European Journal of Operational Research 303 (2022) 78–98

Contents lists available at ScienceDirect

European Journal of Operational Research

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

Discrete Optimization

A scaleable projection-based branch-and-cut algorithm for the *p*-center problem

Elisabeth Gaar[a], Markus Sinnl[a,b,*]

[a] Institute of Production and Logistics Management, Johannes Kepler University Linz, Linz, Austria
[b] JKU Business School, Johannes Kepler University Linz, Linz, Austria

A R T I C L E I N F O

Article history:
Received 16 August 2021
Accepted 5 February 2022
Available online 11 February 2022

Keywords:
Location
p-center problem
Integer programming formulation
Lifting
Min-max objective

A B S T R A C T

The *p*-center problem (pCP) is a fundamental problem in location science, where we are given customer demand points and possible facility locations, and we want to choose p of these locations to open a facility such that the maximum distance of any customer demand point to its closest open facility is minimized. State-of-the-art solution approaches of pCP use its connection to the set cover problem to solve pCP in an iterative fashion by repeatedly solving set cover problems. The classical textbook integer programming (IP) formulation of pCP is usually dismissed due to its size and bad linear programming (LP)-relaxation bounds.

We present a novel solution approach that works on a new IP formulation that can be obtained by a projection from the classical formulation. The formulation is solved by means of branch-and-cut, where cuts for demand points are iteratively generated. Moreover, the formulation can be strengthened with combinatorial information to obtain a much tighter LP-relaxation. In particular, we present a novel way to use lower bound information to obtain stronger cuts. We show that the LP-relaxation bound of our strengthened formulation has the same strength as the best known bound in literature, which is based on a semi-relaxation.

Finally, we also present a computational study on instances from the literature with up to more than 700,000 customers and locations. Our solution algorithm is competitive with highly sophisticated set-cover-based solution algorithms, which depend on various components and parameters.

1. Introduction

The *(vertex) p-center problem* (pCP) is a fundamental problem in location science (Laporte, Nickel, & Saldanha da Gama, 2019; Snyder & Shen, 2011), where we are given customer demand points and potential facility locations, and we want to choose p of these locations to open a facility such that the maximum distance of any customer demand point to its closest open facility is minimized. The pCP is *NP-hard* (Kariv & Hakimi, 1979) and has applications where it is critical that all customer demand points can be reached quickly, i.e., at least one open facility can be reached quickly by each customer demand point. This includes emergency service locations, such as ambulances or fire stations, and also relief actions in humanitarian crisis (Çalık & Tansel, 2013; Jia, Ordóñez, & Dessouky, 2007; Lu & Sheu, 2013). Next to applications in location science, the pCP is also used for clustering of large-scale data (Geraci, Leoncini, Montangero, Pellegrini, & Renda, 2009; Kleindessner, Awasthi, & Morgenstern, 2019; Malkomes, Kusner, Chen, Weinberger, & Moseley, 2015), for feature selection (Meinl, Ostermann, & Berthold, 2011) and in computer vision (Friedler & Mount, 2010).

A formal definition of the pCP is as follows. Given an integer p, a set of customer demand points I with cardinality $|I| = n$, a set of potential facility locations J of cardinality $|J| = m \geq p$ and a distance d_{ij} from a customer demand point i to the potential facility location j for every $i \in I$ and $j \in J$, find a subset $S \subseteq J$ with cardinality $|S| = p$ of facilities to *open* such that the maximum distance between a customer demand point and its closest open facility is minimized, i.e., such that $\max_{i \in I} \min_{j \in S}\{d_{ij}\}$ is minimized.

There exists various integer programming (IP) formulations for the pCP. In this paper we focus on the classical textbook formulation, see for example Daskin (2013). It uses binary variables y_j to indicate whether a facility opens at the potential facility loca-

* Corresponding author at: Institute of Production and Logistics Management, Johannes Kepler University Linz, Linz, Austria.
E-mail addresses: elisabeth.gaar@jku.at (E. Gaar), markus.sinnl@jku.at (M. Sinnl).

https://doi.org/10.1016/j.ejor.2022.02.016
0377-2217/

tion $j \in J$, binary variables x_{ij} to indicate whether a customer demand point $i \in I$ is served by the potential facility location $j \in J$ and a continuous variable z to measure the distance in the objective function.

$$
\begin{aligned}
\text{(PC1)} \quad \min \quad & z && && \text{(1a)}\\
\text{s.t.} \quad & \sum_{j \in J} y_j = p && && \text{(1b)}\\
& \sum_{j \in J} x_{ij} = 1 && \forall i \in I && \text{(1c)}\\
& x_{ij} \le y_j && \forall i \in I, \forall j \in J && \text{(1d)}\\
& \sum_{j \in J} d_{ij} x_{ij} \le z && \forall i \in I && \text{(1e)}\\
& x_{ij} \in \{0,1\} && \forall i \in I, \forall j \in J && \text{(1f)}\\
& y_j \in \{0,1\} && \forall j \in J && \text{(1g)}\\
& z \in \mathbb{R} && && \text{(1h)}
\end{aligned}
$$

In particular, (1b) makes sure that exactly p facilities are open, (1c) guarantees that every customer is assigned to a facility, (1d) ensures that customers are only assigned to open facilities, (1e) forces z to be at least the distance of any customer to its assigned facility, the objective function (1a) minimizes this z and the constraints (1f) and (1g) make certain that x and y are binary. However, this formulation suffers from bad linear programming (LP)-relaxation bounds (see, e.g., Snyder & Shen, 2011) and also has scalability issues, as there are $O(|I| \cdot |J|)$ variables and constraints and large-scale instances from the literature for the pCP usually have $I = J = V$ with $|V|$ in the ten-thousands.

Thus, state-of-the-art exact approaches (see, e.g., Chen & Chen, 2009; Contardo, Iori, & Kramer, 2019) for the pCP use its connection to the *set cover problem (SCP)*. Given a ground set U and a set W of subsets of U, such that the union of the sets in W is U, in the SCP we want to find a subset T of W of minimum cardinality, such that the union of the sets in T is U. The question of whether the optimal objective function value of an instance of the pCP is less than or equal to a given value $\bar{z}$ can be modeled by an instance of the SCP: we set $U = I$, define $W_j = \{i \in I : d_{ij} \le \bar{z}\}$ for every $j \in J$, and set $W = \cup_{j \in J} W_j$. Then the optimal objective function of pCP is less than or equal to $\bar{z}$ if and only if the corresponding set cover instance has an optimal solution that uses at most p sets. State-of-the-art exact approaches for the pCP use this connection and solve the pCP by iteratively solving SCPs for different values of $\bar{z}$. More details on this approach and also other existing IP formulations are given in Section 1.2.

1.1. Contribution and outline

In this paper, we present a novel solution approach that works on an IP formulation that can be obtained by a projection from the classical formulation. The formulation is solved by means of branch-and-cut (B&C), where cuts for customer demand points are iteratively generated. This makes the method suitable for large scale instances, as the complete distance matrix does not need to be kept in memory. Moreover, we show how our formulation can be strengthened with combinatorial information to obtain a much tighter LP-relaxation compared to the LP-relaxation of the classical formulation. This strengthening procedure uses a novel way to use lower bound information to obtain stronger cuts. We also show how our formulation is connected to the SCP and present a computational comparison with state-of-the-art solution algorithms on instances from the literature.

In the remainder of this section, we discuss previous and related work to the pCP. Section 2 contains our new formulation, and Section 3 the strengthening procedure for the inequalities included in our formulation and theoretical results on the connection to the SCP and existing lower bounds from the literature. Section 4 describes implementation details of our B&C based solution algorithm. In Section 5, the computational study is presented. Finally, Section 6 concludes the paper.

1.2. Literature review

The pCP was first mentioned in 1965 by Hakimi (1965) and it was proven to be NP-hard by Kariv & Hakimi (1979) for $p \ge 2$, even in the case that $I = J$ is the set of vertices of a planar graph with maximum degree three and all distances are equal to one. However, there are special cases of pCP that can be solved in polynomial time, for example it can be solved in $O(|I|^2 \log |I|)$ time if $I = J$ is the set of vertices of a tree (Kariv & Hakimi, 1979).

Since the introduction of the pCP there has been an tremendous amount on work on both heuristic and exact solution methods. As our work is on the design of an exact solution algorithm, we focus our literature review on existing exact methods and refer to the recent survey (Garcia-Diaz et al., 2019) for approximation algorithms and heuristics for the pCP.

The first exact solution method for the pCP was proposed by Minieka (1970) and used the above described connection with the SCP. Minieka suggested to start with an arbitrary solution of the pCP and then iteratively solve SCPs in order to find out whether there is a better solution. Let $D = \{d_{ij} : i \in I, j \in J\}$ denote the set of all possible distances and let $d_1, \ldots, d_K$ be the values contained in D, so $D = \{d_1, \ldots, d_K\}$. Clearly, the optimal objective function value of pCP is in D and there are at most $|I| \cdot |J|$ potential optimal values. Minieka's approach can be interpreted as searching for the optimal value in D by first removing some values from D and then going through all remaining values of D one by one starting with the largest one and checking their optimality. Later Garfinkel, Neebe, & Rao (1977) elaborated this approach and proposed to use a heuristic to obtain a starting solution in order to further reduce potential optimal values in D and to then use a binary search with the remaining values of D instead of a linear search.

In the early 2000s a lot of work was done on the pCP. Ilhan & Pinar (2001) introduced a two-phase algorithm based on the idea of Minieka, where they solve feasibility linear programs (LPs) in order to obtain a good lower bound in the first phase, and iteratively solve feasibility SCPs within performing a linear search on D starting from the lower bound in the second phase. Later Al-Khedhairi & Salhi (2005) proposed some enhancements to this approach in order to reduce the number of iterations of the second phase. Ilhan, Özsoy, & Pinar (2002) perform a binary search to detect a suitable lower bound by solving LPs, and then perform a linear search on D. Caruso, Colorni, & Aloi (2003) use the connection to the SCP by providing two heuristics and two exact alogorithms for the pCP based on weak and strong dominance relationships.

In 2004, Elloumi, Labbé, & Pochet (2004) introduced a new IP formulation for the pCP. This formulation has a binary variable y_j for $j \in J$ indicating whether facility j opens, analogously to (PC1). Furthermore there is a binary variable for each value in D that indicates whether the optimal value of pCP is less or equal than this value. Towards this end let $u_k = 0$ if all customers have an open facility with distance at most d_{k-1}, otherwise $u_k = 1$ for all $k \in \{2, \dots, K\}$.

$$\text{(PCE)} \quad \min \quad d_1 + \sum_{k=2}^{K} (d_k - d_{k-1}) u_k \tag{2a}$$

$$\text{s.t.} \quad \sum_{j \in J} y_j \le p \tag{2b}$$

$$\sum_{j \in J} y_j \ge 1 \tag{2c}$$

$$u_k + \sum_{j: d_{ij} < d_k} y_j \ge 1 \quad \forall i \in I, \forall k \in \{2, \dots, K\} \tag{2d}$$

$$u_k \in \{0, 1\} \quad \forall k \in \{2, \dots, K\} \tag{2e}$$

$$y_j \in \{0, 1\} \quad \forall j \in J \tag{2f}$$

Elloumi et al. (2004) show that the lower bound obtained by the LP-relaxation of this IP formulation (PCE) is tighter than the one from the LP-relaxation of the classical IP formulation (PC1) and how the advantages of their formulation can help in the computation of an optimal solution.

Yet another IP formulation was introduced by Çalık & Tansel (2013). They also have the same binary variables y as in (PC1), and additionally to that there is a binary variable for each value in D that indicates whether the optimal value of pCP is exactly this value. Thus, in contrast to the IP formulation from Elloumi et al. (2004) here exactly one of this new binary variables is equal to one. Çalık & Tansel (2013) also investigated the connection of their IP formulation to the one of Elloumi et al. (2004) and provided the best LP-relaxation for the pCP so far. They also utilize their IP formulation within an exact solver for the pCP based on successive restrictions of this new formulation.

Chen & Chen (2009) introduced an LP-relaxation-based iterative SCP-based algorithm, in which they only consider some of the customer demand points in each iteration. So, instead of having few iterations with large sub-problems like previous approaches, they considered many iterations with small sub-problems. Recently, Contardo et al. (2019) have successfully enhanced this idea of considering only a subset of costumer demand points and included it into an exact SCP-based binary search algorithm for the pCP, that allows to solve large scale instances.

Finally, several different variants of the pCP have also been investigated, among them the capacitated pCP, where each potential facilitiy location has a maximum capacity, the continuous pCP, where the potential facility locations are not restricted to a set, and several versions with uncertain parameters. We refer to the chapter on the pCP (Çalık, Labbé, & Yaman, 2019) in Laporte et al. (2019) for more details.

2. A new IP formulation for the *p*-center problem

In this section, we start by presenting our new IP formulation for the p-center problem in Section 2.1. Next, we show that the feasible region of the LP-relaxation of our new formulation can be seen as a projection of the feasible region of the LP-relaxation of (PC1) in Section 2.2. Furthermore, we provide details about how our new formulation can be obtained from (PC1) as Benders decomposition in Section 2.3.

2.1. New IP formulation

Our new IP formulation (PC2) for the p-center problem uses y and z variables with the same meaning as in (PC1), i.e., the binary variables y_j indicate whether a facility opens at location $j \in J$ and the continuous variable z represents the distance in the objective function.

$$\text{(PC2)} \quad \min z \tag{3a}$$

$$\text{s.t.} \quad \sum_{j \in J} y_j = p \tag{3b}$$

$$z \ge d_{ij} - \sum_{j': d_{ij'} < d_{ij}} (d_{ij} - d_{ij'}) y_{j'} \quad \forall i \in I, \forall j \in J \tag{3c}$$

$$y_j \in \{0, 1\} \quad \forall j \in J \tag{3d}$$

$$z \in \mathbb{R} \tag{3e}$$

Next, we give combinatorial arguments for the correctness of (PC2). The constraint (3b) makes sure that exactly p facilities open and (3d) forces y to be binary. The constraints (3c) ensure that z is at least the distance to the nearest open facility for each customer $i \in I$ and work as follows: For each $j \in J$, they ensure that z is at least d_{ij} in case no closer facility j' to customer i (i.e., a facility j' with $d_{ij'} < d_{ij}$) is open. Thus, z has to take at least the distance to the nearest open facility to customer i. Moreover, if for $j \in J$ at least one closer facility is open, then let j' be the closest facility to customer i that is open. Then the right-hand side of (3c) is at most $d_{ij} - (d_{ij} - d_{ij'}) = d_{ij'}$ (maybe even more is subtracted from d_{ij}), thus z is forced to be larger than a value that is at most $d_{ij'}$, which is the distance of customer i to its closest open facility. Thus, constraints (3c) never lead to an overestimation of the distance of a customer to its nearest open facility.

We observe that formulation (PC2) has $O(|J|)$ variables and $O(|I| \cdot |J|)$ constraints. Thus, the number of variables is $O(|I|)$ times less than for formulation (PC1). However, similar to (PC1), the number of constraints can become prohibitive for solving large scale instances. We thus do not solve (PC2) directly, but use a branch-and-cut approach, where a lifted version of constraints (3c) are separated on-the-fly. Section 3 describes the lifting of (3c) and Section 4 discusses the separation.

2.2. Projection-based point of view

Before exploiting (PC2) computationally, we show the connection between (PC1) and (PC2). In particular, we show that the feasible region of the LP-relaxation of (PC2) is a projection of the feasible region of the LP-relaxation of (PC1). Thus, the LP-relaxations of (PC1) and (PC2) give the same bound for the pCP.

Towards this end, keep in mind that we have already argued that both IPs model the pCP. We start by examining their LP-relaxations. Let $\mathcal{P}(PC1)$ be the feasible region of the LP-relaxation of (PC1), i.e.,

$$\mathcal{P}(PC1) = \{(x, y, z) \in \mathbb{R}^{|I| \cdot |J| + |J| + 1} : (1b), (1c), (1d), (1e), \; x_{ij} \ge 0 \quad \forall i \in I \; \forall j \in J, \; 0 \le y_j \le 1 \quad \forall j \in J\}$$

and let $\mathcal{P}(PC2)$ be the feasible region of the LP-relaxation of (PC2), i.e.,

$$\mathcal{P}(PC2) = \{(y, z) \in \mathbb{R}^{|J|+1} : (3b), (3c), \; 0 \le y_j \le 1 \quad \forall j \in J\}.$$

Theorem 1. *The feasible region $\mathcal{P}(PC2)$ is the projection of $\mathcal{P}(PC1)$ to the space of (y, z)-variables, so $\mathcal{P}(PC2) = \{(y, z) \in \mathbb{R}^{|J|+1} : \exists x \in \mathbb{R}^{|I|\cdot|J|} : (x, y, z) \in \mathcal{P}(PC1)\}$.*

Proof. We start by showing that any point in the projection of $\mathcal{P}(PC1)$ is in $\mathcal{P}(PC2)$. Towards that end let $(x^\circ, y^\circ, z^\circ)$ be in $\mathcal{P}(PC1)$. Clearly (y°, z°) fulfills (3b) due to (1b) and $0 \le y_j^\circ \le 1$ for all $j \in J$ by construction, so we only have to show that it fulfills (3c).

Consider some $i \in I$ and $j \in J$. Because of (1d) for the right-hand side of (3c) clearly

$$d_{ij} - \sum_{j':d_{ij'} < d_{ij}} (d_{ij} - d_{ij'}) y_{j'}^\circ \le d_{ij} - \sum_{j':d_{ij'} < d_{ij}} (d_{ij} - d_{ij'}) x_{ij'}^\circ$$

$$= d_{ij}\left(1 - \sum_{j':d_{ij'} < d_{ij}} x_{ij'}^\circ\right) + \sum_{j':d_{ij'} < d_{ij}} d_{ij'} x_{ij'}^\circ$$

holds and because of (1c) this equals

$$d_{ij}\left(\sum_{j':d_{ij'} \ge d_{ij}} x_{ij'}^\circ\right) + \sum_{j':d_{ij'} < d_{ij}} d_{ij'} x_{ij'}^\circ \le \sum_{j' \in J} d_{ij'} x_{ij'}^\circ$$

which is at most z° because of (1e). Thus, the constraints (3c) are fulfilled for any i and j and therefore (y°, z°) is in $\mathcal{P}(PC2)$.

What is left to show is that any point in $\mathcal{P}(PC2)$ is also in the projection of $\mathcal{P}(PC1)$. Let $(y^*, z^*) \in \mathbb{R}^{|J|+1}$ be in $\mathcal{P}(PC2)$. We start by constructing $x^* \in \mathbb{R}^{|I|\cdot|J|}$ and then show that $(x^*, y^*, z^*) \in \mathcal{P}(PC1)$.

For every $i \in I$, let $j_i \in \arg\min_{j \in J}\{d_{ij} : \sum_{j':d_{ij'} < d_{ij}} y_{j'}^* < 1 \text{ and } \sum_{j':d_{ij'} \le d_{ij}} y_{j'}^* \ge 1\}$, i.e., j_i is an index such that all y^* strictly closer than d_{ij_i} to customer i have weight less than 1, and all y^* with distance at most d_{ij_i} to customer i have weight at least 1. In other words y^* reaches weight 1 at distance d_{ij_i} for customer i. Then for every $j \in J$ we set

$$x_{ij}^* = \begin{cases} y_j^* & \text{if } d_{ij} < d_{ij_i}, \\ y_j^*\left(1 - \sum_{j':d_{ij'} < d_{ij_i}} y_{j'}^*\right) / \left(\sum_{j':d_{ij'} = d_{ij_i}} y_{j'}^*\right) & \text{if } d_{ij} = d_{ij_i}, \\ 0 & \text{if } d_{ij} > d_{ij_i}. \end{cases}$$

Next we prove that (x^*, y^*, z^*) is in $\mathcal{P}(PC1)$. Clearly $0 \le y_j^* \le 1$ for all $j \in J$ holds and (1b) is satisfied because of (3b). Furthermore $x_{ij}^* \ge 0$ for all $i \in I$ and $j \in J$ holds by construction. Moreover

$$\sum_{j \in J} x_{ij}^* = \sum_{j:d_{ij} < d_{ij_i}} y_j^* + \left(1 - \sum_{j':d_{ij'} < d_{ij_i}} y_{j'}^*\right) / \left(\sum_{j':d_{ij'} = d_{ij_i}} y_{j'}^*\right) \sum_{j:d_{ij} = d_{ij_i}} y_j^* = 1,$$

so (1c) holds. In addition to that, due to the fact that $\sum_{j':d_{ij'} \le d_{ij_i}} y_{j'}^* \ge 1$ holds, we have that $x_{ij}^* \le y_j^*$, hence (1d) is fulfilled. Eventually we consider the last constraint (1e). For a fixed i, due to (1c) we have

$$\sum_{j \in J} d_{ij} x_{ij}^* = d_{ij_i}\left(1 - \sum_{j \in J} x_{ij}^*\right) + \sum_{j \in J} d_{ij} x_{ij}^* = d_{ij_i} - \sum_{j \in J} (d_{ij_i} - d_{ij}) x_{ij}^*.$$

As a consequence of $x_{ij}^* = 0$ for $d_{ij} > d_{ij_i}$ and $x_{ij}^* = y_j^*$ for $d_{ij} < d_{ij_i}$, this is the same as

$$d_{ij_i} - \sum_{j:d_{ij} < d_{ij_i}} (d_{ij_i} - d_{ij'}) x_{ij'}^* = d_{ij_i} - \sum_{j:d_{ij} < d_{ij_i}} (d_{ij_i} - d_{ij'}) y_{j'}^*,$$

which is at most z^* due to (3c) for i and j_i. Hence we have shown that $\sum_{j \in J} d_{ij} x_{ij}^* \le z^*$, so (1e) holds and therefore (x^*, y^*, z^*) is in $\mathcal{P}(PC1)$. □

Thus, Theorem 1 confirms that the feasible region of the LP-relaxation of (PC2) is in fact a projection of the feasible region of the LP-relaxation of (PC1). The following corollary is a simple consequence.

Corollary 2. *The optimal objective function values of the LP-relaxations of (PC1) and (PC2) coincide. Furthermore any optimal (y^*, z^*) of the LP-relaxation of (PC2) is optimal for the LP-relaxation of (PC1) and vice versa.*

As a consequence of Corollary 2, the LP-relaxations of (PC1) and (PC2) give the same bound for the pCP.

2.3. Reference to Benders decomposition

We now present an alternative approach on how (PC2) can be obtained from (PC1) as Benders decomposition. Towards this end let (PC1 – Rx) be formulation (PC1) with relaxed x variables, i.e., (PC1-Rx) is (PC1) without (1f) and with the constraint $x_{ij} \ge 0$ for all $i \in I$ and for all $j \in J$.

Observation 3. For any feasible solution (x, y, z) of (PC1-Rx), a binary $\tilde{x}$ can be constructed such that $(\tilde{x}, y, z)$ is a feasible solution of (PC1). Note that both solutions have the same objective function value z.

Proof. Let (x, y, z) be a feasible solution of (PC1-Rx). We construct $\tilde{x}$ in the following way. For any $i \in I$, we choose $j_i \in \arg\min_{j \in J}\{d_{ij} : y_j = 1\}$ and define $\tilde{x}_{ij_i} = 1$ and $\tilde{x}_{ij} = 0$ for all $j \in J \setminus \{j_i\}$. Clearly this $\tilde{x}$ is binary.

What is left to show is that $(\tilde{x}, y, z)$ is feasible for (PC1). Clearly (1b) and (1g) hold because (x, y, z) is feasible for (PC1-Rx). Moreover it is easy to see that (1c), (1d) and (1f) are fulfilled by construction. What is left so show is that (1e) is satisfied. For any $i \in I$ we have by construction $\sum_{j \in J} d_{ij}\tilde{x}_{ij} = d_{ij_i}\tilde{x}_{ij'} = d_{ij_i}$ and, because (x, y, z) is feasible for (PC1-Rx), d_{ij_i} equals $d_{ij_i}\sum_{j \in J} x_{ij} = \sum_{j \in J} d_{ij_i} x_{ij}$. Due to the fact that $x_{ij} = 0$ whenever $d_{ij} < d_{ij_i}$ (because in this case $y_j = 0$), this is less or equal to $\sum_{j \in J} d_{ij} x_{ij}$. This is less or equal to z, because (x, y, z) is feasible for (PC1-Rx). Thus (1e) is fulfilled and $(\tilde{x}, y, z)$ is feasible for (PC1). □

As a consequence of Observation 3, the pCP can be solved by using formulation (PC1-Rx) instead of (PC1). Our new formulation (PC2) can be seen as being obtained by applying Benders decomposition to (PC1-Rx) to project out the x-variables. Similar to the Benders decomposition reformulation for the uncapacitated facility location problem (UFL) presented in Fischetti, Ljubić, & Sinnl (2017), the obtained formulation i) is compact, i.e., it has a polynomial number of variables and constraints, and ii) has a combinatorial interpretation, that we have already presented. The main difference between the Benders decomposition for the UFL and our new formulation for the pCP is that in the UFL, there is a variable $z_i \ge 0$ for every customer $i \in I$ to measure the cost for each customer i, and the variables are then summed in the objective function, while in our case we have just a single z as the pCP has the objective function to minimize the maximum distance. We use this connection for an efficient separation of inequalities (3c) (resp., a lifted variant of them) in our branch-and-cut algorithm. Details of the separation are given in Section 4.1. For more background on Benders decomposition we refer to e.g., Chapter 8 of the book (Conforti, Cornuéjols, & Zambelli, 2014) and the surveys (Costa, 2005; Rahmaniani, Crainic, Gendreau, & Rei, 2017).

Note that Observation 3 and Theorem 1 also directly imply the following corollary.

Corollary 4. *The optimal objective function values of (PC1) and (PC2) coincide. Furthermore any optimal y^* of (PC1) is optimal for (PC2) and vice versa.*

Thus, as a byproduct of this different Benders decomposition viewpoint, Corollary 4 implies that (PC2) is indeed an IP formulation for the pCP and allows us to exploit (PC2) computationally.

3. Lifting the new formulation

In this section we first present a method to lift the LP-relaxation of our new IP formulation (PC2) in Section 3.1, which allows us to utilize a known lower bound and obtain a lower bound that is at least as strong as this lower bound by solving an LP. Then we discuss how to compute the best possible lower bound obtainable this way and give a combinatorial interpretation of it in terms of the set cover problem in Section 3.2. In Section 3.3 we show that our best possible lower bound coincides with the currently best lower bound known from the literature. Finally, we show that only some inequalities of the LP are necessary for obtaining the best possible lower bound in Section 3.4. Aspects relevant for computations are discussed in Section 4.

3.1. Exploiting lifted optimality cuts

The key ingredient to our lifting is the following lemma that involves the lifted optimality cuts (L-OPT).

Lemma 5. *Let $LB \geq 0$ be a lower bound on the optimal objective function value of (PC2). Then for any $i \in I$ and $j \in J$ the inequality*

$$z \geq \max\{LB, d_{ij}\} - \sum_{j':d_{ij'} < d_{ij}} \big(\max\{LB, d_{ij}\} - \max\{LB, d_{ij'}\}\big) y_{j'} \tag{L-OPT}$$

is valid for (PC2), i.e., every feasible solution of (PC2) fulfills (L-OPT).

Proof. Let $i \in I$, $j \in J$ and let (y, z) be a feasible solution of (PC2). Then y is a feasible solution to the considered instance of pCP with objective function value z. In particular, every customer demand point $i \in I$ has at most distance z to at least one of the locations indicated in y. As LB is a lower bound on the optimal objective function value of (PC2) we have $LB \leq z$. As a consequence, (y, z) is also a feasible solution of the slightly modified instance of pCP where we replace every distance d_{ij} with $\max\{LB, d_{ij}\}$. Clearly (L-OPT) is nothing else than (3c) for this slightly modified instance, hence (y, z) fulfills (L-OPT) because it is feasible for this slightly modified instance. Therefore (L-OPT) is valid. □

Now we can use the valid inequalities (L-OPT) to find a stronger LP-relaxation of (PC2) by replacing (3c) by (L-OPT) in the following way.

Theorem 6. *Let $LB \geq 0$ be a lower bound on the optimal objective function value of (PC2). Then*

$$\text{(PCLB)} \quad \mathcal{L}(LB) = \min z \tag{4a}$$

$$\text{s.t.} \sum_{j \in J} y_j = p \tag{4b}$$

$$z \geq d_{ij} - \sum_{j':d_{ij'} < d_{ij}} \big(d_{ij} - \max\{LB, d_{ij'}\}\big) y_{j'} \quad \forall i \in I, \forall j \in J : d_{ij} > LB \tag{4c}$$

$$z \geq LB \tag{4d}$$

$$0 \leq y_j \leq 1 \quad \forall j \in J \tag{4e}$$

$$z \in \mathbb{R} \tag{4f}$$

is an LP-relaxation of (PC2). In particular, $\mathcal{L}(LB)$ is a lower bound on the optimal objective function value of (PC2) with $\mathcal{L}(LB) \geq LB$.

Proof. Obviously an LP-relaxation of (PC2) is obtained by relaxing the binary constraints on y. Due to Lemma 5 this LP-relaxation can be strengthen by adding cuts of the form (L-OPT). Let us consider some $i \in I$ and $j \in J$. If $d_{ij} \leq LB$ then (L-OPT) simplifies to (4d), and if $d_{ij} > LB$ then (L-OPT) is equivalent to (4c). Furthermore we can disregard the inequalities (3c) because they are superseded by (L-OPT). □

In a nutshell, Theorem 6 allows us to start with a lower bound LB on (PC2) and solve the LP (PCLB) with $|J| + 1$ variables and a maximum of $|I| \cdot |J| + 2|J| + 2$ constraints, in order to obtain a new lower bound $\mathcal{L}(LB)$ that is as least as good as LB. Our aim is to utilize Theorem 6 in the following way: We start with a lower bound LB on the optimal objective function value of (PC2), for example the minimum of all distance values of D, so $LB = \min\{d : d \in D\}$, and then iteratively improve this lower bound by solving (PCLB) in order to obtain the new lower bound $\mathcal{L}(LB)$ and use this bound as LB in the next iteration. We give a formal definition of this procedure at the end of Section 3.2.

3.2. Our best possible lower bound

The aim of this section is to give a combinatorial interpretation of the best possible lower bound that we can reach with this procedure, i.e., to consider its convergence. Furthermore, we detail how to compute this best possible lower bound. To this end we start with the following lemmata.

Lemma 7. *Let $LB \geq 0$ be a lower bound on the optimal objective function value of (PC2) such that $\mathcal{L}(LB) = LB$ and let $(y^*, z^* = LB)$ be an optimal solution of (PCLB). Then*

$$\sum_{j:d_{ij} \leq LB} y^*_j \geq 1$$

holds for all $i \in I$.

Proof. For any $i \in I$ let $j_i \in \arg\min_{j \in J}\{d_{ij} : d_{ij} > LB\}$, i.e., j_i is the closest facility location to the customer demand point i that has distance greater than LB. Then (4c) for i and $j = j_i$ yields that

$$LB \geq d_{ij_i} - \sum_{j':d_{ij'} < d_{ij_i}} \big(d_{ij_i} - \max\{LB, d_{ij'}\}\big) y^*_{j'}$$

holds. By the choice of j_i we have $\max\{LB, d_{ij'}\} = LB$ for all j', so this implies

$$\big(d_{ij_i} - LB\big) \sum_{j':d_{ij'} < d_{ij_i}} y^*_{j'} \geq d_{ij_i} - LB$$

and together with the fact that $d_{ij'} < d_{ij_i}$ is equivalent to $d_{ij'} \leq LB$ for all j' this yields the desired result

$$\sum_{j':d_{ij'} \leq LB} y^*_{j'} \geq 1.$$

□

Lemma 8. *Let $LB \geq 0$ be a lower bound on the optimal objective function value of (PC2), y such that $0 \leq y \leq 1$ and $z = LB$. If*

$$\sum_{j':d_{ij'} \leq LB} y_{j'} \geq 1 \tag{5}$$

holds for some $i \in I$, then the inequalities (4c) hold for (y, z) for this i and all $j \in J$.

Proof. Let $i \in I$ such that (5) holds. Consider an arbitrary but fixed $j \in J$ with $d_{ij} > LB$. Then clearly

$$d_{ij} - \sum_{j':d_{ij'} < d_{ij}} \big(d_{ij} - \max\{LB, d_{ij'}\}\big) y_{j'} \leq d_{ij} - \sum_{j':d_{ij'} \leq LB} \big(d_{ij} - \max\{LB, d_{ij'}\}\big) y_{j'}$$

$$= d_{ij} - \left(d_{ij} - LB\right) \sum_{j':d_{ij'} \leq LB} y_{j'} \leq d_{ij} - \left(d_{ij} - LB\right) = LB = z$$

holds, and hence (4c) is satisfied for (y, z). □

Lemma 7 and Lemma 8 are the key ingredients of the following theorem.

Theorem 9. *Let $LB \geq 0$ be a lower bound on the optimal objective function value of (PC2). Then $\mathcal{L}(LB) = LB$ holds if and only if there is a fractional set cover solution with radius LB that uses at most p sets, i.e., if and only if there is a feasible solution y^* for*

$$\min \quad \sum_{j \in J} y_j \tag{6a}$$

$$\text{s.t.} \quad \sum_{j:d_{ij} \leq LB} y_j \geq 1 \qquad \forall i \in I \tag{6b}$$

$$0 \leq y_j \leq 1 \qquad \forall j \in J \tag{6c}$$

with objective function value at most p.

Proof. Assume $\mathcal{L}(LB) = LB$ holds, then by Lemma 7 an optimal solution $(y^*, z^* = LB)$ of (PCLB) fulfills (6b), because of (4e) it satisfies (6c) and due to (4b) the optimal objective function value (6a) is at most p.

Now let $z^* = LB$ and assume y^* is feasible for (6b) and (6c) and has objective function value (6a) at most p. Then due to Lemma 8 all inequalities (4c) are fulfilled for (y^*, z^*). It is easy to see that we can obtain $(y^{**}, z^{**} = z^*)$ by arbitrarily choosing some y_j^* in y^* and increase them such that (4b) and (4e) hold and all other constraints of (PCLB) are still satisfied. Therefore $\mathcal{L}(LB) \leq z^{**} = LB$ holds because (y^{**}, z^{**}) is feasible for (PCLB). As a result $\mathcal{L}(LB) = LB$, because $\mathcal{L}(LB) \geq LB$ holds by Theorem 6. □

Theorem 9 implies that if there is a fractional set cover solution with radius LB that uses at most p sets, then the new bound we obtain by solving (PCLB) will not be better than LB. Also the following corollary is an easy consequence of Theorem 9.

Corollary 10. *Let $LB \geq 0$ be a lower bound on the optimal objective function value of (PC2) with $\mathcal{L}(LB) = LB$. Then $\mathcal{L}(LB') = LB'$ holds for all $LB' \geq LB$.*

Proof. By Theorem 9 there is a fractional set cover solution with radius LB that uses at most p sets for (PC2). Then for any $LB' \geq LB$ it follows that there is also a fractional set cover solution with radius LB' that uses at most p sets for (PC2), so by Theorem 9 it holds that $\mathcal{L}(LB') = LB'$. □

Corollary 10 implies that whenever a lower bound LB does not yield a better bound by solving (PCLB), this is also true for all values larger than LB. This gives rise to the following definition.

Definition 11. Let $LB^\# = \min\{LB \in \mathbb{R} : \mathcal{L}(LB) = LB\}$.

Due to Corollary 10 $LB^\#$ is not only the smallest lower bound, for which solving (PCLB) does not improve the lower bound, but $LB^\#$ is also the supremum over all lower bounds, for which solving (PCLB) improves the bound. Next we reduce the possible values for $LB^\#$.

Lemma 12. *It holds that $LB^\# \in D$.*

Proof. It is easy to see that for any $k \in \{1, \dots, K-1\}$ the constraint (6b) is the same for all values of LB with $d_k \leq LB < d_{k+1}$, because whether a variable y_j is present in the sum only depends on whether its distance is less or equal to LB, and d_k and d_{k+1} are two consecutive distances of all distances. As a consequence, the optimal objective function value of (PCLB) is the same for all LB with $d_k \leq LB < d_{k+1}$. This together with Theorem 9 implies that whether $\mathcal{L}(LB) = LB$ holds is the same for all LB with $d_k \leq LB < d_{k+1}$. As $LB^\#$ is defined as the minimum value of LB such that $\mathcal{L}(LB) = LB$, clearly $LB^\#$ coincides with d_k for some $k \in \{1, \dots, K-1\}$ or with d_K, therefore $LB^\# \in D$ holds. □

In order to compute $LB^\#$ we can use the following approach: We start with a lower bound LB on the optimal objective function value of (PC2), for example the minimum of all distance values in D, so $LB = d_1$. Then we iteratively improve this lower bound by first solving (PCLB) in order to obtain the new lower bound $\mathcal{L}(LB)$, and in a second step we set $LB = \min\{d_k \in D : d_k \geq \mathcal{L}(LB)\}$. As soon as we do not improve LB in an iteration anymore, we set $LB^\# = LB$.

It is straight forward to see that the output $LB^\#$ of this procedure gives $LB^\#$ as defined in Definition 11. Note that an infinite series of increases without convergence is not possible since in the procedure LB only takes values of D and there are at most $|I| \cdot |J|$ values in D.

3.3. Comparison of our best possible lower bound with the best lower bound from the literature

Next we want to compare our best bound $LB^\#$ to the currently best lower bound known from the literature LB^*, which was introduced by Elloumi et al. (2004). They obtain LB^* by relaxing the integrality of the variables y_j while keeping the integrality of the variables u_k of their IP formulation (PCE), so LB^* is the optimal objective function value of the following MIP.

$$\begin{array}{llll} (\text{PCE}-\text{R}) & LB^* = \min & (2a) & \\ & \text{s.t.} & (2b)-(2c) & \\ & & 0 \leq y_j \leq 1 & \forall j \in J \end{array}$$

It turns out that our best bound $LB^\#$ and the best known bound from the literature LB^* coincide.

Theorem 13. $LB^\# = LB^*$

Proof. First, we show that $LB^\# \geq LB^*$ holds. Clearly $\mathcal{L}(LB^\#) = LB^\#$ holds, so by Theorem 9 there is a fractional set cover solution y^* with radius $LB^\#$ that uses at most p sets. Furthermore by Lemma 12 we have that $LB^\# \in D$, so there is a k^* such that $LB^\# = d_{k^*}$. We set $u_k^* = 1$ for all $k \leq k^*$ and $u_k^* = 0$ for all $k > k^*$. It is easy to see that (u^*, y^*) is a feasible solution of (PCE-R), whose objective function value equals $d_{k^*} = LB^\#$, thus $LB^\# \geq LB^*$ holds.

Next, we show that $LB^* \geq LB^\#$ holds. Towards this end let (u°, y°) be an optimal solution of (PCE-R). Elloumi et al. (2004) already observed that due to the structure of the problem there is an index $k^\circ \in \{2, \dots, K\}$ such that $u_k^\circ = 1$ for all $k \leq k^\circ$ and $u_k^\circ = 0$ for all $k > k^\circ$. Then constraint (2d) for $k = k^\circ + 1$ implies that $\sum_{j:d_{ij} < d_{k^\circ+1}} y_j^\circ \geq 1$ holds $\forall i \in I$, and hence also $\sum_{j:d_{ij} \leq d_{k^\circ}} y_j^\circ \geq 1$ holds $\forall i \in I$. As a consequence, y° is a fractional set cover solution with radius d_{k° that uses at most p sets. Furthermore d_{k° coincides with LB^*, because LB^* is the optimal objective function value of (PCE-R) and the optimal solution (u°, y°) has objective function value d_{k°. Thus, y° is a fractional set cover solution with radius LB^* that uses at most p sets, so by Theorem 9 it holds that $\mathcal{L}(LB^*) = LB^*$, therefore $LB^* \geq LB^\#$. □

3.4. Inequalities needed for obtaining our best possible lower bound

Now we want to draw our attention to the question of whether all inequalities (4c) are necessary for the convergence result. In fact, the following holds.

Theorem 14. *Let $LB \geq 0$ be a lower bound on the optimal objective function value of (PC2). For any $i \in I$ let $j_i \in \arg\min_{j \in J}\{d_{ij} : d_{ij} > LB\}$,* *i.e., j_i is the closest facility location to the customer demand point i that has distance more than LB. Then*

$$
\begin{aligned}
\mathcal{L}'(LB) = \min \quad & (4a) \\
\text{s.t.} \quad & z \geq d_{ij_i} - \sum_{j': d_{ij'} < d_{ij_i}} \left(d_{ij_i} - \max\{LB, d_{ij'}\}\right) y_{j'} \qquad \forall i \in I \qquad (7) \\
& (4b), (4d) - (4f)
\end{aligned}
$$

is an LP-relaxation of (PC2). In particular, $\mathcal{L}'(LB)$ gives a lower bound on the optimal objective function value of (PC2) with $\mathcal{L}(LB) \geq \mathcal{L}'(LB) \geq LB$. Furthermore $\mathcal{L}'(LB) = LB$ holds if and only if there is a fractional set cover solution with radius LB that uses at most p sets.

Proof. Clearly $\mathcal{L}'(LB)$ is obtained from $\mathcal{L}(LB)$ by removing some inequalities, so $\mathcal{L}(LB) \geq \mathcal{L}'(LB)$ holds. Analogous arguments as the ones in the proof of Theorem 6 prove that $\mathcal{L}'(LB)$ is a lower bound on the optimal objective function value of (PC2). The proof of the remaining statements can be done analogously to the proof of Theorem 9, i.e., with the help of analogous results as Lemma 7 and Lemma 8. □

In other words Theorem 14 implies that including only one of the inequalities (4c) for every customer demand point i, namely (7) for the closest facility location to i that has distance more than LB, is enough to ensure convergence to a fractional set cover solution $LB^{\#}$. This reduces the number of needed constraints in $\mathcal{L}'(LB)$ to $|I| + 2|J| + 2$ instead of potentially $|I| \cdot |J| + 2|J| + 2$ constraints needed in $\mathcal{L}(LB)$. However, it can be beneficial to include (4c) for a customer demand point i for several locations j in order to improve the speed of convergence.

After deriving theoretical properties of our best bound $LB^{\#}$ and showing that it is as good as the best bound from the literature, we next exploit $LB^{\#}$ computationally.

4. Implementation details

We now present our solution algorithm to solve the pCP based on formulation (PC2), where we replace (3c) with the lifted optimality cuts (L-OPT). In a nutshell, in our branch-and-cut algorithm we initialize the model with only a subset of inequalities (L-OPT) (i.e., only for a subset of customers $i \in I$ and potential facility locations $j \in J$) for a given valid lower bound LB. We then iteratively separate inequalities (L-OPT) while also updating the current lower bound LB used in defining (L-OPT) by exploiting the objective value of the current LP-relaxation. In the update of LB we also exploit that in instances from the literature the distances are integral.

We have implemented different strategies for doing the overall separation scheme (e.g., which inequalities to add,...), these strategies are detailed in Section 4.2. Before we discuss these strategies, we describe how a violated inequality (L-OPT) can be separated efficiently in Section 4.1. Finally, Section 4.3 describes a primal heuristic which is called during the B&C and driven by the optimal solution of the current LP-relaxation.

Our solution algorithm was implemented in C++ and it is available online under https://msinnl.github.io/pages/instancescodes.html. IBM ILOG CPLEX 12.10 with default settings was used as B&C framework. We apply the described separation schemes in the `UserCutCallback` of CPLEX, which gets called for fractional optimal solutions of the LP-relaxations within the B&C tree. In the `LazyConstraintCallback`, which gets called for optimal integer solutions, we simply add a single violated inequality (L-OPT) if any exits. This is done, as most of the integer solutions encountered are produced by internal CPLEX heuristics (and are not just integral LP-relaxations) and are very different to the current optimal solution of the LP-relaxations. As a consequence, adding multiple inequalities (L-OPT) for such solutions is usually not useful to help to improve the optimal objective value the LP-relaxation.

4.1. Separation of a violated inequality (L-OPT)

Let (y^*, z^*) be a (fractional) optimal solution of the LP-relaxation at a B&C node (the LP-relaxation of (PC2), where we replace (3c) with the lifted optimality cuts (L-OPT) for a subset of all customers $i \in I$ and potential facility locations $j \in J$), LB a given lower bound on the objective value and $i \in I$ a given customer. In order to obtain an efficient separation procedure, we can leverage that the cuts for a fixed customer are similar to the Benders cuts for the uncapacitated facility location problem as discussed in Section 2.3. Thus we can use an adapted separation procedure of the one presented in Fischetti et al. (2017) to find the most violated inequality, if there is a violated inequality for the customer $i \in I$ and the lower bound LB.

The procedure works as follows: Let $d'_{ij} = d_{ij}$ if $d_{ij} > LB$ and $d'_{ij} = LB$ otherwise. Sort the facilities $j \in J$ in ascending order according to d'_{ij}. Note that we only need to consider facilities with $y^*_j > 0$ for this sorting as the other facilities do not contribute to the potential violation. In the following, we assume that the locations are ordered in this way, i.e., we have $d'_{i1} \leq \ldots \leq d'_{i|J|}$. Let the *critical location* j_i be the index such that $\sum_{j=1}^{j_i-1} y^*_j < 1 \leq \sum_{j=1}^{j_i} y^*_j$. The maximum violation (i.e., the largest right-hand side value for the current y^*) is then given by

$$
d_{ij_i} - \sum_{j: y^*_j = 1 \text{ and } d_{ij} < d_{ij_i}} \left(d_{ij_i} - \max\{LB, d_{ij}\}\right) y^*_j
$$

and the inequality (L-OPT) with maximum violation is

$$
z \geq d_{ij_i} - \sum_{j: d_{ij} < d_{ij_i}} \left(d_{ij_i} - \max\{LB, d_{ij}\}\right) y_j.
$$

For more details on why this procedure gives the largest right-hand side value we refer to Fischetti et al. (2017).

We note that this procedure allows us to calculate j_i and hence the maximal violation for each customer i in an efficient way without the need of knowing all the distances d_{ij}. As mentioned above, we only need to know the distances d_{ij} for each $j \in J$ with $y^*_j > 0$ in order to determine j_i, and the number of such j is usually rather small compared to $|J|$. This is important for large-scale instances, where the distance matrix cannot be stored in memory and the distances need to be computed on-the-fly when needed. Note however, that for adding such an inequality to the model for a customer i, all the distances from i to any j need to be known.

4.2. Details of the overall separation schemes

We have implemented two different separation schemes, which differ in the way the violated inequalities that are added at a separation round are selected. Note that due to the size of the encountered instances, always adding all the violated inequalities becomes prohibitive. Moreover, previously added violated inequalities (L-OPT) can also become redundant, because during the course of the algorithm, we iteratively obtain larger values for LB, and thus new inequalities (L-OPT) which are strictly stronger than previously added ones can be derived. Moreover, as detailed in Section 4.1, the maximal violation of any cut of the form (L-OPT) can be calculated efficiently for each customer, while for adding an inequality, all the distances need to be known, which is computationally much more costly for large-scale instances. Hence, a carefully engineered separation scheme is needed in order to obtain good performance.

Our two schemes are denoted by `maxViolated` and `fixedCustomer`. In both schemes, we use the option purgeable of CPLEX when adding a violated inequality in the root-node of the B&C tree. This setting allows CPLEX to automatically remove a previously violated inequality if CPLEX decides (by an internal mechanism) that this inequality is "useless" for the subsequent solution process. To add a violated inequality at any other node of the B&C tree, we use the method addLocal of CPLEX, which adds the inequality not in a global fashion, but only for the current subtree of the B&C tree. This allows us to use the lower bound of the subtree as LB within the inequalities (L-OPT) instead of using the global lower bound[1]

Separation scheme `maxViolated`. In this scheme, we simply add the most violated inequality (L-OPT) for the maxNumCutsRoot (maxNumCutsTree) customers that have the largest maximum violation at each separation round. We use at most maxNumSepRoot (maxNumSepTree) separation rounds. If the lower bound value does not improve more than $\varepsilon = 1e-5$ for maxNoImprovements separation rounds at a node in the B&C tree (including the root-node), we stop the separation at this node. In this scheme, we do not add any inequalities (L-OPT) at initialization, and set LB to zero (the best deducible lower bound without additional computational effort) in the beginning. Algorithm 1 summarizes the separation scheme which is implemented in the UserCutCallback.

Separation scheme `fixedCustomer`. In this scheme, we restrict the separation to a subset $\hat{I} \subseteq I$ of the customers, which we then iteratively grow during the course of the algorithm. The idea behind this separation scheme is that for solving the problem to optimality, it can be enough to focus on a subset of the customers due to the min-max structure of the objective function. This is also used in the set-cover-based approaches (Chen & Chen, 2009; Contardo et al., 2019), where the set cover problem is solved (to optimality) on only a subset of the customers, and then it is checked, if any of the not-considered customers would change the objective function value. If yes, the set cover problem is then adapted with new customers and resolved. Furthermore, note that in this scheme, we sometimes use that $I = J$ as is customary in the instances from the literature. We mention whenever we use that condition and also detail how to proceed if $I \neq J$.

Our separation scheme tries to follow a similar avenue within our B&C framework. In particular, we add the most violated inequality (L-OPT) for all customers in $\hat{I} \subseteq I$ in each separation round and use at most maxNumSepRoot (maxNumSepTree) separation rounds. If the lower bound value does not improve more than $\varepsilon = 1e-5$ for maxNoImprovements separation rounds at a node in the B&C tree (including the root-node), we stop the separation at this node.

The procedure to initialize the set $\hat{I}$ is given as follows. We select a sample $\hat{I}$ of $p+1$ customers using the following greedy algorithm: We randomly choose an initial customer $i \in I$ and add it to the empty set $\hat{I}$. We then iteratively grow $\hat{I}$ by adding the customer $i \in I \setminus \hat{I}$ which has the maximum distance to its closest customer in $\hat{I}$. This is done until $|\hat{I}| = p+1$.

This procedure to initialize the set $\hat{I}$ exploits the assumption that $I = J$ by using the distances between pairs of customers, which are not necessary available if $I \neq J$. In the case that $I \neq J$ and these distances are available, this procedure can be used in the same way. If $I \neq J$ and these distances are not available, then one can iteratively add to $\hat{I}$ the customer $i \in I \setminus \hat{I}$ which has the maximum value of $\min_{\hat{i} \in \hat{I}} \min_{j \in J} \{d_{ij} + d_{\hat{i}j}\}$, or $|\hat{I}|$ can be initialized with $p+1$ random customers.

As initial value for LB we use $\min_{i \in \hat{I}, j \in J: j \neq i} \{d_{ij}\}$. This is a valid lower bound, since initially $|\hat{I}| = p+1$. Thus at least one customer

Algorithm 1: Separation scheme maxViolated

Input: instance $(I = J, d, p)$ of the pCP , LP-relaxation (y^*, z^*) of current B&C node, ID nodeID of current B&C node, ID prevNodeID of previous B&C node, counter nodeIterations, counter bound NotImprovedCount, LP-relaxation bound z^*_{prev} of the previous iteration

if $nodeID = prevNodeID$ **then**
  if $z^* - z^*_{prev} < \varepsilon$ **then**
    boundNotImprovedCount $\leftarrow$ boundNotImprovedCount $+ 1$
    if $boundNotImprovedCount = maxNoImprovements$ **then**
      return
else
  boundNotImprovedCount $\leftarrow 0$
  nodeIterations $\leftarrow 0$
maxNumCuts $\leftarrow$ maxNumCutsRoot
maxNumSep $\leftarrow$ maxNumSepRoot
if $nodeID \neq rootNode$ **then**
  maxNumCuts $\leftarrow$ maxNumCutsTree
  maxNumSep $\leftarrow$ maxNumSepTree
if $nodeIterations > maxNumSep$ **then**
  return
$z^*_{prev} \leftarrow z^*$
$LB \leftarrow \lceil z^* \rceil$
customerAndViolation $\leftarrow \emptyset$
for $i \in I$ **do**
  calculate maximal violation of inequalities (L-OPT) for customer i and given lower bound LB as described in Section 4.1
  if $violation > 0$ **then**
    customerAndViolation $\leftarrow$ customerAndViolation $\cup \{(i, \text{violation})\}$
sort customerAndViolation in descending order according to violation
for $\ell \in \{1, \ldots, \min\{maxNumCuts, size(customerAndViolation)\}\}$ **do**
  $i \leftarrow$ customer from customerAndViolation$[\ell]$
  if $nodeID = rootNode$ **then**
    add the maximal violated inequalitiy (L-OPT) for customer i and lower bound LB with the option purgeable
  else
    add the maximal violated inequalitiy (L-OPT) for customer i and lower bound LB as locally valid cut with using addLocal

$i \in \hat{I}$ cannot be chosen as open facility (which is in any case only possible if $i \in J$) in a feasible solution, and so its minimum distance must be the one to another location $j \neq i$ with $j \in J$. Note that in the case $I \neq J$ the condition $j \neq i$ can become redundant. Furthermore observe that as in the scheme maxViolated, we initialize LB with the best deducible lower bound without additional computational effort.

The customers to grow $\hat{I}$ are selected as follows: At each separation iteration, take the customer $i \in I \setminus \hat{I}$ which induces the most violated inequality (L-OPT) (if any exist) and add it to $\hat{I}$. Moreover, if there are more than maxNoImprovementsFixed con-

[1] Unfortunately, CPLEX does not allow to use the purgeable option when using addLocal.

secutive iterations, in which the lower bound did not improve, we increase $\hat{I}$ more aggressively by adding more customers. The selection of these additional customers is again driven by maximum violation of (L-OPT), but in order to find a diverse set of customers does not consider all $i \in I \setminus \hat{I}$ as candidates, but a subset $\bar{I} \subseteq I \setminus \hat{I}$. To build $\bar{I}$, initially we set $\bar{I} = I \setminus \hat{I}$. Whenever $d_{ij} \leq LB$ holds for some $j \in I = J$ for a customer i we just added to $\hat{I}$, we remove j from $\bar{I}$. Note that in this case j occurs in the maximum violated inequality (L-OPT) for customer i. The idea behind this is that if i and j are close to each other, then the already added inequality for customer i may be quite similar to the inequality we would add for j, thus we prevent adding the inequality for j.

Note that for constructing $\bar{I}$ we exploit that $I = J$, again by using the distances between pairs of customers. If these are available for $I \neq J$, our approach can still be used, only j does not appear in the maximum violated inequality (L-OPT) for customer i in this case. If $I \neq J$ and these distances are not available, then one could either delete all customers $\hat{i}$ for which there is a location $\hat{j}$ such that $d_{i\hat{j}} + d_{\hat{i}\hat{j}} \leq LB$, or omit the deletion of customers of $\bar{I}$ completely.

Algorithm 2, Part 1 (continued in Algorithm 3, Part 2) summarizes the separation scheme which is implemented in the `UserCutCallback`.

4.3. Primal heuristic

In order to obtain primal solutions, we have implemented a greedy heuristic in the `HeuristicCallback` of CPLEX. It uses the optimal solution (y^*, z^*) of the current LP-relaxation at a B&C node. Let S^H be the indices of the facilities to open in the solution produced by our heuristic. To construct S^H, we sort all the locations with $y^*_j > 0$ in descending order and then iterate through the sorted list. Whenever adding a location j from the list to S^H would improve the objective function value of the heuristic solution (which initially is set to ∞), we add j to S^H. This is done until $|S^H| = p$. If needed, we make multiple passes through the list.

5. Computational results

The runs were made on a single core of an Intel Xeon E5-2670v2 machine with 2.5 GHz and 32GB of RAM, and all CPLEX settings were left on their default values. The timelimit for the runs was set to 1800 s. The following parameter values were used for our algorithm, they were determined in preliminary computations:

- `maxNumCutsRoot`: 100
- `maxNumCutsTree`: 50
- `maxNumSepRoot`: 1000
- `maxNumSepTree`: 1
- `maxNoImprovements`: 100
- `maxNoImprovementsFixed`: 5

5.1. Instances

We used two sets of instances from the literature to evaluate the performance of our algorithm, these sets are denoted by `pmedian` and `TSPLIB` and details are given below.

- `pmedian`: This is an instance set used in Çalık & Tansel (2013), Chen & Chen (2009), Contardo et al. (2019). The set contains 40 instances. All instances have all customer demand points as potential facility locations, so $I = J = V$ holds. The number of customer demand points and hence also the number of potential facility locations $|I| = |J| = |V|$ is between 100 and 900, and p is between 5 and 200. For the concrete values of $|V|$ and p for each instance see Table 1.

Algorithm 2: (Part 1 of 2) Separation scheme `fixed Customer`

Input: instance $(I = J, d, p)$ of the pCP , LP-relaxation (y^*, z^*) of current B&C node, ID `nodeID` of current B&C node, ID `prevNodeID` of previous B&C node, counter `nodeIterations`, counter `boundNotImprovedCount`, counter `boundNotImprovedCountFixed`, LP-relaxation bound z^*_{prev} of the previous iteration, current subset of customers $\hat{I}$

```
if nodeID = prevNodeID then
    if z* − z*_prev < ε then
        boundNotImprovedCount ← boundNotImprovedCount + 1
        boundNotImprovedCountFixed ← boundNotImprovedCountFixed + 1
        if boundNotImprovedCount = maxNoImprovements then
            return
else
    boundNotImprovedCount ← 0
    boundNotImprovedCountFixed ← 0
    nodeIterations ← 0
maxNumSep ← maxNumSepRoot
if nodeID ≠ rootNode then
    maxNumSep ← maxNumSepTree
if nodeIterations > maxNumSep then
    return
z*_prev ← z*
LB ← ⌈z*⌉
customerAndViolation ← ∅
for i ∈ Î do
    calculate maximal violation of inequalities (L-OPT)
    for customer i and given lower bound LB as
    described in Section 4.1
    if violation > 0 then
        customerAndViolation ←
        customerAndViolation ∪ {(i, violation)}
sort customerAndViolation in descending order
according to violation
for ℓ ∈ {1, ..., size(customerAndViolation)} do
    i ← customer from customerAndViolation[ℓ]
    if nodeID = rootNode then
        add the maximal violated inequalitiy (L-OPT) for
        customer i and lower bound LB with the option
        purgeable
    else
        add the maximal violated inequalitiy (L-OPT) for
        customer i and lower bound LB as locally valid
        cut with using addLocal
```

continued in Algorithm 2, Part 2 of 2

- `TSPLIB`: This set contains larger instances and is based on the TSP-library (Reinelt, 1991). The number of customer demand points and hence also the number of potential facility locations $|I| = |J| = |V|$ is between 1621 and 744710, for the concrete values of $|V|$ see the tables in the appendix. The complete set of 44 instances was used in Contardo et al. (2019). In other works (Çalık & Tansel, 2013; Chen & Chen, 2009;

Algorithm 3: (Part 2 of 2) Separation scheme `fixed Customer`

```
continued from Algorithm 2, Part 1 of 2
customerAndViolationOutside ← ∅
for i ∈ I \ Î do
    calculate maximal violation of inequalities (L-OPT) for customer i and given lower bound LB as described in Section 4.1
    if violation > 0 then
        customerAndViolationOutside ← customerAndViolationOutside ∪ {(i, violation)}
sort customerAndViolationOutside in descending order according to the violation
if boundNotImprovedCountFixed < maxNoImprovementsFixed then
    if size(customerAndViolationOutside) ≥ 1 then
        mostViolatedCustomer ← customer from customerAndViolationOutside[1]
        Î ← Î ∪ {mostViolatedCustomer}
        if nodeID = rootNode then
            add the maximal violated inequalitiy (L-OPT) for customer mostViolatedCustomer and lower bound LB with the option purgeable
        else
            add the maximal violated inequalitiy (L-OPT) for customer mostViolatedCustomer and lower bound LB as locally valid cut with using addLocal
else
    boundNotImprovedCountFixed ← 0
    Ī ← I \ Î
    for ℓ ∈ {1, ..., size(customerAndViolationOutside)} do
        i ← customer from customerAndViolationOutside[ℓ]
        if i ∈ Ī then
            Î ← Î ∪ {i}
            if nodeID = rootNode then
                add the maximal violated inequalitiy (L-OPT) for customer i and lower bound LB with the option purgeable
            else
                add the maximal violated inequalitiy (L-OPT) for customer i and lower bound LB as locally valid cut with using addLocal
            for j with d_ij ≤ LB (this implies that j occurs in the maximal violated inequalitiy (L-OPT) for customer i and lower bound LB) do
                Ī ← Ī \ {j}
```

Elloumi et al., 2004) used only two or three (smaller) instances (with up to 3038 vertices) to test the scalability of the respective algorithms.[2] In Contardo et al. (2019) the instances were used with $p = 2, 3, 5, 10, 15, 20, 25, 30$, while in the other works also larger values of p were considered. In the instances all customer demand points are given as two-dimensional coordinates, and the Euclidean distance rounded down to the nearest integer is used as distance. This follows previous literature.

5.2. Results

We first give a comparison of our approaches with the lifted inequalities (L-OPT) with a B&C using our new formulation (PC2) without the lifting on instance set `pmedian`. Then we describe the results obtained by using our approaches with the lifted inequalities (L-OPT) for the larger instance set `TSPLIB`.

Results for the instance set `pmedian`. In Table 1, we compare the following six different configurations:

- `mVnoL`: separation scheme `maxViolated` and inequalities (3c)
- `fCnoL`: separation scheme `fixedCustomer` and inequalities (3c)
- `mVL`: separation scheme `maxViolated` and lifted inequalities (L-OPT)
- `fCL`: separation scheme `fixedCustomer` and lifted inequalities (L-OPT)
- `mVLH`: configuration `mVL` and the primal heuristic described in Section 4.3
- `fCLH`: configuration `fCL` and the primal heuristic described in Section 4.3

In Table 1, we see the huge effect of the lifted inequalities (L-OPT). We report the runtime in seconds ($t[s]$), the optimality gap ($g[\%]$, which is computed as $\frac{UB-LB}{UB} \cdot 100$, where LB is the obtained lower bound and UB is the obtained upper bound) and the number of B&C nodes ($\#BC$). Without lifting, only 11 of the 40 instances are solvable within the timelimit of 1800 s, while with lifting all the instances can be solved to optimality within four seconds runtime. Thus, our lifting has an immense positive effect. For these instances, the primal heuristic has no discernible effect, as they already become solvable very fast when using the lifting. Also the choice of the separation scheme (`maxViolated` or `fixedCustomer`) does not influence the quality of the results a lot. We also note that most of the instances can be solved within the root-node of the B&C algorithm when using our lifting.

Results for the instance set `TSPlib`. In Figs. 1-2, we show plots of the runtime and optimality gap aggregated by p for configurations `mVL`, `fCL`, `mVLH` and `fCLH`, i.e., the by far best configurations for the instance set `pmedian` that contains smaller instances. We used the same values for p as Contardo et al. (2019). Detailed results for the setting `fCLH` can be found in the appendix in the Tables A.2–A.9, were we also include the results obtained by Contardo et al. (2019). From the figures, we see that the instances are becoming harder to solve when p becomes larger, which is in tune with the computational experiments of Contardo et al. (2019). We note that for the settings `mVL` and `mVLH` the code did not terminate for all instances, as the available memory was not enough. In general, the settings `fCL` and `fCLH` perform better than `mVL` and `mVLH`. An explanation for this could be that by focusing the separation of inequalities (L-OPT) on the subset $\hat{I}$, the separation procedure becomes more stable, which could allow CPLEX to easier decide which inequalities to remove for further iterations. Hence, the LP-relaxations becomes smaller and easier to solve, and also not so memory-intensive.

To further illustrate the different behavior of both separation schemes, Fig. 3 shows the obtained lower bound values at the root node of the B&C tree against the runtime for settings `mVLH` and `fCLH` for the `TSPlib` instance `rl11849` for $p = 5$. While both settings manage to solve this instance to opimality in the root node, we see that `fCLH` converges much faster. The behavior of

[2] In Chen & Chen (2009) also other smaller TSPlib instances with less than 1000 vertices were used, but they are not included in the set used in Contardo et al. (2019).

Table 1
Detailed results for the `pmedian` instances.

name	$\|V\|$	p	mVnoL			fCnoL			mVL			fCL			mVLH			fCLH		
			$t[s]$	$g[\%]$	#BC	$t[s]$	$g[\%]$	#BC	$t[s]$	$g[\%]$	#BC	$t[s]$	$g[\%]$	#BC	$t[s]$	$g[\%]$	#BC	$t[s]$	$g[\%]$	#BC
pmed1	100	5	57.72	0.00	115,902	52.96	0.00	114,523	0.14	0.00	45	0.11	0.00	46	0.15	0.00	71	0.10	0.00	27
pmed2	100	10	1,198.51	0.00	2,384,887	1695.92	0.00	3,342,277	0.04	0.00	0	0.05	0.00	0	0.05	0.00	0	0.05	0.00	0
pmed3	100	10	1,163.21	0.00	1,862,933	729.00	0.00	1,250,156	0.04	0.00	0	0.04	0.00	0	0.05	0.00	0	0.05	0.00	0
pmed4	100	20	TL	10.81	1,616,178	TL	10.81	1,877,802	0.05	0.00	0	0.05	0.00	0	0.05	0.00	0	0.05	0.00	0
pmed5	100	33	TL	8.33	2,454,807	TL	8.33	2,367,363	0.05	0.00	0	0.05	0.00	0	0.05	0.00	0	0.05	0.00	0
pmed6	200	5	52.98	0.00	60,870	61.15	0.00	79,822	0.11	0.00	3	0.11	0.00	3	0.12	0.00	3	0.10	0.00	3
pmed7	200	10	TL	9.38	1,139,346	TL	7.81	1,294,446	0.09	0.00	0	0.08	0.00	0	0.09	0.00	0	0.08	0.00	0
pmed8	200	20	TL	20.00	778,085	TL	20.00	794,555	0.11	0.00	0	0.10	0.00	0	0.12	0.00	0	0.10	0.00	0
pmed9	200	40	TL	18.92	543,618	TL	18.92	581,900	0.14	0.00	0	0.15	0.00	0	0.14	0.00	0	0.16	0.00	0
pmed10	200	67	TL	25.00	488,041	TL	25.00	560,078	0.13	0.00	0	0.15	0.00	0	0.13	0.00	0	0.16	0.00	0
pmed11	300	5	73.09	0.00	109,946	60.72	0.00	89,820	0.16	0.00	2	0.13	0.00	0	0.16	0.00	2	0.13	0.00	0
pmed12	300	10	TL	5.88	979,558	TL	3.92	1,110,223	0.17	0.00	0	0.16	0.00	0	0.16	0.00	0	0.15	0.00	0
pmed13	300	30	TL	16.67	402,537	TL	22.22	383,634	0.21	0.00	0	0.24	0.00	0	0.22	0.00	0	0.24	0.00	0
pmed14	300	60	TL	23.08	260,750	TL	23.08	252,826	0.36	0.00	2	0.36	0.00	0	0.37	0.00	2	0.37	0.00	0
pmed15	300	100	TL	27.78	221,453	TL	27.78	216,826	0.33	0.00	0	0.36	0.00	0	0.34	0.00	0	0.38	0.00	0
pmed16	400	5	24.67	0.00	23,852	22.56	0.00	22,601	0.27	0.00	0	0.25	0.00	0	0.28	0.00	0	0.26	0.00	0
pmed17	400	10	TL	7.69	618,267	TL	7.69	672,741	0.32	0.00	0	0.30	0.00	0	0.32	0.00	0	0.33	0.00	0
pmed18	400	40	TL	21.43	247,230	TL	21.43	280,508	0.49	0.00	0	0.62	0.00	9	0.50	0.00	0	0.69	0.00	15
pmed19	400	80	TL	22.22	155,689	TL	22.22	158,464	0.72	0.00	0	0.63	0.00	0	0.74	0.00	0	0.65	0.00	0
pmed20	400	133	TL	23.08	130,236	TL	23.08	135,130	0.57	0.00	0	0.61	0.00	0	0.57	0.00	0	0.64	0.00	0
pmed21	500	5	381.19	0.00	337,602	377.07	0.00	334,501	0.48	0.00	0	0.47	0.00	0	0.48	0.00	0	0.47	0.00	0
pmed22	500	10	TL	7.89	501,977	TL	7.89	512,749	0.67	0.00	17	0.86	0.00	89	0.75	0.00	61	0.97	0.00	130
pmed23	500	50	TL	22.73	182,841	TL	22.73	159,898	0.79	0.00	0	0.85	0.00	6	0.81	0.00	0	0.87	0.00	6
pmed24	500	100	TL	26.67	102,612	TL	26.67	102,117	0.77	0.00	0	0.83	0.00	0	0.78	0.00	0	0.84	0.00	0
pmed25	500	167	TL	27.27	75,362	TL	27.27	79,665	0.92	0.00	0	0.98	0.00	0	0.95	0.00	0	1.02	0.00	0
pmed26	600	5	180.59	0.00	115,119	183.90	0.00	119,306	0.84	0.00	2	0.80	0.00	1	0.86	0.00	15	0.80	0.00	1
pmed27	600	10	TL	9.38	533,068	TL	9.38	496,703	0.82	0.00	0	0.82	0.00	0	0.82	0.00	0	0.83	0.00	0
pmed28	600	60	TL	22.22	132,018	TL	22.22	144,764	1.04	0.00	0	1.05	0.00	0	1.06	0.00	0	1.08	0.00	0
pmed29	600	120	TL	23.08	75,833	TL	23.08	78,090	1.17	0.00	0	1.23	0.00	0	1.20	0.00	0	1.26	0.00	0
pmed30	600	200	TL	22.22	51,632	TL	22.22	54,290	1.34	0.00	0	1.42	0.00	0	1.37	0.00	0	1.47	0.00	0
pmed31	700	5	746.29	0.00	401,934	479.78	0.00	234,253	1.24	0.00	0	1.20	0.00	0	1.22	0.00	0	1.19	0.00	0
pmed32	700	10	TL	10.34	347,742	TL	10.34	368,273	1.39	0.00	10	1.59	0.00	60	1.40	0.00	10	1.53	0.00	36
pmed33	700	70	TL	20.00	86,212	TL	20.00	92,720	1.63	0.00	0	1.78	0.00	0	1.53	0.00	0	1.77	0.00	0
pmed34	700	140	TL	27.27	57,991	TL	27.27	55,186	1.75	0.00	0	1.75	0.00	0	1.78	0.00	0	1.74	0.00	0
pmed35	800	5	66.68	0.00	41,432	69.04	0.00	41,664	1.76	0.00	3	1.74	0.00	0	1.72	0.00	0	1.74	0.00	0
pmed36	800	10	TL	7.41	339,494	TL	7.41	380,960	1.92	0.00	9	1.92	0.00	5	1.93	0.00	9	1.95	0.00	5
pmed37	800	80	TL	26.67	79,136	TL	26.67	67,520	2.44	0.00	0	2.26	0.00	0	2.47	0.00	0	2.27	0.00	0
pmed38	900	5	43.10	0.00	23,546	31.83	0.00	16,249	2.50	0.00	0	2.47	0.00	0	2.50	0.00	0	2.48	0.00	0
pmed39	900	10	TL	4.35	364,938	TL	4.35	338,705	2.70	0.00	28	2.53	0.00	0	2.56	0.00	0	2.55	0.00	0
pmed40	900	90	TL	23.08	55,589	TL	23.08	54,009	3.44	0.00	0	3.26	0.00	0	3.45	0.00	0	3.30	0.00	0

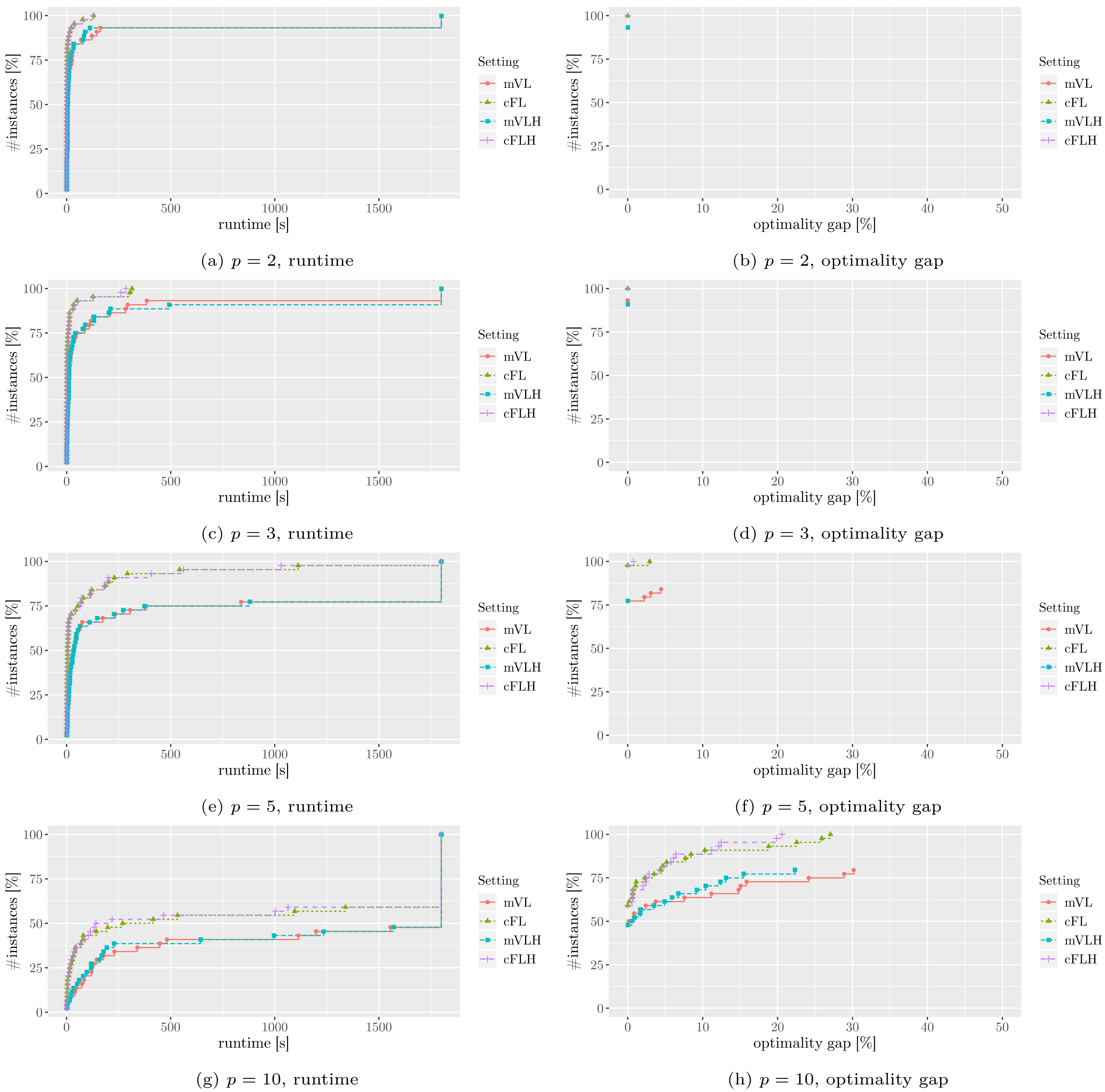


Fig. 1. Plots for runtime and optimality gap for $p = 2, 3, 5, 10$ for the `TSPLIB` instances. If `#instances[%]` in the optimality gap plot does not sum up to 100%, this means that some runs did not finish due to memory issues.

the lower bound in this instance is a typical behavior we observed for both strategies.

Figure 4 a and b show the number of added inequalities (L-OPT) for all `TSPlib` instances with $p = 2$ and $p = 5$ which could be solved by all four considered settings. Theses figures further support that with the scheme `fixedCustomer` much less inequalities need to be added. We also see that for larger p much more inequalities need to be added. For the scheme `fixedCustomer` the maximum number of added inequalities for $p = 2$ is slightly over 500, while for $p = 5$ it is around 10,000. For the scheme `maxViolated` this number grows from around 4,000 for $p = 2$ to around 80,000 for $p = 5$. We note that the number of added inequalities is not necessarily the same as the number of inequalities which are in the final LP-relaxations, as the CPLEX cut management may delete added violated inequalities during the course of the B&C.

From the Figs. 1–2, we also see that there is some effect of using the primal heuristic, in particular regarding the optimality gap. This is consistent when comparing our obtained lower and upper bound values with the values obtained by Contardo et al. (2019), i.e., from the comparison we see that our approach provides good lower bounds (sometimes even improving on the lower bound of

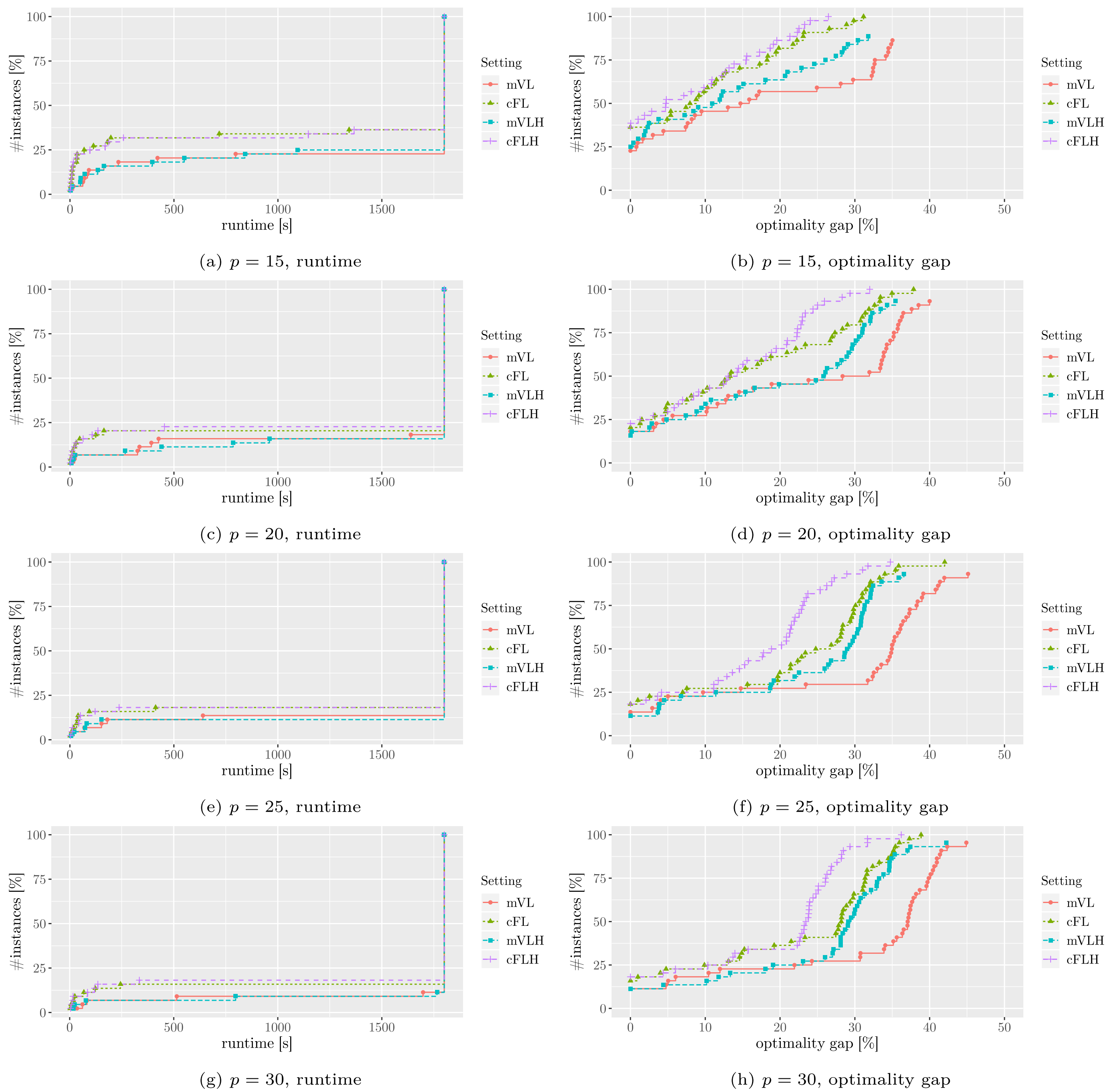


Fig. 2. Plots for runtime and optimality gap for $p = 15, 20, 25, 30$ for the TSPLIB instances. If `#instances[%]` in the optimality gap plot does not sum up to 100%, this means that some runs did not finish due to memory issues.

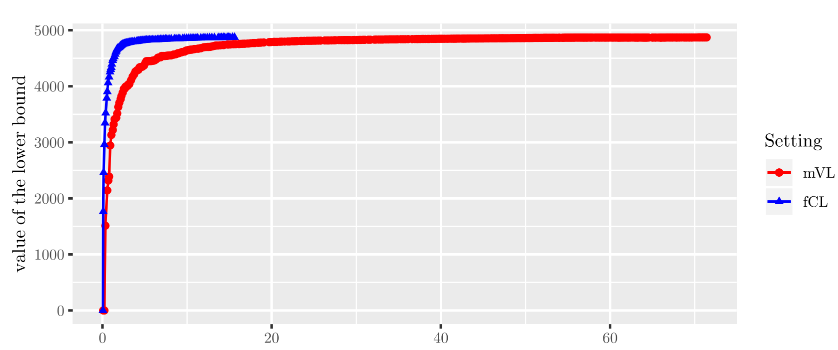
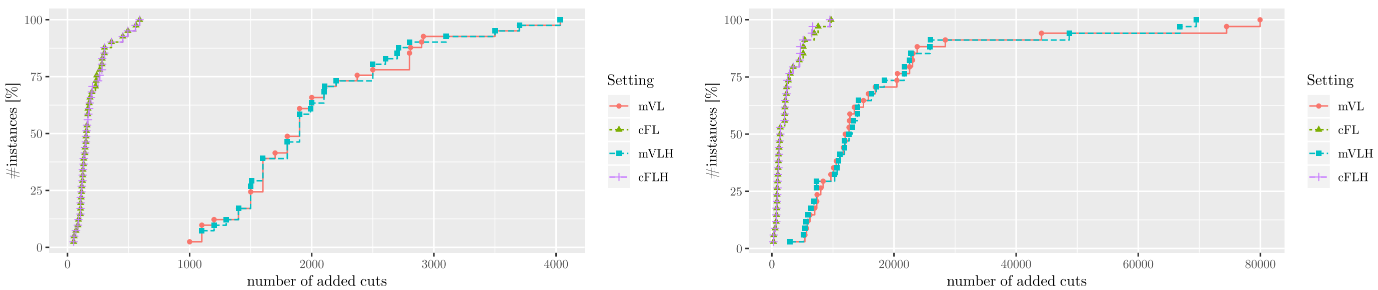


Fig. 3. Behavior of the lower bound at the root node for the instance `rl11849` with $p = 5$ and our two separation schemes .



(a) $p = 2$, 41 instances (b) $p = 5$, 34 instances

Fig. 4. Plots for the number of added inequalities (L-OPT) for all TSPLIB instances which were solved to optimality with all considered settings.

Contardo et al. (2019), which they obtained within a timelimit of 24 h), while the upper bounds for some instances could still be improved. This again shows the important contribution of our lifting procedure.

From the results in Tables A.2–A.9 we can also see that in some instances for larger p our approach scales worse compared to the approach of Contardo et al. (2019). We believe that this could be due to the combination of the following factors: i) our method is a “classical” B&C approach, thus the strength of the LP-relaxation is of course crucial. For larger p the feasible region of the LP-relaxation is getting larger, and thus the LP-relaxation may also become weaker; ii) state-of-the-art methods such as Contardo et al. (2019) work by iteratively resolving set cover problems. For these problems, the used MIP-solvers may be able to use additional preprocessing routines, general-purpose cuts or other speed-up tricks, which are nowadays built in such solvers; iii) Contardo et al. (2019) also uses a specific preprocessing which is applied before each set cover problem. The procedure is not described, thus it is also clear, if such a procedure could be directly transferable into our B&C approach; iv) in some cases the obtained primal bound is the bottleneck for our algorithm, even though we already implemented a primal heuristic.

6. Conclusions

In this paper, we present a novel solution approach for the p-center problem, which is a fundamental problem in location science. The solution approach works on a new integer programming formulation that can be obtained by a projection from the classical formulation. The formulation is solved by means of branch-and-cut, where cuts for customer demand points are iteratively generated. This makes the method suitable for large scale instances. Moreover, the formulation can be strengthened with combinatorial information to obtain a much tighter LP-relaxation. In particular, we present a novel way to use lower bound information to obtain stronger cuts. We show that the obtained formulation has the same lower bound as the best known relaxations from the literature (Çalık & Tansel, 2013; Elloumi et al., 2004). In contrast to these bounds from the literature, which are based on semi-relaxations, i.e., some variables are kept integer in the relaxation, our relaxation only contains continuous variables. Moreover, our approach can be easily implemented in a single branch-and-cut algorithm, while existing state-of-the-art approaches for the p-center problem consist of iterative solution algorithms, where set cover problems are solved repeatedly (Chen & Chen, 2009; Contardo et al., 2019). We describe different strategies on how to separate our lifted cuts, and also present a primal heuristic which uses the information provided by the linear programming relaxation. The solution algorithm is made available online.

In order to assess the efficiency of our proposed approach, we conducted a computational study on instances from the literature with up to more than 700,000 customers and locations. The results show that for many instances, our solution algorithm is competitive with highly sophisticated solution frameworks from the literature.

There are various avenues for further work. Further research regarding potential additional valid inequalities to strengthen the LP-relaxation could be interesting. Another direction could be the development of better primal heuristics to be used within the branch-and-cut algorithm, as for many instances our lower bound was quite good, while the primal bound still had some room for improvement. However, the large-scale nature of the considered instances may make this a challenging task, as even applying simple local search operator could become quite time consuming. A hybridization of our approach with existing set cover-based approaches could also be interesting, because the results show that for some instances our approach works better, while for other instances set-cover-based approaches work better. The investigation if preprocessing procedures from set-cover-based approaches can be transferred to our approach could also be a fruitful topic. It could also be worthwhile to investigate, if there are certain instance-characteristics which may favor a particular approach. Moreover, as our solution approach directly models the problem with optimality-type cuts, our approach could be more suitable when considering stochastic or robust versions of the pCP, compared to the set cover-based approaches, which may no be easily adaptable to such settings.

Acknowledgments

The research was supported by the Linz Institute of Technology (Project LIT-2019-7-YOU-211) and the JKU Business School.

Appendix A. Detailed results for the instance set TSPlib

In the following Tables A.2–A.9 we provide a comparison with the results obtained by Contardo et al. (2019) using an Intel Xeon E5462 with 2.8 Ghz and 16GB RAM. In their runs, the timelimit was set to 24 h (indicated by TL2 in the tables), while we have a timelimit of 30 min (indicated by TL in the tables). Whenever at least one of the two versions finished, the faster runtime is bold. If both versions did not finish, the better upper and the better lower bound is bold.

Table A.1
Detailed results for the TSPlib instances with $p = 2$.

name	$\|V\|$	fCLH LB	fCLH UB	fCLH $g[\%]$	fCLH #BC	fCLH $t[s]$	Contardo et al. (2019) LB	Contardo et al. (2019) UB	Contardo et al. (2019) $t[s]$
rw1621	1,621	714	714	0.00	0	**0.11**	714	714	9.50
u1817	1,817	1,061	1,061	0.00	0	**0.20**	1,061	1,061	12.10
rl1889	1,889	6,931	6,931	0.00	0	**0.16**	6,931	6,931	10.20
mu1979	1,979	3,284	3,284	0.00	0	**0.09**	3,284	3,284	10.50
pr2392	2,392	6,060	6,060	0.00	0	**0.28**	6,060	6,060	13.10
d15112-modif-2500	2,500	9,234	9,234	0.00	0	**0.30**	9,234	9,234	12.70
pcb3038	3,038	1,734	1,734	0.00	0	**0.27**	1,734	1,734	10.00
nu3496	3,496	2,140	2,140	0.00	0	**0.21**	2,140	2,140	10.20
ca4663	4,663	25,781	25,781	0.00	0	**0.53**	25,781	25,781	17.90
rl5915	5,915	7,385	7,385	0.00	0	**0.52**	7,385	7,385	17.00
rl5934	5,934	7,004	7,004	0.00	0	**0.97**	7,004	7,004	20.50
tz6117	6,117	4,582	4,582	0.00	0	**0.85**	4,582	4,582	15.80
eg7146	7,146	4,643	4,643	0.00	0	**0.49**	4,643	4,643	14.20
pla7397	7,397	310,664	310,664	0.00	0	**0.43**	310,664	310,664	15.30
ym7663	7,663	3,807	3,807	0.00	0	**0.25**	3,807	3,807	12.50
pm8079	8,079	1,686	1,686	0.00	0	**0.45**	1,686	1,686	13.40
ei8246	8,246	1,813	1,813	0.00	0	**0.57**	1,813	1,813	18.30
ar9152	9,152	9,725	9,725	0.00	0	**0.68**	9,725	9,725	20.50
ja9847	9,847	10,099	10,099	0.00	0	**0.30**	10,099	10,099	18.70
gr9882	9,882	3,757	3,757	0.00	0	**0.71**	3,757	3,757	13.70
kz9976	9,976	11,325	11,325	0.00	0	**1.91**	11,325	11,325	17.50
fi10639	10,639	4,922	4,922	0.00	0	**0.79**	4,922	4,922	16.70
rl11849	11,849	7,298	7,298	0.00	0	**1.11**	7,298	7,298	21.20
usa13509	13,509	175,750	175,750	0.00	0	**0.86**	175,750	175,750	17.90
brd14051	14,051	2,970	2,970	0.00	0	**1.38**	2,970	2,970	15.20
mo14185	14,185	3,698	3,698	0.00	0	**1.27**	3,698	3,698	16.20
ho14473	14,473	1,834	1,834	0.00	0	**0.96**	1,834	1,834	15.80
d15112	15,112	9,406	9,406	0.00	0	**2.05**	9,406	9,406	17.40
it16862	16,862	4,677	4,677	0.00	0	**1.01**	4,677	4,677	16.80
d18512	18,512	3,301	3,301	0.00	0	**2.51**	3,301	3,301	16.40
vm22775	22,775	4,266	4,266	0.00	0	**1.59**	4,266	4,266	20.50
sw24978	24,978	4,960	4,960	0.00	0	**1.91**	4,960	4,960	25.10
fyg28534	28,534	448	448	0.00	0	**3.00**	448	448	18.50
bm33708	33,708	5,167	5,167	0.00	0	**2.92**	5,167	5,167	21.60
pla33810	33,810	330,462	330,462	0.00	0	**6.95**	330,462	330,462	64.30
bby34656	34,656	474	474	0.00	0	**2.99**	474	474	15.10
pba38478	38,478	469	469	0.00	0	**2.90**	469	469	25.50
ch71009	71,009	19,988	19,988	0.00	0	**5.63**	19,988	19,988	27.10
pla85900	85,900	436,008	436,008	0.00	0	**21.56**	436,008	436,008	136.10
sra104815	104,814	908	908	0.00	0	**16.21**	908	908	51.10
usa115475	115,475	17,745	17,745	0.00	0	**10.22**	17,745	17,745	47.90
ara238025	238,025	1,484	1,484	0.00	0	**39.25**	1,484	1,484	89.80
lra498378	498,378	5,888	5,888	0.00	0	**81.50**	5,888	5,888	423.70
lrb744710	744,710	1,930	1,930	0.00	0	**127.01**	1,930	1,930	241.00

Table A.2
Detailed results for the TSPlib instances with $p = 3$.

name	$\|V\|$	fCLH LB	fCLH UB	fCLH $g[\%]$	fCLH #BC	fCLH $t[s]$	Contardo et al. (2019) LB	Contardo et al. (2019) UB	Contardo et al. (2019) $t[s]$
rw1621	1,621	624	624	0.00	0	**0.18**	624	624	10.80
u1817	1,817	895	895	0.00	0	**0.23**	895	895	9.90
rl1889	1,889	6,066	6,066	0.00	0	**0.35**	6,066	6,066	12.30
mu1979	1,979	2,326	2,326	0.00	0	**0.14**	2,326	2,326	10.10
pr2392	2,392	5,413	5,413	0.00	0	**1.36**	5,413	5,413	17.10
d15112-modif-2500	2,500	7,952	7,952	0.00	0	**0.38**	7,952	7,952	13.10
pcb3038	3,038	1,519	1,519	0.00	0	**0.61**	1,519	1,519	12.40
nu3496	3,496	1,513	1,513	0.00	0	**0.62**	1,513	1,513	12.60
ca4663	4,663	22,680	22,680	0.00	2	**1.56**	22,680	22,680	20.70
rl5915	5,915	6,377	6,377	0.00	0	**1.62**	6,377	6,377	18.50
rl5934	5,934	6,005	6,005	0.00	0	**1.29**	6,005	6,005	16.80
tz6117	6,117	4,171	4,171	0.00	0	**3.06**	4,171	4,171	21.60
eg7146	7,146	3,702	3,702	0.00	0	**0.55**	3,702	3,702	15.30
pla7397	7,397	279,242	279,242	0.00	0	**2.36**	279,242	279,242	32.50
ym7663	7,663	3,107	3,107	0.00	0	**0.67**	3,107	3,107	17.10
pm8079	8,079	1,382	1,382	0.00	0	**0.62**	1,382	1,382	14.50
ei8246	8,246	1,522	1,522	0.00	0	**1.34**	1,522	1,522	15.90
ar9152	9,152	8,196	8,196	0.00	0	**1.23**	8,196	8,196	24.80

(continued on next page)

Table A.2 (*continued*)

name	$\|V\|$	fCLH LB	fCLH UB	fCLH g[%]	fCLH #BC	fCLH t[s]	Contardo et al. (2019) LB	Contardo et al. (2019) UB	Contardo et al. (2019) t[s]
ja9847	9,847	7,872	7,872	0.00	0	**0.64**	7,872	7,872	13.50
gr9882	9,882	3,170	3,170	0.00	0	**1.11**	3,170	3,170	14.90
kz9976	9,976	8,244	8,244	0.00	0	**1.02**	8,244	8,244	16.10
fi10639	10,639	3,742	3,742	0.00	0	**1.69**	3,742	3,742	13.50
rl11849	11,849	6,452	6,452	0.00	0	**5.10**	6,452	6,452	24.20
usa13509	13,509	134,489	134,489	0.00	0	**1.18**	134,489	134,489	23.50
brd14051	14,051	2,426	2,426	0.00	0	**2.14**	2,426	2,426	17.90
mo14185	14,185	2,992	2,992	0.00	0	**1.71**	2,992	2,992	16.70
ho14473	14,473	1,534	1,534	0.00	0	**1.81**	1,534	1,534	19.00
d15112	15,112	8,154	8,154	0.00	0	**5.08**	8,154	8,154	32.60
it16862	16,862	3,724	3,724	0.00	0	**1.09**	3,724	3,724	20.50
d18512	18,512	2,914	2,914	0.00	0	**6.41**	2,914	2,914	30.20
vm22775	22,775	3,135	3,135	0.00	0	**2.86**	3,135	3,135	21.80
sw24978	24,978	4,120	4,120	0.00	0	**2.99**	4,120	4,120	24.20
fyg28534	28,534	400	400	0.00	0	**10.90**	400	400	34.60
bm33708	33,708	4,213	4,213	0.00	0	**7.25**	4,213	4,213	35.10
pla33810	33,810	302,160	302,160	0.00	0	**31.67**	302,160	302,160	154.70
bby34656	34,656	420	420	0.00	0	**11.43**	420	420	43.30
pba38478	38,478	413	413	0.00	0	**9.88**	413	413	33.20
ch71009	71,009	13,292	13,292	0.00	0	**13.96**	13,292	13,292	59.90
pla85900	85,900	399,677	399,677	0.00	0	**127.69**	399,677	399,677	577.30
sra104815	104,814	688	688	0.00	0	**35.93**	688	688	97.70
usa115475	115,475	13,530	13,530	0.00	0	**13.33**	13,530	13,530	64.30
ara238025	238,025	1,166	1,166	0.00	0	**53.05**	1,166	1,166	153.30
lra498378	498,378	4,995	4,995	0.00	0	**260.76**	4,995	4,995	689.40
lrb744710	744,710	1,702	1,702	0.00	0	**284.90**	1,702	1,702	567.10

Table A.3
Detailed results for the TSPlib instances with $p = 5$.

name	$\|V\|$	fCLH LB	fCLH UB	fCLH g[%]	fCLH #BC	fCLH t[s]	Contardo et al. (2019) LB	Contardo et al. (2019) UB	Contardo et al. (2019) t[s]
rw1621	1,621	414	414	0.00	0	**0.33**	414	414	12.50
u1817	1,817	715	715	0.00	0	**1.22**	715	715	15.90
rl1889	1,889	4,792	4,792	0.00	0	**1.74**	4,792	4,792	16.30
mu1979	1,979	1,877	1,877	0.00	0	**0.56**	1,877	1,877	14.90
pr2392	2,392	3,827	3,827	0.00	0	**4.51**	3,827	3,827	33.90
d15112-modif-2500	2,500	5,856	5,856	0.00	0	**3.44**	5,856	5,856	16.90
pcb3038	3,038	1,064	1,064	0.00	0	**1.79**	1,064	1,064	21.20
nu3496	3,496	1,123	1,123	0.00	0	**1.11**	1,123	1,123	14.60
ca4663	4,663	16,837	16,837	0.00	0	**1.51**	16,837	16,837	16.10
rl5915	5,915	4,554	4,554	0.00	0	**5.30**	4,554	4,554	27.00
rl5934	5,934	4,792	4,792	0.00	0	**4.85**	4,792	4,792	33.40
tz6117	6,117	2,918	2,918	0.00	0	**8.45**	2,918	2,918	37.10
eg7146	7,146	2,590	2,590	0.00	0	**2.27**	2,590	2,590	21.90
pla7397	7,397	174,542	174,542	0.00	0	**2.22**	174,542	174,542	32.70
ym7663	7,663	2,043	2,043	0.00	0	**0.95**	2,043	2,043	14.00
pm8079	8,079	938	938	0.00	0	**2.04**	938	938	18.40
ei8246	8,246	1,042	1,042	0.00	0	**2.71**	1,042	1,042	18.00
ar9152	9,152	6,752	6,752	0.00	0	**3.29**	6,752	6,752	21.40
ja9847	9,847	4,503	4,503	0.00	0	**0.98**	4,503	4,503	18.60
gr9882	9,882	2,151	2,151	0.00	0	**6.48**	2,151	2,151	37.00
kz9976	9,976	5,995	5,995	0.00	0	**4.70**	5,995	5,995	29.70
fi10639	10,639	2,739	2,739	0.00	0	**6.88**	2,739	2,739	23.30
rl11849	11,849	4,873	4,873	0.00	0	**20.29**	4,873	4,873	53.70
usa13509	13,509	103,671	103,671	0.00	0	**9.52**	103,671	103,671	53.40
brd14051	14,051	1,822	1,822	0.00	0	**4.61**	1,822	1,822	22.70
mo14185	14,185	2,143	2,143	0.00	0	**4.86**	2,143	2,143	30.30
ho14473	14,473	1,112	1,112	0.00	0	**4.39**	1,112	1,112	20.70
d15112	15,112	5,890	5,890	0.00	0	**42.21**	5,890	5,890	77.40
it16862	16,862	2,811	2,811	0.00	0	**3.13**	2,811	2,811	33.10
d18512	18,512	2,073	2,073	0.00	0	**65.95**	2,073	2,073	68.40
vm22775	22,775	2,269	2,269	0.00	0	**6.20**	2,269	2,269	41.70
sw24978	24,978	3,022	3,022	0.00	0	**14.66**	3,022	3,022	45.90
fyg28534	28,534	261	261	0.00	0	**64.21**	261	261	164.00
bm33708	33,708	2,747	2,747	0.00	0	**9.95**	2,747	2,747	53.80
pla33810	33,810	203,597	203,597	0.00	0	**179.93**	203,597	203,597	769.50

(*continued on next page*)

Table A.3 (*continued*)

name	$\lvert V \rvert$	fCLH LB	fCLH UB	fCLH $g[\%]$	fCLH #BC	fCLH $t[s]$	Contardo et al. (2019) LB	Contardo et al. (2019) UB	Contardo et al. (2019) $t[s]$
bby34656	34,656	286	286	0.00	0	**118.60**	286	286	256.70
pba38478	38,478	313	313	0.00	0	**71.79**	313	313	174.10
ch71009	71,009	10,944	10,944	0.00	11	**200.89**	10,944	10,944	269.30
pla85900	85,900	269,544	269,544	0.00	0	**1030.73**	269,544	269,544	3212.00
sra104815	104,814	508	508	0.00	0	**115.84**	508	508	237.20
usa115475	115,475	10,414	10,414	0.00	0	**196.31**	10,414	10,414	562.20
ara238025	238,025	855	855	0.00	5	**406.24**	855	855	578.20
lra498378	498,378	3,259	3,259	0.00	0	**561.74**	3,259	3,259	2250.30
lrb744710	744,710	1,167	1,176	0.77	0	TL	1,170	1,170	**4781.70**

Table A.4
Detailed results for the TSPlib instances with $p = 10$.

name	$\lvert V \rvert$	fCLH LB	fCLH UB	fCLH $g[\%]$	fCLH #BC	fCLH $t[s]$	Contardo et al. (2019) LB	Contardo et al. (2019) UB	Contardo et al. (2019) $t[s]$
rw1621	1,621	285	285	0.00	0	**1.21**	285	285	15.80
u1817	1,817	458	458	0.00	6	**16.71**	458	458	42.30
rl1889	1,889	3,078	3,101	0.74	584	TL	3,101	3,101	**151.80**
mu1979	1,979	1,161	1,161	0.00	0	**0.61**	1,161	1,161	14.60
pr2392	2,392	2,581	2,581	0.00	550	1001.96	2,581	2,581	**166.10**
d15112-modif-2500	2,500	3,705	3,705	0.00	45	49.04	3,705	3,705	**45.90**
pcb3038	3,038	729	729	0.00	169	**133.16**	729	729	175.40
nu3496	3,496	757	757	0.00	0	**4.59**	757	757	20.70
ca4663	4,663	10,499	10,499	0.00	0	**2.15**	10,499	10,499	27.00
rl5915	5,915	3,104	3,262	4.84	52	TL	3,137	3,137	**361.80**
rl5934	5,934	3,078	3,143	2.07	105	TL	3,092	3,092	**337.50**
tz6117	6,117	1,902	1,902	0.00	0	**89.14**	1,902	1,902	180.30
eg7146	7,146	1,826	1,826	0.00	0	**3.35**	1,826	1,826	35.30
pla7397	7,397	121,968	121,968	0.00	0	**21.07**	121,968	121,968	104.80
ym7663	7,663	1,447	1,447	0.00	0	**5.87**	1,447	1,447	36.60
pm8079	8,079	653	653	0.00	0	**6.36**	653	653	27.40
ei8246	8,246	722	722	0.00	5	**114.22**	722	722	225.20
ar9152	9,152	4,273	4,273	0.00	54	218.28	4,273	4,273	**117.70**
ja9847	9,847	2,766	2,766	0.00	0	**3.48**	2,766	2,766	28.50
gr9882	9,882	1,372	1,372	0.00	0	**8.71**	1,372	1,372	54.40
kz9976	9,976	4,155	4,155	0.00	0	**13.12**	4,155	4,155	61.80
fi10639	10,639	1,855	1,855	0.00	0	**37.92**	1,855	1,855	97.40
rl11849	11,849	3,158	3,179	0.66	31	TL	3,164	3,164	**385.00**
usa13509	13,509	67,075	67,075	0.00	30	**465.62**	67,075	67,075	503.80
brd14051	14,051	1,257	1,265	0.63	30	TL	1,265	1,265	**181.20**
mo14185	14,185	1,405	1,405	0.00	0	**65.60**	1,405	1,405	211.60
ho14473	14,473	738	738	0.00	6	140.97	738	738	**119.50**
d15112	15,112	3,760	4,005	6.12	5	TL	3,785	3,785	**599.70**
it16862	16,862	1,574	1,574	0.00	0	**16.45**	1,574	1,574	87.80
d18512	18,512	1,332	1,371	2.84	94	TL	1,340	1,340	**2060.10**
vm22775	22,775	1,315	1,315	0.00	0	**31.50**	1,315	1,315	134.30
sw24978	24,978	2,061	2,061	0.00	0	**103.94**	2,061	2,061	155.90
fyg28534	28,534	175	176	0.57	36	TL	176	176	**3137.30**
bm33708	33,708	1,907	1,907	0.00	0	1063.64	1,907	1,907	**1015.40**
pla33810	33,810	135,441	168,992	19.85	0	TL	136,517	136,517	**6825.70**
bby34656	34,656	192	197	2.54	2	TL	192	192	**3449.20**
pba38478	38,478	206	232	11.21	2	TL	207	207	**6831.70**
ch71009	71,009	7,162	7,599	5.75	0	TL	7,183	7,183	**1595.20**
pla85900	85,900	176,333	200,715	12.15	0	TL	180,497	180,497	**19575.70**
sra104815	104,814	349	373	6.43	0	TL	354	354	**7059.50**
usa115475	115,475	6,700	6,866	2.42	0	TL	6,736	6,736	**4784.00**
ara238025	238,025	549	573	4.19	0	TL	552	552	**7953.80**
lra498378	498,378	2,160	2,468	12.48	0	TL	2,232	2,232	**16623.10**
lrb744710	744,710	748	942	20.59	0	TL	**795**	**810**	TL2

Table A.5
Detailed results for the TSPlib instances with $p = 15$.

		fCLH					Contardo et al. (2019)		
name	$\|V\|$	LB	UB	$g[\%]$	#BC	$t[s]$	LB	UB	$t[s]$
rw1621	1,621	217	217	0.00	8	**6.62**	217	217	27.50
u1817	1,817	359	359	0.00	737	1366.66	359	359	**386.20**
rl1889	1,889	2,384	2,384	0.00	264	1146.25	2,384	2,384	**181.60**
mu1979	1,979	868	868	0.00	0	**0.92**	868	868	17.50
pr2392	2,392	2,011	2,110	4.69	394	TL	2,039	2,039	**1581.30**
d15112-modif-2500	2,500	2,972	2,972	0.00	30	**95.50**	2,972	2,972	110.70
pcb3038	3,038	567	701	19.12	276	TL	578	578	**3474.90**
nu3496	3,496	604	604	0.00	12	**31.38**	604	604	54.40
ca4663	4,663	8,296	8,296	0.00	0	**3.80**	8,296	8,296	36.20
rl5915	5,915	2,403	2,768	13.19	186	TL	2,441	2,441	**6266.20**
rl5934	5,934	2,392	2,393	0.04	83	TL	2,393	2,393	**821.20**
tz6117	6,117	1,508	1,727	12.68	459	TL	1,528	1,528	**764.70**
eg7146	7,146	1,308	1,308	0.00	0	**3.81**	1,308	1,308	44.70
pla7397	7,397	95,270	95,270	0.00	0	**186.73**	95,270	95,270	217.50
ym7663	7,663	1,054	1,054	0.00	0	**16.26**	1,054	1,054	67.60
pm8079	8,079	517	517	0.00	0	**9.59**	517	517	52.30
ei8246	8,246	575	655	12.21	265	TL	579	579	**2268.70**
ar9152	9,152	3,250	3,284	1.04	9	TL	3,250	3,250	**397.70**
ja9847	9,847	1,990	1,990	0.00	0	**9.36**	1,990	1,990	65.10
gr9882	9,882	1,038	1,038	0.00	0	**12.58**	1,038	1,038	74.40
kz9976	9,976	3,261	3,261	0.00	0	**169.08**	3,261	3,261	433.10
fi10639	10,639	1,456	1,484	1.89	182	TL	1,461	1,461	**1030.10**
rl11849	11,849	2,424	3,129	22.53	10	TL	2,462	2,462	**14808.20**
usa13509	13,509	51,476	57,780	10.91	6	TL	52,178	52,178	**4740.90**
brd14051	14,051	954	1,058	9.83	248	TL	966	966	**1832.50**
mo14185	14,185	1,110	1,166	4.80	166	TL	1,118	1,118	**692.60**
ho14473	14,473	577	647	10.82	105	TL	585	585	**826.10**
d15112	15,112	3,008	3,239	7.13	0	TL	3,037	3,037	**27897.80**
it16862	16,862	1,237	1,237	0.00	20	257.02	1,237	1,237	**142.10**
d18512	18,512	1,066	1,119	4.74	0	TL	1,075	1,075	**12403.90**
vm22775	22,775	1,099	1,099	0.00	0	**33.74**	1,099	1,099	265.60
sw24978	24,978	1,631	1,775	8.11	19	TL	1,637	1,637	**1751.50**
fyg28534	28,534	139	168	17.26	0	TL	**142**	**151**	TL2
bm33708	33,708	1,469	1,512	2.84	1	TL	1,475	1,475	**3952.00**
pla33810	33,810	106,684	137,325	22.31	0	TL	**110019**	**126933**	TL2
bby34656	34,656	152	189	19.58	0	TL	**155**	**164**	TL2
pba38478	38,478	158	208	24.04	0	TL	**160**	**163**	TL2
ch71009	71,009	5,564	6,459	13.86	0	TL	5,664	5,665	**13621.50**
pla85900	85,900	136,667	185,853	26.47	0	TL	**142938**	**159484**	TL2
sra104815	104,814	273	347	21.33	0	TL	277	277	**15204.60**
usa115475	115,475	5,090	6,259	18.68	0	TL	**5288**	**5419**	TL2
ara238025	238,025	432	511	15.46	0	TL	**441**	**458**	TL2
lra498378	498,378	1,647	2,147	23.29	0	TL	1,755	1,755	**44562.60**
lrb744710	744,710	586	694	15.56	0	TL	**626**	**653**	TL2

Table A.6
Detailed results for the TSPlib instances with $p = 20$.

		fCLH					Contardo et al. (2019)		
name	$\|V\|$	LB	UB	$g[\%]$	#BC	$t[s]$	LB	UB	$t[s]$
rw1621	1,621	186	186	0.00	376	**21.39**	186	186	34.80
u1817	1,817	306	310	1.29	6,425	TL	309	309	**2171.80**
rl1889	1,889	2,060	2,201	6.41	632	TL	2,089	2,089	**405.00**
mu1979	1,979	751	751	0.00	0	**1.15**	751	751	26.20
pr2392	2,392	1,694	1,864	9.12	383	TL	1,736	1,736	**7500.10**
d15112-modif-2500	2,500	2,519	2,965	15.04	227	TL	2,573	2,573	**1795.90**
pcb3038	3,038	480	618	22.33	124	TL	493	493	**54802.10**
nu3496	3,496	519	519	0.00	358	454.56	519	519	**128.30**
ca4663	4,663	7,024	7,024	0.00	0	**11.04**	7,024	7,024	57.00
rl5915	5,915	2,036	2,509	18.85	11	TL	2,083	2,083	**76684.10**
rl5934	5,934	2,068	2,412	14.26	68	TL	2,100	2,100	**4367.20**
tz6117	6,117	1,266	1,306	3.06	155	TL	1,278	1,278	**1988.70**
eg7146	7,146	972	972	0.00	0	**3.16**	972	972	51.30
pla7397	7,397	78,014	83,860	6.97	19	TL	78,817	78,817	**608.20**
ym7663	7,663	899	899	0.00	5	**63.54**	899	899	152.10
pm8079	8,079	430	430	0.00	5	**29.19**	430	430	91.40
ei8246	8,246	488	533	8.44	295	TL	497	497	**33150.50**
ar9152	9,152	2,653	2,818	5.86	80	TL	2,695	2,695	**6067.80**
ja9847	9,847	1,724	1,724	0.00	0	**8.76**	1,724	1,724	66.70
gr9882	9,882	877	877	0.00	4	**106.71**	877	877	235.60

(continued on next page)

Table A.6 (*continued*)

name	$\|V\|$	fCLH LB	fCLH UB	fCLH g[%]	fCLH #BC	fCLH t[s]	Contardo et al. (2019) LB	Contardo et al. (2019) UB	Contardo et al. (2019) t[s]
kz9976	9,976	2,762	3168	12.82	45	TL	2,790	2,790	**1540.20**
fi10639	10,639	1,228	1,500	18.13	16	TL	1,245	1,245	**15688.60**
rl11849	11,849	2,080	2,629	20.88	0	TL	**2119**	**2273**	TL2
usa13509	13,509	43,592	56,610	23.00	0	TL	**44740**	**46719**	TL2
brd14051	14,051	796	909	12.43	70	TL	803	803	**16721.90**
mo14185	14,185	914	1,022	10.57	23	TL	919	919	**3597.10**
ho14473	14,473	492	566	13.07	82	TL	497	497	**1470.20**
d15112	15,112	2,539	3,359	24.41	0	TL	**2581**	**2717**	TL2
it16862	16,862	1,059	1,059	0.00	0	**134.58**	1,059	1,059	563.60
d18512	18,512	902	1,218	25.94	0	TL	**912**	**969**	TL2
vm22775	22,775	919	967	4.96	42	TL	932	932	**1491.90**
sw24978	24,978	1,403	1,662	15.58	0	TL	1,421	1,421	**14835.00**
fyg28534	28,534	118	154	23.38	0	TL	**128**	**138**	TL2
bm33708	33,708	1,245	1,455	14.43	3	TL	1,257	1,257	**17808.70**
pla33810	33,810	89,123	118,855	25.02	0	TL	**91302**	**106076**	TL2
bby34656	34,656	**128**	159	19.50	0	TL	**128**	**138**	TL2
pba38478	38,478	**136**	175	22.29	0	TL	**136**	**148**	TL2
ch71009	71,009	4,637	5,868	20.98	0	TL	**4798**	**5250**	TL2
pla85900	85,900	115,361	163,300	29.36	0	TL	**119643**	**136984**	TL2
sra104815	104,814	227	294	22.79	0	TL	**232**	**236**	TL2
usa115475	115,475	4,198	5,855	28.30	0	TL	**4453**	**4747**	TL2
ara238025	238,025	359	466	22.96	0	TL	**372**	**407**	TL2
lra498378	498,378	1,342	1,726	22.25	0	TL	**1442**	**1573**	TL2
lrb744710	744,710	474	697	31.99	0	TL	**524**	**580**	TL2

Table A.7
Detailed results for the TSPlib instances with $p = 25$.

name	$\|V\|$	fCLH LB	fCLH UB	fCLH g[%]	fCLH #BC	fCLH t[s]	Contardo et al. (2019) LB	Contardo et al. (2019) UB	Contardo et al. (2019) t[s]
rw1621	1,621	166	166	0.00	8	**12.83**	166	166	38.70
u1817	1,817	265	275	3.64	1766	TL	272	272	**1875.80**
rl1889	1,889	1,833	2,063	11.15	294	TL	1,866	1,866	**1154.80**
mu1979	1,979	639	639	0.00	0	**2.11**	639	639	30.40
pr2392	2,392	1,492	1,956	23.72	135	TL	1,520	1,520	**23767.70**
d15112-modif-2500	2,500	2,205	2,581	14.57	225	TL	2,243	2,243	**5266.20**
pcb3038	3,038	425	545	22.02	104	TL	**433**	**470**	TL2
nu3496	3,496	441	441	0.00	11	**41.79**	441	441	116.80
ca4663	4,663	5,966	5,966	0.00	13	121.12	5,966	5,966	**118.20**
rl5915	5,915	1,786	2,201	18.86	20	TL	**1823**	**1916**	TL2
rl5934	5,934	1,825	2,113	13.63	19	TL	1,850	1,850	**25300.40**
tz6117	6,117	1,121	1,426	21.39	237	TL	**1152**	**1258**	TL2
eg7146	7,146	855	855	0.00	0	**9.76**	855	855	79.10
pla7397	7,397	68,964	78,142	11.75	35	TL	69,508	69,508	**776.80**
ym7663	7,663	785	819	4.15	20	TL	785	785	**189.40**
pm8079	8,079	367	367	0.00	0	**50.52**	367	367	184.80
ei8246	8,246	421	532	20.86	71	TL	**429**	**461**	TL2
ar9152	9,152	2,302	2,717	15.27	50	TL	2,355	2,355	**30531.70**
ja9847	9,847	1,362	1,362	0.00	0	**43.02**	1,362	1,362	154.80
gr9882	9,882	772	772	0.00	59	**236.54**	772	772	532.30
kz9976	9,976	2,439	2,800	12.89	72	TL	2,479	2,479	**4666.40**
fi10639	10,639	1,075	1,400	23.21	10	TL	**1103**	**1173**	TL2
rl11849	11,849	1,823	2,506	27.25	0	TL	**1838**	**2099**	TL2
usa13509	13,509	37,471	46,954	20.20	0	TL	**38150**	**40578**	TL2
brd14051	14,051	693	843	17.79	7	TL	**703**	**737**	TL2
mo14185	14,185	806	907	11.14	11	TL	818	818	**42123.30**
ho14473	14,473	433	547	20.84	99	TL	436	436	**11941.10**
d15112	15,112	2,201	2,877	23.50	0	TL	**2233**	**2447**	TL2
it16862	16,862	939	1,140	17.63	6	TL	951	951	**5284.40**
d18512	18,512	792	1029	23.03	0	TL	**795**	**881**	TL2
vm22775	22,775	798	815	2.09	1	TL	799	799	**3102.50**
sw24978	24,978	1,209	1,567	22.85	0	TL	**1233**	**1285**	TL2
fyg28534	28,534	**103**	133	22.56	0	TL	102	**112**	TL2
bm33708	33,708	1,069	1,358	21.28	0	TL	**1106**	**1140**	TL2
pla33810	33,810	80,661	108,074	25.37	0	TL	**81800**	**88861**	TL2
bby34656	34,656	**113**	144	21.53	0	TL	**113**	**128**	TL2
pba38478	38,478	**118**	160	26.25	0	TL	**118**	**134**	TL2
ch71009	71,009	4,013	5,491	26.92	0	TL	**4145**	**4321**	TL2
pla85900	85,900	103,546	145,695	28.93	0	TL	**107527**	**124693**	TL2
sra104815	104,814	203	241	15.77	0	TL	**206**	**216**	TL2
usa115475	115,475	3,649	5,347	31.76	0	TL	**3808**	**4453**	TL2
ara238025	238,025	307	393	21.88	0	TL	**319**	**357**	TL2
lra498378	498,378	1,109	1,608	31.03	0	TL	**1228**	**1339**	TL2
lrb744710	744,710	415	636	34.75	0	TL	**454**	**533**	TL2

Table A.8
Detailed results for the `TSPlib` instances with $p = 30$.

name	$\|V\|$	fCLH LB	fCLH UB	fCLH $g[\%]$	fCLH #BC	fCLH $t[s]$	Contardo et al. (2019) LB	Contardo et al. (2019) UB	Contardo et al. (2019) $t[s]$
rw1621	1,621	147	147	0.00	32	**14.84**	147	147	64.80
u1817	1,817	240	251	4.38	2,438	TL	241	241	**4228.30**
rl1889	1,889	1,638	1,885	13.10	226	TL	1,657	1,657	**1603.40**
mu1979	1,979	552	552	0.00	0	**4.21**	552	552	34.40
pr2392	2,392	1,351	1765	23.46	362	TL	**1379**	**1471**	TL2
d15112-modif-2500	2,500	1,980	2,604	23.96	414	TL	2,029	2,029	**50627.40**
pcb3038	3,038	382	508	24.80	235	TL	**386**	**412**	TL2
nu3496	3,496	396	396	0.00	43	**135.53**	396	396	205.70
ca4663	4,663	5,361	5,361	0.00	0	**85.75**	5,361	5,361	152.60
rl5915	5,915	1,618	2,210	26.79	10	TL	**1624**	**1853**	TL2
rl5934	5,934	1,631	2,116	22.92	60	TL	**1658**	**1812**	TL2
tz6117	6,117	1,001	1,304	23.24	224	TL	**1025**	**1142**	TL2
eg7146	7,146	744	744	0.00	0	**8.53**	744	744	76.80
pla7397	7,397	62,057	71,915	13.71	0	TL	63,770	63,770	**20045.10**
ym7663	7,663	703	748	6.02	0	TL	715	715	**536.60**
pm8079	8,079	330	330	0.00	3	**119.69**	330	330	459.80
ei8246	8,246	384	508	24.41	42	TL	**386**	**412**	TL2
ar9152	9,152	2,050	2,648	22.58	14	TL	2,080	2,080	**10145.70**
ja9847	9,847	1,220	1,220	0.00	0	**16.65**	1,220	1,220	268.30
gr9882	9,882	677	677	0.00	9	**334.72**	677	677	1024.60
kz9976	9,976	2,191	2,820	22.30	44	TL	2,230	2,230	**11820.20**
fi10639	10,639	967	1,251	22.70	10	TL	**974**	**1017**	TL2
rl11849	11,849	**1649**	2,255	26.87	0	TL	1,641	**1855**	TL2
usa13509	13,509	34,137	45,591	25.12	0	TL	**34771**	**37036**	TL2
brd14051	14,051	618	812	23.89	4	TL	**620**	**668**	TL2
mo14185	14,185	719	936	23.18	0	TL	**732**	**767**	TL2
ho14473	14,473	389	511	23.87	30	TL	396	396	**29545.80**
d15112	15,112	1,994	2,662	25.09	0	TL	**2009**	**2254**	TL2
it16862	16,862	849	947	10.35	57	TL	854	854	**16851.80**
d18512	18,512	**712**	963	26.06	0	TL	**712**	**786**	TL2
vm22775	22,775	690	802	13.97	7	TL	696	696	**13315.20**
sw24978	24,978	1,083	1,514	28.47	0	TL	**1096**	**1215**	TL2
fyg28534	28,534	**93**	122	23.77	0	TL	92	**108**	TL2
bm33708	33,708	930	1,221	23.83	0	TL	**968**	**1065**	TL2
pla33810	33,810	70,365	95,203	26.09	0	TL	**71480**	**83187**	TL2
bby34656	34,656	101	137	26.28	0	TL	**102**	**133**	TL2
pba38478	38,478	**107**	149	28.19	0	TL	**107**	**125**	TL2
ch71009	71,009	3,581	4,947	27.61	0	TL	**3724**	**4098**	TL2
pla85900	85,900	90,212	127,681	29.35	0	TL	**93871**	**114244**	TL2
sra104815	104,814	182	216	15.74	0	TL	**186**	**204**	TL2
usa115475	115,475	3,258	4,770	31.70	0	TL	**3391**	**3874**	TL2
ara238025	238,025	277	386	28.24	0	TL	**288**	**368**	TL2
lra498378	498,378	1,015	1,486	31.70	0	TL	**1073**	**1192**	TL2
lrb744710	744,710	370	580	36.21	0	TL	**403**	**475**	TL2

References

Al-Khedhairi, A., & Salhi, S. (2005). Enhancements to two exact algorithms for solving the vertex *p*-center problem. *Journal of Mathematical Modelling and Algorithms, 4*(2), 129–147.
Caruso, C., Colorni, A., & Aloi, L. (2003). Dominant, an algorithm for the *p*-center problem. *European Journal of Operational Research, 149*(1), 53–64.
Çalık, H., Labbé, M., & Yaman, H. (2019). *P*-center problems. In *Location science* (pp. 51–65). Springer.
Çalık, H., & Tansel, B. C. (2013). Double bound method for solving the *p*-center location problem. *Computers & Operations Research, 40*(12), 2991–2999.
Chen, D., & Chen, R. (2009). New relaxation-based algorithms for the optimal solution of the continuous and discrete *p*-center problems. *Computers & Operations Research, 36*(5), 1646–1655.
Conforti, M., Cornuéjols, G., Zambelli, G., et al. (2014). *Integer programming*: vol. 271. Springer.
Contardo, C., Iori, M., & Kramer, R. (2019). A scalable exact algorithm for the vertex *p*-center problem. *Computers & Operations Research, 103*, 211–220.
Costa, A. M. (2005). A survey on benders decomposition applied to fixed-charge network design problems. *Computers & Operations Research, 32*(6), 1429–1450.
Daskin, M. S. (2013). *Network and discrete location: Models, algorithms, and applications* (2nd ed). John Wiley & Sons.
Elloumi, S., Labbé, M., & Pochet, Y. (2004). A new formulation and resolution method for the *p*-center problem. *INFORMS Journal on Computing, 16*(1), 84–94.
Fischetti, M., Ljubić, I., & Sinnl, M. (2017). Redesigning benders decomposition for large-scale facility location. *Management Science, 63*(7), 2146–2162.
Friedler, S. A., & Mount, D. M. (2010). Approximation algorithm for the kinetic robust k-center problem. *Computational Geometry, 43*(6-7), 572–586.
Garcia-Diaz, J., Menchaca-Mendez, R., Menchaca-Mendez, R., Hernández, S. P., Pérez-Sansalvador, J. C., & Lakouari, N. (2019). Approximation algorithms for the vertex *k*-center problem: Survey and experimental evaluation. *IEEE Access, 7*, 109228–109245.
Garfinkel, R., Neebe, A., & Rao, M. (1977). The *m*-center problem: Minimax facility location. *Management Science, 23*(10), 1133–1142.
Geraci, F., Leoncini, M., Montangero, M., Pellegrini, M., & Renda, M. E. (2009). K-Boost: A scalable algorithm for high-quality clustering of microarray gene expression data. *Journal of Computational Biology, 16*(6), 859–873.
Hakimi, S. L. (1965). Optimum distribution of switching centers in a communication network and some related graph theoretic problems. *Operations Research, 13*, 462–475.
Ilhan, T., Özsoy, F. A., & Pinar, M. (2002). An efficient exact algorithm for the vertex *p*-center problem and computational experiments for different set covering subproblems. *Technical Report*. Bilkent University, Department of Industrial Engineering, Technical Report. http://www.optimization-online.org/DB_HTML/2002/12/588.html
Ilhan, T., & Pinar, M. (2001). An efficient exact algorithm for the vertex *p*-center problem. *Technical Report*. Bilkent University, Department of Industrial Engineering. http://www.optimization-online.org/DB_HTML/2001/09/376.html
Jia, H., Ordóñez, F., & Dessouky, M. (2007). A modeling framework for facility location of medical services for large-scale emergencies. *IIE Transactions, 39*(1), 41–55.
Kariv, O., & Hakimi, S. L. (1979). An algorithmic approach to network location problems. I: The *p*-centers. *SIAM Journal on Applied Mathematics, 37*(3), 513–538.
Kleindessner, M., Awasthi, P., & Morgenstern, J. (2019). Fair *k*-center clustering for data summarization. In *International Conference on Machine Learning* (pp. 3448–3457). PMLR.
(2019). *Location science*. In G. Laporte, S. Nickel, & F. Saldanha da Gama (Eds.) (2nd ed). Springer.

Lu, C.-C., & Sheu, J. B. (2013). Robust vertex *p*-center model for locating urgent relief distribution centers. *Computers & Operations Research, 40*(8), 2128–2137.
Malkomes, G., Kusner, M. J., Chen, W., Weinberger, K. Q., & Moseley, B. (2015). Fast distributed *k*-center clustering with outliers on massive data. *Advances in Neural Information Processing Systems, 28*, 1063–1071.
Meinl, T., Ostermann, C., & Berthold, M. R. (2011). Maximum-score diversity selection for early drug discovery. *Journal of Chemical Information and Modeling, 51*(2), 237–247.
Minieka, E. (1970). The *m*-center problem. *SIAM Review, 12*, 138–139.
Rahmaniani, R., Crainic, T. G., Gendreau, M., & Rei, W. (2017). The Benders decomposition algorithm: A literature review. *European Journal of Operational Research, 259*(3), 801–817.
Reinelt, G. (1991). TSPLIB-A traveling salesman problem library. *ORSA Journal on Computing, 3*(4), 376–384.
Snyder, L. V., & Shen, Z. J. M. (2011). *Fundamentals of supply chain theory*. Wiley Online Library.