An exact solution framework for the multiple gradual cover location problem

Eduardo Álvarez-Miranda *[1] and Markus Sinnl†[2]

[1]Department of Industrial Engineering, Universidad de Talca, Curicó, Chile
[2]Department of Statistics and Operations Research, Faculty of Business, Economics and Statistics, University of Vienna, Vienna, Austria

Abstract

Facility and covering location models are key elements in many decision aid tools in logistics, supply chain design, telecommunications, public infrastructure planning, and many other industrial and public sectors. In many applications, it is likely that customers are not dichotomously covered by facilities, but gradually covered according to, e.g., the distance to the open facilities. Moreover, customers are not served by a single facility, but by a collection of them, which jointly serve them. In this paper we study the recently introduced *multiple gradual cover location problem* (MGCLP). The MGCLP addresses both of the issues described above.

We provide four different mixed-integer programming formulations for the MGCLP, all of them exploiting the submodularity of the objective function and developed a branch-and-cut framework based one these formulations. The framework is further enhanced by starting and primal heuristics and initialization procedures.

The computational results show that our approach allows to effectively address different sets of instances. We provide optimal solution values for 13 instances from literature, where the optimal solution was not known, and additionally provide improved solution values for seven instances. Many of these instances can be solved within a minute. We also analyze the dependence of the solution-structure on instance-characteristics.

1. Introduction and motivation

Facility and covering location models are key elements in many decision aid tools in logistics, supply chain design, telecommunications, public infrastructure planning, and many other industrial and public sectors. Classical models typically aim at constructing location decisions that ensure that all (or as many as possible) customers are covered by as few facilities as possible. The reader is referred to [Daskin, 2013, Laporte et al., 2015] for two fundamental textbooks of location theory and science.

As explained in Berman et al. [2018], Drezner and Drezner [2014], the use of a dichotomic scheme for for modeling coverage (i.e., a given facility either covers a given client or not), as well as the common consideration that clients are served by a single facility, are two aspects that not necessarily respond well to real-world needs. Instead, in practical contexts, a facility might cover only a *portion* of the demand of a customer, and multiple facilities would *jointly* or *cooperatively* serve a customer simultaneously (for instance, each of them would cover a portion of its demand).

The first issue, partial or gradual coverage, has been studied already since the 80's (see, e.g., [Church and Roberts, 1983]), and the most common way for modeling partial covering is by a distance-based approach.

*ealvarez@utalca.cl
†markus.sinnl@univie.ac.at

In these approaches, two critical distances, say r and R ($r \leq R$), are defined. If the distance between a facility and an allocated customer, say d, does not exceed r, then the customer is *fully* covered. On the contrary, if such distance is greater than R, then the customer is *not* covered at all. For intermediate distances ($r < d < R$), the coverage diminishes *gradually* according to some d-dependent function. For example, [Thompson, 1982] proposes, for a super-drugstore chain, coverage functions depending on the distance from the facility and city size. Likewise, in [Jones and Simmons, 1993] a list of rules for location of a retail facility are provided, which is based on the use of r and R distances; the authors explain that such rules appear to be very common in the industry (see [Berman and Krass, 2002], for further details). In this retail industry examples, as well as in any other context, the coverage level is typically viewed as a decreasing function of the distance to the facility. As a matter of fact, there are several alternatives for this function; relevant examples, covering the efforts devoted to this topic over the last 25 years, can be found, e.g., in [Berman and Krass, 1998, 2002, Berman et al., 2003, 2009b, Drezner et al., 2004, Drezner and Drezner, 2014, Eiselt and Marianov, 2009, Ghosh et al., 1995] and [Karasakal and Karasakal, 2004].

The second issue, *multiple (joint)* coverage, has also been addressed before in the literature. One of the first works consolidating the concepts on joint and partial coverage, and its implication in facilities deployment is presented in [Berman et al., 2009a]. In that paper, the authors explain that multiple location models have been proposed before in the literature, and give special emphasis on applications related to the location of siren stations [Current and O'Kelly, 1992, Wei et al., 2006], and cell phone towers [Akella et al., 2005]. It is shown that an adequate model of cooperation ensures that the coverage of a client is not necessarily performed only by the closest facility, but rather by a mixed of them based on their proximity. The authors study the connection between joint coverage and *backup* facilities (see, e.g., [Hogan and ReVelle, 1986], for an early reference on this issue). Based on this connection, one could also link joint coverage with *reliable* covering models, which have been proposed when dealing with uncertainty in the set of available facilities, the set of customers (or, eventually, their demand), or both. A relevant example can be found in [Snyder and Daskin, 2005], where the authors proposed a model in which every customer is allocated to a *main* facility and to a set of *back-up* facilities; these back-up facilities shall cover the demand of the customer in case of failure of the main facility (which is likely to be the closest one). Extensions and variants, following equivalent concepts, can be found in [Albareda-Sambola et al., 2011, Cui et al., 2010, Li et al., 2013, Shen et al., 2011].

Although the concepts of gradual coverage and multiple (joint) facility location have been studied already for decades, it was only recently that they were combined into a single modeling framework aiming at maximizing the total joint cover of all customers; this corresponds to the *multiple gradual cover location problem* (MGCLP), which has been proposed by [Berman et al., 2018]. The combination is achieved by the optimization of the *joint coverage function* introduced in [Drezner and Drezner, 2014], whose details are provided in the following section. In [Berman et al., 2018], the authors motivate the MGCLP and propose different (heuristic) algorithmic strategies for solving it (see Section2.1 for further details).

Contribution and Outline In this paper, we present four exact mixed-integer programming (MIP) approaches for the MGCLP. These approaches are based on two key elements: (i) they exploit the submodularity of the objective function, and (ii) the use of an exponential number of constraints, which can be separated efficiently and are used in a branch-and-cut framework. We also introduce preprocessing based on domination between facilities and our solution framework also contains starting and primal heuristics. Our approaches allow the optimal solution of 13 instances from literature, which have not been solved to optimality before. Many of these instances can be solved within a few seconds.

The paper is organized as follows. In Section 2, we present the formal definition of the joint coverage function and the MGCLP, and present the four MIP formulations. In Section 3, we describe the implementation details of the branch-and-cut algorithms used to solve the corresponding MIP instances; cut separation, starting and primal heuristics, and preprocessing. Furthermore, we also present a series of structural properties of optimal solutions, which enable an effective initialization. Computational results are given in Section 4. Finally, concluding remarks are outlined in Section 5.

2. Mixed-Integer Programming formulations for the MGCLP

2.1 The joint coverage function

The joint coverage function, presented in [Drezner and Drezner, 2014], and later used in [Berman et al., 2018], is based on probabilistic arguments. Let I denote the set of potential facility location and J the set of customers. Let $0 \leq f_{ij} \leq 1$ be the coverage that customer $j \in J$ receives from facility $i \in I$ (with $f_{ij} = 1$ meaning that the customer gets completely covered, and $f_{ij} = 0$ meaning that the customer does not get covered at all by the facility). Additionally, let $0 \leq \theta \leq 1$ be a given weighting parameter. The *joint coverage function (JCF)* for a customer $j \in J$ and a given set of facilities I' reads as

$$p_j(\theta, I') = \theta\Big(\max_{i \in I'} f_{ij}\Big) + (1-\theta)\Big(1 - \prod_{i \in I'}(1 - f_{ij})\Big) \tag{JCF}$$

In this function f_{ij} gets interpreted as the *probability of full coverage*. The function combines two extreme cases, namely that coverage events are correlated with a correlation coefficient of one, and that coverage events are independent; this combination is *controlled* by the parameter θ. Note that in contrast to many other location problem, with this function, co-location (i.e., opening more than one facility at a potential location) may be beneficial.

Let $w_j \geq 0$ be weights for each customer $j \in J$, and let K be a given (integer) number of facilities to be opened. Since, theoretically, we could open the K facilities in a single location, we define I^K as the set of potential facility locations, having K copies of each facility in I (to account for potential co-location). The *multiple gradual cover location problem (MGCLP)* is defined as follows [Berman et al., 2018].

$$W^*(\theta) = \max_{I' \subseteq I^K, |I'| \leq K} \quad W(\theta, I') = \sum_{j \in J} w_j p_j(\theta, I'), \tag{MGCLP}$$

i.e., in the MGCLP we want to open K facilities (potentially more than one at a given location) as to maximize the weighted sum of the JCF over all customer. As stated above, the MGCLP was introduced in Berman et al. [2018], where a greedy algorithm, an ascent and tabu search heuristic and an approximative MIP-based approach were presented. Berman et al. [2018] show that the greedy algorithm has an approximation guarantee of $1 - 1/e$ by proving that the objective function $W(\theta, I')$ is *nondecreasing* and *submodular* and combining it with the result of Nemhauser et al. [1978] (see Section 3.2 for details). The MIP-approach uses the tangent-line approximation (TLA) method for twice-differentiable concave objective functions, introduced in Aboolian et al. [2007]. In this approach, the objective function gets approximated by L line segments. In their computational study the authors choose three different L such that the TLA objective is within one, five, and ten percent of optimality, respectively.

2.2 MIP models for the MGCLP

In this section, we present four different MIP formulations, $(F1)$-$(F4)$, for the MGCLP, and exploit that the objective function $W(\theta, I')$ with $I' \in I^K$ is a nondecreasing and submodular function. Let $\Phi(S)$ be a real valued set-function over the subsets S of a ground-set N. Let $\rho_n(S) = \Phi(S \cup \{n\}) - \Phi(S)$ for all $S \subset N$ and $n \in N$, i.e., the marginal gain achieved by adding element n to the set S. Such a function $\Phi(S)$ is a nondecreasing and submodular function, iff $\Phi(T) \leq \Phi(S) + \sum_{n \in T \setminus S} \rho_n(S), \forall S, T \subseteq N$ (see, e.g., Nemhauser et al. [1978], which also presents additional equivalent definitions)

The four formulations built-on each other and are based on Nemhauser and Wolsey [1981], where it is shown that for any nondecreasing submodular function maximization problems can be formulated as MIPs by introducing an additional (continuous) variable η. This variable is used to measure the value of the objective function. The correctness of the objective value is ensured by an exponential family of cuts on η, as encoded by expression (SCuts). Let $\mathbf{z} \in \{0,1\}^{|N|}$ denote the characteristic vector of the set N. The family of cuts is given by

$$\eta \leq \Phi(S) + \sum_{i \in N \setminus S} \rho_i(S) z_i, \quad \forall S \subseteq N \tag{SCuts}$$

Note that while all of these cuts are valid, for correctness for problems with a cardinality constraint of value K on the size of S (like the MGCLP), it is enough to add all cuts for $|S| = K$. As shown in [Berman et al., 2018], the objective function $W(\theta, I')$ of the MGCLP is nondecreasing and submodular. In particular, it is the (weighted) sum of the nondecreasing and submodular functions $p_j(\theta, I')$, which in turn are a convex combination of two nondecreasing and submodular functions, namely $\max_{i \in I'} f_{ij}$ and $1 - \prod_{i \in I'}(1 - f_{ij})$.

Let $\mathbf{x} \in \{0,1\}^{|I| \times K}$ be a binary vector such that $x_i^k = 1$, if the k-th facility ($1 \leq k \leq K$) at location $i \in I$ is opened (recall that co-location is possible); and $x_i^k = 0$, otherwise. When needed, we write i^k to refer to the k-th facility at location i and identify an element of I^K by (i, k). The following constraint (CARD) ensures, that at most K facilities are opened.

$$\sum_{i \in I, 1 \leq k \leq K} x_i^k \leq K. \tag{CARD}$$

Additionally, in order to exclude symmetric solutions with regard to co-location, the following constraints can be used

$$x_i^k \leq x_i^{k+1}, \quad \forall i \in I, 1 \leq k \leq K-1; \tag{SYM}$$

these constraint ensure that the $(k+1)$-th facility in a location is only opened, if the k facilities with lower k have been opened before. These constraints are not necessary for correctness of the formulations, but proved very helpful in preliminary computations and thus are used in all formulations.

For a given set of facilities $I' \subset I^K$ and a given vector $\mathbf{x}$, let $\phi(\mathbf{x}, I')$ denote the right-hand-side of cuts (SCuts) when applied to the objective function $W(\theta, I')$ of the MGCLP. Note that the coefficients ρ_i for $(i,k) \in I^K \setminus I'$ of a cut can be easily calculated by calculating $p_j(\theta, I' \cup \{i\}) - p_j(\theta, I')$ for each customer $j \in J$ and summing up, i.e., we obtain

$$\phi(\mathbf{x}, I') = W(\theta, I') + \sum_{(i,k) \in I^K \setminus I} \Big(\sum_{j \in J} (p_j(\theta, I' \cup \{i\}) - p_j(\theta, I')) \Big) x_i^k. \tag{1}$$

Using this notation, (F1.1)-(F1.3) gives a first formulation $(F1)$;

$$(F1) \quad W^*(\theta) = \max \quad \eta \tag{F1.1}$$

$$\eta \leq \phi(\mathbf{x}, I'),\ \forall I' \in I^K : |I'| = K \tag{F1.2}$$

$$\text{(CARD)},\ \text{(SYM) and } \mathbf{x} \in \{0,1\}^{|I| \times K}. \tag{F1.3}$$

The objective function (F1.1) and constraints (F1.2) ensure that the objective is correct. As there is an exponential number of constraints (F1.2), our strategy for tackling this formulation relies on adding them on-the-fly when they are violated, within a branch-and-cut scheme (the separation of the constraints is discussed in Section 3.1).

The second formulation, $(F2)$, encoded by constraints (F2.1)-(F2.3), is given by

$$(F2) \quad W^*(\theta) = \max \quad \sum_{j \in J} \eta_j \tag{F2.1}$$

$$\eta_j \leq \phi_j(\mathbf{x}, I'),\ \forall j \in J, \forall I' \in I^K : |I'| = K \tag{F2.2}$$

$$\text{(CARD)},\ \text{(SYM), and } \mathbf{x} \in \{0,1\}^{|I| \times K}. \tag{F2.3}$$

In this second formulation, we exploit the fact that the objective function of the MGCLP *decomposes* by customer (as it it the sum of the functions $p_j(\theta, I')$ for each customer). Thus, instead of a single-variable η to measure the objective, we use continuous variables η_j for each $j \in J$, and have the sum of these variables in the objective function. Hence, each variable η_j now has an individual family of cuts $\phi_j(\mathbf{x}, I')$ of type (SCuts), which ensures that the part in the objective contributed by the coverage of customer j is correct.

In the third formulation $(F3)$, we further exploit the decomposability of the objective function $W(\theta, I')$. We model the $\max_{i \in I'} f_{ij}$ component of each $p_j(\theta, I')$ by using additional variables and only use cuts to deal with the $1 - \prod_{i \in I'}(1 - f_{ij})$ component. Note that in the MIP model of Berman et al. [2018], the max-part was formulated in a similar way, while the second part was approximated using the TLA approach. Let $\mathbf{y} \in \{0,1\}^{|I| \times |J|}$ be a vector of binary variables such that $y_{ij} = 1$ if the maximum of f_{ij} for a $j \in J$ is obtained by opening facility i, and $y_{ij} = 0$, otherwise. Let $\phi_j^P(\mathbf{x}, I')$ denote the cuts of type (SCuts) for $j \in J$ associated with the $1 - \prod_{i \in I'}(1 - f_{ij})$-part of the objective. We obtain the following formulation $(F3)$;

$$(F3) \quad W^*(\theta) = \max \quad \sum_{i \in I} \sum_{j \in J} \theta w_j f_{ij} y_{ij} + \sum_{j \in J} \eta_j \tag{F3.1}$$

$$\eta_j \leq \phi_j^P(\mathbf{x}, I'),\ \forall j \in J, \forall I' \in I^K : |I'| = K \tag{F3.2}$$

$$\sum_{i \in I} y_{ij} \leq 1, \quad \forall j \in J \tag{F3.3}$$

$$y_{ij} \leq x_i^1, \quad \forall i \in I, \forall j \in J \tag{F3.4}$$

$$\text{(SYM), (CARD)},\ \mathbf{x} \in \{0,1\}^{|N| \times K} \text{ and } \mathbf{y} \in \{0,1\}^{|I| \times |J|} \tag{F3.5}$$

In this MIP model, cuts (F3.2) ensure that the product-part of the objective function is correctly measured. Constraints (F3.3) make sure that only one facility can contribute to the max-part of any customer. Moreover, constraints (F3.4) model the fact that, if a facility at some location wants to contribute to the max-part, it must be opened; in particular, the first facility (of the K copies) at this location must be opened. Hence, these constraints complement the ordering imposed by the symmetry constraints (SYM).

Finally, in formulation $(F4)$, we again exploit the decomposability of the objective function. Compared to $(F3)$, we also model the max-part of the objective by using cuts, i.e., both parts of the objective are now modeled using cuts. To this end, for each customer $j \in J$, we introduce two continuous variables, η_j^M and η_j^P, to measure the contribution of the max-part and product-part, respectively. The cuts for the max-part actually have a nice form and are of polynomial size (see Nemhauser and Wolsey [1981] for further details): Assume that for a given customer j, the f_{ij} values are ordered in a nondecreasing order, i.e., $f_{|I|j} \geq f_{(|I|-1)j} \geq \ldots \geq f_{1j} \geq f_{0j}$, with $f_{|I|-0j}$ defined to be zero. Let $(\cdot)^+ = \max\{0, \cdot\}$. Then these cuts are of the form

$$\eta_j^M \leq \theta w_j \Big(f_{rj} + \sum_{i \in I} (f_{ij} - f_{rj})^+ x_i^1 \Big), \quad r = 0, \ldots, |I| - 1.$$

Note that only the first facilities for each location are involved in the cuts. Although they are of polynomial size, there can still be many of these cuts; hence, their separation is also embedded within a branch-and-cut fashion. Let $\phi_j^M(\mathbf{x}, r)$ denote these cuts for $r = 0, \ldots, |I| - 1$. Using these cuts, we obtain the formulation (F4);

$$(F4) \quad W^*(\theta) = \max \quad \sum_{j \in J} (\eta_j^M + \eta_j^P) \tag{F4.1}$$

$$\eta_j^P \leq \phi_j^P(\mathbf{x}, I'),\ \forall j \in J, \forall I' \in I^K : |I'| = K \tag{F4.2}$$

$$\eta_j^M \leq \phi_j^M(\mathbf{x}, r),\ \forall j \in J, r = 0, \ldots, |I| - 1 \tag{F4.3}$$

$$\text{(SYM), (CARD) and } \mathbf{x} \in \{0,1\}^{|N| \times K}. \tag{F4.4}$$

3. Implementation details of the branch-and-cut algorithms

All the four formulation have an exponential number of constraints ((F1.2), (F2.2), (F3.2) and (F4.2)), for ensuring the correctness of the objective function. These cuts (and also the polynomial-sized family (F4.3)) are separated on-the-fly using branch-and-cut approaches. In this section, separation of the cuts is described, as well as further ingredients of the branch-and-cut approaches.

3.1 Separation of cuts

The separation of cuts (F1.2), is performed as follows. Let $(\tilde{\mathbf{x}}, \tilde{\eta})$ be the values of the LP-relaxation at a given branch-and-bound node. If $\tilde{\mathbf{x}}$ is binary, then an exact separation of the cuts can be done by calculating $W(\theta, \bar{I})$, where $\bar{I} = \{i^k : \tilde{x}_i^k = 1\}$, i.e., the open facilities induced by $\tilde{\mathbf{x}}$. If $W(\theta, \bar{I}) > \tilde{\eta}$, the current LP-solution violates (F1.2), and we add the cut induced by $\bar{I}$.

Note that the exact separation of cuts for the case of binary $\tilde{\mathbf{x}}$ is enough to ensure correctness of our approach. Nonetheless, we also implemented a heuristic separation for the case of $\tilde{\mathbf{x}}$ being fractional. In this case, we sort the facilities i^k non-increasingly according to the values of $\tilde{x}_i^k$ and construct $\bar{I}$ by taking the K first facilities of the sorting. This solution is then used to induce the corresponding cut (which is added, if violated).

The separation of the other cuts, (F2.2), (F3.2), (F4.2) and (F4.3), is performed in an equivalent manner.

3.2 Starting heuristic and primal heuristic

To initialize the branch-and-cut framework, we add a feasible starting solution. This starting solution is constructed using the greedy $(1 - 1/e)$ approximation algorithm of [Nemhauser et al., 1978], which is also used in [Berman et al., 2018] as one of the tested approaches. We also try to improve the solution constructed by the greedy algorithm with a local search. In the local search phase, for each opened facility in the solution, we try to replace it with another one, if it gives an improved solution value. We pass through all opened facilities (starting with the one added last by the greedy algorithm) and repeat this procedure, until no improvement is found in a round of iterations. The starting heuristic is outlined in Algorithm 1.

input : instance (I, J, f, θ, K) of the MGCLP
output: feasible solution S
1 $S \leftarrow \emptyset$
2 **for** $k = 1$ **to** K **do**
3 $\quad i^* = \arg\max_{i \in I} W(\theta, S \cup \{i\})$
4 $\quad S \leftarrow S \cup \{i^*\}$
5 `improve` $\leftarrow true$
6 **while** *improve* **do**
7 $\quad$ `improve` $\leftarrow false$
8 $\quad$ **for** $k = K$ **to** 1 **do**
9 $\quad\quad S' \leftarrow S \setminus S[k]$
10 $\quad\quad$ **for** $i \in I$ **do**
11 $\quad\quad\quad$ **if** $W(\theta, S' \cup \{i\}) > W(\theta, S)$ **then**
12 $\quad\quad\quad\quad S \leftarrow S' \cup \{i\}$
13 $\quad\quad\quad\quad$ `improve` $\leftarrow true$
14 $\quad\quad\quad\quad$ break

Algorithm 1: Greedy Heuristic of [Nemhauser et al., 1978] applied to the MGCLP, complemented by a local search phase.

To speed-up the evaluation of $\arg\max_{i \in I} W(\theta, S \cup \{i\})$ in line 3, we use *lazy evaluation* (see Leskovec et al. [2007]), which exploits the submodularity of W, according to the following rule. In the first iteration, we calculate $W(\theta, \{i\})$ for each $i \in I$. We add i^* to S (due to the arg max criterion), and also store all the facilities in a priority queue, sorted by decreasing values of $W(\theta, \{i\})$. In the remaining iterations, instead of calculating $W(\theta, S \cup \{i\})$ for each $i \in I$, we start by calculating $\rho' = W(\theta, S \cup \{i'\})$, for the top-element i' in the priority queue. We then compare the value ρ' with the stored value ρ'' (of the second element i'' in the priority queue). If $\rho' \geq \rho''$, we have that i' gives the arg max, since due to submodularity, the values of $W(\theta, S \cup \{i\})$ are non-increasing when the size of S increases. If $\rho' < \rho''$, we re-insert i' in the priority queue with value ρ', and repeat the procedure for i''.

During the branch-and-cut, we use a modified version of Algorithm 1 as primal heuristic. In this modified version, in line 3, we use $\arg\max_{i \in I} \tilde{x}_i^k W(\theta, S \cup \{i^k\})$, with $\tilde{\mathbf{x}}$ being a (eventually integer) solution at a given node of the branch-and-bound tree. In order to save computation time on unnecessary runs of the local search, we store all solution values of (intermediate) solutions found during previous runs of the heuristic in a hash-map, and stop the run of the local search, if the currently constructed solution has the same value as a previously encountered solution.

3.3 Preprocessing and initialization

Due to co-location, the number of variables in the models can be very large, as there could be instances where, e.g., the optimal solution only uses one location, and all K facilities are opened there. In this section, we give results that allow removing of some of the potential co-locations by showing that they will never be used in an optimal solution (in our framework, we set all variables associated which such co-locations to zero).

For the results presented below, we assume that the ordering constraint (SYM) is part of the corresponding MIP model; hence, given a location i and positions l, k with $k < l$ a facility i^k will always be opened *before* a facility with higher superscript i^l. The first result is given by the following theorem.

Theorem 1. *Let $i \in I$ be a facility location with each $f_{ij} = 1$ or $f_{ij} = 0$, $j \in J$. Then in an optimal solution, at most one facility will be opened at location i.*

Proof. Opening another facility at this location will neither improve the max-part of the objective function, nor the product-part. □

Complementary, the next result exploits that the objective function $W(\theta, I')$ is a nondecreasing and submodular function. For a facility $i \in I$ an and integer k, let $\mathcal{I}^k = \{i^{k'} : k' \leq k\}$, i.e., be the set of the first k facilities at this location, with $\mathcal{I}^0 = \emptyset$.

Theorem 2. *Let UB^{K-k} denote an upper bound for the objective value for any solution with $K - k$ open facilities and z be the value of a feasible solution. If $W(\theta, \mathcal{I}^k) + UB^{K-k} < z$, then no facilities will be opened in positions k to K at location i in any optimal solution.*

Proof. Since $W(\theta, \cdot)$ is a nondecreasing and submodular function we have that $\sum_{k' \leq k} \Big(\sum_{j \in J} (p_j(\theta, \mathcal{I}^k) - p_j(\theta, \mathcal{I}^{k-1})) \Big) = W(\theta, \mathcal{I}^k)$ is an upper bound on the marginal gain, which can be achieved by opening the first k facilities at location i. Thus, taking any solution where $K - k$ facilities have been opened and then opening the first k facilities at i, we can never get a solution with objective at least z, and hence, such a solution cannot be optimal. □

To use this results, an upper bound UB^{K-k} on the objective value for any solution with $k' = K - k$ open facilities is needed. We use two different ways to calculate such a bound, and then use the smaller value. The first way is to use the greedy $1 - (1/e)$ approximation algorithm (see Section 3.2). Let $z^{k'}$ be the value of the solution constructed at step k', as the algorithm has an $1 - (1/e)$ approximation guarantee, we have that $UB^{k'} \leq \frac{z^{k'}}{1-(1/e)}$. The second way consists of calculating the marginal gain $\sum_{j \in J} (p_j(\theta, \mathcal{I}^l) - p_j(\theta, \mathcal{I}^{l-1}))$ for all i and l up to k', sorting the resulting values in a nonincreasing way, and then summing up the first k' values.

In addition to above results, which allow the removing of co-locations, there is also the following dominance result, which allows the removal of facility locations.

Theorem 3. *Let $i, i' \in I$ be two facilities, with $f_{ij} \geq f_{i'j}$ for each $j \in J$. Then, in an optimal solution, no facility will be opened at location i'.*

Proof. Suppose there is a solution S' where a facility is opened at location i'. Construct another solution S, where the open facility at i' is replaced with an open facility at location i. Since $f_{ij} \geq f_{i'j}$ for each $j \in J$, the objective value of S is at least as large as the one of S'. □

Moreover, we also add all cuts induced by $I' = \emptyset$ to initialize our framework. The efficacy of this initialization is evaluated in the following section,

4. Computational results

The branch-and-cut framework was implemented in C++ using CPLEX 12.7, which was left at default settings. The runs were carried out on an Intel Xeon E5 v4 CPU with 2.2 GHz and 6GB memory and using a single thread. The timelimit for a run was set to 600 seconds.

4.1 Instance description

To evaluate the effectiveness of our approach, we used the same instances as in Berman et al. [2018]. They are based on 40 p-median instances from the OR-library Beasly [1990]. Each node of the instances is a customer and also a potential facility location, and the weights are uniform. The instances have up to 900 nodes and K is up to 200 (see Tables 1-6 for the values for each instance). To define the coverage rates f_{ij}, $i \in I, j \in J$ a linear decline function is used to convert the distances d_{ij} of an instance to values f_{ij}. In particular, given two threshold values $r \geq 0$ and $R > r$, the values f_{ij} are defined as follows:

$$f_{ij} = \begin{cases} 1 & \text{if } d_{ij} \leq r \\ 1 - \frac{d_{ij}-r}{R-r} & \text{if } r < d_{ij} < R \\ 0 & \text{if } d_{ij} \geq R \end{cases} .$$

In Berman et al. [2018], the authors use $r = 5$ and $R = 20$, along with $\theta = 0.2$. In addition to this, we also tested on instances using $r = 10$ and $R = 25$ (thus, having a larger number of $f_{ij} > 0$) and also $\theta \in \{0.2, 0.5, 0.8\}$. In total, this gives 240 instances. We refer to the resulting instance set as pm-r-R-θ, e.g., pm-5-20-0.2 are the instances used in Berman et al. [2018].

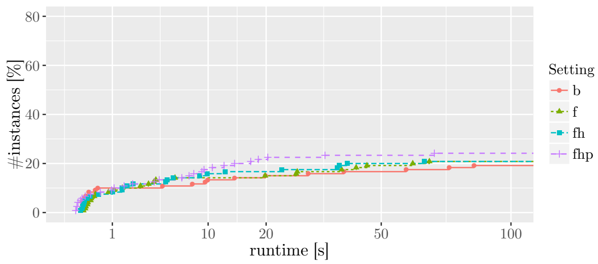
4.2 Assessing the effectiveness of the proposed strategies

First, we give an overview on the performance of the different formulations and also of the different ingredients of our framework (i.e., separation on fractional solutions, preprocessing, initialization, and heuristics). In particular, for each of the formulation, we tested the following four configurations:

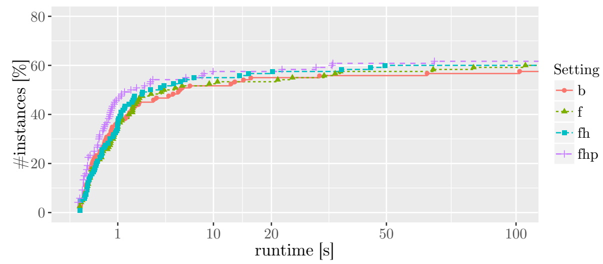
- `b`: In this basic setting, we only use the separation for integer solutions, and do not use the preprocessing, initialization, nor the (starting and primal) heuristic.
- `f`: In this setting, we also do the separation for fractional solutions.
- `fh`: This is setting `f` together with the starting heuristic and the primal heuristic.
- `fhp`: This is setting `fh` together with the preprocessing and also the initialization, i.e., all ingredients of our framework are turned on.

The computational study was carried out on all instances with up to 400 nodes (these are 120 instances). In Figures 1a-1d we report the performance profile plots of the runtime to optimality, while in Figures 2a-2d we report the performance profile plots of the attained optimality gap $g[\%]$ (calculated as $100 \cdot (UB - z^*)/(z^*)$, where UB is the upper bound and z^* is the value of the best solution found) for all formulations, instances and settings.

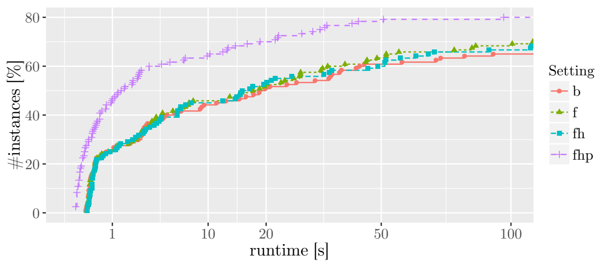
From the results reported in the performance profiles two relevant conclusions can de drawn: First, in terms of running times and attained gaps, the (F4) formulation seems to be the most effective one. With the (F4) model runtimes below ten seconds can be achieved for more than 70% of the instances, while using (F1), only around 10% of the instances can be solved to optimality within ten seconds. Furthermore, when looking



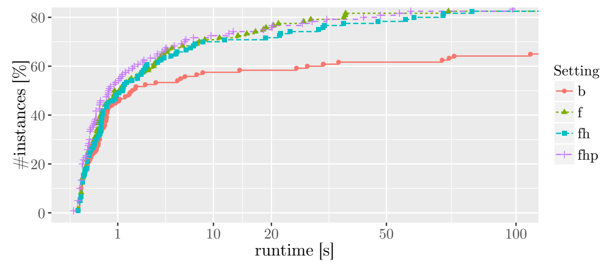
(a) Runtime performance of (F1)



(b) Runtime performance of (F2)

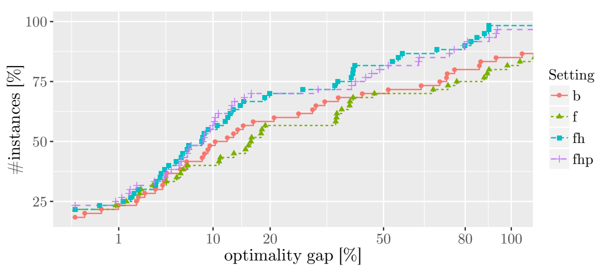


(c) Runtime performance of (F3)

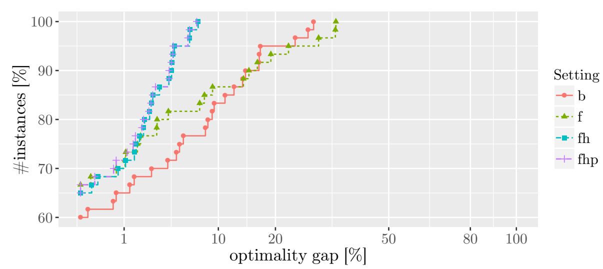


(d) Runtime performance of (F4)

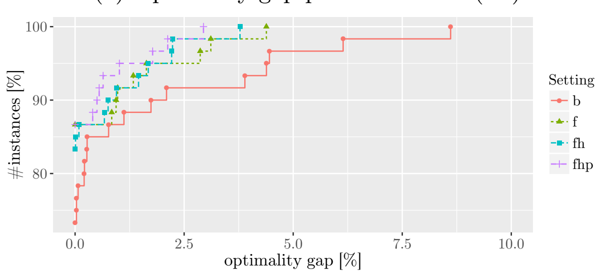
Figure 1: Performance profiles of runtimes for (F1)-(F4) and different settings



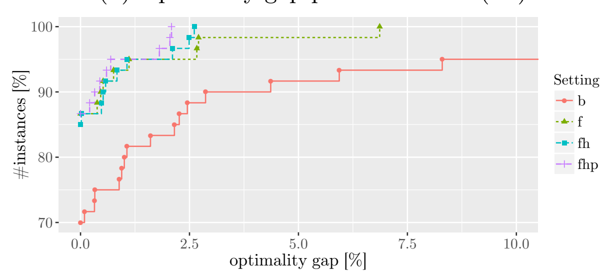
(a) Optimality gap performance of (F1)



(b) Optimality gap performance of (F2)



(c) Optimality gap performance of (F3)



(d) Optimality gap performance of (F4)

Figure 2: Performance profiles of the attained optimality gaps for (F1)-(F4) and different settings

at the attained gaps, the situation is similar; while the branch-and-cut based on the (F4) model is capable of attaining optimality gaps below 2.5% for all instances with the fhp configuration, the branch-and-cut corresponding to (F1) computes such gaps only for about 25% of the instances with the fhp configuration.

Secondly, the different algorithmic enhancements do improve the overall performance of the corresponding branch-and-cut algorithm. In other words, by going from b to fhp we get shorter runtimes and smaller optimality gaps. Moreover, the impact of such enhancements, seems to be stronger for the (F3) and (F4) formulations than for the (F1) and (F2)Ȧdditionally, when focusing on the plots corresponding to the (F3) and (F4) models, it seems that incorporating the separation on fractional solutions brings the best marginal improvement (i.e., b with respect to f), specially in terms of the attained gaps. However, in both cases, the results show that the approach combining all features, fhp, presents the best performance (in terms of runtimes and attained gaps). Due to the dominance of the (F4) formulation with the fhp configuration, the results presented in remainder of this section correspond to this setting.

4.3 Further details on algorithmic performance and solutions nature

In Tables 1-6, we report detailed results for each instance set using formulation (F4) with configuration fhp; each instance set is induced by a particular setting of r, R and θ (as explained in Section 4.1). In these Tables, we report for each instance (identified by the number in column "id") the number of nodes (column "$|V|$", recall that in these instances we have $I = J = V$), the number of $f_{..} = 1$ (column "$\#C1$"), the number of fractional $f_{..} > 0$ (column "$\#CP$"), the runtime in seconds (column "$t[s]$", with TL indicating that the timelimit of 600 seconds was reached), the upper bound (column "UB"), the value of the best solution found (column "z^*", in bold when it corresponds to the optimum value), the optimality gap (column "$g[\%]$"), the number of nodes in the branch-and-bound tree (column "$\#BBn$"), the time spent at the root node (column "$t_r[s]$"), the upper bound at the root node (column "UB_r"), the optimality gap at the root node (column "$g_r[\%]$", calculated as $100 \cdot (UB_r - z^*)/(z^*)$), the runtime of the starting heuristic (column "$t_H[s]$") the value of the starting heuristic solution (column "z_H", in bold when it coincides to the optimum value), the primal gap between this solution and the best solution found (column "$g_H[\%]$", calculated as $(z_H - z^*)/(z_H)$), the number of locations with more than one opened facility in the best found solution (column "$\#CL$"), and the maximum number of facilities opened at a single location in the best found solution (column "mCL").

From the results reported in this table, we can observe that the computational difficulty is strongly influenced by the value of $|V|$ (i.e., size of the problem), instances with $|V| \geq 500$ can rarely be solved to optimality (as can be seen from column "$t[s]$" and "$g[\%]$"). Moreover, for the case of the instance set pm-5-20-0.2, one of the instances with $|V| = 900$ could not be solved due to memory limit issues. Additionally, it is interesting to point out that, for a given value of $|V|$, increasing the value of K has a clear effect on the effectiveness of the algorithm. On the one hand, for the smaller instances ($|V| \leq 400$), increasing K results in an increase of the runtimes (which is particularly clear for $|V| = 400$); on the other hand, for larger instances (which typically reach the timelimit), increasing the value of K results in an improvement of the attained optimality gaps (as can be seen from column "$g[\%]$"). Along the same line, the number of explored branch-and-bound nodes (shown in column "$\#BBn$"), also presents an interesting behavior. For small instances ($|V| \leq 200$), very few nodes are explored, most likely because the initialization and the separation allow to compute very tight dual bounds at a very early stage of the optimization process. Likewise, for larger instances ($|V| \geq 700$), also very few nodes are explored; however, in these cases, it is due to the large size of the induced linear programming models which results in a more time consuming separation process. Such behavior is verified by the by longer runtimes required to process the root node (column "$t_r[s]$") and the poorer quality of the root-node solutions (which can be seen from columns "UB_r" and "$g_r[\%]$"). Therefore, it is only for intermediate size instances ($300 \leq |V| \leq 600$), that more branch-and-bound nodes are explored; this is due to the moderate size of the resulting linear programming models and a less (computationally) expensive cut separation.

From columns "$t_H[s]$", "z^H" and "$g_H[\%]$", we can clearly observe that the implemented starting heuristic is capable of computing remarkably good solutions for the six groups of instances. Moreover, on the contrary to the measures discussed in the previous paragraph, the performance of the starting heuristic, specially the quality of the computed solutions (measured by the values reported in column "$g_H[\%]$") seems to be insensitive to the values of $|V|$ and K . Recall that the starting heuristic builds upon one of the approaches outlined in Berman et al. [2018], which is based on the greedy approximation algorithm of [Nemhauser et al., 1978] for submodular optimization problems.

In Berman et al. [2018], the authors test their approach only on instances from set `pm-5-20-0.2`. When comparing their results with those attained by our approach, and reported in Table 1, we observe the following facts: (i) while in Berman et al. [2018], optimality is proven for 5 instances, we manage to prove it for 18 instances; (ii) we improve the solution value obtained by Berman et al. [2018] for 7 additional instances; and, (iii) our primal-dual nature of our approach allows to always account with a certificate of quality of the attained solution.

Analyzing the solution-characteristics We will now analyze and discuss key characteristics of the computed solutions, and how they are influenced by the instances structure, which is given by r, R and θ, and $|V|$ and K.

Tables 1, 2 and 3 report the solutions of instance sets `pm-5-20-0.2`, `pm-5-20-0.5` and `pm-5-20-0.8`, respectively; hence, in thee instances we have that $R = 20$, $r = 5$, and their only difference is the value of θ (0.2, 0.5 and 0.8 respectively). From columns z^*, we can clearly see that the computed solutions in these three sets have very similar objective values. The deployment of the co-locations deploy on the attained solutions is also very similar; the values reported in columns $\#CL$ and mCL, let us conclude that different values of θ do not necessarily influence on the need of co-locating facilities at different number locations (columns $\#CL$), nor on the number of co-located facilities (columns mCL). In a sense, this result is counterintuitive; due to the definition of JCF, one would expect that by increasing the value of θ, we would reduce the number of location where more than one facility is opened (and, complementary, the number of opened facilities in such locations).

On the contrary to the above described behavior, for the instance sets `pm-10-25-0.2`, `pm-10-25-0.5` and `pm-10-25-0.8` (which are given by $R = 25$ and $r = 10$), it is possible observe a moderate influence of the value of θ, as can be seen from Tables 4, 5 and 6, respectively. As expected, smaller values of θ ($\theta = 0.2$) lead to solutions with more locations hosting multiple facilities and more facilities located at those locations (see columns $\#CL$ and mCL), when compared to having greater values of θ ($\theta = 0.8$). Furthermore, the influence of θ can be also observed when comparing the attained objective function values (columns "z^*"); as expected, greater values of θ lead to less expensive solutions, as co-location and joint coverage are less emphasized (as when having smaller values of θ).

The difference in the behavior of the solutions of instance sets `pm-5-20-0.2`, `pm-5-20-0.5` and `pm-5-20-0.8`, with respect to those of `pm-10-25-0.2`, `pm-10-25-0.5` and `pm-10-25-0.8`, is likely to be explained by the different values of r and R. While for the first group of instances R is four times larger than r, for the second, R is only 2.5 times larger than r. Additionally, for the first group, the value of r is half the value of r for second group. Thus, for the first group of instances, there is a smaller number of facility-customer-pairs which would result in full coverage (see $\#C1$). For example, in the instance number 40, there are 3552 such candidates when $r = 5$, while there are 14636 for $r = 10$. Moreover, also the combinations providing partial coverage are most of the time more numerous in the second group of instances. Thus, the first group of instances gives less choices and the problem resembles a little more a classical maximum coverage problem, while the second group with more available connections potentially allows more exploitation of the benefits of partial coverage which are enhanced by the possibility of having more than one facility located at a given location.

Table 1: Detailed results for instance set `pm-5-20-0.2`.

id	$\lvert V \rvert$	K	$\#C1$	$\#CP$	$t[s]$	UB	z^*	$g[\%]$	$\#BBn$	$t_r[s]$	UB_r	$g_r[\%]$	$t_H[s]$	z^H	$g_H[\%]$	$\#CL$	mCL
1	100	5	114	64	0.01	14.60000	**14.60000**	0.000	0	0.01	14.60000	0.000	0.00	**14.60000**	0.000	0	1
2	100	10	124	76	0.02	26.79200	**26.79200**	0.000	0	0.02	26.79200	0.000	0.00	**26.79200**	0.000	0	1
3	100	10	116	100	0.07	25.65333	**25.65333**	0.000	0	0.07	25.65333	0.000	0.00	**25.65333**	0.000	0	1
4	100	20	106	56	0.05	35.43200	**35.43200**	0.000	0	0.05	35.43200	0.000	0.00	**35.43200**	0.000	0	1
5	100	33	130	114	0.14	62.21778	**62.21778**	0.000	1	0.14	62.21778	0.000	0.00	**62.21778**	0.000	0	1
6	200	5	296	430	0.02	30.13333	**30.13333**	0.000	0	0.02	30.13333	0.000	0.00	**30.13333**	0.000	0	1
7	200	10	292	592	0.26	50.48124	**50.48124**	0.000	4	0.25	50.56167	0.159	0.01	**50.48124**	0.000	0	1
8	200	20	278	418	0.13	69.79514	**69.79514**	0.000	0	0.13	69.79514	0.000	0.00	**69.79514**	0.000	0	1
9	200	40	302	502	1.37	118.10412	**118.10412**	0.000	36	1.00	118.35876	0.216	0.02	117.95745	0.124	3	2
10	200	67	328	900	58.07	158.93399	**158.93399**	0.000	1246	6.02	160.91652	1.247	0.09	**158.93399**	0.000	4	3
11	300	5	534	2358	0.61	62.52753	**62.52753**	0.000	10	0.44	63.20786	1.088	0.01	**62.52753**	0.000	0	1
12	300	10	474	1654	0.37	80.00622	**80.00622**	0.000	5	0.33	80.11857	0.140	0.00	**80.00622**	0.000	0	1
13	300	30	500	2004	29.07	154.42877	**154.42877**	0.000	475	4.08	157.06122	1.705	0.06	154.42506	0.002	0	1
14	300	60	502	1586	**TL**	207.97964	207.54577	0.209	4575	12.74	209.92442	1.146	0.15	206.64859	0.434	0	1
15	300	100	518	2000	**TL**	252.19124	250.45467	0.693	1327	28.12	253.87769	1.367	0.33	250.32876	0.050	2	2
16	400	5	874	6918	2.53	108.96806	**108.96806**	0.000	26	1.16	111.27877	2.121	0.01	108.03455	0.864	0	1
17	400	10	840	5902	4.37	147.92027	**147.92027**	0.000	55	1.89	150.69428	1.875	0.02	**147.92027**	0.000	0	1
18	400	40	738	4070	**TL**	243.01593	238.04555	2.088	1307	24.92	246.37574	3.499	0.10	237.38353	0.279	1	2
19	400	80	816	4932	**TL**	329.00121	323.15514	1.809	170	101.76	329.23856	1.883	0.94	323.07301	0.025	2	2
20	400	133	760	4170	**TL**	375.11226	367.57563	2.050	156	147.06	375.25593	2.089	1.50	366.89330	0.186	3	2
21	500	5	1186	12832	2.26	149.16160	**149.16160**	0.000	13	1.53	150.93946	1.192	0.01	**149.16160**	0.000	0	1
22	500	10	1090	9960	38.68	178.49851	**178.49851**	0.000	278	4.35	184.47554	3.349	0.03	176.32300	1.234	0	1
23	500	50	1196	11222	**TL**	377.38754	363.88111	3.712	157	83.84	377.80670	3.827	0.75	363.65744	0.062	3	2
24	500	100	1174	11328	**TL**	451.08139	439.76165	2.574	21	303.16	451.40047	2.647	1.79	439.13596	0.142	2	2
25	500	167	1258	13962	**TL**	489.27113	482.60720	1.381	8	469.44	489.36607	1.400	5.67	482.41662	0.040	4	2
26	600	5	1706	30160	76.97	221.74392	**221.74392**	0.000	167	6.78	234.29056	5.658	0.04	**221.74392**	0.000	0	1
27	600	10	1764	32768	**TL**	323.17551	305.54231	5.771	144	34.65	327.64050	7.232	0.09	305.49935	0.014	1	2
28	600	60	1716	33584	**TL**	519.85051	503.69890	3.207	3	499.19	520.03866	3.244	2.68	503.69889	0.000	1	2
29	600	120	1714	28968	**TL**	575.12023	560.94685	2.527	0	604.38	575.12023	2.527	4.96	560.88227	0.012	5	2
30	600	200	1570	21010	**TL**	593.24141	590.50481	0.463	0	604.15	593.24141	0.463	12.59	590.27327	0.039	6	2
31	700	5	2542	64420	**TL**	331.11567	314.32281	5.343	221	14.60	342.73355	9.039	0.04	314.32281	0.000	0	1
32	700	10	2218	50474	**TL**	397.36246	374.87069	6.000	76	32.41	398.47059	6.295	0.13	374.87069	0.000	0	1
33	700	70	2268	49250	**TL**	640.18003	621.39845	3.022	0	607.97	640.18003	3.022	3.40	620.85758	0.087	2	2
34	700	140	2450	58282	**TL**	686.32210	679.62662	0.985	0	627.53	686.32210	0.985	4.92	679.45459	0.025	1	2
35	800	5	3424	143422	**TL**	498.76943	460.69826	8.264	22	72.53	500.36055	8.609	0.05	460.69826	0.000	0	1
36	800	10	3052	91980	**TL**	517.48254	479.24997	7.978	25	68.11	517.94425	8.074	0.19	479.24997	0.000	0	1
37	800	80	2762	71842	**TL**	745.44345	727.12707	2.519	0	604.20	745.44345	2.519	4.68	726.84905	0.038	0	1
38	900	5	4520	224438	**TL**	619.22839	563.03405	9.981	9	108.97	620.35317	10.180	0.13	561.05866	0.352	0	1
39	900	10	4450	227826	**TL**	724.97571	682.84189	6.170	7	204.66	725.47869	6.244	0.20	682.84189	0.000	0	1
40	900	80	*	*	*not solution available due to memory limit*												

Table 2: Detailed results for instance set `pm-5-20-0.5`.

id	$\lvert V \rvert$	K	$\#C1$	$\#CP$	$t[s]$	UB	z^*	$g[\%]$	$\#BBn$	$t_r[s]$	UB_r	$g_r[\%]$	$t_H[s]$	z^H	$g_H[\%]$	$\#CL$	mCL
1	100	5	114	64	0.01	14.60000	**14.60000**	0.000	0	0.01	14.60000	0.000	0.00	**14.60000**	0.000	0	1
2	100	10	124	76	0.02	26.72000	**26.72000**	0.000	0	0.02	26.72000	0.000	0.00	**26.72000**	0.000	0	1
3	100	10	116	100	0.04	25.63333	**25.63333**	0.000	0	0.04	25.63333	0.000	0.00	**25.63333**	0.000	0	1
4	100	20	106	56	0.05	35.42000	**35.42000**	0.000	0	0.05	35.42000	0.000	0.00	**35.42000**	0.000	0	1
5	100	33	130	114	0.13	62.11111	**62.11111**	0.000	1	0.13	62.11111	0.000	0.00	**62.11111**	0.000	0	1
6	200	5	296	430	0.02	30.13333	**30.13333**	0.000	0	0.02	30.13333	0.000	0.00	**30.13333**	0.000	0	1
7	200	10	292	592	0.22	50.32578	**50.32578**	0.000	0	0.22	50.32578	0.000	0.00	**50.32578**	0.000	0	1
8	200	20	278	418	0.14	69.66030	**69.66030**	0.000	0	0.14	69.66030	0.000	0.01	**69.66030**	0.000	0	1
9	200	40	302	502	0.63	117.59007	**117.59007**	0.000	20	0.54	117.65543	0.056	0.04	**117.59007**	0.000	3	2
10	200	67	328	900	2.44	157.89400	**157.89400**	0.000	19	1.80	158.13614	0.153	0.07	157.77459	0.076	3	3
11	300	5	534	2358	0.35	62.07970	**62.07970**	0.000	1	0.35	62.07970	0.000	0.00	**62.07970**	0.000	0	1
12	300	10	474	1654	0.34	79.62889	**79.62889**	0.000	0	0.34	79.62889	0.000	0.00	**79.62889**	0.000	0	1
13	300	30	500	2004	3.57	152.75187	**152.75187**	0.000	45	2.15	153.45741	0.462	0.05	152.72967	0.015	1	2
14	300	60	502	1586	12.33	205.58821	**205.58821**	0.000	143	5.23	206.21113	0.303	0.13	204.76453	0.402	0	1
15	300	100	518	2000	99.1	247.41430	**247.41430**	0.000	1160	11.22	248.39762	0.397	0.19	246.78161	0.256	2	2
16	400	5	874	6918	1.24	107.40504	**107.40504**	0.000	8	0.98	108.24489	0.782	0.01	106.29763	1.042	0	1
17	400	10	840	5902	2.12	145.77517	**145.77517**	0.000	14	1.48	146.76176	0.677	0.01	**145.77517**	0.000	0	1
18	400	40	738	4070	**TL**	234.84788	233.81358	0.442	2675	14.13	237.47458	1.566	0.17	233.32576	0.209	0	1
19	400	80	816	4932	**TL**	318.65316	317.62369	0.324	848	49.02	319.45887	0.578	0.53	317.23561	0.122	6	2
20	400	133	760	4170	**TL**	364.64616	362.48410	0.596	250	61.27	365.12343	0.728	1.84	361.74052	0.206	4	2
21	500	5	1186	12832	1.17	146.17600	**146.17600**	0.000	0	1.17	146.17600	0.000	0.02	**146.17600**	0.000	0	1
22	500	10	1090	9960	14.48	174.86157	**174.86157**	0.000	92	3.64	177.70824	1.628	0.04	172.41694	1.418	0	1
23	500	50	1196	11222	**TL**	360.21888	354.87291	1.506	306	64.50	361.15181	1.769	0.45	354.03810	0.236	0	1
24	500	100	1174	11328	**TL**	434.69475	430.74256	0.918	142	100.63	435.07190	1.005	1.23	430.71258	0.007	3	2
25	500	167	1258	13962	**TL**	478.22622	476.03060	0.461	55	242.32	478.31624	0.480	5.67	476.03059	0.000	3	2
26	600	5	1706	30160	26.56	214.31495	**214.31495**	0.000	70	5.54	221.24374	3.233	0.06	**214.31495**	0.000	0	1
27	600	10	1764	32768	**TL**	299.53133	290.93944	2.953	181	20.42	304.76118	4.751	0.05	290.34134	0.206	0	1
28	600	60	1716	33584	**TL**	495.15376	487.18178	1.636	22	229.52	495.38495	1.684	1.70	486.87124	0.064	0	1
29	600	120	1714	28968	**TL**	553.70093	547.87855	1.063	19	297.53	553.83774	1.088	2.77	547.72427	0.028	3	3
30	600	200	1570	21010	**TL**	584.76703	583.39450	0.235	3	539.43	584.81751	0.244	8.68	583.26436	0.022	4	2
31	700	5	2542	64420	560.85	302.45176	**302.45176**	0.000	434	16.12	319.85636	5.755	0.06	**302.45176**	0.000	0	1
32	700	10	2218	50474	**TL**	367.70970	357.83233	2.760	202	27.44	371.81872	3.909	0.09	357.83233	0.000	1	2
33	700	70	2268	49250	**TL**	609.70806	600.68651	1.502	6	421.93	609.81106	1.519	1.75	600.51548	0.028	1	2
34	700	140	2450	58282	**TL**	668.80616	665.46411	0.502	0	612.04	668.80616	0.502	10.67	665.35981	0.016	6	2
35	800	5	3424	143422	**TL**	457.12331	432.06141	5.801	32	35.08	457.97255	5.997	0.05	431.86545	0.045	0	1
36	800	10	3052	91980	**TL**	478.35183	453.18283	5.554	48	43.35	478.79706	5.652	0.22	453.18283	0.000	0	1
37	800	80	2762	71842	**TL**	712.12631	704.15161	1.133	1	583.37	712.12631	1.133	6.85	704.08931	0.009	1	2
38	900	5	4520	224438	**TL**	562.74327	526.27128	6.930	18	136.65	563.82824	7.136	0.13	526.27128	0.000	0	1
39	900	10	4450	227826	**TL**	663.37077	637.70118	4.025	8	185.74	664.02808	4.128	0.19	637.70118	0.000	0	1
40	900	90	3552	140956	**TL**	833.23948	825.40900	0.949	0	605.08	833.23948	0.949	4.00	824.80465	0.073	2	2

Table 3: Detailed results for instance set `pm-5-20-0.8`.

id	$\lvert V\rvert$	K	$\#C1$	$\#CP$	$t[s]$	UB	z^*	$g[\%]$	$\#BBn$	$t_r[s]$	UB_r	$g_r[\%]$	$t_H[s]$	z^H	$g_H[\%]$	$\#CL$	mCL
1	100	5	114	64	0	14.60000	**14.60000**	0.000	0	0.00	14.60000	0.000	0.00	**14.60000**	0.000	0	1
2	100	10	124	76	0.02	26.64800	**26.64800**	0.000	0	0.02	26.64800	0.000	0.00	**26.64800**	0.000	0	1
3	100	10	116	100	0.04	25.61333	**25.61333**	0.000	0	0.04	25.61333	0.000	0.00	**25.61333**	0.000	0	1
4	100	20	106	56	0.04	35.40800	**35.40800**	0.000	0	0.04	35.40800	0.000	0.00	**35.40800**	0.000	0	1
5	100	33	130	114	0.11	62.00444	**62.00444**	0.000	1	0.11	62.00444	0.000	0.00	**62.00444**	0.000	0	1
6	200	5	296	430	0.03	30.13333	**30.13333**	0.000	0	0.03	30.13333	0.000	0.01	**30.13333**	0.000	0	1
7	200	10	292	592	0.12	50.20587	**50.20587**	0.000	0	0.12	50.20587	0.000	0.00	**50.20587**	0.000	0	1
8	200	20	278	418	0.14	69.54412	**69.54412**	0.000	0	0.14	69.54412	0.000	0.01	**69.54412**	0.000	0	1
9	200	40	302	502	0.65	117.07603	**117.07603**	0.000	10	0.47	117.08809	0.010	0.03	**117.07603**	0.000	3	2
10	200	67	328	900	1.18	157.13121	**157.13121**	0.000	1	1.18	157.13121	0.000	0.07	157.11760	0.009	4	2
11	300	5	534	2358	0.15	61.63188	**61.63188**	0.000	0	0.15	61.63188	0.000	0.01	**61.63188**	0.000	0	1
12	300	10	474	1654	0.23	79.25156	**79.25156**	0.000	0	0.23	79.25156	0.000	0.01	**79.25156**	0.000	0	1
13	300	30	500	2004	1.56	151.10075	**151.10075**	0.000	5	1.49	151.16044	0.040	0.06	151.08654	0.009	1	2
14	300	60	502	1586	3.59	203.81306	**203.81306**	0.000	20	2.95	203.87750	0.032	0.21	203.15461	0.324	1	2
15	300	100	518	2000	13.02	244.77680	**244.77680**	0.000	97	6.01	244.96591	0.077	0.18	244.12746	0.266	3	3
16	400	5	874	6918	0.55	105.84202	**105.84202**	0.000	0	0.55	105.84202	0.000	0.02	**105.84202**	0.000	0	1
17	400	10	840	5902	0.9	143.63007	**143.63007**	0.000	0	0.90	143.63007	0.000	0.02	**143.63007**	0.000	0	1
18	400	40	738	4070	20.18	230.27463	**230.27463**	0.000	114	10.18	230.82854	0.241	0.26	229.51506	0.331	1	2
19	400	80	816	4932	42.98	312.67871	**312.67871**	0.000	107	24.40	312.92945	0.080	1.17	312.42754	0.080	2	2
20	400	133	760	4170	98.38	357.99786	**357.99786**	0.000	252	38.16	358.28921	0.081	2.15	357.35165	0.181	1	2
21	500	5	1186	12832	0.79	143.19040	**143.19040**	0.000	0	0.79	143.19040	0.000	0.03	**143.19040**	0.000	0	1
22	500	10	1090	9960	5.15	171.22463	**171.22463**	0.000	18	3.78	171.74588	0.304	0.06	168.88678	1.384	0	1
23	500	50	1196	11222	119.93	346.94339	**346.94339**	0.000	186	36.89	347.94152	0.288	0.65	346.04908	0.258	0	1
24	500	100	1174	11328	**TL**	422.49787	422.36762	0.031	648	60.20	423.10762	0.175	1.66	421.74503	0.148	2	3
25	500	167	1258	13962	**TL**	470.51314	470.46275	0.011	400	166.26	470.69439	0.049	8.47	470.14281	0.068	2	2
26	600	5	1706	30160	8.35	206.88598	**206.88598**	0.000	15	4.69	209.12648	1.083	0.05	**206.88598**	0.000	0	1
27	600	10	1764	32768	195.75	278.78555	**278.78555**	0.000	305	15.16	284.42413	2.023	0.10	278.73863	0.017	0	1
28	600	60	1716	33584	**TL**	474.72976	472.49510	0.473	157	134.12	475.10421	0.552	1.56	472.25887	0.050	1	2
29	600	120	1714	28968	**TL**	538.06209	536.95262	0.207	83	231.42	538.29481	0.250	3.30	536.77142	0.034	1	2
30	600	200	1570	21010	**TL**	577.59212	577.39683	0.034	76	303.27	577.68397	0.050	7.36	576.88482	0.089	1	2
31	700	5	2542	64420	69.28	290.58070	**290.58070**	0.000	76	10.41	298.01495	2.558	0.04	**290.58070**	0.000	0	1
32	700	10	2218	50474	215.16	340.81293	**340.81293**	0.000	249	18.60	346.15909	1.569	0.09	**340.81293**	0.000	1	2
33	700	70	2268	49250	**TL**	584.36914	581.92421	0.420	60	208.39	584.68282	0.474	2.18	581.26311	0.114	3	2
34	700	140	2450	58282	**TL**	653.81786	653.34890	0.072	14	457.78	653.91650	0.087	8.14	652.87986	0.072	2	2
35	800	5	3424	143422	467.3	403.50618	**403.50618**	0.000	113	32.17	416.18687	3.143	0.06	**403.50618**	0.000	0	1
36	800	10	3052	91980	**TL**	437.67703	427.42371	2.399	101	34.33	440.39391	3.035	0.16	425.81790	0.377	0	1
37	800	80	2762	71842	**TL**	685.88115	683.34521	0.371	8	421.27	685.97471	0.385	3.00	683.24213	0.015	2	2
38	900	5	4520	224438	**TL**	508.25104	490.73630	3.569	25	62.31	509.28211	3.779	0.25	490.73630	0.000	0	1
39	900	10	4450	227826	**TL**	604.44502	592.56047	2.006	25	98.40	605.03571	2.105	0.21	592.56047	0.000	0	1
40	900	90	3552	140956	**TL**	802.88504	800.98919	0.237	1	608.49	802.88504	0.237	4.90	800.76111	0.028	1	2

Table 4: Detailed results for instance set `pm-10-25-0.2`.

id	$\lvert V\rvert$	K	$\#C1$	$\#CP$	$t[s]$	UB	z^*	$g[\%]$	$\#BBn$	$t_r[s]$	UB_r	$g_r[\%]$	$t_H[s]$	z^H	$g_H[\%]$	$\#CL$	mCL
1	100	5	138	60	0.01	17.53333	**17.53333**	0.000	0	0.01	17.53333	0.000	0.00	**17.53333**	0.000	0	1
2	100	10	152	78	0.04	31.79597	**31.79597**	0.000	0	0.04	31.79597	0.000	0.00	31.63828	0.498	1	2
3	100	10	136	110	0.04	32.00000	**32.00000**	0.000	0	0.04	32.00000	0.000	0.00	**32.00000**	0.000	0	1
4	100	20	126	66	0.04	43.55556	**43.55556**	0.000	0	0.04	43.55556	0.000	0.00	**43.55556**	0.000	0	1
5	100	33	174	116	0.18	70.43111	**70.43111**	0.000	0	0.18	70.43111	0.000	0.01	**70.43111**	0.000	1	2
6	200	5	408	628	0.1	41.22133	**41.22133**	0.000	0	0.10	41.22133	0.000	0.00	**41.22133**	0.000	0	1
7	200	10	414	906	0.35	67.98513	**67.98513**	0.000	2	0.33	68.01286	0.041	0.00	**67.98513**	0.000	0	1
8	200	20	360	658	0.26	93.38347	**93.38347**	0.000	0	0.26	93.38347	0.000	0.01	**93.38347**	0.000	0	1
9	200	40	418	686	4.27	140.75464	**140.75464**	0.000	118	1.89	141.30842	0.393	0.06	140.73299	0.015	1	2
10	200	67	526	1372	98.39	184.06172	**184.06172**	0.000	1378	8.88	185.75057	0.918	0.08	183.91174	0.082	3	2
11	300	5	1004	3968	0.62	96.19941	**96.19941**	0.000	5	0.59	96.47004	0.281	0.01	**96.19941**	0.000	0	1
12	300	10	814	2756	1.03	121.76729	**121.76729**	0.000	15	0.82	122.23208	0.382	0.01	**121.76729**	0.000	0	1
13	300	30	902	3342	48.08	200.24834	**200.24834**	0.000	474	9.66	202.68305	1.216	0.07	200.24169	0.003	0	1
14	300	60	804	2722	**TL**	247.77258	247.41956	0.143	2809	27.04	249.51119	0.845	0.27	247.19514	0.091	0	1
15	300	100	910	3470	**TL**	289.10723	285.98499	1.092	573	46.68	289.89205	1.366	0.45	285.62724	0.125	4	2
16	400	5	1974	13784	6.31	181.42063	**181.42063**	0.000	32	2.03	186.76530	2.946	0.02	**181.42063**	0.000	0	1
17	400	10	1738	11266	197.47	220.59393	**220.59393**	0.000	1298	6.95	227.75208	3.245	0.06	219.60947	0.448	0	1
18	400	40	1430	7758	**TL**	314.32964	308.97821	1.732	564	47.49	316.39468	2.400	0.24	308.83571	0.046	2	2
19	400	80	1644	9618	**TL**	379.62734	375.64126	1.061	182	106.09	380.00260	1.161	0.72	375.55034	0.024	3	2
20	400	133	1462	8014	**TL**	397.89812	397.15671	0.187	150	147.13	398.04788	0.224	2.69	397.08061	0.019	5	2
21	500	5	2918	27626	13.56	252.42390	**252.42390**	0.000	38	4.25	258.07630	2.239	0.03	**252.42390**	0.000	0	1
22	500	10	2496	21494	**TL**	289.36879	284.23641	1.806	606	10.15	296.97951	4.483	0.13	284.23641	0.000	0	1
23	500	50	2772	24348	**TL**	453.67706	443.81430	2.222	53	154.77	453.92340	2.278	0.54	443.60046	0.048	1	2
24	500	100	2810	23810	**TL**	494.49860	490.89659	0.734	33	208.13	494.53532	0.741	1.65	490.86898	0.006	3	3
25	500	167	3078	29866	**TL**	499.72439	499.69513	0.006	100	327.18	499.73327	0.008	7.88	499.64449	0.010	4	3
26	600	5	5086	70238	**TL**	389.77965	370.60402	5.174	108	22.70	393.75697	6.247	0.03	370.60402	0.000	0	1
27	600	10	5420	75078	**TL**	466.45326	446.98031	4.357	42	70.18	467.18071	4.519	0.10	445.93657	0.234	0	1
28	600	60	5338	78458	**TL**	583.44767	574.68799	1.524	3	459.65	583.61116	1.553	1.67	574.18047	0.088	3	2
29	600	120	4982	67654	**TL**	597.90520	597.17802	0.122	8	416.18	597.92809	0.126	4.69	596.95331	0.038	2	2
30	600	200	4284	48514	28.02	600.00000	**600.00000**	0.000	0	28.02	600.00000	0.000	3.74	**600.00000**	0.000	7	7
31	700	5	8624	152102	**TL**	543.71703	510.57203	6.492	37	68.55	545.00485	6.744	0.06	508.89662	0.329	0	1
32	700	10	7294	123992	**TL**	577.08107	553.66841	4.229	24	107.81	577.50451	4.305	0.24	552.92943	0.134	1	2
33	700	70	7110	121930	**TL**	692.47546	687.95411	0.657	2	541.67	692.55299	0.668	3.29	687.75055	0.030	1	2
34	700	140	8044	141988	**TL**	700.00000	699.79615	0.029	0	603.37	700.00000	0.029	10.28	699.77804	0.003	2	2
35	800	5	14838	312066	**TL**	697.75670	668.81970	4.327	7	133.40	698.54679	4.445	0.08	668.81970	0.000	0	1
36	800	10	10904	224412	**TL**	703.35094	677.49941	3.816	11	204.02	704.00226	3.912	0.20	677.25554	0.036	0	1
37	800	80	9072	186296	**TL**	794.07320	790.62270	0.436	0	606.86	794.07320	0.436	8.17	790.62265	0.000	0	1
38	900	5	21526	466312	**TL**	820.78375	788.21215	4.132	4	223.07	821.39778	4.210	0.20	788.21215	0.000	0	1
39	900	10	21248	471774	**TL**	856.26718	841.15019	1.797	3	283.25	856.53510	1.829	0.39	841.15012	0.000	0	1
40	900	90	14636	345298	**TL**	898.40506	897.00137	0.156	0	610.10	898.40506	0.156	5.69	896.88462	0.013	2	2

Table 5: Detailed results for instance set `pm-10-25-0.5`.

id	$\lvert V \rvert$	K	$\#C1$	$\#CP$	$t[s]$	UB	z^*	$g[\%]$	$\#BBn$	$t_r[s]$	UB_r	$g_r[\%]$	$t_H[s]$	z^H	$g_H[\%]$	$\#CL$	mCL
1	100	5	138	60	0.01	17.53333	**17.53333**	0.000	0	0.01	17.53333	0.000	0.00	**17.53333**	0.000	0	1
2	100	10	152	78	0.03	31.69748	**31.69748**	0.000	0	0.03	31.69748	0.000	0.00	31.57393	0.391	1	2
3	100	10	136	110	0.03	32.00000	**32.00000**	0.000	0	0.03	32.00000	0.000	0.00	**32.00000**	0.000	0	1
4	100	20	126	66	0.04	43.52222	**43.52222**	0.000	0	0.04	43.52222	0.000	0.00	**43.52222**	0.000	0	1
5	100	33	174	116	0.11	70.34444	**70.34444**	0.000	0	0.11	70.34444	0.000	0.01	**70.34444**	0.000	1	2
6	200	5	408	628	0.1	41.11333	**41.11333**	0.000	0	0.10	41.11333	0.000	0.00	**41.11333**	0.000	0	1
7	200	10	414	906	0.26	67.61570	**67.61570**	0.000	0	0.26	67.61570	0.000	0.01	67.44711	0.250	0	1
8	200	20	360	658	0.2	93.11467	**93.11467**	0.000	0	0.20	93.11467	0.000	0.01	**93.11467**	0.000	0	1
9	200	40	418	686	1.1	140.16165	**140.16165**	0.000	18	0.82	140.30235	0.100	0.03	139.76427	0.284	1	2
10	200	67	526	1372	7.34	182.75316	**182.75316**	0.000	110	3.40	183.20081	0.245	0.07	182.57485	0.098	5	2
11	300	5	1004	3968	0.36	95.44963	**95.44963**	0.000	3	0.35	95.53896	0.094	0.01	**95.44963**	0.000	0	1
12	300	10	814	2756	0.84	120.57956	**120.57956**	0.000	6	0.78	120.60948	0.025	0.02	120.45310	0.105	0	1
13	300	30	902	3342	6.45	198.40521	**198.40521**	0.000	32	4.41	199.15613	0.378	0.05	198.08131	0.164	0	1
14	300	60	804	2722	20.1	245.20222	**245.20222**	0.000	168	9.37	245.90362	0.286	0.26	244.24898	0.390	0	1
15	300	100	910	3470	**TL**	283.28413	283.16395	0.042	2353	17.30	284.29512	0.399	0.34	282.57714	0.208	4	2
16	400	5	1974	13784	4	177.56289	**177.56289**	0.000	19	2.09	180.59247	1.706	0.02	**177.56289**	0.000	0	1
17	400	10	1738	11266	26.74	216.97120	**216.97120**	0.000	192	6.02	220.71472	1.725	0.04	**216.97120**	0.000	0	1
18	400	40	1430	7758	**TL**	304.79825	304.49173	0.101	1755	32.50	307.60140	1.021	0.25	303.77542	0.236	3	2
19	400	80	1644	9618	**TL**	372.06546	371.85967	0.055	1366	45.23	372.97701	0.300	1.09	371.23688	0.168	3	2
20	400	133	1462	8014	246.9	395.68115	**395.68115**	0.000	341	91.60	395.88009	0.050	2.56	395.52344	0.040	4	2
21	500	5	2918	27626	5.4	246.63994	**246.63994**	0.000	14	3.17	248.88136	0.909	0.02	**246.63994**	0.000	0	1
22	500	10	2496	21494	186.94	277.37275	**277.37275**	0.000	500	9.22	285.04695	2.767	0.18	**277.37275**	0.000	0	1
23	500	50	2772	24348	**TL**	441.46235	436.65058	1.102	189	81.81	441.95910	1.216	0.60	436.48554	0.038	1	2
24	500	100	2810	23810	**TL**	488.30956	486.76514	0.317	114	127.53	488.41310	0.339	1.74	486.27600	0.101	4	2
25	500	167	3078	29866	**TL**	499.37030	499.36950	0.000	369	334.94	499.37952	0.002	4.58	499.24489	0.025	2	2
26	600	5	5086	70238	**TL**	361.65548	356.70251	1.389	181	23.46	371.64846	4.190	0.03	356.70251	0.000	0	1
27	600	10	5420	75078	**TL**	443.23151	431.38015	2.747	90	43.41	444.48826	3.039	0.10	431.38015	0.000	0	1
28	600	60	5338	78458	**TL**	569.99048	565.97899	0.709	27	281.00	570.11752	0.731	2.08	565.86250	0.021	1	2
29	600	120	4982	67654	**TL**	595.09777	594.54609	0.093	26	343.76	595.15516	0.102	3.74	594.08655	0.077	0	1
30	600	200	4284	48514	24.62	600.00000	**600.00000**	0.000	0	24.62	600.00000	0.000	1.96	**600.00000**	0.000	3	3
31	700	5	8624	152102	**TL**	512.46549	490.90752	4.391	60	34.01	513.49092	4.600	0.04	489.27721	0.333	0	1
32	700	10	7294	123992	**TL**	547.60250	533.11776	2.717	46	61.53	547.86623	2.766	0.27	533.11776	0.000	1	2
33	700	70	7110	121930	**TL**	682.62810	679.97642	0.390	9	417.71	682.75339	0.408	3.44	679.74519	0.034	1	2
34	700	140	8044	141988	**TL**	699.59917	699.59490	0.001	0	624.71	699.59917	0.001	7.37	699.45761	0.020	1	2
35	800	5	14838	312066	**TL**	660.25472	640.36231	3.106	11	85.58	660.87949	3.204	0.08	640.36231	0.000	0	1
36	800	10	10904	224412	**TL**	668.68489	649.23970	2.995	20	139.68	669.17449	3.070	0.25	649.22013	0.003	0	1
37	800	80	9072	186296	**TL**	785.75330	783.32037	0.311	0	614.14	785.75330	0.311	4.85	783.10236	0.028	3	2
38	900	5	21526	466312	**TL**	778.29020	755.52010	3.014	7	178.76	779.30087	3.148	0.10	755.52010	0.000	0	1
39	900	10	21248	471774	**TL**	825.87441	813.77446	1.487	5	277.99	826.24229	1.532	0.42	813.77440	0.000	0	1
40	900	90	14636	345298	**TL**	894.19899	893.13174	0.119	0	609.33	894.19899	0.119	5.87	892.75694	0.042	0	1

Table 6: Detailed results for instance set pm-10-25-0.8.

id	$\|V\|$	K	$\#C1$	$\#CP$	$t[s]$	UB	z^*	$g[\%]$	$\#BBn$	$t_r[s]$	UB_r	$g_r[\%]$	$t_H[s]$	z^H	$g_H[\%]$	$\#CL$	mCL
1	100	5	138	60	0.01	17.53333	**17.53333**	0.000	0	0.01	17.53333	0.000	0.00	**17.53333**	0.000	0	1
2	100	10	152	78	0.02	31.59899	**31.59899**	0.000	0	0.02	31.59899	0.000	0.00	31.50957	0.284	1	2
3	100	10	136	110	0.03	32.00000	**32.00000**	0.000	0	0.03	32.00000	0.000	0.00	**32.00000**	0.000	0	1
4	100	20	126	66	0.04	43.48889	**43.48889**	0.000	0	0.04	43.48889	0.000	0.00	**43.48889**	0.000	0	1
5	100	33	174	116	0.1	70.25778	**70.25778**	0.000	0	0.10	70.25778	0.000	0.00	**70.25778**	0.000	1	2
6	200	5	408	628	0.08	41.00533	**41.00533**	0.000	0	0.08	41.00533	0.000	0.00	**41.00533**	0.000	0	1
7	200	10	414	906	0.25	67.24628	**67.24628**	0.000	0	0.25	67.24628	0.000	0.01	67.13884	0.160	0	1
8	200	20	360	658	0.14	92.84587	**92.84587**	0.000	0	0.14	92.84587	0.000	0.00	**92.84587**	0.000	0	1
9	200	40	418	686	0.7	139.58466	**139.58466**	0.000	0	0.70	139.58466	0.000	0.04	139.26263	0.231	1	2
10	200	67	526	1372	1.77	182.01744	**182.01744**	0.000	1	1.77	182.01744	0.000	0.05	181.66281	0.195	3	2
11	300	5	1004	3968	0.26	94.69985	**94.69985**	0.000	0	0.26	94.69985	0.000	0.01	**94.69985**	0.000	0	1
12	300	10	814	2756	0.55	119.39182	**119.39182**	0.000	0	0.55	119.39182	0.000	0.02	**119.39182**	0.000	0	1
13	300	30	902	3342	3.01	196.56208	**196.56208**	0.000	3	2.99	196.56635	0.002	0.08	196.46042	0.052	0	1
14	300	60	804	2722	5.18	243.46959	**243.46959**	0.000	19	4.47	243.58466	0.047	0.20	242.57910	0.367	1	2
15	300	100	910	3470	9.65	281.15156	**281.15156**	0.000	6	9.10	281.16753	0.006	0.22	280.41637	0.262	4	2
16	400	5	1974	13784	1.88	173.70516	**173.70516**	0.000	5	1.79	173.93763	0.134	0.02	**173.70516**	0.000	0	1
17	400	10	1738	11266	5.93	213.34848	**213.34848**	0.000	21	4.02	214.36070	0.474	0.05	**213.34848**	0.000	0	1
18	400	40	1430	7758	18.48	300.60222	**300.60222**	0.000	25	15.00	300.85103	0.083	0.21	299.69177	0.304	1	2
19	400	80	1644	9618	32.75	368.98454	**368.98454**	0.000	40	26.47	369.05720	0.020	1.05	368.73928	0.067	3	2
20	400	133	1462	8014	54.25	394.49127	**394.49127**	0.000	25	48.42	394.51863	0.007	2.39	394.18127	0.079	3	2
21	500	5	2918	27626	2.14	240.85597	**240.85597**	0.000	1	2.14	240.85597	0.000	0.01	**240.85597**	0.000	0	1
22	500	10	2496	21494	23.61	270.50910	**270.50910**	0.000	52	8.54	273.78255	1.210	0.23	269.90640	0.223	0	1
23	500	50	2772	24348	**TL**	431.11067	430.59969	0.119	680	55.02	432.50301	0.442	0.86	429.34958	0.291	2	2
24	500	100	2810	23810	102.06	483.42587	**483.42587**	0.000	94	62.16	483.60746	0.038	1.20	483.31644	0.023	2	2
25	500	167	3078	29866	**TL**	499.06847	499.06780	0.000	480	293.78	499.07072	0.001	3.96	498.64504	0.085	5	2
26	600	5	5086	70238	111.75	343.15861	**343.15861**	0.000	94	12.16	350.42676	2.118	0.04	342.80100	0.104	0	1
27	600	10	5420	75078	**TL**	418.49400	417.26694	0.294	296	30.06	422.97740	1.369	0.09	416.29616	0.233	0	1
28	600	60	5338	78458	**TL**	560.06552	558.94592	0.200	139	161.65	560.47718	0.274	1.42	558.38867	0.100	3	2
29	600	120	4982	67654	**TL**	592.56902	592.41242	0.026	91	273.00	592.71083	0.050	3.75	591.87448	0.091	2	2
30	600	200	4284	48514	21.5	600.00000	**600.00000**	0.000	0	21.50	600.00000	0.000	2.53	**600.00000**	0.000	3	2
31	700	5	8624	152102	**TL**	475.09728	471.24301	0.818	111	33.75	482.82600	2.458	0.04	470.24484	0.212	0	1
32	700	10	7294	123992	**TL**	516.14680	512.62608	0.687	135	42.78	519.29985	1.302	0.32	512.62607	0.000	1	2
33	700	70	7110	121930	**TL**	674.32202	673.19323	0.168	20	372.46	674.42293	0.183	2.23	672.06564	0.168	3	2
34	700	140	8044	141988	**TL**	699.35981	699.35796	0.000	0	616.80	699.35981	0.000	6.50	699.19755	0.023	3	2
35	800	5	14838	312066	**TL**	623.45089	611.90493	1.887	30	69.20	624.34885	2.034	0.07	611.72016	0.030	0	1
36	800	10	10904	224412	**TL**	635.07708	621.43791	2.195	32	80.34	635.54000	2.269	0.19	620.29056	0.185	0	1
37	800	80	9072	186296	**TL**	778.47803	777.14445	0.172	0	609.69	778.47803	0.172	5.96	776.88145	0.034	2	2
38	900	5	21526	466312	**TL**	736.92875	723.50305	1.856	13	111.86	737.79255	1.975	0.13	723.50304	0.000	0	1
39	900	10	21248	471774	**TL**	796.72085	787.65539	1.151	12	159.43	797.01301	1.188	0.20	787.65527	0.000	0	1
40	900	90	14636	345298	**TL**	890.67134	889.55816	0.125	0	606.54	890.67134	0.125	6.92	889.11200	0.050	2	2

5. Conclusions and future work

In this paper we studied the recently introduced *multiple gradual cover location problem* (MGCLP, see [Berman et al., 2018]). The MGCLP addresses, simultaneously, two issues that have been identified as relevant in practical facility location applications: gradual coverage and potential co-location of facilities. We presented four different mixed integer programming formulations for the MGCLP, all of them exploiting the submodularity of the objective function. Furthermore, we designed and implemented a branch-and-cut framework based one these formulations. Our framework is further enhanced by additional cut separation strategies, starting and primal heuristics and initialization procedures.

From an algorithmic perspective, the computational results show that our approach allows to effectively address different sets of instances. We provide optimal solution values for 13 instances from literature, where the optimal solution was not known, and additionally provide improved solution values for seven instances. We also analyzed the dependence of the solution-structure on instance-characteristics. The reported results show that the MGCLP possesses a great capability for allowing decision makers to design their facility deployment strategies according to different coverage capacities or customer preferences. Interestingly, the results show that although multiple location of facilities indeed occurs (there are even cases where seven facilities are located at the same location), the magnitude of the co-location is rather small.

A wide class of strategical and tactical decisions in many operations research settings correspond to facility location deployment. Incorporating modeling features such as partial and joint coverage along with co-location of facilities, establishes a new path for closing the gap between academic optimization tools and real-world location problems. For future work, it would be interesting to study how different partial coverage functions can be included into the proposed models, or how the different degrees of uncertainty (e.g., location of customers) can be incorporated within the proposed modeling and algorithmic frameworks.

Acknowledgements

E.A.-M. acknowledges the support of the Chilean Council of Scientific and Technological Research, CONICYT, through the grant FONDECYT N.1180670 and through the Complex Engineering Systems Institute (ICM-FIC:P-05-004-F, CONICYT:FB0816). The research of M.S. was supported by the Austrian Research Fund (FWF, Project P 26755-N19).

References

R. Aboolian, O. Berman, and D. Krass. Competitive facility location model with concave demand. *European Journal of Operational Research*, 181(2):598–619, 2007.

M. Akella, R. Batta, E. Delmelle, P. Rogerson, A. Blatt, and G. Wilson. Base station location and channel allocation in a cellular network with emergency coverage requirements. *European Journal of Operational Research*, 164(2):301–323, 2005.

M. Albareda-Sambola, E. Fernández, and F. Saldanha-da Gama. The Facility Location problem with Bernoulli demands. *Omega*, 39(3):335–345, 2011.

J.E. Beasly. Collection of test data sets for a variety of operations research (OR) problems, 1990. URL `http://people.brunel.ac.uk/~mastjjb/jeb/info.html`.

O. Berman and D. Krass. Flow intercepting spatial interaction model: a new approach to optimal location of competitive facilities. *Location Science*, 6(1):41–65, 1998.

O. Berman and D. Krass. The generalized maximal covering location problem. *Computers & Operations Research*, 29(6):563–581, 2002.

O. Berman, D. Krass, and Z. Drezner. The gradual covering decay location problem on a network. *European Journal of Operational Research*, 151(3):474–480, 2003.

O. Berman, Z. Drezner, and D. Krass. Cooperative cover location problems: the planar case. *IIE Transactions*, 42(3):232–246, 2009a.

O. Berman, Z. Drezner, D. Krass, and G. Wesolowsky. The variable radius covering problem. *European Journal of Operational Research*, 196(2):516–525, 2009b.

O. Berman, Z. Drezner, and D. Krass. The multiple gradual cover location problem. *Journal of the Operational Research Society*, pages 1–10, 2018.

R. Church and K. Roberts. Generalized coverage models and public facility location. In *Papers of the regional science association*, volume 53, pages 117–135. Springer, 1983.

T. Cui, Y. Ouyang, and Z. Shen. Reliable facility location design under the risk of disruptions. *Operations Research*, 58(4):998–1011, 2010.

J. Current and M. O'Kelly. Locating emergency warning sirens. *Decision Sciences*, 23(1):221–234, 1992.

M. Daskin. *Network and Discrete Location: Models, Algorithms, and Applications.* Wiley, 2nd edition, 2013.

T. Drezner and Z. Drezner. The maximin gradual cover location problem. *OR spectrum*, 36(4):903–921, 2014.

Z. Drezner, G. Wesolowsky, and T. Drezner. The gradual covering problem. *Naval Research Logistics (NRL)*, 51(6):841–855, 2004.

H. Eiselt and V. Marianov. Gradual location set covering with service quality. *Socio-Economic Planning Sciences*, 43(2):121–130, 2009.

A. Ghosh, S. McLaffery, and C. Craig. Multifacility retail networks. In Z. Drezner, editor, *Facility Location*, pages 301–330. Springer, 1995.

K. Hogan and C. ReVelle. Concepts and applications of backup coverage. *Management Science*, 32(11): 1434–1444, 1986.

K. Jones and J. Simmons, editors. *Location, location, location: analyzing the retail environment.* Nelson, 1st edition, 1993.

O. Karasakal and E. Karasakal. A maximal covering location model in the presence of partial coverage. *Computers & Operations Research*, 31(9):1515–1526, 2004.

G. Laporte, S. Nickel, and F. Saldanha da Gama, editors. *Location Science.* Springer, 1st edition, 2015.

J. Leskovec, A. Krause, C. Guestrin, C. Faloutsos, J. VanBriesen, and N. Glance. Cost-effective outbreak detection in networks. In *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 420–429. ACM, 2007.

Q. Li, B. Zeng, and A. Savachkin. Reliable facility location design under disruptions. *Computers & OR*, 40 (4):901–909, 2013.

G. Nemhauser and L. Wolsey. Maximizing submodular set functions: formulations and analysis of algorithms. In *North-Holland Mathematics Studies*, volume 59, pages 279–301. Elsevier, 1981.

G. Nemhauser, L. Wolsey, and M. Fisher. An analysis of approximations for maximizing submodular set functionsi. *Mathematical Programming*, 14(1):265–294, 1978.

Z. Shen, R. Zhan, and J. Zhang. The reliable facility location problem: Formulations, heuristics, and approximation algorithms. *INFORMS Journal on Computing*, 23(3):470–482, 2011.

L. Snyder and M. Daskin. Reliability models for facility location: The expected failure cost case. *Transportation Science*, 39(3):400–416, 2005.

J. Thompson, editor. *Site selection*. Lebhar-Friedman, 1st edition, 1982.

H. Wei, A. Murray, and N. Xiao. Solving the continuous space p-centre problem: planning application issues. *IMA Journal of Management Mathematics*, 17(4):413–425, 2006.