3,307**	*	**3,307**	*	**3.62**
kroE100	3,090	3,090	**3,090**	*	**3,090**	230.10	3,082	0.26	0.50	**3,090**	*	39.57	**3,090**	*	**3,090**	*	**11.31**
rd100	3,359	3,359	**3,359**	*	**3,359**	**0.20**	**3,359**	*	0.50	**3,359**	*	30.80	**3,359**	*	**3,359**	*	0.36
eil101	3,655	3,655	**3,655**	*	**3,655**	153.00	**3,655**	*	0.82	**3,655**	*	26.19	**3,655**	*	**3,655**	*	**4.15**
lin105	3,544	3,544	**3,544**	*	**3,544**	67.30	3,530	0.40	1.10	**3,544**	*	36.22	**3,544**	*	**3,544**	*	**2.51**
pr107	2,667	2,667	**2,667**	*	**2,667**	0.60	**2,667**	*	1.05	**2,667**	*	69.67	**2,667**	*	**2,667**	*	**0.20**
gr120	4,371	4,371	**4,371**	*	**4,371**	35.80	4,356	0.34	1.37	**4,371**	*	40.41	**4,371**	*	**4,371**	*	**6.57**
pr124	3,917	3,917	**3,917**	*	**3,917**	**0.50**	3,899	0.46	1.34	**3,917**	*	55.25	**3,917**	*	**3,917**	*	1.07
bier127	5,383	5,383	**5,383**	*	**5,383**	58.80	5,381	0.04	1.71	5,366	0.32	23.01	**5,383**	*	**5,383**	*	**0.96**
pr136	4,309	4,309	**4,309**	*	**4,309**	2.10	**4,309**	*	1.15	**4,309**	*	35.63	**4,309**	*	**4,309**	*	**1.25**
gr137	4,286	4,286	**4,286**	*	**4,286**	196.90	4,099	4.36	3.09	**4,286**	*	639.80	**4,286**	*	**4,286**	*	**10.65**
pr144	4,003	4,003	**4,003**	*	**4,003**	90.40	3,965	0.95	3.02	3,969	0.85	100.20	**4,003**	*	**4,003**	*	**32.23**
kroA150	4,918	4,918	**4,918**	*	**4,918**	241.40	4,902	0.33	1.26	**4,918**	*	80.06	**4,918**	*	**4,918**	*	**60.43**
kroB150	4,869	4,869	**4,869**	*	**4,869**	24.80	**4,869**	*	1.19	**4,869**	*	61.96	**4,869**	*	**4,869**	*	**16.94**
pr152	4,279	4,279	**4,279**	*	**4,279**	2.20	4,245	0.79	3.47	**4,279**	*	67.41	**4,279**	*	**4,279**	*	**1.85**
u159	4,960	4,960	**4,960**	*	**4,960**	192.20	4,941	0.38	1.44	4,950	0.20	109.59	**4,960**	*	**4,960**	*	**14.96**
rat195	5,791	5,791	**5,791**	*	**5,791**	128.80	5,703	1.52	1.55	5,782	0.16	263.23	**5,791**	*	**5,791**	*	**46.09**
d198	6,670	6,670	**6,670**	*	**6,670**	**74.20**	6,660	0.15	7.33	6,661	0.13	88.47	**6,670**	*	**6,670**	*	298.24
kroA200	6,547	6,547	**6,547**	*	**6,547**	68.70	6,534	0.20	1.71	**6,547**	*	116.11	**6,547**	*	**6,547**	*	**16.18**
kroB200	6,419	6,419	**6,419**	*	**6,419**	34.70	6,278	2.20	1.97	6,413	0.09	189.98	**6,419**	*	**6,419**	*	**20.62**
gr202	7,789	7,789	**7,789**	*	**7,789**	**85.70**	**7,789**	*	8.77	7,719	0.90	188.27	**7,789**	*	**7,789**	*	139.90
ts225	6,834	6,834	**6,834**	*	**6,834**	**6.60**	6,819	0.22	1.47	6,782	0.76	394.00	**6,834**	*	**6,834**	*	95.22
tsp225	6,987	6,987	**6,987**	*	**6,987**	174.50	6,936	0.73	1.87	6,980	0.10	299.73	**6,987**	*	**6,987**	*	**54.09**
pr226	6,662	6,662	**6,662**	*	**6,662**	**74.10**	6,658	0.06	7.29	**6,662**	*	201.68	**6,662**	*	**6,662**	*	2,894.81
gr229	9,177	9,177	**9,177**	*	**9,177**	182.60	9,174	0.03	13.19	**9,177**	*	1,379.35	**9,177**	*	**9,177**	*	**16.67**
gil262	8,321	8,321	**8,321**	*	**8,321**	89.60	8,175	1.75	3.47	8,269	0.62	487.41	**8,321**	*	**8,321**	*	**64.63**
pr264	6,654	6,654	**6,654**	*	**6,654**	23.00	6,173	7.23	5.94	**6,654**	*	314.27	**6,654**	*	**6,654**	*	**13.33**
a280	8,428	8,428	**8,428**	*	**8,428**	**103.80**	8,304	1.47	2.85	8,404	0.28	215.31	**8,428**	*	**8,428**	*	519.95
pr299	9,182	9,182	**9,182**	*	**9,182**	**426.50**	9,112	0.76	3.23	9,147	0.38	393.12	**9,182**	*	**9,182**	*	623.34
lin318	10,923	10,923	**10,923**	*	**10,923**	862.40	10,866	0.52	8.29	10,801	1.12	370.64	**10,923**	*	**10,923**	*	**367.53**
rd400	13,652	13,652	**13,652**	*	**13,652**	**293.50**	13,442	1.54	6.80	13,562	0.66	1,174.91	**13,652**	*	**13,652**	*	769.66
average				*		**92.89**		0.63	2.38		0.15	173.77		*		*	136.63

Table B.18
Generation 2, $n > 400$.

Instance	Best		FST				EA4OP			ALNS			RB&C				
	LB	UB	LB	GGap	UB	Time	LB	GGap	Time	LB	GGap	Time	LB	GGap	UB	OGap	Time
fl417	11,933	12,294	11,894	0.33	**12,294**	**18,000.00**	11,787	1.22	16.73	11,923	0.08	2,144.94	**11,933**	*	12,387	3.67	**18,000.00**
gr431	18,318	18,318	**18,318**	*	**18,318**	**969.50**	18,287	0.17	51.38	**18,318**	*	2,740.82	**18,318**	*	**18,318**	*	2,809.41
pr439	16,171	16,171	**16,171**	*	**16,171**	**1,298.30**	16,085	0.53	11.77	16,128	0.27	629.44	**16,171**	*	**16,171**	*	3,765.86
pcb442	14,484	14,484	**14,484**	*	**14,484**	**6,259.10**	14,273	1.46	6.83	14,411	0.50	4,410.74	**14,484**	*	**14,484**	*	13,760.94
d493	16,995	17,007	.	.	.	.	16,729	1.57	17.15	16,820	1.03	6,231.42	**16,995**	*	**17,007**	0.07	**18,000.00**
att532	19,635	19,800	19,598	0.19	**19,800**	**18,000.00**	19,265	1.88	23.43	19,465	0.87	1,564.89	**19,635**	*	**19,800**	0.83	**18,000.00**
ali535	21,954	21,954	**21,954**	*	**21,954**	**2,099.70**	21,910	0.20	95.05	21,761	0.88	1,537.87	**21,954**	*	21,973	0.09	18,000.00
pa561	19,576	19,576	**19,576**	*	**19,576**	**1,487.10**	18,894	3.48	23.45	19,092	2.47	790.31	**19,576**	*	**19,576**	*	1,961.95
u574	19,351	19,351	**19,351**	*	**19,351**	**612.50**	18,966	1.99	16.33	19,028	1.67	5,389.10	**19,351**	*	**19,351**	*	1,026.82
rat575	18,251	18,251	**18,251**	*	**18,251**	**931.10**	17,705	2.99	14.97	17,984	1.46	2,089.02	**18,251**	*	**18,251**	*	9,616.70
p654	17,900	21,566	17,160	4.13	**21,566**	**18,000.00**	17,821	0.44	42.82	**17,900**	*	18,000.00	17,753	0.82	22,248	20.20	**18,000.00**
d657	21,503	21,503	**21,503**	*	**21,503**	2,682.40	21,162	1.59	22.90	21,231	1.26	4,161.44	**21,503**	*	**21,503**	*	**554.67**
gr666	26,514	26,569	.	.	.	.	26,336	0.67	136.48	25,971	2.05	1,024.22	**26,514**	*	**26,569**	0.21	**18,000.00**
u724	24,223	24,223	**24,223**	*	**24,223**	**5,830.50**	23,793	1.78	28.71	23,878	1.42	5,755.06	**24,223**	*	**24,223**	*	9,829.42
rat783	25,474	25,474	.	.	.	.	24,861	2.41	32.36	24,987	1.91	6,622.62	**25,474**	*	**25,474**	*	**12,246.90**
dsj1000	35,835	35,915	35,772	0.18	35,917	**18,000.00**	34,463	3.83	83.34	34,641	3.33	18,000.00	**35,835**	*	**35,915**	0.22	**18,000.00**
pr1002	33,030	33,092	27,066	18.06	.	**18,000.00**	31,746	3.89	46.19	32,120	2.76	18,000.00	**33,030**	*	**33,092**	0.19	**18,000.00**
u1060	36,151	36,291	.	.	.	.	35,110	2.88	77.78	35,284	2.40	18,000.00	**36,151**	*	**36,291**	0.39	**18,000.00**
vm1084	40,777	40,952	40,687	0.22	40,954	**18,000.00**	40,308	1.15	55.67	40,240	1.32	18,000.00	**40,777**	*	**40,952**	0.43	**18,000.00**
pcb1173	37,035	37,100	.	.	.	.	35,826	3.26	69.94	35,946	2.94	18,000.00	**37,035**	*	**37,100**	0.18	**18,000.00**
d1291	37,778	37,854	.	.	.	.	35,153	6.95	289.25	36,815	2.55	18,000.00	**37,778**	*	**37,854**	0.20	**18,000.00**
rl1304	42,275	42,359	.	.	.	.	40,561	4.05	97.68	40,893	3.27	12,853.40	**42,275**	*	**42,359**	0.20	**18,000.00**
rl1323	43,377	43,450	43,347	0.07	**43,450**	**18,000.00**	41,459	4.42	89.78	41,210	5.00	18,000.00	**43,377**	*	**43,450**	0.17	**18,000.00**
nrw1379	46,676	46,787	.	.	.	.	45,602	2.30	117.51	45,576	2.36	18,000.00	**46,676**	*	**46,787**	0.24	**18,000.00**
fl1400	56,692	64,298	53,222	6.12	64,726	**18,000.00**	56,258	0.77	794.15	**56,692**	*	18,000.00	54,124	4.53	**64,298**	15.82	**18,000.00**
u1432	46,946	47,018	.	.	.	.	44,810	4.55	100.91	44,982	4.18	18,000.00	**46,946**	*	**47,018**	0.15	**18,000.00**
fl1577	45,505	50,154	.	.	.	.	**45,505**	*	334.28	41,148	9.57	18,000.00	45,326	0.39	**50,154**	9.63	**18,000.00**
d1655	49,319	53,083	.	.	.	.	47,211	4.27	683.17	**49,319**	*	18,000.00	46,158	6.41	**53,083**	13.05	**18,000.00**
vm1748	68,042	68,303	.	.	.	.	66,685	1.99	195.85	66,636	2.07	18,000.00	**68,042**	*	**68,303**	0.38	**18,000.00**
u1817	54,245	54,554	.	.	.	.	50,366	7.15	734.39	51,676	4.74	18,000.00	**54,245**	*	**54,554**	0.57	**18,000.00**
rl1889	63,308	64,425	52,047	17.79	.	**18,000.00**	60,084	5.09	286.07	60,928	3.76	18,000.00	**63,308**	*	**64,425**	1.73	**18,000.00**
d2103	63,426	63,426	.	.	.	.	57,202	9.81	682.28	61,636	2.82	18,000.00	**63,426**	*	**63,426**	*	**16,593.51**
u2152	64,649	64,775	53,976	16.51	.	**18,000.00**	60,211	6.86	1,164.38	61,052	5.56	18,000.00	**64,649**	*	**64,775**	0.19	**18,000.00**
u2319	80,914	81,139	72,790	10.04	.	**18,000.00**	78,102	3.48	447.06	77,610	4.08	18,000.00	**80,914**	*	**81,139**	0.28	**18,000.00**
pr2392	72,843	78,237	64,577	11.35	.	**18,000.00**	71,018	2.51	440.57	71,851	1.36	18,000.00	**72,843**	*	**78,237**	6.89	**18,000.00**
pcb3038	97,902	97,995	83,951	14.25	.	**18,000.00**	91,842	6.19	820.37	91,457	6.58	18,000.00	**97,902**	*	**97,995**	0.09	**18,000.00**
fl3795	103,397	142,895	.	.	.	.	**103,397**	*	4,788.96	102,642	0.73	18,000.00	98,998	4.25	**142,895**	30.72	**18,000.00**
fnl4461	147,109	150,189	.	.	.	.	140,424	4.54	2,618.15	135,515	7.88	18,000.00	**147,109**	*	**150,189**	2.05	**18,000.00**
rl5915	184,424	197,729	.	.	.	.	176,678	4.20	5,512.40	173,500	5.92	18,000.00	**184,424**	*	**197,729**	6.73	**18,000.00**
rl5934	187,034	196,805	.	.	.	.	171,649	8.23	5,757.80	166,368	11.05	18,000.00	**187,034**	*	**196,805**	4.96	**18,000.00**
pla7397	281,977	297,246	.	.	.	.	272,452	3.38	18,000.00	266,038	5.65	18,000.00	**281,977**	*	**297,246**	5.14	**18,000.00**
average				4.51		**11,644.10**		3.13	1,093.37		2.87	12,827.93		0.40		3.06	15,369.91

Table B.19
Generation 3, $n \leq 400$.

Instance	Best		FST				EA4OP			ALNS			RB&C				
	LB	UB	LB	GGap	UB	Time	LB	GGap	Time	LB	GGap	Time	LB	GGap	UB	OGap	Time
att48	1,049	1,049	**1,049**	*	**1,049**	38.50	**1,049**	*	0.26	**1,049**	*	7.18	**1,049**	*	**1,049**	*	**1.17**
gr48	1,480	1,480	**1,480**	*	**1,480**	**0.20**	**1,480**	*	0.13	**1,480**	*	8.87	**1,480**	*	**1,480**	*	0.72
hk48	1,764	1,764	**1,764**	*	**1,764**	**0.00**	**1,764**	*	0.22	**1,764**	*	8.51	**1,764**	*	**1,764**	*	0.06
eil51	1,399	1,399	**1,399**	*	**1,399**	**0.20**	1,398	0.07	0.22	**1,399**	*	6.87	**1,399**	*	**1,399**	*	1.46
berlin52	1,036	1,036	**1,036**	*	**1,036**	124.70	1,034	0.19	0.64	**1,036**	*	12.84	**1,036**	*	**1,036**	*	**4.61**
brazil58	1,702	1,702	**1,702**	*	**1,702**	**0.00**	**1,702**	*	0.71	**1,702**	*	11.09	**1,702**	*	**1,702**	*	**0.02**
st70	2,108	2,108	**2,108**	*	**2,108**	**0.40**	**2,108**	*	0.31	**2,108**	*	9.65	**2,108**	*	**2,108**	*	0.49
eil76	2,467	2,467	**2,467**	*	**2,467**	**0.40**	**2,467**	*	0.36	**2,467**	*	20.48	**2,467**	*	**2,467**	*	2.96
pr76	2,430	2,430	**2,430**	*	**2,430**	**0.20**	**2,430**	*	0.57	**2,430**	*	20.43	**2,430**	*	**2,430**	*	1.07
gr96	3,170	3,170	**3,170**	*	**3,170**	61.50	3,166	0.13	1.41	3,166	0.13	15.22	**3,170**	*	**3,170**	*	**5.66**
rat99	2,908	2,908	**2,908**	*	**2,908**	4.90	–	–	–	–	–	–	**2,908**	*	**2,908**	*	**3.01**
kroA100	3,211	3,211	**3,211**	*	**3,211**	63.30	3,180	0.97	0.38	**3,211**	*	32.31	**3,211**	*	**3,211**	*	**1.81**
kroB100	2,804	2,804	**2,804**	*	**2,804**	0.60	2,785	0.68	0.51	**2,804**	*	35.83	**2,804**	*	**2,804**	*	**0.35**
kroC100	3,155	3,155	**3,155**	*	**3,155**	**1.50**	**3,155**	*	0.44	**3,155**	*	34.67	**3,155**	*	**3,155**	*	1.82
kroD100	3,167	3,167	**3,167**	*	**3,167**	10.70	3,141	0.82	0.58	**3,167**	*	31.08	**3,167**	*	**3,167**	*	**0.70**
kroE100	3,049	3,049	**3,049**	*	**3,049**	1.50	**3,049**	*	0.47	**3,049**	*	31.96	**3,049**	*	**3,049**	*	**1.36**
rd100	2,926	2,926	**2,926**	*	**2,926**	113.20	2,923	0.10	0.48	**2,926**	*	16.35	**2,926**	*	**2,926**	*	**23.20**
eil101	3,345	3,345	**3,345**	*	**3,345**	29.80	**3,345**	*	0.56	**3,345**	*	28.61	**3,345**	*	**3,345**	*	**1.37**
lin105	2,986	2,986	**2,986**	*	**2,986**	51.90	2,973	0.44	2.09	**2,986**	*	38.24	**2,986**	*	**2,986**	*	**16.02**
pr107	1,877	1,877	**1,877**	*	**1,877**	**660.90**	1,802	4.00	0.82	**1,877**	*	65.16	**1,877**	*	**1,877**	*	3,297.37
gr120	3,779	3,779	**3,779**	*	**3,779**	**1.50**	3,748	0.82	1.36	3,777	0.05	37.94	**3,779**	*	**3,779**	*	2.65
pr124	3,557	3,557	**3,557**	*	**3,557**	**1,021.50**	3,455	2.87	0.88	**3,557**	*	99.87	**3,557**	*	**3,557**	*	4,507.38
bier127	2,365	2,365	**2,365**	*	**2,365**	79.90	2,361	0.17	2.62	2,361	0.17	49.9	**2,365**	*	**2,365**	*	**40.07**
pr136	4,390	4,390	**4,390**	*	**4,390**	86.70	**4,390**	*	1.13	**4,390**	*	61.84	**4,390**	*	**4,390**	*	**30.50**
gr137	3,954	3,954	**3,954**	*	**3,954**	**8.60**	**3,954**	*	1.88	**3,954**	*	637.09	**3,954**	*	**3,954**	*	14.01
pr144	3,745	3,745	**3,745**	*	**3,745**	**112.60**	3,700	1.20	2.41	3,744	0.03	112.92	**3,745**	*	**3,745**	*	116.68
kroA150	5,039	5,039	**5,039**	*	**5,039**	330.70	5,019	0.40	1.07	5,037	0.04	104.23	**5,039**	*	**5,039**	*	**46.43**
kroB150	5,314	5,314	**5,314**	*	**5,314**	107.60	**5,314**	*	1.04	**5,314**	*	63.05	**5,314**	*	**5,314**	*	**28.53**
pr152	3,905	3,905	**3,905**	*	**3,905**	1,122.40	3,902	0.08	3.63	3,539	9.37	184.38	**3,905**	*	**3,905**	*	**83.51**
u159	5,272	5,272	**5,272**	*	**5,272**	52.20	**5,272**	*	0.95	**5,272**	*	94.27	**5,272**	*	**5,272**	*	**8.59**
rat195	6,195	6,195	**6,195**	*	**6,195**	49.90	–	–	–	–	–	–	**6,195**	*	**6,195**	*	**33.56**
d198	6,320	6,320	**6,320**	*	**6,320**	**286.10**	6,290	0.47	7.15	**6,320**	*	105.7	**6,320**	*	**6,320**	*	461.18
kroA200	6,123	6,123	**6,123**	*	**6,123**	122.30	6,114	0.15	1.72	6,118	0.08	232.2	**6,123**	*	**6,123**	*	**92.41**
kroB200	6,266	6,266	**6,266**	*	**6,266**	40.10	6,213	0.85	1.78	**6,266**	*	188.77	**6,266**	*	**6,266**	*	**3.87**
gr202	8,616	8,616	**8,616**	*	**8,616**	**224.80**	8,605	0.13	10.45	8,564	0.60	57.88	**8,616**	*	**8,616**	*	315.26
ts225	7,575	7,575	**7,575**	*	**7,575**	171.20	**7,575**	*	1.14	**7,575**	*	450.25	**7,575**	*	**7,575**	*	**6.62**
tsp225	7,740	7,740	**7,740**	*	**7,740**	150.30	–	–	–	–	–	–	**7,740**	*	**7,740**	*	**38.61**
pr226	6,993	6,993	**6,993**	*	**6,993**	**32.60**	6,908	1.22	8.01	**6,993**	*	177.59	**6,993**	*	**6,993**	*	1,170.00
gr229	6,328	6,328	**6,328**	*	**6,328**	**10.20**	6,297	0.49	11.66	**6,328**	*	1,298.8	**6,328**	*	**6,328**	*	42.63
gil262	9,246	9,246	**9,246**	*	**9,246**	133.40	9,094	1.64	3.94	9,210	0.39	649.54	**9,246**	*	**9,246**	*	**83.29**
pr264	8,137	8,137	**8,137**	*	**8,137**	**20.70**	8,068	0.85	3.63	**8,137**	*	357.8	**8,137**	*	**8,137**	*	186.59
a280	9,774	9,774	**9,774**	*	**9,774**	213.30	8,684	11.15	3.22	8,789	10.08	378.8	**9,774**	*	**9,774**	*	**126.80**
pr299	10,343	10,343	**10,343**	*	**10,343**	**363.60**	9,959	3.71	3.95	10,233	1.06	549.11	**10,343**	*	**10,343**	*	913.13
lin318	10,368	10,368	**10,368**	*	**10,368**	534.80	10,273	0.92	6.33	10,337	0.30	528.2	**10,368**	*	**10,368**	*	**327.58**
rd400	13,223	13,223	**13,223**	*	**13,223**	293.20	13,088	1.02	7.74	13,122	0.76	727.58	**13,223**	*	**13,223**	*	**214.40**
average				*		**149.66**		0.85	2.35		0.55	180.55		*		*	272.43

Table B.20
Generation 3, $n > 400$.

Instance	Best		FST				EA4OP			ALNS			RB&C				
	LB	UB	LB	GGap	UB	Time	LB	GGap	Time	LB	GGap	Time	LB	GGap	UB	OGap	Time
fl417	14,220	14,220	**14,220**	*	**14,220**	**6,227.60**	14,186	0.24	12.45	**14,220**	*	1,131.05	14,219	0.01	14,387	1.17	18,000.00
gr431	10,911	10,911	**10,911**	*	**10,911**	**1,046.90**	10,817	0.86	54.50	10,907	0.04	2,411.45	**10,911**	*	**10,911**	*	7,814.17
pr439	15,176	15,296	15,160	0.11	**15,296**	**18,000.00**	15,097	0.52	10.96	15,080	0.63	1,328.74	**15,176**	*	15,331	1.01	**18,000.00**
pcb442	14,819	14,819	**14,819**	*	14,839	18,000.00	14,522	2.00	6.58	14,695	0.84	1,192.19	**14,819**	*	**14,819**	*	**11,574.76**
d493	25,167	25,188	**25,167**	*	**25,188**	**18,000.00**	24,981	0.74	19.18	24,849	1.26	3,829.32	**25,167**	*	25,195	0.11	**18,000.00**
att532	15,498	15,498	**15,498**	*	**15,498**	933.20	15,342	1.01	22.75	15,335	1.05	4,533.36	**15,498**	*	**15,498**	*	**318.44**
ali535	9,414	9,472	.	.	.	.	9,328	0.91	94.09	9,308	1.13	13,313.5	**9,414**	*	**9,472**	0.61	**18,000.00**
pa561	14,482	14,482	**14,482**	*	**14,482**	10,543.80	–	–	–	–	–	–	**14,482**	*	**14,482**	*	**2,539.41**
u574	20,064	20,064	**20,064**	*	**20,064**	**1,409.30**	19,691	1.86	19.77	19,841	1.11	1,671.01	**20,064**	*	**20,064**	*	2,693.59
rat575	20,109	20,109	**20,109**	*	**20,109**	1,426.50	–	–	–	–	–	–	**20,109**	*	**20,109**	*	**929.99**
p654	24,492	24,518	**24,492**	*	31,914	**18,000.00**	24,130	1.48	18.54	24,427	0.27	7,543.02	**24,492**	*	**24,518**	0.11	**18,000.00**
d657	24,562	24,562	**24,562**	*	**24,562**	**4,053.30**	23,772	3.22	21.89	23,829	2.98	4,600.87	**24,562**	*	**24,562**	*	8,777.39
gr666	17,023	17,048	17,020	0.02	**17,048**	**18,000.00**	16,902	0.71	143.87	16,709	1.84	2,734.75	**17,023**	*	17,060	0.22	**18,000.00**
u724	28,348	28,348	**28,348**	*	**28,348**	**5,870.60**	27,932	1.47	29.26	28,033	1.11	12,058.6	**28,348**	*	**28,348**	*	10,332.54
rat783	27,566	27,566	**27,566**	*	**27,566**	7,232.30	–	–	–	–	–	–	**27,566**	*	**27,566**	*	**3,812.98**
dsj1000	31,434	31,454	.	.	.	.	30,943	1.56	79.18	31,040	1.25	15,962	**31,434**	*	**31,454**	0.06	**18,000.00**
pr1002	39,526	39,526	39,449	0.19	39,545	18,000.00	38,762	1.93	47.30	38,502	2.59	18,000	**39,526**	*	**39,526**	*	**13,955.69**
u1060	37,492	37,569	.	.	.	.	36,570	2.46	75.88	36,598	2.38	18,000	**37,492**	*	**37,569**	0.20	**18,000.00**
vm1084	37,669	37,669	37,653	0.04	37,694	18,000.00	37,508	0.43	54.21	37,178	1.30	3,286.89	**37,669**	*	**37,669**	*	**8,710.50**
pcb1173	41,257	41,257	.	.	.	.	40,069	2.88	66.16	40,513	1.80	18,000	**41,257**	*	**41,257**	*	**15,133.74**
d1291	41,509	42,153	30,106	27.47	.	**18,000.00**	38,132	8.14	299.87	39,919	3.83	18,000	**41,509**	*	**42,153**	1.53	**18,000.00**
rl1304	41,881	42,075	40,478	3.35	.	**18,000.00**	41,214	1.59	81.11	41,679	0.48	18,000	**41,881**	*	**42,075**	0.46	**18,000.00**
rl1323	47,213	47,384	44,458	5.84	.	**18,000.00**	46,641	1.21	93.53	45,500	3.63	8,544.44	**47,213**	*	**47,384**	0.36	**18,000.00**
nrw1379	42,920	42,975	.	.	.	.	–	–	–	–	–	–	**42,920**	*	**42,975**	0.13	**18,000.00**
fl1400	57,470	59,491	54,792	4.66	67,053	**18,000.00**	57,226	0.42	599.81	**57,470**	*	18,000	54,661	4.89	**59,491**	8.12	**18,000.00**
u1432	47,778	47,895	.	.	.	.	46,657	2.35	138.02	47,242	1.12	18,000	**47,778**	*	**47,895**	0.24	**18,000.00**
fl1577	45,935	48,809	.	.	.	.	45,692	0.53	295.62	**45,935**	*	18,000	45,768	0.36	**48,809**	6.23	**18,000.00**
d1655	62,048	62,945	51,168	17.53	.	**18,000.00**	58,728	5.35	674.25	60,956	1.76	18,000	**62,048**	*	**62,945**	1.43	**18,000.00**
vm1748	71,885	72,010	68,979	4.04	.	**18,000.00**	70,958	1.29	225.29	71,244	0.89	18,000	**71,885**	*	**72,010**	0.17	**18,000.00**
u1817	63,639	67,670	52,186	18.00	.	**18,000.00**	**63,639**	*	1,302.35	63,016	0.98	18,000	63,618	0.03	**67,670**	5.99	**18,000.00**
rl1889	70,065	71,106	43,374	38.09	.	**18,000.00**	68,422	2.34	244.97	68,096	2.81	18,000	**70,065**	*	**71,106**	1.46	**18,000.00**
d2103	82,787	82,973	76,035	8.16	.	**18,000.00**	77,333	6.59	1,168.90	81,081	2.06	18,000	**82,787**	*	**82,973**	0.22	**18,000.00**
u2152	74,007	78,066	52,091	29.61	.	**18,000.00**	73,400	0.82	1,619.61	72,733	1.72	18,000	**74,007**	*	**78,066**	5.20	**18,000.00**
u2319	79,351	81,050	**79,351**	*	81,619	**18,000.00**	78,113	1.56	569.76	79,130	0.28	18,000	79,343	0.01	**81,050**	2.11	**18,000.00**
pr2392	85,409	90,261	60,225	29.49	.	**18,000.00**	84,094	1.54	422.73	85,084	0.38	18,000	**85,409**	*	**90,261**	5.38	**18,000.00**
pcb3038	106,928	112,006	96,356	9.89	.	**18,000.00**	104,667	2.11	917.39	105,337	1.49	18,000	**106,928**	*	**112,006**	4.53	**18,000.00**
fl3795	97,707	116,792	.	.	.	.	**97,707**	*	3,158.89	95,580	2.18	18,000	89,218	8.69	**116,792**	23.61	**18,000.00**
fnl4461	146,995	152,562	.	.	.	.	–	–	–	–	–	–	**146,995**	*	**152,562**	3.65	**18,000.00**
rl5915	203,695	217,366	.	.	.	.	199,336	2.14	5,593.23	201,814	0.92	18,000	**203,695**	*	**217,366**	6.29	**18,000.00**
rl5934	212,021	229,405	.	.	.	.	207,385	2.19	5,881.87	203,667	3.94	18,000	**212,021**	*	**229,405**	7.58	**18,000.00**
pla7397	322,285	334,885	.	.	.	.	320,744	0.48	18,000.00	312,645	2.99	18,000	**322,285**	*	**334,885**	3.76	**18,000.00**
average				6.78		**13,749.78**		1.80	1,168.44		1.47	12,837.26		0.34		2.24	14,843.74

Table B.21
Minimum, average and maximum values of lower bounds and upper bounds for RB&C, $n > 400$.

Instance	Generation 1						Generation 2						Generation 3					
	LB			UB			LB			UB			LB			UB		
	min	av.	max	min	av.	max	min	av.	max	min	av.	max	min	av.	max	min	av.	max
fl417	227	228	228	231	233	239	11,914	11,928	11,933	12,387	12,383	12,965	14,210	14,215	14,219	14,387	14,386	14,388
gr431	350	350	350	350	350	350	18,318	18,318	18,318	18,318	18,318	18,318	10,896	10,906	10,911	10,911	10,919	10,938
pr439	313	313	313	313	313	313	16,171	16,171	16,171	16,171	16,171	16,171	15,150	15,161	15,176	15,331	15,341	15,361
pcb442	251	251	251	251	251	251	14,429	14,476	14,484	14,484	14,496	14,568	14,806	14,815	14,819	14,819	14,826	14,845
d493	320	320	320	320	320	320	16,988	16,993	16,995	17,007	17,010	17,011	25,150	25,164	25,167	25,195	25,211	25,216
att532	363	363	363	363	363	363	19,610	19,622	19,635	19,800	19,802	19,806	15,498	15,498	15,498	15,498	15,498	15,498
ali535	425	425	425	426	427	427	21,949	21,952	21,954	21,973	21,980	21,993	9,257	9,389	9,414	9,472	9,472	9,473
pa561	357	357	357	357	357	357	19,576	19,576	19,576	19,576	19,576	19,576	14,482	14,482	14,482	14,482	14,482	14,482
u574	354	354	354	354	354	354	19,351	19,351	19,351	19,351	19,351	19,351	20,064	20,064	20,064	20,064	20,064	20,064
rat575	322	322	322	322	322	322	18,251	18,251	18,251	18,251	18,251	18,251	20,109	20,109	20,109	20,109	20,109	20,109
p654	311	329	342	396	420	430	16,760	17,153	17,753	22,248	23,839	25,772	24,492	24,492	24,492	24,518	24,817	26,403
d657	386	386	386	386	386	386	21,503	21,503	21,503	21,503	21,503	21,503	24,545	24,557	24,562	24,562	24,574	24,605
gr666	503	503	503	503	503	503	26,467	26,499	26,514	26,569	26,575	26,616	16,993	17,015	17,023	17,060	17,065	17,069
u724	439	439	439	439	439	439	24,064	24,188	24,223	24,223	24,235	24,259	28,287	28,341	28,348	28,348	28,351	28,371
rat783	438	438	438	438	438	438	25,291	25,412	25,474	25,474	25,499	25,511	27,566	27,566	27,566	27,566	27,566	27,566
dsj1000	656	656	656	656	656	656	35,762	35,803	35,835	35,915	35,918	35,922	31,424	31,428	31,434	31,454	31,459	31,460
pr1002	606	606	606	606	606	606	32,950	33,005	33,030	33,092	33,088	33,115	39,504	39,521	39,526	39,526	39,536	39,548
u1060	656	658	660	660	661	661	35,911	36,042	36,151	36,291	36,267	36,309	37,414	37,454	37,492	37,569	37,574	37,577
vm1084	777	777	777	777	777	777	40,720	40,747	40,777	40,952	40,959	40,962	37,667	37,669	37,669	37,669	37,679	37,702
pcb1173	675	675	675	675	675	675	36,972	37,013	37,035	37,100	37,104	37,108	41,252	41,256	41,257	41,257	41,300	41,335
d1291	711	714	715	715	715	715	37,649	37,724	37,778	37,854	37,856	37,864	40,508	41,021	41,509	42,153	42,176	42,186
rl1304	802	802	802	802	802	802	41,840	42,171	42,275	42,359	42,363	42,369	41,548	41,728	41,881	42,075	42,085	42,094
rl1323	812	814	814	814	814	816	43,131	43,324	43,377	43,450	43,449	43,466	46,886	47,078	47,213	47,384	47,402	47,417
nrw1379	811	814	815	817	816	817	46,317	46,542	46,676	46,787	46,790	46,792	42,869	42,889	42,920	42,975	42,978	42,983
fl1400	944	993	1,003	1,084	1,100	1,122	49,774	51,622	54,124	64,298	64,505	64,717	52,471	53,677	54,661	59,491	60,643	61,317
u1432	744	748	754	764	765	765	46,450	46,730	46,946	47,018	47,025	47,037	47,005	47,575	47,778	47,895	48,003	48,644
fl1577	892	894	897	900	901	902	43,627	44,510	45,326	50,154	51,351	51,912	41,744	43,322	45,768	48,809	49,706	51,334
d1655	757	854	922	924	928	959	42,470	43,682	46,158	53,083	53,216	53,470	51,376	56,559	62,048	62,945	63,527	65,094
vm1748	1,267	1,272	1,276	1,282	1,282	1,282	67,499	67,750	68,042	68,303	68,311	68,320	71,678	71,775	71,885	72,010	72,023	72,029
u1817	978	982	983	983	983	984	53,152	53,817	54,245	54,554	54,557	54,563	56,826	59,419	63,618	67,670	67,837	67,982
rl1889	1,209	1,218	1,226	1,226	1,229	1,229	62,009	63,216	63,308	64,425	64,438	64,448	69,194	69,592	70,065	71,106	71,132	71,148
d2103	1,197	1,198	1,200	1,200	1,202	1,203	63,350	63,397	63,426	63,426	63,428	63,428	59,444	79,489	82,787	82,973	82,980	82,985
u2152	1,148	1,150	1,151	1,151	1,151	1,152	61,913	63,667	64,649	64,775	64,779	64,783	69,556	70,859	74,007	78,066	78,395	78,594
u2319	1,161	1,164	1,170	1,171	1,171	1,171	80,733	80,925	80,914	81,139	81,143	81,147	78,944	79,144	79,343	81,050	81,346	81,613
pr2392	1,287	1,301	1,316	1,415	1,424	1,433	69,108	70,788	72,843	78,237	78,486	78,683	84,604	84,895	85,409	90,261	90,849	91,279
pcb3038	1,713	1,718	1,727	1,730	1,730	1,730	97,624	97,789	97,902	97,995	97,998	98,000	105,837	106,473	106,928	112,006	112,433	112,698
fl3795	1,446	1,667	1,965	2,249	2,303	2,346	83,307	92,105	98,998	142,895	143,273	144,004	85,258	87,174	89,218	116,792	118,555	119,805
fnl4461	2,514	2,523	2,541	2,570	2,570	2,570	145,486	146,114	147,109	150,189	150,195	150,200	144,953	145,494	146,995	152,562	152,950	153,237
rl5915	3,475	3,549	3,593	3,786	3,786	3,787	176,501	180,424	184,424	197,729	199,104	202,253	196,346	199,558	203,695	217,366	219,184	220,548
rl5934	3,528	3,595	3,632	3,752	3,752	3,753	179,939	182,320	187,034	196,805	196,899	197,214	203,602	206,779	212,021	229,405	231,124	232,486
pla7397	5,198	5,230	5,289	5,657	5,664	5,668	273,669	278,230	281,977	297,246	297,530	297,815	316,534	319,604	322,285	334,885	335,307	335,551

Table C.22
Common parameters used by the RB&C algorithm. The values for the different parameters are manually chosen based on our preliminary experiments and previous values in the literature.

Parameter	Value	Description
ZERO	10^{-7}	Sensibility of fractional numbers
ADD_CUT_BATCH	250	Maximum number of cuts added to the LP_0 at once
ADD_MIN_VIOL	10^{-6}	Minimum violation of a cut to include it in the LP_0
SUBLOOP_IMPR	1%	Minimum improvement to repeat the subloops (Section 4.2)
ADD_SEC_PER_SET	50	Amount of SECs considered for each subset (Section 4.2)
ADD_PATH_MAX	500	Maximum cuts for Path inequalities separation (Section 3.6)
ADD_EGH_EPSILON	0.3	Epsilon value for the EGH blossom heuristic (Section 4.2.2)
PRICE_MAX_ADD	200	Maximum number of variables added to the LP_0 (Section 4.3)
PRICE_RC_THRESH	10^{-5}	Minimum penalty of a variable to add to the LP_0 (Section 4.3)
DEL_DUST_VAR	10^{-3}	Minimum y value to consider an edge as active (Section 4.3)
DEL_DUST_CUT	10^{-3}	Maximum slack value to consider a cut as active (Section 4.3)
DEL_MAX_AGE_CUT	5	Consecutive inactivity to delete a cut from the LP_0 (Section 4.3)
DEL_MAX_AGE_VAR	100	Consecutive inactivity to delete an edge from the LP_0 (Section 4.3)
XHEUR_GREEDY_XMIN	0.3	Use arcs larger than this value in PB primal heuristic (Section 4.5)
XHEUR_EA4OP_POP_SIZE	10	Population size for EA4OP (Section 4.5)
XHEUR_EA4OP_D2D	5	Iterations before checking feasibility in EA4OP (Section 4.5)
XHEUR_EA4OP_NPAR	3	Number of parents preselected in EA4OP (Section 4.5)

Table D.23
Results of the REFERENCE and alternative configurations for RB&C on the root node. The values are the mean relative differences to the best overall achieved, in percentages. In italics, the values of the alternatives that are worse than those obtained by the REFERENCE configuration are shown.

Strategy	Gap								
	Gen1			Gen2			Gen3		
	LB	UB	Time	LB	UB	Time	LB	UB	Time
REFERENCE	0.33	0.00	41.90	0.18	0.01	49.90	0.26	0.01	43.55
-SRK	0.03	0.00	*63.36*	*0.25*	0.01	*71.30*	0.19	*0.02*	*62.04*
SRK=C1C2S3	*0.35*	0.00	40.82	*0.28*	0.01	36.53	0.16	0.00	43.16
-CC STRATS	*0.34*	0.00	31.76	*0.23*	*0.02*	40.09	*0.31*	0.01	41.26
-EPH	*0.38*	*0.13*	10.72	*0.34*	*0.13*	11.15	*0.35*	*0.21*	15.28
-EGH	*0.37*	*0.02*	33.24	0.15	*0.04*	37.85	*0.36*	*0.02*	*48.14*
+FST	0.20	0.00	*57.27*	0.18	0.00	*60.14*	0.18	0.00	*59.61*
EPH EGH=FST	*0.54*	*0.09*	6.85	0.18	*0.10*	26.66	*0.41*	*0.23*	*44.72*
-CYCLE COVER	*0.45*	0.00	34.50	0.11	0.01	49.66	0.18	0.01	*48.60*
-EDGE COVER	0.11	0.00	*43.48*	0.17	0.01	*52.93*	*0.27*	0.01	*49.51*
-PATH	*0.58*	0.00	38.27	0.15	0.01	39.81	0.21	*0.03*	*44.86*
+VERTEX	0.24	0.00	*46.06*	0.18	0.01	*52.46*	*0.37*	0.01	*47.95*
SEP=TWO	0.25	0.00	*61.07*	0.18	0.01	*67.42*	0.26	0.01	*61.98*

Appendix C. Common parameters

In Table C.22, we detail the values of the common parameters for all the simulations of the RB&C algorithm. They were chosen inspired by the parameters used in Applegate et al. (2007) and our preliminary experiments for the OP.

Appendix D. Initialization of the RB&C algorithm: detailed results

D1. Evaluation of components in the root of the tree

In Table D.23 we summarize the mean relative difference to the best achieved LB and UB and the mean relative difference to the best-performing configuration in terms of running time for RB&C on the root node for the instances analyzed in Section 5.1. The REFERENCE configuration is never dominated by any other alternative configuration in the multi-objective sense. Furthermore, we can observe that the LB on the root node of the REFERENCE outperforms most of the alternative configurations and the UB on the root node of the REFERENCE configuration is not worse than almost all the alternative configurations.

D2. Lower bounds in the root of the tree

In Tables D.24 and D.25 we reproduce for medium-sized and large-sized instances, respectively, the relative difference (in percentage) between the best known solution obtained with RB&C and (i) the EA4OP meta-heuristic implemented during the initialization (EA4OP column) and (ii) the best lower bound of the initialization (root column) in the first and second columns, respectively. In the third ones (status column) we show: OP, when the optimality of the root solution is verified in the initialization phase (i.e., initial lower and upper bounds are equal); BR, when the branching is performed; and TL, when the RB&C algorithm stops after the initialization phase because running time limit is achieved. This way, we can compare the initial lower bounds obtained after the EA4OP and the whole initialization phase with respect to the final lower bounds. In the last row, we report the average relative differences by generation. Globally, in relation with the best known solution, the EA4OP is at 12.64% and 25.46% for medium-sized and large-sized instances, respectively, while the lower bound obtained at the end of the whole initial phase is at 0.63% and 1.64%, respectively. Finally, note that in 27 out of 135 medium-sized instances, optimality is verified in the initial phase, and for all the large-sized instances branching is required.

Table D.24
Relative difference between the best known solution and (i) the output of the initial EA4OP meta-heuristic (EA4OP), (ii) lower bound after the initial phase of the RB&C algorithm (root) and status after the initialization (OP: optimality verified, BR: branching required) for medium-sized instances ($n \leq 400$).

Instance	Generation 1			Generation 2			Generation 3		
	EA4OP	root	status	EA4OP	root	status	EA4OP	root	status
att48	3.23	0.00	OP	10.08	0.00	OP	4.58	0.48	BR
gr48	9.68	0.00	OP	4.20	0.91	BR	8.45	0.07	BR
hk48	6.67	0.00	OP	9.85	0.00	BR	8.45	0.00	BR
eil51	10.34	0.00	OP	8.54	0.36	BR	2.79	0.07	BR
berlin52	2.70	0.00	OP	6.96	0.00	BR	6.95	0.19	BR
brazil58	6.52	0.00	OP	5.50	0.00	OP	0.00	0.00	OP
st70	2.33	0.00	OP	6.21	0.52	BR	14.71	0.90	BR
eil76	17.02	0.00	OP	8.20	0.00	BR	7.42	0.00	BR
pr76	6.12	0.00	OP	8.35	0.00	BR	22.84	0.41	BR
gr96	9.38	0.00	OP	3.03	0.06	BR	8.71	0.54	BR
rat99	11.54	3.85	BR	8.87	0.54	BR	8.84	0.79	BR
kroA100	10.71	0.00	BR	17.96	0.00	BR	8.66	0.06	BR
kroB100	15.52	1.72	BR	15.24	0.25	BR	11.77	0.00	BR
kroC100	21.43	0.00	OP	14.66	0.92	BR	13.12	0.00	BR
kroD100	13.56	0.00	OP	12.73	0.24	BR	12.88	0.00	BR
kroE100	12.28	3.51	BR	5.83	2.49	BR	14.50	0.52	BR
rd100	11.48	0.00	OP	11.97	0.42	BR	2.84	0.31	BR
eil101	4.69	0.00	OP	11.46	0.57	BR	17.73	0.06	BR
lin105	28.79	0.00	BR	2.57	0.31	BR	6.80	1.11	BR
pr107	0.00	0.00	OP	0.00	0.00	OP	6.45	0.43	BR
gr120	10.67	1.33	BR	7.07	0.75	BR	13.18	0.05	BR
pr124	20.00	0.00	OP	21.27	0.41	BR	13.13	4.69	BR
bier127	9.71	0.00	BR	6.67	0.00	BR	9.26	0.42	BR
pr136	7.04	0.00	OP	18.10	0.00	BR	15.60	0.39	BR
gr137	19.75	0.00	OP	11.64	1.98	BR	12.85	0.03	BR
pr144	33.77	0.00	BR	11.87	1.67	BR	10.28	0.93	BR
kroA150	10.47	1.16	BR	8.78	1.00	BR	8.77	0.04	BR
kroB150	13.79	3.45	BR	13.27	0.06	BR	16.80	0.06	BR
pr152	3.90	1.30	BR	7.46	0.23	BR	21.51	2.77	BR
u159	9.68	1.08	BR	7.84	0.38	BR	13.56	0.34	BR
rat195	18.63	2.94	BR	14.16	1.05	BR	18.95	0.26	BR
d198	9.76	2.44	BR	4.15	1.93	BR	10.59	2.22	BR
kroA200	11.11	2.56	BR	21.92	0.00	BR	18.72	0.03	BR
kroB200	11.76	0.84	BR	15.06	0.09	BR	24.93	0.00	BR
gr202	11.03	0.00	BR	9.77	0.26	BR	11.11	0.32	BR
ts225	8.87	0.00	BR	11.40	0.28	BR	16.55	0.00	BR
tsp225	20.93	1.55	BR	13.47	1.07	BR	19.46	0.40	BR
pr226	30.16	2.38	BR	8.17	1.58	BR	5.65	0.34	BR
gr229	9.09	0.00	OP	15.95	0.01	BR	14.66	0.00	BR
gil262	22.78	0.00	OP	7.62	0.30	BR	21.11	0.05	BR
pr264	44.70	0.00	OP	16.43	0.00	OP	28.34	0.02	BR
a280	20.41	4.76	BR	20.93	0.95	BR	16.79	0.00	BR
pr299	19.14	2.47	BR	10.26	4.02	BR	28.50	0.01	BR
lin318	16.59	0.00	OP	13.30	0.66	BR	17.29	0.57	BR
rd400	32.78	0.84	BR	21.63	0.21	BR	18.23	0.33	BR
average	13.83	0.85		10.90	0.59		13.21	0.45	

Table D.25
Relative difference between the best known solution and (i) the output of the initial EA4OP meta-heuristic (EA4OP), (ii) lower bound after the initial phase of the RB&C algorithm (root) and status after the initialization (BR: branching required or TL: time limit achieved) for large-sized instances ($n > 400$).

Instance	Generation 1			Generation 2			Generation 3		
	EA4OP	root	status	EA4OP	root	status	EA4OP	root	status
fl417	37.72	1.32	BR	28.74	3.44	BR	13.55	0.12	BR
gr431	3.43	0.29	BR	4.21	0.22	BR	17.23	0.87	BR
pr439	18.21	0.00	BR	23.32	1.00	BR	11.57	1.88	BR
pcb442	23.51	0.80	BR	19.57	0.99	BR	27.75	0.40	BR
d493	29.06	0.00	BR	21.35	0.34	BR	12.64	0.32	BR
att532	15.43	0.28	BR	24.28	0.81	BR	22.51	0.57	BR
ali535	10.35	0.71	BR	8.06	0.10	BR	12.26	3.57	BR
pa561	23.81	1.12	BR	20.18	0.17	BR	29.25	1.43	BR
u574	25.99	0.56	BR	13.85	0.91	BR	29.91	1.16	BR
rat575	17.39	0.31	BR	20.86	1.64	BR	29.33	0.07	BR
p654	15.79	0.58	BR	16.99	2.86	BR	4.94	0.63	BR
d657	17.36	0.52	BR	14.69	0.44	BR	30.35	0.22	BR
gr666	19.88	0.20	BR	8.87	0.36	BR	19.12	0.38	BR
u724	15.95	1.37	BR	16.76	0.32	BR	26.64	0.64	BR
rat783	30.59	1.83	BR	22.01	1.40	BR	17.90	0.48	BR
dsj1000	23.17	0.91	BR	20.76	0.51	BR	23.86	0.06	BR
pr1002	31.68	0.99	BR	24.54	1.40	BR	25.50	0.02	BR
u1060	18.33	2.88	BR	26.22	1.45	BR	25.13	0.19	BR
vm1084	10.30	0.13	BR	9.38	0.38	BR	24.24	0.02	BR
pcb1173	27.41	2.07	BR	28.03	0.71	BR	36.55	0.27	BR
d1291	35.10	15.94	BR	35.80	0.47	BR	42.60	7.48	BR
rl1304	27.43	0.37	BR	24.70	0.73	BR	11.78	3.08	BR
rl1323	21.25	1.23	BR	16.83	0.33	BR	25.22	0.62	BR
nrw1379	27.24	0.98	BR	27.20	1.30	BR	29.89	0.17	BR
fl1400	24.53	0.00	TL	9.83	0.00	TL	9.70	0.37	TL
u1432	16.84	2.79	BR	25.09	1.19	BR	20.45	0.08	BR
fl1577	43.59	0.78	BR	33.00	0.00	TL	21.54	0.00	TL
d1655	40.13	16.92	BR	27.15	2.17	TL	43.19	11.23	BR
vm1748	19.04	0.94	BR	11.27	0.87	BR	18.40	0.38	BR
u1817	33.37	9.56	BR	31.30	13.48	BR	32.12	0.00	TL
rl1889	18.84	2.85	BR	19.14	2.75	BR	15.79	0.85	BR
d2103	42.17	0.83	BR	38.69	0.10	BR	48.51	26.26	BR
u2152	39.53	12.34	BR	36.25	7.56	BR	38.13	0.00	TL
u2319	12.14	1.11	BR	19.61	0.56	BR	25.29	0.00	TL
pr2392	19.00	0.38	TL	17.66	0.00	TL	31.47	0.00	TL
pcb3038	29.42	0.81	BR	32.20	0.25	BR	28.66	0.00	TL
fl3795	37.05	0.00	TL	25.00	0.00	TL	28.01	0.00	TL
fnl4461	29.75	1.22	BR	30.40	0.88	BR	34.21	0.00	TL
rl5915	34.43	0.00	TL	26.99	0.00	TL	18.91	0.00	TL
rl5934	40.42	0.00	TL	31.44	0.00	TL	28.30	0.00	TL
pla7397	37.17	0.00	TL	29.98	0.00	TL	38.36	0.00	TL
average	25.46	2.10		22.49	1.27		25.14	1.56	

In Tables D.26 and D.27 we compare for medium- and large-sized instances, respectively, the number of cases in which each Branch-and-Bound algorithm obtains better solutions in terms of quality and running times for the root node. We have again restricted to those instances in which FST actually returns a solution. In general, the RB&C outperforms the results of the FST for large-sized instances, while the latter obtains better root node results for smaller instances, what clearly shows the scaling behaviour of our approach.

Table D.26
Comparison for the root node of medium-sized instances of the number of obtained optimal solutions (OPT), number of best-known solutions (LB), number of best upper bounds (UB) and number of instances in which the considered B&C algorithm is faster than the competitor (Time).

		OPT		LB		UB		Time	
	#	FST	RB&C	FST	RB&C	FST	RB&C	FST	RB&C
Gen1	45	**34**	22	**11**	3	**14**	0	**37**	5
Gen2	45	**19**	4	**30**	3	**40**	0	**25**	15
Gen3	45	**17**	1	**28**	7	**42**	1	**22**	21
All	135	**70**	27	**69**	13	**96**	1	**84**	41

Table D.27
Comparison for the root node of large-sized instances of the number of obtained optimal solutions (OPT), number of best-known solutions (LB), number of best upper bounds (UB) and number of instances in which the considered B&C algorithm is faster than the competitor (Time) in the instances that FST does return a solution.

		OPT		LB		UB		Time	
	#	FST	RB&C	FST	RB&C	FST	RB&C	FST	RB&C
Gen1	21	1	1	2	**19**	4	**8**	5	**14**
Gen2	22	0	0	5	**17**	**12**	7	1	**19**
Gen3	29	0	0	8	**21**	10	**16**	4	**19**
All	72	1	1	15	**57**	26	**31**	10	**52**

References

Angelelli, E., Archetti, C., Filippi, C., & Vindigni, M. (2017). The probabilistic orienteering problem. *Computers & Operations Research, 81*, 269–281.

Angelelli, E., Bazgan, C., Speranza, M. G., & Tuza, Z. (2014). Complexity and approximation for traveling salesman problems with profits. *Theoretical Computer Science, 531*, 54–65.

Applegate, D. L., Bixby, R. E., Chvatal, V., & Cook, W. J. (2007). *The traveling salesman problem: A computational study (princeton series in applied mathematics)*. Princeton, NJ, USA: Princeton University Press.

Archetti, C., Corberán, A., Plana, I., Sanchis, J. M., & Speranza, M. G. (2016). A branch-and-cut algorithm for the orienteering arc routing problem. *Computers & Operations Research, 66*, 95–104.

Archetti, C., Speranza, M. G., Corberán, A., Sanchis, J. M., & Plana, I. (2014). The team orienteering arc routing problem. *Transportation Science, 48*(3), 442–457.

Balas, E. (1975). Facets of the knapsack polytope. *Mathematical Programming, 8*(1), 146–164.

Balas, E. (1989). The prize collecting traveling salesman problem. *Networks, 19*(6), 621–636.

Bauer, P. (1997). The circuit polytope: Facets. *Mathematics of Operations Research, 22*(1), 110–145.

Bianchessi, N., Mansini, R., & Speranza, M. G. (2018). A branch-and-cut algorithm for the team orienteering problem. *International Transactions in Operational Research, 25*(2), 627–635.

Boussier, S., Feillet, D., & Gendreau, M. (2007). An exact algorithm for team orienteering problems. *4OR quarterly Journal of the Belgian, French and Italian Operations Research Societies, 5*, 211–230.

Dang, D.-C., El-Hajj, R., & Moukrim, A. (2013). A branch-and-cut algorithm for solving the team orienteering problem. In C. Gomes, & M. Sellmann (Eds.), *Integration of AI and OR techniques in constraint programming for combinatorial optimization problems, Berlin, Heidelberg* (pp. 332–339). Springer Berlin Heidelberg.

Dell'Amico, M., Maffioli, F., & Värbrand, P. (1995). On prize-collecting tours and the asymmetric travelling salesman problem. *International Transactions in Operational Research, 2*(3), 297–308.

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

Fischetti, M., Salazar-González, J. J., & Toth, P. (1998). Solving the orienteering problem through branch-and-cut. *INFORMS Journal on Computing, 10*, 133–148.

Gendreau, M., Laporte, G., & Semet, F. (1998). A branch-and-cut algorithm for the undirected selective traveling salesman problem. *Networks, 32*, 263–273.

Goldberg, A. V., & Tsioutsiouliklis, K. (2001). Cut tree algorithms: An experimental study. *Journal of Algorithms, 38*(1), 51–83.

Golden, B. L., Levy, L., & Vohra, R. (1987). The orienteering problem. *Naval Research Logistics, 34*, 307–318.

Grötschel, M., & Holland, O. (1991). Solution of large-scale symmetric travelling salesman problems. *Mathematical Programming, 51*(1), 141–202.

Gunawan, A., Lau, H. C., & Vansteenwegen, P. (2016). Orienteering problem: A survey of recent variants, solution approaches and applications. *European Journal of Operational Research, 255*, 315–332.

Keshtkaran, M., Ziarati, K., Bettinelli, A., & Vigo, D. (2015). Enhanced exact solution methods for the Team Orienteering Problem. *International Journal of Production Research, 54*(2), 591–601.

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

Kobeaga, G., Merino, M., & Lozano, J. A. (2021). On solving cycle problems with branch-and-cut: extending shrinking and exact subcycle elimination separation algorithms. *Annals of Operations Research, 305*, 107–136.

Laporte, G., & Martello, S. (1990). The selective travelling salesman problem. *Discrete Applied Mathematics, 26*(2), 193–207.

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

Padberg, M., & Hong, S. (1980). *On the symmetric travelling salesman problem: A computational study* (pp. 78–107). Springer Berlin Heidelberg, Berlin, Heidelberg.

Poggi, M., Viana, H., & Uchoa, E. (2010). The team orienteering problem: Formulations and branch-cut and price. In T. Erlebach, & M. Lübbecke (Eds.), *10th workshop on algorithmic approaches for transportation modelling, optimization, and systems (ATMOS'10), volume 14 of openaccess series in informatics (OASIcs), Dagstuhl, Germany* (pp. 142–155). Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik.

Ramesh, R., Yoon, Y.-S., & Karwan, M. H. (1992). An optimal algorithm for the orienteering tour problem. *ORSA Journal on Computing, 4*(2), 155–165.

Riera-Ledesma, J., & Salazar-González, J. J. (2017). Solving the team orienteering arc routing problem with a column generation approach. *European Journal of Operational Research, 262*(1), 14–27.

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

Sun, Y., Wang, S., Shen, Y., Li, X., Ernst, A. T., & Kirley, M. (2022). Boosting ant colony optimization via solution prediction and machine learning. *Computers & Operations Research, 143*, 105–769.

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

Vansteenwegen, P., & Gunawan, A. (2019). *State-of-the-art solution techniques for OP and TOP* (pp. 41–66). Springer International Publishing, Cham.

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

Waring, J., Lindvall, C., & Umeton, R. (2020). Automated machine learning: Review of the state-of-the-art and opportunities for healthcare. *Artificial Intelligence in Medicine, 104*, 101822.