# Revisiting Stackelberg p-median problem with user preferences

Yun Hui Lin[a], Qingyun Tian[b,*], Dongdong He[c], Yuan Wang[d]

[a]*Institute of High Performance Computing (IHPC), Agency for Science, Technology and Research (A*STAR), 1 Fusionopolis Way, #16-16 Connexis, Singapore 138632, Republic of Singapore*
[b]*School of Civil and Environmental Engineering, Nanyang Technological University, Singapore*
[c]*School of Economics and Management, Dalian University of Technology, China*
[d]*School of Business, Singapore University of Social Sciences, Singapore*

**Abstract**

The p-median facility location problem with user preferences (PUP) studies an operator that locates $p$ facilities to serve customers/users in a cost-efficient manner, upon anticipating customer preferences and facility choices. The problem can be visualized as a leader-follower game in which the operator is the leader who opens facilities, whereas the customer is the follower who observes the operator's location decision at first and then seeks services from the most preferred open facility. Despite that a considerable number of solution methodologies have been proposed, many of them are heuristic methods whose solution quality cannot be easily verified. Moreover, due to the hardness of the problems, existing exact approaches have limited performance. Motivated by these observations, we discuss various exact branch-and-cut solution approaches. In particular, we develop two approaches that have not been applied to PUP thus far. The first one explores the bilevel structure of the problem and derives an effective bilevel feasibility cut, whereas the second one is based on Benders decomposition, which is further accelerated using analytical Benders separation and heuristic separation. Using a broad testbed and benchmarks, we demonstrate the efficiency of our algorithms. Finally, we conduct sensitivity analysis to draw additional implications and to highlight the importance of considering user preferences when they exist.

*Keywords:* Location, User preference, Stackelberg game, Decomposition algorithm, Bilevel feasibility

## 1. Introduction

The classical facility location problem (FLP) studies the strategic planning of an operator who provides services to customers. The problem involves the determination of a set of open facilities from candidate sites, aiming to minimize the cost of satisfying the demand of customers/or maximize some objectives of interest (e.g., to maximize the profit of the operator or to cover as many customers as possible). To date, a large number of models have been proposed to address the FLPs under various settings. For comprehensive discussions of models, algorithms, and applications, we refer readers to Melo et al. (2009) and Daskin (2011).

*Corresponding author.
*Email addresses:* liny@ihpc.a-star.edu.sg (Yun Hui Lin), qingyun001@e.ntu.edu.sg (Qingyun Tian), hedd@dlut.edu.cn (Dongdong He), wangyuan@suss.edu.sg (Yuan Wang)

In the FLP literature, it is commonly assumed that customers (users) are allocated to their closest facilities or the operator acts as a *centralized* decision-maker that assigns customers to facilities and serves them in a cost-efficient fashion. However, this allocation/assignment rule is relatively simple and does not account for customer preferences. In many scenarios, customers may not prefer the assigned facility. They may instead seek services from some other facilities, from which it could be costly for the operator to serve them. This implies that there could be a mismatch between the preferences of the customers and the desirable customer allocation/assignment of the operator. Therefore, when locating the facilities, the operator needs to consider this mismatch, giving rise to a variant of the FLP that is often referred to as the facility location problem with user preferences (FLPUP) or with order (Camacho-Vallejo et al., 2014b; Hansen et al., 2004; Marić et al., 2012; Vasilev et al., 2009; Vasilyev and Klimentova, 2010). When the number of facilities to be set up is fixed, the problem is called the p-median problem with user preferences (PUP) (Casas-Ramírez and Camacho-Vallejo, 2017; Camacho-Vallejo et al., 2014a).

In the context of FLPUP and PUP, customers are studied as individual decision-makers that select facilities to serve them based on their preferences. The resultant problem is visualized from a fully *decentralized* perspective and thus can be characterized as a Stackelberg leader-follower game, in which the operator is the leader and the customer is the follower. Here, we briefly illustrate the decision process of both the operator and the customer. At the initial stage, customers and candidate facilities are known as discrete nodes. At Stage 1, the operator selects locations from the candidate sites. Customers, as followers, observe the operator's decisions at first and then request services from facilities at Stage 2. Finally, the operator proceeds to serve customers according to the requests. It is worth noting that when locating facilities at Stage 1, the operator needs to anticipate how customers choose facilities at Stage 2 because the decentralized customers want to be served by the most preferred facilities that are not necessarily cost-efficient from the operator's side. Customers' requests could thus have a significant impact on the operators' service profile and the service provision cost at Stage 3. Since facilities are generally built for a long-term purpose and it will be costly and difficult to alter the decision, it is important for the operator to consider customer preferences and anticipate customers' choices.

This paper investigates a PUP, whose the mathematical model is computationally challenging. The main purpose of this paper is to briefly revisit existing exact solution approaches and then develop efficient algorithms for large-scale PUP instances. Below, we briefly review the related literature with a focus on the solution approaches to position our contributions.

**Literature review.** The first study that considered customer preferences is Hanjoul and Peeters (1987) where each customer has a preference ordering toward the set of potential locations, and the simple plant location problem was presented with a set of constraints on preference ordering. A greedy heuristic algorithm was proposed to conduct the numerical tests on multiple instances. Since then, there is a proliferation of research on the FLPUP and PUP.

Although the problem has a Stackelberg structure (as illustrated in Figure 1), single-level formulation indeed exists. A typical exact approach thus consists of formulating the FLPUP and PUP as a single-level mixed-integer linear programming (MILP) and then solving the resultant MILP using existing solvers (possibly strengthened by some valid inequalities). In general, the formulation can be divided into two research streams. The first one is based on the primal-dual information of the lower-level problem and leverages the KKT conditions to recast the bilevel model into a mixed-integer program with complementarity constraints. Then the non-convex complementarity constraints are linearized, giving rise to a MILP that can be directly solved by

off-the-shelf solvers (e.g., see Camacho-Vallejo et al. (2014a); Casas-Ramírez and Camacho-Vallejo (2017); Cao and Chen (2006); Lotfi et al. (2021)). However, due to the use of the KKT conditions, such an approach needs to introduce a large number of variables (i.e., the dual variables) that dramatically increases the formulation size. As a consequence, its performance is typically limited. The second stream, essentially, utilizes the *closest assignment constraints* (CACs) to subtly rewrite the lower-level problem into a set of linear constraints without additional variables. For a comprehensive discussion on the CACs, we refer readers to Espejo et al. (2012). To date, a considerable number of studies have adopted the CAC-based reformulation. For example, Cánovas et al. (2007) proposed a strengthened CAC to speed up solving the model of Hanjoul and Peeters (1987). Vasilev et al. (2009) provided several valid inequalities when the bilevel problem was reformulated into a MILP using CACs and demonstrated the tightness of the inequalities in terms of their linear relaxations and integrality gaps. Then Vasilyev and Klimentova (2010) implemented a branch-and-cut method based on the proposed family of valid inequalities in Vasilev et al. (2009). A recent application can be found in Casas-Ramírez and Camacho-Vallejo (2017).

We point out that the above approaches are exact and can yield the proven optimal solution, provided that the reformulated MILP model can be solved optimally. However, the MILP model itself is a challenging problem that cannot scale well on the problem size, rendering these approaches computational prohibitive for large-scale instances. Therefore, heuristic approaches are gaining popularity. Recent years have witnessed the use of Lagrangian-based heuristics for various problems involving customer preferences (e.g., see Cabezas and García (2018); Cabezas et al. (2021); Lee and Lee (2012)). Moreover, there is an explosion of related research on metaheuristics. Marić et al. (2012) proposed three metaheuristic methods for solving FLPUP, i.e., Particle Swarm Optimization, Simulated Annealing, and a combination of Reduced and Basic Variable Neighborhood Search Method. To solve a large-scale bilevel FLPUP, Camacho-Vallejo et al. (2014b) developed a population-based evolutionary algorithm (EA). A similar EA was applied to solve a bilevel facility location problem with cardinality constraints and preferences (Calvete et al., 2020). Casas-Ramírez et al. (2018) obtained approximate solutions to the problem by using a cross-entropy method to search for improved location decisions. In Casas-Ramírez and Camacho-Vallejo (2017), the authors investigated the PUP model and designed a scatter search metaheuristic algorithm to efficiently handle large-scale problems. In the context of maximum coverage, Díaz et al. (2017) presented a bilevel model considering that customers will choose the facilities within a coverage radius. They proposed a greedy randomized adaptive search procedure (GRASP) heuristic and a hybrid GRASP-Tabu heuristic to find near-optimal solutions. Recently, Mrkela and Stanimirović (2021) investigated a similar problem as in Díaz et al. (2017). Instead of locating $p$ facilities, they considered a limited budget for locating facilities, and an efficient Variable Neighborhood Search was proposed as the solution approach.

**Our contributions.** Despite that there is a considerable number of research works dedicated to developing solution methodologies for the FLPUP and PUP, many of them are heuristic methods whose solution quality cannot be easily verified. Moreover, due to the hardness of the problems, existing exact approaches based on single-level reformulation techniques have limited performance. Such an observation motivates this paper to develop efficient exact algorithms that can be applied to solve large-scale PUP models. In summary, our contributions are three-fold.

(i) *A branch-and-cut approach leveraging bilevel feasibility cut*. Essentially, PUP models belong to a type of mixed-integer bilevel program. By an equivalent reformulation of the lower-level problem, we derive a bilevel feasibility cut, which can effectively cut off bilevel infeasible solutions.

We also demonstrate that such a cut can be easily embedded into the branch-and-cut framework.

(ii) *A branch-and-cut approach based on Benders decomposition*. By exploring the structure of a CAC-based MILP model, we propose a branch-and-cut algorithm, where we project out the high-dimensional preference-related variables and solve a master problem defined only on the low-dimensional location-related decision space to generate cutting planes. Moreover, to further accelerate the solution process, we design an analytical approach for Benders separation.

(iii) *Computational studies and managerial implications*. Using a broad testbed, we conduct extensive computational experiments to show the efficiency of our proposed approaches. We also perform sensitivity analysis using instances from a standard benchmark library. Our analysis reveals that when user preferences exist, the operator must consider these preferences when anticipating customer choices; otherwise, the operator could suffer from a substantially higher cost, and opening more facilities could ironically result in additional service costs.

The rest of the paper is organized as follows. We introduce the problem and the model in Section 2 and widely used exact solution approaches in the literature in Section 3. We then develop the Benders decomposition and the acceleration technique in Section 5, followed by the development of the branch-and-cut approach based on bilevel feasibility cut in Section 4. Extensive numerical studies are conducted in Section 6 to demonstrate the effectiveness of our algorithm and provide additional managerial implications. Finally, we conclude the paper in Section 7.

## 2. Model formulation

This section describes the model formulation of the p-median problem with user preferences (PUP). Table 1 provides the main notations.

Table 1: Notation table.

| | | |
|---|---|---|
| Sets | | |
| $I$ | : | the set of customers. |
| $J$ | : | the set of candidate facilities. $\|J\| = n$ |
| Parameters | | |
| $p$ | : | the number of open facilities |
| $c_{ij}$ | : | the cost of serving customer $i$ by facility $j$, $\forall i \in I, j \in J$ |
| $g_{ij}$ | : | the disutility of facility $j$ to customer $i$, $\forall i \in I, j \in J$ |
| $\pi_{ij}$ | : | ranking of disutility of facility $j$ among all candidate facilities to customer $i$, $\forall i \in I, j \in J$ |
| Decision Variables | | |
| $x_j$ | : | binary. 1, if facility $j$ is open; 0, otherwise, $\forall j \in J$ |
| $y_{ij}$ | : | binary. 1, if customer $i$ seeks the service from facility $j$; 0, otherwise, $\forall i \in I, j \in J$ |

### 2.1. Stackelberg visualization

The virtual decision process of PUP is as explained in the introduction. We use $J$ to denote the set of candidate facilities and $I$ to denote the set of customers. At Stage 1, the operator will open $p$ facilities, selecting from set $J$. We define a binary variable $x_j, \forall j \in J$, which is 1 if facility $j$ is open; 0, otherwise. After facilities are built, customers then determine which facilities to seek services from at Stage 2. To reflect customers' choices, we define a binary variable $y_{ij}$ such that $y_{ij} = 1$ if customer $i$ requests services from facility $j$, $\forall i \in I, j \in J$.

Following the standard assumption in the FLPUP and the PUP literature, we assume that facility $j$ has a disutility to customer $i$, i.e., $g_{ij}$, and customer $i$ prefers facility $j$ over facility $k$, if

$g_{ij} < g_{ik}$. In other words, when both facility $j$ and facility $k$ are open, customer $i$ will not select facility $k$ if $g_{ij} < g_{ik}$. Moreover, customer preferences for facilities are different, i.e., elements in the vector $g_{i\cdot}$ are distinct ($g_{ij} \neq g_{ik}$ if $j \neq k$). With these definitions, given a location decision $x$, customer preferences to the facilities can be modeled by the following problem

$$\min_{y} \sum_{i\in I}\sum_{j\in J} g_{ij}y_{ij} \tag{1a}$$

$$\text{st.} \sum_{j\in J} y_{ij} = 1 \qquad \forall i \in I \tag{1b}$$

$$y_{ij} \leq x_j \qquad \forall i \in I, j \in J \tag{1c}$$

$$y_{ij} \in \{0,1\} \quad \forall i \in I, j \in J \tag{1d}$$

which describes the decision problem at Stage 2. The objective function imposes that customers will minimize their disutilities. This is equivalent to state that he/she will request the service from the most preferred facility located by the operator.

Given the solution $y$ from the above problem, the operator incurs a service provision cost of $\phi = \sum_{i\in I}\sum_{j\in J} c_{ij}y_{ij}$ at Stage 3, where $c_{ij}$ is the cost of serving customer $i$ from facility $j$ (possibly weighted by the demand size or the relative "importance" of the customer). Note that the operator also wants to minimize the total service cost; therefore, when locating facilities at Stage 1, the operator needs to solve the following problem

$$\min_{x,y} \phi = \sum_{i\in I}\sum_{j\in J} c_{ij}y_{ij} \tag{2a}$$

$$\text{st.} \sum_{j\in J} x_j = p \tag{2b}$$

$$x_j \in \{0,1\} \quad \forall j \in J \tag{2c}$$

**[PUP]**

$$\min_{y} \sum_{i\in I}\sum_{j\in J} g_{ij}y_{ij} \tag{2d}$$

$$\text{st.} \sum_{j\in J} y_{ij} = 1 \quad \forall i \in I \tag{2e}$$

$$y_{ij} \leq x_j \qquad \forall i \in I, j \in J \tag{2f}$$

$$y_{ij} \in \{0,1\} \quad \forall i \in I, j \in J \tag{2g}$$

which is a *mixed-integer bilevel program*, where the *upper-level problem* (ULP) is (2a)-(2c), which defines the objective and the decision space of the operator; whereas the *lower-level problem* (LLP) is (2d)-(2g), which is exactly Problem (1). For the ease of exposition we will refer to Problem (1) as the LLP from here onwards.

Throughout this paper, we assume that elements in the vector $g_{i\cdot}$ are distinct. Under such a distinct disutility assumption, an operator's decision will only lead to a unique optimal solution of $y$ (i.e., Problem (1) has a unique optimal solution). This assumption is somewhat restrictive. However, it appears to be reasonable for various scenarios. For example, if the disutility of a facility to a customer is represented by the distance, then it is very unlikely to observe two facilities that have exactly the same distance away from the customer. Note that, without this assumption, [PUP] can be seen as an "optimistic" bilevel problem, in which customers are assumed to "cooperate" with the operator, i.e., in the presence of multiple optimal solutions, customers will select

the one that leads to a lowest cost of the operator.

To facilitate our discussion, we define the following set

$$\Omega^U = \{x \mid (2b) - (2c)\} \qquad \Omega = \{(x, y) \mid (2b) - (2c), (2e) - (2g)\}$$

where $\Omega^U$ is the decision space of the operator and represents the feasible region of the upper-level variable. $\Omega$ is the joint decision space of the operator and the customer, containing all constraints in [PUP] except for the embedded minimization restriction (2d).

## 3. Single-level reformulation approaches

The bilevel structure of [PUP] renders the problem computationally challenging. This section thus discusses single-level reformulation approaches for [PUP] to circumvent the difficulties of handling Problem (1) in [PUP]. Note that in the literature of facility location with user preferences, the *primal-dual reformulation* has been used (Casas-Ramírez and Camacho-Vallejo, 2017; Casas-Ramírez et al., 2018; Lotfi et al., 2021). The fundamental idea is to replace Problem (1) by its KKT conditions, thereby recasting [PUP] into a single-level primal-dual reformulation model (PDRM). We provide the PDRM formulation in Appendix A for self-interested readers. Note that this approach introduces a large number of new variables into the formulation (i.e., there are $|I| \cdot (2|J| + 1)$ dual variables in the KKT conditions) and a significant number of additional constraints, which, unfortunately, increase the problem size by a large margin and render the approach computationally intractable even for moderate-sized instances. In this section, we discuss alternative approaches based the so-call *closest assignment constraint*.

### *3.1. Reformulation based on closest assignment constraint*

To start with, we first observe that in the lower-level problem in [PUP], customer $i$ selects the open facility with lowest disutility. Clearly, if an open facility has a higher disutility than another open facility, then it will not be selected by the customer. This indicates that it is the ranking (not necessarily the value) of $g_i$ that matters for customer' choices. Therefore, for customer $i$, we can replace its $g_i$ by the associated ranking vector $\pi_i$, in which if $g_{ij} > g_{ik}$, then we set $\pi_{ij} > \pi_{ik}$. Naturally, the minimum and maximum values of $\pi_i$ are 1 and $n$ (the number of candidate facilities).

Using $\pi$, minimizing the objective function (2d) can be represented by the following inequality

$$\sum_{k \in J} \pi_{ik} y_{ik} \leq (\pi_{ij} - n) x_j + n \quad \forall i \in I, j \in J \tag{3}$$

To see the equivalence, if facility $j$ is not open ($x_j = 0$), then the right hand side becomes $n$, meaning that (3) is inactive. If facility $j$ is open ($x_j = 1$), then (3) imposes that, among all the possible allocations, customer $i$ will be allocated to the facility with the minimum ranking value. Essentially, this constraint belongs to a type of *closest assignment constraints* (CACs), referring to as *BDTW* in Espejo et al. (2012). Based on it, we have the following single-level reformulation model

for [PUP]

$$\min \sum_{i\in I}\sum_{j\in J} c_{ij}y_{ij} \tag{4a}$$

**[SRM-BDTW]** $$\text{st.} \sum_{k\in J} \pi_{ik}y_{ik} \leq (\pi_{ij}-n)x_j + n \quad \forall i \in I, j \in J \tag{4b}$$

$$(x, y) \in \Omega \tag{4c}$$

We point out that other types of CACs that can possibly be used to reformulate [PUP]. They have been discussed in detail in Espejo et al. (2012). Among all CACs, we choose (4b) since it is the one that is used in the same problem studied by Casas-Ramírez and Camacho-Vallejo (2017). Moreover, it only involves $O(|I|\cdot|J|)$ constraints, whereas others may contain up to $O(|I|\cdot|J|\cdot|J|)$ constraints. Indeed, there do exist other CACs with $O(|I|\cdot|J|)$ constraints. In particular, the following one is proved to yield tight continuous relaxation

$$\sum_{k\in J:\pi_{ik}>\pi_{ij}} y_{ik} \leq 1 - x_j \quad \forall i \in I, j \in J \tag{5}$$

which is referred to *WF* in Espejo et al. (2012). This constraint states that if facility $j$ is open (i.e., $x_j = 1$), then those facilities that have higher disutilities than $\pi_{ij}$ will not be selected by customer $i$, effectively imposing that customer $i$ will be assigned to the most preferred open facility. The following single-level model thus arises

$$\min \sum_{i\in I}\sum_{j\in J} c_{ij}y_{ij} \tag{6a}$$

**[SRM-WF]** $$\text{st.} \sum_{k\in J:\pi_{ik}>\pi_{ij}} y_{ik} \leq 1 - x_j \quad \forall i \in I, j \in J \tag{6b}$$

$$(x, y) \in \Omega \tag{6c}$$

### *3.2. Branch-and-cut enhancement*

Since both [SRM-BDTW] and [SRM-WF] are mixed-integer linear programs (MILPs), they are ready to be solved by modern MILP solvers (in this paper, we utilize Gurobi 9.5 as our solver). As shown in Casas-Ramírez and Camacho-Vallejo (2017), such a "plug-in" approach cannot handle large-scale problems well. Fortunately, based on the reformulation structure, it is possible that we further enhance the performance of [SRM-BDTW] and [SRM-WF] by implementing the associated CACs as cutting planes that are generated on-the-fly and added to the problem only as needed within Gurobi's branch-and-cut (BC) framework.

Specifically, we start solving [SRM-BDTW] (or [SRM-WF]) without Constraint (4b) (or Constraint (6b)) using Gurobi. At integer nodes of the BC searching tree (i.e., the current LP relaxation turns out to be integer), we extract the current solution $(\bar{x}, \bar{y})$. Then, we check whether $(\bar{x}, \bar{y})$ violates Constraint (4b) (or Constraint (6b)). If violation is observed, then we add the corresponding CAC to the searching tree as a *lazy cut* using the *callback* function of Gurobi.

We refer to the approaches as `BC-BDTW` and `BC-WF` to indicate that the associated CACs are implemented in a BC manner. Although the implementation is rather straightforward, it can speed up the computation by a large margin. Appendix B reports detailed computational results, which illustrate the benefit of BC approaches.

## 4. A new branch-and-cut approach based on bilevel feasibility cut

In this section, we present an alternative single-level BC approach by exploring the bilevel structure of [PUP] with an effective feasibility cut.

### *4.1. Algorithm framework*

We start with the following result. For a solution $(x, y)$ in the joint decision space of the operator and the customer, i.e., $(x, y) \in \Omega$, Problem (1) can be restated as

$$\max_{y} \sum_{i\in I}\sum_{j\in J} \theta_{ij} y_{ij} x_j \tag{7a}$$

$$\text{st.} \sum_{j\in J} y_{ij} = 1 \qquad \forall i \in I \tag{7b}$$

$$y_{ij} \in \{0, 1\} \qquad \forall i \in I, j \in J \tag{7c}$$

where $\theta_{ij} = n - \pi_{ij} + 1, \forall i \in I, j \in J$.

To verify this reformulation, note that, by definition, $\theta_{ij} > 0$. If $g_{ij} < g_{ik}$ (thus $\pi_{ij} < \pi_{ik}$), then $\theta_{ij} > \theta_{ik}$. Now, the constraints impose that each customer should be "assigned" to one facility. Since we are maximizing the objective function, when $x_j = 0$, $y_{ij}$ will be 0 in the optimal solution because setting $y_{ij} = 1$ cannot contribute to the growth in the objective. When $x_j = 1$, $y_{ij}$ can be 1. To maximize the objective, the $y_{ij}$ corresponding to the largest $\theta_{ij}$ should be set to 1, meaning that the facility with the lowest disutility to customer $i$ (the smallest $g_{ij}$ or $\pi_{ij}$) will be selected, which is exactly the goal of Problem (1).

In fact, if we interpret $\theta_{ij}$ as the "utility" of facility $j$ to customer $i$, then Problem (7) says that *customers will maximize their utilities through selecting the facilities to seek services from*. Similar to Lin and Tian (2023), the purpose of using Problem (7) to model customer choices is to get rid of the "decision-dependent" feasible region of LLP. In Problem (1), the feasible region is affected by the variable in the ULP (i.e., $x$). By contrast, in Problem (7), the feasible region, denoted by $\Omega^L$, does not depend on the operator decision any more. We can thus express the decision space of customers as

$$\Omega^L = \{y \mid ((7b) - (7c)\}$$

Mathematically, the upper-level variable now only appears in the objective function of Problem (7). This property is important for our algorithm development.

**Definition 1.** *A solution $(x, y)$ is bilevel feasible if given an operator decision $x$, $y$ is an optimal solution to Problem (7).*

Essentially, for a bilevel feasible solution $(x, y)$, $y$ reflects the best choices of the customers under the location decision $x$. Certainly, $(x, y)$ is feasible for [PUP] if and only if it is bilevel feasible. For the operator, minimizing the service provision cost is equivalent to searching for a bilevel feasible solution to minimize the objective function in [PUP].

The core question becomes how to represent the bilevel feasibility in a mathematical manner such that when a solution is not bilevel feasible, we can easily identify it and then eliminate it using some specialized cutting planes. It turns out that the bilevel feasibility can be expressed as the following condition with an exponential number of constraints that match the size of $\Omega^L$.

**Lemma 1.** *A solution $(x, y) \in \Omega$ is bilevel feasible if*

$$\sum_{j \in J} \theta_{ij} y_{ij} \geq \sum_{j \in J} \theta_{ij} \bar{y}_{ij} x_j \quad \forall i \in I, \bar{y} \in \Omega^L \tag{8}$$

*Proof.* For a solution $(x, y) \in \Omega$, if it is bilevel feasible solution, then it must satisfy

$$\sum_{i \in I} \sum_{j \in J} \theta_{ij} y_{ij} x_j \geq \sum_{i \in I} \sum_{j \in J} \theta_{ij} \bar{y}_{ij} x_j \quad \forall \bar{y} \in \Omega^L \tag{9}$$

The "$\forall \bar{y} \in \Omega^L$" statement says that we enumerate all possible customer choices in $\Omega^L$. In this case, if the above condition holds, then $y$ is the customer choice that maximizes the objective function of Problem (7) given the location decision $x$. In other words, $y$ is an optimal solution to Problem (7) and thus, $(x, y)$ is bilevel feasible by definition. Moreover, in the left hand side of (9), the bilinear term $y_{ij} x_j$ can be replaced by $y_{ij}$ because the condition $y_{ij} \leq x_j$ is included in $\Omega$. Finally, both (9) and $\Omega^L$ are separate for each customer; therefore, we can impose up to $|I|$ constraints (i.e., one for each customer) for each $\bar{y}$, giving rise to the formulation in (8). □

In fact, for each $\bar{y}$, we can treat Constraint (8) as a *bilevel feasible cut* (BFC), which cuts off the bilevel infeasible solutions to recover bilevel feasibility. Now, we can rewrite [PUP] as the following single-level problem

$$\begin{aligned}
& \min \ \sum_{i \in I} \sum_{j \in J} c_{ij} y_{ij} && && \text{(10a)} \\
\textbf{[EP]} \qquad & \text{st.} \ \sum_{j \in J} \theta_{ij} y_{ij} \geq \sum_{j \in J} \theta_{ij} \bar{y}_{ij} x_j && \forall i \in I, \bar{y} \in \Omega^L && \text{(10b)} \\
& (x, y) \in \Omega && && \text{(10c)}
\end{aligned}$$

We call it the *extended problem* ([EP]). Such a formulation arises because $\Omega^L$ does not depend on $x$ and thus, we are able to enemerate all feasible points in $\Omega^L$ in advance (this is why we derive Problem (7) to replace the LLP in Problem (1)). Theoretically, solving [EP] yields the optimal solution to [PUP]. However, there are in total $|\Omega^L| \times |I|$ number of BFCs. Certainly, it is impractical to solve [EP] directly owing to the problem size.

We thus resort to the relaxed version of [EP] with partial enumeration, i.e.,

$$\begin{aligned}
& \min \ \sum_{i \in I} \sum_{j \in J} c_{ij} y_{ij} && && \text{(11a)} \\
\textbf{[rEP]} \qquad & \text{st.} \ \sum_{j \in J} \theta_{ij} y_{ij} \geq \sum_{j \in J} \theta_{ij} \bar{y}_{ij} x_j && \forall i \in I, \bar{y} \in \Pi && \text{(11b)} \\
& (x, y) \in \Omega && && \text{(11c)}
\end{aligned}$$

where set $\Pi$ is such that $\Pi \subset \Omega^L$. Our philosophy is to maintain a small number of BFCs and then generate BFCs as needed. Obviously, solving [rEP] gives us a lower bound to [PUP] since [rEP] is a relaxation of [EP]. On the other hand, given $x$, $\bar{y}$ can be obtained by solving Problem (7). This $\bar{y}$ allows us to (possibly) update the upper bound as $\sum_{i \in I} \sum_{j \in J} c_{ij} \bar{y}_{ij}$. It can also be used to define a set of BFCs, which can be added to [rEP] to cut off bilevel infeasible solutions and improve the lower bound. Through subsequently generating BFCs, we can select an optimal bilevel feasible solution $(x^*, y^*)$ to [PUP].

*4.2. Branch-and-cut implementation*

The implementation is straightforward. We start solving [rEP] with $\Pi = \emptyset$. In the BC searching tree, when we visit an integer node, we read the solution $(\tilde{x}, \tilde{y})$ at the node. Then, we obtain the optimal customer choices $\bar{y}$ under $\tilde{x}$ by the easy sorting process described in Section 5. Now, we check the bilevel feasibility of $(\tilde{x}, \tilde{y})$. If we observe $\sum_{j\in J} \theta_{ij}\tilde{y}_{ij} < \sum_{j\in J} \theta_{ij}\bar{y}_{ij}\tilde{x}_j$, then $(\tilde{x}, \tilde{y})$ from the searching tree is not bilevel feasible for customer $i$ since $\tilde{y}$ does not lead to the maximum objective of Problem (7) under $\tilde{x}$. To recover bilevel feasibility, we add the corresponding BFC defined by $\bar{y}$ into [rEP] to cut off $(\tilde{x}, \tilde{y})$ using *lazy-cut callback* function of Gurobi.

## 5. Branch-and-cut approach based on Benders decomposition

*5.1. Standard branch-and-cut Benders decomposition*

In this section, we develop a BC algorithm based on Benders decomposition to expedite solving [SRM-WF]. Our decomposition is inspired by the following observation: In the [SRM-WF] formulation, $y_{ij} \in \{0, 1\}$ can be relaxed to $y_{ij} \geq 0$ without changing the solution and the objective. This follows directly from the assumption of distinct disutilites. It implies that when the operator's decision is made, the remaining subproblem will become a linear program (LP), which admits strong duality. Therefore, we can design a decomposition algorithm leveraging the dual information of the subproblem, which is exactly the idea of Benders decomposition.

In our approach, we only maintain the location decision $x$ in a master problem by projecting out the continuous variable $y$, which is used to reflect the customer preferences and compute the service cost. Specifically, we define the Benders master problem as

$$\min \sum_{i\in I} w_i \tag{12a}$$

$$\textbf{[MP]} \quad \text{st. } w_i \geq \Phi_i(x) \quad \forall i \in I \tag{12b}$$

$$x \in \Omega^U \tag{12c}$$

where $\Phi_i(x)$ is obtained by solving the following subproblem

$$\Phi_i(x) = \min \sum_{j\in J} c_{ij} y_{ij} \tag{13a}$$

$$\text{st. } \sum_{j\in J} y_{ij} = 1 \qquad (\mu_i) \tag{13b}$$

$$\textbf{[SP}_i(x)\textbf{]} \qquad y_{ij} \leq x_j \quad \forall j \in J \qquad (\lambda_{ij}) \tag{13c}$$

$$\sum_{k\in J:\pi_{ik}>\pi_{ij}} y_{ik} \leq 1 - x_j \quad \forall j \in J \quad (v_{ij}) \tag{13d}$$

$$y_{ij} \geq 0 \quad \forall j \in J \tag{13e}$$

Note that $\Phi_i(x)$ is a convex function over $x$; therefore, [MP] is a mixed-integer convex optimization problem, and we can solve it by approximating $\Phi_i(x)$ with its linear under-estimators, i.e., the Benders cuts.

Given an integer solution $\bar{x}$ from [MP], the Benders cut can be obtained by solving the dual problem of [SP$_i(\bar{x})$]. Let $\mu_i$ and $\lambda_{ij}$ and $v_{ij}$ in (13) be the dual variables associated with the con-

straints. The dual subproblem is given by

$$\max\ \mu_i - \sum_{j\in J} v_{ij} + \sum_{j\in J}(v_{ij} - \lambda_{ij})\bar{x}_j \tag{14a}$$

$$\textbf{[DSP}_i(\bar{x})\textbf{]} \qquad \text{st. } c_{ij} - \mu_i + \lambda_{ij} + \sum_{k\in J:\pi_{ik}<\pi_{ij}} v_{ik} \geq 0 \quad \forall j \in J \tag{14b}$$

$$\lambda_{ij} \geq 0 \quad \forall j \in J \tag{14c}$$

$$v_{ij} \geq 0 \quad \forall j \in J \tag{14d}$$

Let $(\bar{\lambda}, \bar{v}, \bar{\mu})$ be the optimal solution to the above LP. The following Benders cut defined at $\bar{x}$ arises.

$$w_i \geq \Phi_i(x) \geq \bar{\mu}_i - \sum_{j\in J} \bar{v}_{ij} + \sum_{j\in J}(\bar{v}_{ij} - \bar{\lambda}_{ij})x_j \quad \forall i \in I \tag{15}$$

We can now solve [MP] using the following MILP formulation, i.e., the *relaxed master problem*,

$$\min\ \sum_{i\in I} w_i \tag{16a}$$

$$\textbf{[rMP]} \qquad \text{st. } w_i \geq \bar{\mu}_i - \sum_{j\in J} \bar{v}_{ij} + \sum_{j\in J}(\bar{v}_{ij} - \bar{\lambda}_{ij})x_j \quad \forall i \in I, (\bar{\lambda}, \bar{v}, \bar{\mu}) \in \Xi \tag{16b}$$

$$x \in \Omega^U \tag{16c}$$

where $\Xi$ is the set of dual variables. We can expand the set, each time we have an integer solution $\bar{x}$ and solve the corresponding [DSP$_i(\bar{x})$]. These dual variables define Benders cuts that cut off non-optimal solutions and lead us to the proven optimal solution.

Modern advanced solvers, such as Gurobi, use BC algorithms to solve (mixed-)integer programs. Apart from the general-purpose cuts that are embedded within the solvers, one can design a problem-specific separation function that uses a solution of [rMP] as an input and generates violated cuts, which are inserted into the BC searching tree to improve the relaxation bound and/or cut off non-optimal solutions. State-of-the-art Benders decomposition algorithms for facility location problems are typically implemented within solvers' BC frameworks (Fischetti et al., 2016; Lin and Tian, 2021; Taherkhani et al., 2020). We thus utilize this approach as well. More specifically, we use Gurobi 9.5.2 and rely on its default branching rules and embedded cutting planes. We initialize the master problem [rMP] without Benders cuts (i.e., $\Xi = \emptyset$). At integer nodes, we retrieve the integer value of $\bar{x}$ and generate the corresponding Benders cuts as in (15) by solving [DSP$_i(\bar{x})$]. The Benders separation is performed within the *callback* function of Gurobi. Cuts are added as *lazy cuts* to the current node relaxation only if they violate the current solution $(\bar{x}, \bar{w})$ by more than $10^{-5}$ of the relative violation, defined as the absolute violation of the cut divided by the current $\bar{w}_i$ value. Due to the convexity of $\Phi_i$, these inserted cuts are globally valid and will eventually lead us to the optimal solution of [SRM-WF] with a zero MIP gap.

### *5.2. Accelerating Benders approach through analytical integer separation*

The above Benders decomposition generates Benders cuts by solving the dual subproblem [DSP$_i(\bar{x})$] using external solvers, each time an integer node is visited during the BC searching tree. However, [DSP$_i(\bar{x})$] itself is a large-scale LP. For some problems, solving [DSP$_i(\bar{x})$] to obtain an optimal dual variable is not trivial and may take up to several seconds. Noting that we

typically need to visit a large number of integer nodes before the MIP gap vanishes, significant computational time could thus be consumed on Benders separation, causing a bottleneck in the BC algorithm.

Fortunately, by carefully exploring the structure of [SP$_i(\bar{x})$] and [DSP$_i(\bar{x})$], an optimal dual solution can be generated analytically, allowing for expediting the algorithm by a large margin. This is supported by the result in the following lemma.

**Lemma 2.** *Given an integer solution $\bar{x}$ from [rMP], let $\bar{\tau} = \{j \in J \mid \bar{x}_j = 1\}$ and $m_i = \arg\min_{j \in \bar{\tau}} \pi_{ij}$. An optimal dual solution to [DSP$_i(\bar{x})$] can be computed analytically as*

$$\mu_i = c_{im_i} \tag{17a}$$

$$v_{im_i} = \left[\max_{k \in \bar{\tau} \setminus \{m_i\}} (\mu_i - c_{ik})\right]_+ \tag{17b}$$

$$v_{ij} = 0 \qquad \forall j \in \setminus \{m_i\} \tag{17c}$$

$$\lambda_{ij} = (1 - \bar{x}_j)\left[\mu_i - c_{ij} - \mathcal{I}(\pi_{im_i} < \pi_{ij}) \cdot v_{im_i}\right]_+ \qquad \forall j \in J \tag{17d}$$

*where $[z]_+ = \max\{z, 0\}$, and $\mathcal{I}(z)$ is the indicator function, which is 1 if statement $z$ is true; 0, otherwise.*

*Proof.* To show that (17) is an optimal dual solution, we need to prove that it is feasible and optimal to **[DSP**$_i(\bar{x})$**]**.

**(i) Feasibility**. We first show that the dual solution in (17) is feasible. Since $\lambda$ and $v$ are negative by definition, the remaining task is to check whether (14b) holds. Plugging (17) into the left-hand-side of (14b) leads us to

$$c_{ij} - c_{im_i} + \mathcal{I}(\pi_{im_i} < \pi_{ij}) \cdot v_{im_i} + (1 - \bar{x}_j)\left[c_{im_i} - c_{ij} - \mathcal{I}(\pi_{im_i} < \pi_{ij}) \cdot v_{im_i}\right]_+ \tag{18}$$

If $\bar{x}_j = 0$, then we have[1]

$$c_{ij} - c_{im_i} + \mathcal{I}(\pi_{im_i} < \pi_{ij}) \cdot v_{im_i} + \left[c_{im_i} - c_{ij} - \mathcal{I}(\pi_{im_i} < \pi_{ij}) \cdot v_{im_i}\right]_+ \geq 0 \qquad \forall j \in \bar{\tau} \tag{19}$$

If $\bar{x}_j = 1$, then we have

$$c_{ij} - c_{im_i} + \mathcal{I}(\pi_{im_i} < \pi_{ij}) \cdot v_{im_i} \tag{20}$$

$$= c_{ij} - c_{im_i} + \left[\max_{k \in \bar{\tau} \setminus \{m_i\}} (c_{im_i} - c_{ik})\right]_+ \geq c_{ij} - c_{im_i} + \left[c_{im_i} - c_{ij}\right]_+ \geq 0 \quad \forall j \in J \setminus \bar{\tau} \tag{21}$$

Therefore, the dual solution in (17) is feasible.

**(ii) Optimality**. We resort to strong duality. Clearly, the primal objective under $\bar{x}$ is $c_{im_i}$. For the dual objective, plugging (17) into (14a) gives rise to

$$\mu_i - \sum_{j \in J} v_{ij} + \sum_{j \in J} (v_{ij} - \lambda_{ij})\bar{x}_j = c_{m_i} - v_{im_i} + \sum_{j \in \bar{\tau}} (v_{ij} - \lambda_{ij}) = c_{m_i} - v_{im_i} + v_{im_i} = c_{m_i} \tag{22}$$

Therefore, the dual objective is equal to the primal objective. According to strong duality, we conclude that the dual solution is optimal. □

[1] Note that for two scalar numbers $a \geq 0, b \geq 0$, it holds that $a - b + [b - a]_+ \geq 0$

Now, given an integer $\bar{x}$, the optimal dual variables can be obtained according to Lemma 2, which only involves a sorting algorithm (i.e., determining the most preferred open facility $m_i$ for customer $i$) and some elemental matrix manipulation. Therefore, the analytical separation is very efficient. We emphasize that the integrality of $\bar{x}$ is important for the analytical separation because Lemma 2 holds only if $\bar{x}$ is an integer point. When we implement the accelerated Benders approach, analytical separation of Benders cuts is performed for every integer node.

### *5.3. Heuristic separation at fractional nodes*

It is worth noting that the applicability of our separation is not restricted to integer nodes. At fractional nodes, we can use primal heuristics to construct a feasible integer solution of $x$ and then separate Benders cuts accordingly. In this paper, our primal heuristics is based on a greedy scheme: given a fractional solution $(\tilde{x}, \tilde{w})$, we sort the $\tilde{x}$ according to its values from the largest to the smallest. Then we set the largest $P$ elements in $\tilde{x}$ to 1 and the remaining elements to 0. This ensures that the resulting integer solution $\bar{x}$ satisfies Constraint $\sum_{j\in J} x_j = p$ and thus is feasible.

We then implement our analytical separation in Section 5.2 using $\bar{x}$ and obtain the associated dual variables $(\bar{\lambda}, \bar{\mu}, \bar{v})$. If we observe that for $i \in I$, the current solution $(\tilde{x}, \tilde{w})$ violates the associated Benders cut by more than $10^{-5}$ of the relative violation, i.e.,

$$\tilde{w}_i \cdot (1 + 10^{-5}) < \bar{\mu}_i - \sum_{j\in J} \bar{v}_{ij} + \sum_{j\in J} (\bar{v}_{ij} - \bar{\lambda}_{ij})\tilde{x}_j \tag{23}$$

then we add the cut to the branch-and-cut searching tree. We refer to such primal heuristics and separation as *heuristic separation*. It is performed for every cut pass at the root node of the searching tree and once every 1000 nodes at the others.

## 6. Numerical study

This section presents numerical experiments. We first conduct extensive computational experiments using a broad testbed to demonstrate the computational efficiency of the proposed approaches. Then we discuss the importance of considering user preferences in facility location problems using instances from a standard library. All experiments are done on a 32 GB memory MacBook Pro with Apple M1 Max processor using Python as the programming language.

### *6.1. Experiment setup*

We refer to the BC approach based on bilevel feasibility cuts in Section 4 as `BC-BF` and the BC approach based on Benders cuts in Section 5 as `BC-Benders`. Furthermore, we also test the performance of `BC-BDTW` and `BC-WF` presented in Section 2. They serve as benchmarks.

We use three datasets with different scales and structures.

**RND**. Instances that are generated using Python with fixed random seeds. Candidate facilities and customers are generated from a two-dimensional uniform distribution $[0, 100]^2$. The distance between facility $j$ and customer $i$ (i.e., $l_{ij}$) is represented by the Euclidean distance. Each customer has a demand $d_i$, following a uniform distribution $[0, 10]$. The cost of serving customer $i$ from facility $j$ is computed as $c_{ij} = d_i \cdot l_{ij}$. The disutility $g_{ij}$ is randomly generated from $[(1-\delta)\cdot l_{ij}, (1+\delta)\cdot l_{ij}]$, where $0 \le \delta < 1$. That is, we allow $g_{ij}$ to deviate $100\delta\%$ from $l_{ij}$, and thus, the nearest facility is not necessarily the most attractive facility to customers. Furthermore, we consider the following combinations of instance sizes and parameters: $(|I|, |J|) =$

$\{(200, 150), (300, 100), (400, 150), (500, 100), (600, 150), (800, 100)\}$, $P = \{5, 10, 15, 20\}$ , $\delta = \{0.3, 0.5\}$. We have $6 \times 4 \times 2 = 48$ instances in this dataset.

**ORLIB**. Instances with 1000 customers that are taken from ORLib's facility location benchmark set, i.e., *capa*, *capb*, *capc*. They can be found at `http://people.brunel.ac.uk/~mastjjb/jeb/orlib/uncapinfo.html`. The service cost $c_{ij}$ and the demand $d_i$ are given. We compute $l_{ij} = c_{ij}/d_i$. Similar to RND instances, we generate $g_{ij}$ from $[(1-\delta)\cdot l_{ij}, (1+\delta)\cdot l_{ij}]$, where $0 \leq \delta < 1$.

**PMPUP**. A testbed for *p-Median Problem with Users Preferences* from benchmark library *Discrete Location Problems* (see `http://www.math.nsc.ru/AP/benchmarks/Bilevel/bilevel-eng.html`). We recode the dataset using Python. Files can be provided upon request. In total, there are 30 instances (i.e., *inst-333, inst-433, inst-533, ..., inst-3233*).

We now introduce notations that will be used in our later discussion: *CPU[s]* denotes the computational runtime in seconds. Throughout this section, the time limit for solving an instance is 7200 seconds. *t-gp[%]* stands for the relative exit gap in percentages upon termination, which is computed as |*zbb-zopt*|/|*zbb*| $\times$ 100, where *zopt* is the value of the current best solution and *zbb* is the best bound. An instance is solved optimally if *zbb=zop* or *t-gp* $< 0.01\%$; $\#N$ is the number of BC nodes explored by Gurobi. Finally, *r-gp[%]* is the root gap, which reflects the tightness of the relaxation bound right after processing the root node. It is computed as $|rrb - zopt|\,/rrb \times 100$, where $rrb$ is the root node relaxation bound.

*6.2. Results analysis on RND instances*

We start with the result analysis on the 48 RND instances. Table 2 provides the computational results. Among all approaches, `BC-BDTW` is the slowest one. For our proposed `BC-Benders`, the average computational time is 12.1 seconds under $\delta = 0.3$ and 58.8 seconds under $\delta = 0.5$. Obviously, it significantly outperforms `BC-BDTW` and `BC-WF`. Finally, `BC-BF` is slightly faster than `BC-WF`, with average runtimes 51.1 seconds versus 64.2 seconds under $\delta = 0.3$ and 527.1 seconds versus 546.3 seconds under $\delta = 0.5$.

We summarize the main results of the above table in Figure 1. The figure reports the boxplot of the computational time (in log-scale) for the above three approaches under two values of $\delta$. The line in the box indicates the median of the runtime under a specific value of $\delta$. Note that a larger $\delta$ value generally indicates that customer preferences differ more from the operator's unit service costs. The difficulty of the instances certainly depends on $\delta$, and intuitively, when $\delta = 0$, the model reduces to the standard facility location problem without user preferences, which can be much more efficiently solved. Thus, we expect that under a larger $\delta$ value, the instances will be more challenging. Apparently, the results in Figure 1 agree with our intuition as we observe that the instances require a longer runtime when $\delta = 0.5$.

*6.3. Results analysis on ORLIB instances*

Our next experiment is to investigate the performance of the approaches on the large-scale ORLIB structured data (*capa, capb, capc*).

Table 3 reports the results. We have the following observations: (i) Consistent with the previous finding, the runtime increases with $\delta$, i.e., when the user preferences differ more from the costs of the operator, the instances become harder. In particular, when $\delta = 0.5$ and $p = 10$, the *capc* instance is so challenging that `BC-BDTW` fails to solve it to optimality within the time limit; (ii) For

Table 2: Computational results of the RND instances.

| $\delta$ | $\lvert I\rvert - \lvert J\rvert$ | $p$ | BC-BDTW | | | | BC-WF | | | | BC-BF | | | | BC-Benders | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | CPU[s] | #N | r-gp[%] | $\phi$ | CPU[s] | #N | r-gp[%] | $\phi$ | CPU[s] | #N | r-gp[%] | $\phi$ | CPU[s] | #N | r-gp[%] | $\phi$ |
| 0.3 | 200-150 | 5 | 10.9 | 135 | 0.49 | 19883 | 11.5 | 50 | 0.49 | 19883 | 7.8 | 1 | 0.00 | 19883 | 5.3 | 1097 | 14.51 | 19883 |
| | | 10 | 7.2 | 1 | 0.00 | 13538 | 6.3 | 1 | 0.00 | 13538 | 4.4 | 1 | 0.00 | 13538 | 4.7 | 1504 | 18.23 | 13538 |
| | | 15 | 10.4 | 1 | 0.00 | 10701 | 5.6 | 1 | 0.00 | 10701 | 4.2 | 1 | 0.00 | 10701 | 4.2 | 958 | 3.78 | 10701 |
| | | 20 | 9.7 | 186 | 0.35 | 9215 | 18.1 | 14 | 0.14 | 9215 | 4.6 | 158 | 0.46 | 9215 | 4.5 | 1754 | 5.27 | 9215 |
| | 300-100 | 5 | 14.6 | 1 | 0.00 | 29993 | 14.7 | 53 | 0.57 | 29993 | 15.7 | 239 | 0.52 | 29993 | 5.3 | 970 | 14.13 | 29993 |
| | | 10 | 12.9 | 65 | 0.34 | 20404 | 11.0 | 22 | 0.30 | 20404 | 7.1 | 67 | 0.36 | 20404 | 4.9 | 1136 | 14.12 | 20404 |
| | | 15 | 6.6 | 1 | 0.00 | 16523 | 5.9 | 1 | 0.00 | 16523 | 9.7 | 109 | 0.51 | 16523 | 4.8 | 1085 | 12.98 | 16523 |
| | | 20 | 11.2 | 297 | 0.57 | 14479 | 8.3 | 1 | 0.00 | 14479 | 6.5 | 637 | 0.59 | 14479 | 4.9 | 2162 | 15.07 | 14479 |
| | 400-150 | 5 | 25.4 | 433 | 1.18 | 39161 | 31.3 | 129 | 1.02 | 39161 | 27.1 | 545 | 1.11 | 39161 | 10.5 | 2071 | 21.75 | 39161 |
| | | 10 | 51.3 | 1467 | 0.86 | 27180 | 43.3 | 288 | 0.97 | 27180 | 24.7 | 1350 | 1.05 | 27180 | 12.2 | 2959 | 16.52 | 27180 |
| | | 15 | 31.8 | 715 | 0.67 | 21666 | 40.4 | 1 | 0.00 | 21666 | 46.8 | 151 | 0.28 | 21666 | 12.0 | 3086 | 15.22 | 21666 |
| | | 20 | 33.8 | 82 | 0.25 | 18494 | 49.8 | 48 | 0.21 | 18494 | 20.4 | 176 | 0.24 | 18494 | 11.4 | 1788 | 15.92 | 18494 |
| | 500-100 | 5 | 74.2 | 148 | 0.75 | 49530 | 53.5 | 92 | 0.65 | 49530 | 34.5 | 125 | 0.82 | 49530 | 8.6 | 826 | 13.87 | 49530 |
| | | 10 | 37.3 | 1 | 0.00 | 33744 | 29.7 | 1 | 0.00 | 33744 | 36.8 | 1 | 0.00 | 33744 | 7.8 | 875 | 15.58 | 33744 |
| | | 15 | 24.6 | 1 | 0.00 | 27148 | 24.6 | 106 | 0.35 | 27148 | 12.1 | 215 | 0.49 | 27148 | 8.9 | 2073 | 14.91 | 27148 |
| | | 20 | 23.5 | 1033 | 0.39 | 23445 | 20.0 | 1 | 0.00 | 23445 | 13.5 | 1 | 0.00 | 23445 | 7.9 | 2735 | 13.51 | 23445 |
| | 600-150 | 5 | 165.8 | 3625 | 1.72 | 58678 | 125.2 | 600 | 1.71 | 58678 | 66.6 | 2400 | 1.77 | 58678 | 23.8 | 2493 | 21.98 | 58678 |
| | | 10 | 159.7 | 7658 | 1.41 | 40372 | 147.2 | 940 | 1.39 | 40372 | 178.6 | 5460 | 1.41 | 40372 | 28.2 | 8119 | 14.57 | 40372 |
| | | 15 | 166.8 | 1232 | 1.04 | 32156 | 63.4 | 455 | 1.03 | 32156 | 167.2 | 1417 | 1.08 | 32156 | 29.8 | 3685 | 17.22 | 32156 |
| | | 20 | 110.5 | 1069 | 0.86 | 27380 | 76.9 | 249 | 0.67 | 27380 | 118.6 | 936 | 0.87 | 27380 | 17.7 | 7273 | 13.77 | 27380 |
| | 800-100 | 5 | 70.4 | 67 | 0.36 | 77394 | 171.9 | 1 | 0.00 | 77394 | 74.2 | 21 | 0.33 | 77394 | 12.0 | 432 | 8.78 | 77394 |
| | | 10 | 98.3 | 939 | 0.92 | 54848 | 194.0 | 142 | 0.86 | 54848 | 103.1 | 1063 | 0.92 | 54848 | 17.0 | 1495 | 16.31 | 54848 |
| | | 15 | 373.2 | 1925 | 1.27 | 44534 | 267.1 | 416 | 0.67 | 44534 | 125.9 | 2589 | 1.30 | 44534 | 26.4 | 2224 | 17.71 | 44534 |
| | | 20 | 109.5 | 2149 | 0.97 | 38222 | 121.5 | 377 | 0.79 | 38222 | 115.1 | 2490 | 1.03 | 38222 | 16.8 | 5703 | 14.73 | 38222 |
| | | **Avg** | **68.3** | **968** | **0.60** | **31195** | **64.2** | **166** | **0.49** | **31195** | **51.1** | **840** | **0.63** | **31195** | **12.1** | **2438** | **14.60** | **31195** |
| 0.5 | 200-150 | 5 | 21.5 | 3848 | 3.13 | 20494 | 32.8 | 313 | 3.05 | 20494 | 18.2 | 1704 | 3.22 | 20494 | 7.3 | 2920 | 24.12 | 20494 |
| | | 10 | 53.7 | 3880 | 3.79 | 14120 | 44.8 | 2096 | 3.65 | 14120 | 64.4 | 3409 | 3.69 | 14120 | 13.8 | 7872 | 26.47 | 14120 |
| | | 15 | 56.7 | 2933 | 3.33 | 11208 | 32.8 | 1019 | 3.02 | 11208 | 23.1 | 7643 | 3.44 | 11208 | 12.2 | 5025 | 10.58 | 11208 |
| | | 20 | 38.2 | 1707 | 2.32 | 9536 | 23.4 | 531 | 2.87 | 9536 | 11.8 | 2869 | 2.75 | 9536 | 7.1 | 2085 | 9.88 | 9536 |
| | 300-100 | 5 | 24.6 | 2259 | 3.54 | 31004 | 32.5 | 443 | 3.49 | 31004 | 19.6 | 2013 | 3.55 | 31004 | 8.4 | 1430 | 18.79 | 31004 |
| | | 10 | 78.6 | 2001 | 3.07 | 21133 | 22.1 | 343 | 3.14 | 21133 | 21.2 | 4610 | 3.18 | 21133 | 8.0 | 3701 | 19.79 | 21133 |
| | | 15 | 108.2 | 2642 | 3.22 | 17225 | 37.7 | 596 | 3.03 | 17225 | 35.5 | 6077 | 3.24 | 17225 | 11.6 | 3296 | 23.12 | 17225 |
| | | 20 | 51.5 | 1808 | 2.23 | 15005 | 22.3 | 503 | 2.26 | 15005 | 45.0 | 1714 | 2.25 | 15005 | 7.5 | 2704 | 18.92 | 15005 |
| | 400-150 | 5 | 280.8 | 3422 | 3.82 | 40361 | 225.4 | 1818 | 3.75 | 40361 | 240.8 | 4980 | 3.89 | 40361 | 49.9 | 5037 | 24.75 | 40361 |
| | | 10 | 772.0 | 22251 | 4.01 | 28185 | 687.3 | 4400 | 3.84 | 28185 | 402.7 | 6709 | 4.00 | 28185 | 74.1 | 19937 | 22.79 | 28185 |
| | | 15 | 468.1 | 11211 | 4.25 | 22685 | 268.2 | 11447 | 4.52 | 22685 | 448.1 | 13802 | 4.42 | 22685 | 58.1 | 25040 | 22.82 | 22685 |
| | | 20 | 239.7 | 2820 | 3.03 | 19234 | 142.4 | 2621 | 2.94 | 19234 | 189.8 | 4096 | 3.19 | 19234 | 30.0 | 12335 | 21.68 | 19234 |
| | 500-100 | 5 | 206.7 | 3776 | 4.54 | 51628 | 110.5 | 1321 | 4.58 | 51628 | 62.1 | 3365 | 4.67 | 51628 | 19.4 | 3717 | 19.12 | 51628 |
| | | 10 | 217.3 | 2624 | 4.33 | 35360 | 107.5 | 1690 | 4.29 | 35360 | 210.7 | 3505 | 4.38 | 35360 | 39.6 | 6666 | 21.19 | 35360 |
| | | 15 | 143.3 | 3192 | 2.79 | 28220 | 70.2 | 496 | 2.57 | 28220 | 86.7 | 5382 | 2.86 | 28220 | 18.5 | 5611 | 22.95 | 28220 |
| | | 20 | 173.0 | 3461 | 2.61 | 24630 | 118.2 | 1788 | 2.33 | 24630 | 158.9 | 3058 | 2.68 | 24630 | 22.3 | 6584 | 19.75 | 24630 |
| | 600-150 | 5 | 1763.4 | 25944 | 5.19 | 60915 | 2158.2 | 10364 | 5.18 | 60915 | 1979.4 | 50923 | 5.22 | 60915 | 129.6 | 12645 | 29.68 | 60915 |
| | | 10 | 5859.8 | 151443 | 4.91 | 42018 | 2964.7 | 19695 | 4.98 | 42018 | 2456.4 | 55908 | 5.10 | 42018 | 317.4 | 57194 | 21.97 | 42018 |
| | | 15 | 3526.2 | 90586 | 4.35 | 33501 | 2578.8 | 39956 | 4.35 | 33501 | 2093.3 | 39974 | 4.29 | 33501 | 153.4 | 40570 | 24.53 | 33501 |
| | | 20 | 1995.0 | 30365 | 4.25 | 28639 | 1304.8 | 22527 | 4.09 | 28639 | 1160.4 | 23908 | 4.28 | 28639 | 124.5 | 31774 | 21.02 | 28639 |
| | 800-100 | 5 | 421.9 | 3245 | 4.34 | 80811 | 272.3 | 806 | 4.25 | 80811 | 192.1 | 2677 | 4.41 | 80811 | 33.2 | 2797 | 19.44 | 80811 |
| | | 10 | 835.0 | 10056 | 5.07 | 57385 | 515.1 | 3721 | 5.08 | 57385 | 778.0 | 14388 | 5.16 | 57385 | 99.9 | 13309 | 22.87 | 57385 |
| | | 15 | 937.5 | 12311 | 4.89 | 46577 | 817.7 | 7516 | 4.57 | 46577 | 1243.4 | 15155 | 5.01 | 46577 | 84.3 | 15400 | 24.33 | 46577 |
| | | 20 | 1060.8 | 13462 | 4.01 | 40018 | 522.6 | 6954 | 3.55 | 40018 | 709.4 | 9218 | 4.25 | 40018 | 81.2 | 22853 | 22.57 | 40018 |
| | | **Avg** | **805.6** | **17135** | **3.79** | **32496** | **546.3** | **5957** | **3.72** | **32496** | **527.1** | **11962** | **3.88** | **32496** | **58.8** | **12938** | **21.38** | **32496** |

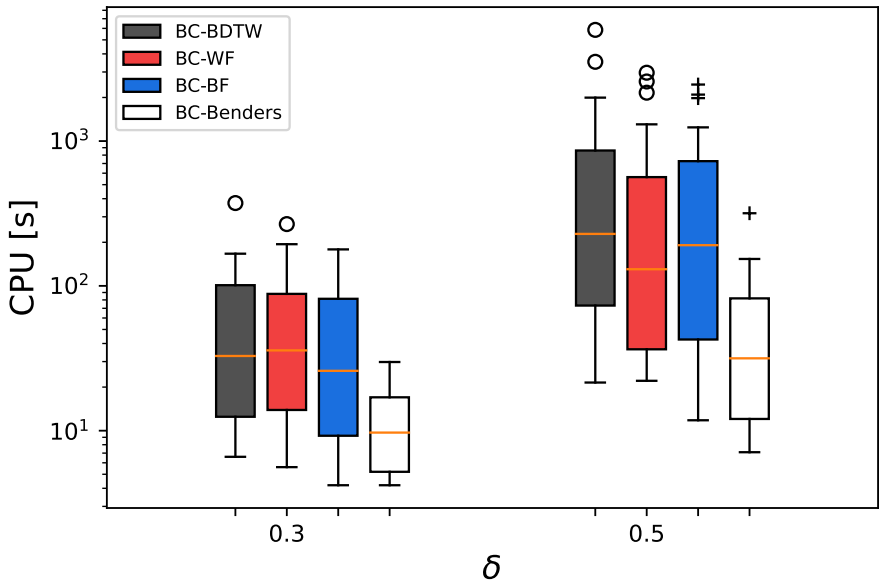



Figure 1: Boxplot of computational time in log-scale of RND instances.

all problem instances, `BC-Benders` is the fastest approach. In terms of computational time, `BC-Benders` is on average more than 10 times faster than `BC-WF`. (iii) Once again, we observe that `BC-BF`, on average, takes a slightly shorter runtime than `BC-WF`.

Based on the results of the two experiments, we conclude that our proposed `BC-Benders` indeed significantly outperforms the benchmark approaches. Moreover, our single-level based branch-and-cut approach `BC-BF` is comparably efficient when benchmarking it with `BC-WF`.

Table 3: Computational results of the ORLIB instances.

| inst. | $\delta$ | $p$ | BC-BDTW | | | | | BC-WF | | | | | BC-BF | | | | | BC-Benders | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | CPU[s] | t-gp[%] | #N | r-gp[%] | $\phi$ | CPU[s] | t-gp[%] | #N | r-gp[%] | $\phi$ | CPU[s] | t-gp[%] | #N | r-gp[%] | $\phi$ | CPU[s] | t-gp[%] | #N | r-gp[%] | $\phi$ |
| capa | 0.3 | 5 | 147.5 | 0 | 876 | 1.23 | 10252897 | 118.8 | 0 | 96 | 1.21 | 10252897 | 122.6 | 0 | 558 | 1.23 | 10252897 | 19.3 | 0 | 1147 | 14.64 | 10252897 |
| | 0.3 | 7 | 296.3 | 0 | 1307 | 1.13 | 8646381 | 397.4 | 0 | 361 | 1.02 | 8646381 | 323.4 | 0 | 1630 | 1.13 | 8646381 | 24.2 | 0 | 2407 | 15.29 | 8646381 |
| | 0.3 | 10 | 333.1 | 0 | 2670 | 1.45 | 7175161 | 249.9 | 0 | 501 | 1.40 | 7175161 | 261.2 | 0 | 1996 | 1.46 | 7175161 | 28.1 | 0 | 3626 | 19.46 | 7175161 |
| | 0.5 | 5 | 693.7 | 0 | 7911 | 4.82 | 10665607 | 702.8 | 0 | 2334 | 4.80 | 10665607 | 1065.4 | 0 | 9523 | 4.97 | 10665607 | 62.2 | 0 | 5034 | 23.78 | 10665607 |
| | 0.5 | 7 | 3434.7 | 0 | 45642 | 5.64 | 9079951 | 3183.8 | 0 | 12006 | 5.61 | 9079951 | 2395.5 | 0 | 40387 | 5.76 | 9079951 | 269.6 | 0 | 19989 | 23.45 | 9079951 |
| | 0.5 | 10 | 4111.5 | 0 | 48052 | 5.49 | 7502149 | 2071.3 | 0 | 10923 | 5.42 | 7502149 | 2944.6 | 0 | 34287 | 5.58 | 7502149 | 315.7 | 0 | 30960 | 28.68 | 7502149 |
| capb | 0.3 | 5 | 97.1 | 0 | 156 | 0.86 | 10324019 | 101.2 | 0 | 97 | 0.81 | 10324019 | 99.6 | 0 | 180 | 0.90 | 10324019 | 18.7 | 0 | 796 | 15.78 | 10324019 |
| | 0.3 | 7 | 264.6 | 0 | 6595 | 1.35 | 8707406 | 142.7 | 0 | 274 | 1.35 | 8707406 | 182.0 | 0 | 3785 | 1.35 | 8707406 | 20.9 | 0 | 1806 | 18.45 | 8707406 |
| | 0.3 | 10 | 231 | 0 | 2110 | 1.42 | 7162656 | 141.1 | 0 | 508 | 1.40 | 7162656 | 117.5 | 0 | 1743 | 1.48 | 7162656 | 24.7 | 0 | 4201 | 14.26 | 7162656 |
| | 0.5 | 5 | 1542.7 | 0 | 13308 | 5.25 | 10818027 | 1004.3 | 0 | 2702 | 5.20 | 10818027 | 1033.2 | 0 | 11680 | 5.31 | 10818027 | 147.1 | 0 | 6640 | 21.47 | 10818027 |
| | 0.5 | 7 | 2994.7 | 0 | 48402 | 5.50 | 9102576 | 2376.3 | 0 | 8274 | 5.40 | 9102576 | 2842.1 | 0 | 42091 | 5.54 | 9102576 | 219.1 | 0 | 11261 | 21.17 | 9102576 |
| | 0.5 | 10 | 2548.3 | 0 | 13174 | 4.95 | 7467031 | 1139.8 | 0 | 3903 | 4.93 | 7467031 | 1261.7 | 0 | 8972 | 5.13 | 7467031 | 174.7 | 0 | 16550 | 24.84 | 7467031 |
| capc | 0.3 | 5 | 248.1 | 0 | 1815 | 1.43 | 10230758 | 285.9 | 0 | 269 | 1.41 | 10230758 | 292.5 | 0 | 699 | 1.42 | 10230758 | 22.2 | 0 | 1202 | 17.43 | 10230758 |
| | 0.3 | 7 | 97.6 | 0 | 297 | 1.12 | 8544506 | 314.3 | 0 | 157 | 0.83 | 8544506 | 169.9 | 0 | 335 | 1.07 | 8544506 | 27.2 | 0 | 2363 | 15.79 | 8544506 |
| | 0.3 | 10 | 248.3 | 0 | 636 | 0.90 | 7018687 | 361.0 | 0 | 533 | 0.90 | 7018687 | 240.6 | 0 | 1036 | 0.96 | 7018687 | 27.4 | 0 | 3467 | 15.22 | 7018687 |
| | 0.5 | 5 | 3072.8 | 0 | 29321 | 5.54 | 10700077 | 1397.1 | 0 | 3553 | 5.58 | 10700077 | 2185.8 | 0 | 23834 | 5.63 | 10700077 | 253.7 | 0 | 12027 | 22.19 | 10700077 |
| | 0.5 | 7 | 1324.8 | 0 | 11960 | 4.99 | 8908130 | 2319.5 | 0 | 5108 | 4.94 | 8908130 | 1019.3 | 0 | 11512 | 5.03 | 8908130 | 170.4 | 0 | 11274 | 22.54 | 8908130 |
| | 0.5 | 10 | 7200 | 1.16 | 101683 | 6.30 | 7449553 | 7132.2 | 0 | 56561 | 6.31 | 7448788 | 4968.0 | 0 | 60958 | 6.37 | 7448788 | 430.4 | 0 | 56128 | 24.45 | 7448788 |
| | | **Avg** | **1604.8** | **0.06** | **18662** | **3.30** | **8875310** | **1302.2** | **0** | **6009** | **3.25** | **8875267** | **1195.8** | **0** | **14178** | **3.35** | **8875267** | **125.0** | **0** | **10604** | **19.94** | **8875267** |

*6.4. Managerial implication*

In our final experiment, we use the PMPUP dataset to investigate the importance of integrating user preferences into the facility location problem when preferences indeed exist. Specifically, we compare the total service costs of the operator (i.e., the objective function $\phi$) under two scenarios: (i) the operator considers user preferences and anticipates user's choices when locating facilities. This scenario is exactly the single-level models presented in Section 2; (ii) the operator ignores user preferences and locates facilities based on the cost matrix $c$. In this scenario, the operator's decision is based on the classical p-Median problem (see Chapter 6 of Daskin (2011)).

To facilitate our discussion, we introduce the following notations. Let $x_{wt}$ be the location decision when the operator locates facilities *without* considering user preferences. $x_{wt}$ can be obtained by setting $g = c$ in the model. The operator's *actual total service cost* $\phi_{wt}$ is evaluated by holding the location decision at $x_{wt}$. Therefore, $\phi_{wt}$ stands for the total service cost when the decision is made based on the assumption that user preferences do not exist (or are mistakenly assumed to coincide with the operator's service costs). Moreover, the *optimal total service cost* $\phi$ is the optimal objective function value when the operator indeed accounts for user preferences (i.e., the best objective). Then, we define the relative cost increase $\Delta[\%]$ as

$$\Delta = \frac{\phi_{wt} - \phi}{\phi} \times 100\% \tag{24}$$

which specifies the *relative additional cost incurred when user preferences exist but are ignored by the operator when locating facilities*.

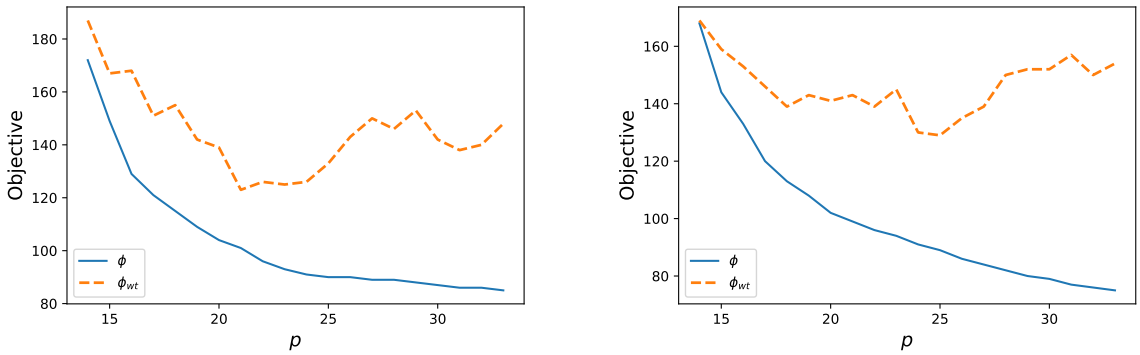
Our experiment is conducted on the PMPUP dataset since the original problem presented in the benchmark library has a similar structure to ours. Table 4 reports the operator's total service costs under three values of $p$. The first $p$ value is set to 14 since the original dataset imposes $p = 14$ for all instances (except for inst-533, marked with $*$ in the table, where $p = 13$). We observe that for the original instances (i.e., the column "$p = 14$"), the average $\Delta$ is 2.44%; therefore, ignoring user preferences will lead to an additional 2.44% of the average service cost. Furthermore, we observe that $\Delta$ increases dramatically when $p$ increases. In particular, the average $\Delta$ under $p = 30$ blows up to $84.24\%$. This result is astonishing as it indicates that the operator will almost double its service cost; therefore, the preferences must be considered if they exist.

Another interesting observation is that when user preferences are ignored, opening more facilities may lead to a higher service cost. For example, in inst-333, when the operator opens 20 facilities, $\phi_{wt}$ is 139; however, when 10 more facilities are open, $\phi_{wt}$ increases to 142. To give a direct view, we plot how $\phi$ and $\phi_{wt}$ evolve with $p$ in Figure 2. The figure shows $\phi_{wt}$ could increase when $p$ increases. On the contrary, we can clearly observe the decreasing trend of $\phi$, stating that when the preferences are considered in the operator's decision stage, opening more facilities will indeed reduce the service cost.

To summarize, the above experiment demonstrates that the operator must take user preferences into consideration and correctly anticipate user's choices; otherwise, the operator could suffer from a substantially higher cost, and opening more facilities could unexpectedly result in additional service costs as well. These results further justify the usefulness of the p-median problem with user preferences.

Table 4: Service costs with and without the consideration of user preferences ($\phi$ and $\phi_{wt}$).

| inst. | $p = 14$ | | | $p = 20$ | | | $p = 30$ | | |
|---|---|---|---|---|---|---|---|---|---|
| | $\phi_{wt}$ | $\phi$ | $\Delta$[%] | $\phi_{wt}$ | $\phi$ | $\Delta$[%] | $\phi_{wt}$ | $\phi$ | $\Delta$[%] |
| *333* | 187 | 172 | 8.72 | 139 | 104 | 33.65 | 142 | 87 | 63.22 |
| *433* | 156 | 156 | 0 | 129 | 100 | 29.00 | 134 | 71 | 88.73 |
| *533** | 188 | 188 | 0 | 131 | 98 | 33.67 | 133 | 73 | 82.19 |
| *633* | 172 | 165 | 4.24 | 135 | 102 | 32.35 | 130 | 77 | 68.83 |
| *733* | 160 | 159 | 0.63 | 114 | 91 | 25.27 | 135 | 64 | 110.94 |
| *833* | 181 | 170 | 6.47 | 142 | 108 | 31.48 | 118 | 81 | 45.68 |
| *933* | 160 | 160 | 0 | 146 | 102 | 43.14 | 142 | 77 | 84.42 |
| *1033* | 165 | 159 | 3.77 | 121 | 93 | 30.11 | 128 | 69 | 85.51 |
| *1133* | 174 | 163 | 6.75 | 115 | 98 | 17.35 | 123 | 85 | 44.71 |
| *1233* | 167 | 163 | 2.45 | 111 | 89 | 24.72 | 109 | 63 | 73.02 |
| *1333* | 169 | 168 | 0.60 | 141 | 102 | 38.24 | 152 | 79 | 92.41 |
| *1433* | 176 | 172 | 2.33 | 138 | 102 | 35.29 | 132 | 81 | 62.96 |
| *1533* | 152 | 152 | 0 | 115 | 90 | 27.78 | 108 | 64 | 68.75 |
| *1633* | 156 | 156 | 0 | 127 | 93 | 36.56 | 134 | 72 | 86.11 |
| *1733* | 160 | 152 | 5.26 | 138 | 94 | 46.81 | 128 | 73 | 75.34 |
| *1833* | 154 | 154 | 0 | 139 | 91 | 52.75 | 157 | 63 | 149.21 |
| *1933* | 160 | 158 | 1.27 | 127 | 92 | 38.04 | 127 | 69 | 84.06 |
| *2033* | 161 | 161 | 0 | 109 | 96 | 13.54 | 144 | 75 | 92.00 |
| *2133* | 176 | 166 | 6.02 | 138 | 108 | 27.78 | 141 | 83 | 69.88 |
| *2233* | 155 | 154 | 0.65 | 110 | 91 | 20.88 | 155 | 75 | 106.67 |
| *2333* | 160 | 155 | 3.23 | 144 | 98 | 46.94 | 153 | 76 | 101.32 |
| *2433* | 159 | 155 | 2.58 | 142 | 103 | 37.86 | 155 | 81 | 91.36 |
| *2533* | 147 | 147 | 0 | 136 | 91 | 49.45 | 119 | 68 | 75.00 |
| *2633* | 156 | 156 | 0 | 113 | 94 | 20.21 | 146 | 73 | 100.00 |
| *2733* | 173 | 159 | 8.81 | 119 | 102 | 16.67 | 133 | 76 | 75.00 |
| *2833* | 164 | 161 | 1.86 | 132 | 95 | 38.95 | 115 | 73 | 57.53 |
| *2933* | 152 | 152 | 0 | 117 | 91 | 28.57 | 119 | 65 | 83.08 |
| *3033* | 162 | 157 | 3.18 | 120 | 93 | 29.03 | 105 | 62 | 69.35 |
| *3133* | 155 | 155 | 0 | 109 | 87 | 25.29 | 165 | 63 | 161.90 |
| *3233* | 162 | 155 | 4.52 | 128 | 97 | 31.96 | 137 | 77 | 77.92 |
| **Avg** | **164.0** | **159.1** | **2.24** | **127.5** | **96.5** | **32.1** | **134.0** | **73.2** | **84.24** |




(a) Inst-333 (b) Inst-1333

Figure 2: Service costs with and without the consideration of user preferences versus the number of open facilities.

## 7. Conclusion

This paper studied the exact solution approach for the p-median facility location problem with user preferences (PUP). By exploring the problem structure, we proposed two branch-and-cut algorithms. One explores the bilevel structure of the model and leverages a specialized bilevel feasibility cut to progressively recover the bilevel feasibility and update the incumbent, whereas the other is based on Benders decomposition, where Benders separation at integer nodes of the searching tree was accelerated using analytical formulas, and at fractional nodes, an effective heuristic separation was implemented to further enhance the algorithm performance. Using a broad testbed, our computational experiments demonstrated that our proposed Benders approach significantly outperforms the benchmark approaches. Moreover, our branch-and-cut approach based the bilevel feasibility cut is comparably efficient when comparing it with the best benchmark. We also conducted sensitivity analysis and observed that when user preferences indeed exist, the operator must consider customer preferences and correctly anticipate the choices to avoid an unnecessarily high service provision cost. Furthermore, ignoring the preferences may lead to an ironic situation where opening more facilities could result in additional service costs.

There are potential future research directions. Since the operator needs to forecast customer preferences to anticipate the choices, there exists the possibility that the forecasted preference is subject to estimation errors. In this case, we should build a stochastic model or/and a robust optimization model to consider the preference uncertainty of customers. Moreover, we assume fully rationale behaviors; however, this might not be the case in some scenarios because customers may only be bounded rationale, and it is worth investigating the impact of bounded rationale behaviors on the location planning. We would like to leave the model formulation and the algorithm development for future research.

As a final remark, we briefly discuss the generality of the solution approaches employed in this paper. In the PUP, it is assumed that each customer will select or be assigned to exactly one facility. This assumption is mathematically represented by (2e) , where the right-hand side equals one to signify a unique choice. The single-level reformulation approaches based closest assignment constraints and the subsequent Benders decomposition effectively leverage this assumption. However, when the right-hand side of the equation is different from one (e.g., two, implying that each customer is assigned to two facilities), these approaches lose their applicability, as they are tailored specifically for addressing the unique assignment assumption. By contrast, the bilevel feasibility cut, derived by exploring the bilevel structure, retains its applicability as the bilevel feasibility condition outlined in Lemma 1 is independent of the number of facilities to which a customer is assigned. Therefore, the bilevel approach exhibits greater generality, despite that it is slower than the Benders decomposition when they are both applied to the customized PUP model.

## Appendix A. Primal-dual reformulation model

This appendix presents a single-level reformulation model for [PUP], which relies on the primal-dual optimality conditions of the lower-level problem. Similar approaches can be found in Casas-Ramírez and Camacho-Vallejo (2017), Casas-Ramírez et al. (2018). Here, we briefly describe the reformulation procedure. Given the operator's location decision $x$, the lower-level problem is

$$\min_{y \in \mathbb{R}_+} \sum_{i \in I} \sum_{j \in J} \pi_{ij} y_{ij} \tag{A.1a}$$

$$\text{st.} \sum_{j \in J} y_{ij} = 1 \quad \forall i \in I \qquad (\alpha_i) \tag{A.1b}$$

$$y_{ij} \leq x_j \quad \forall i \in I, j \in J \qquad (\beta_{ij}) \tag{A.1c}$$

The $\alpha_i$ and $\beta_{ij}$ in the parentheses are the dual variables associated with the constraints. We can then rewrite the lower-level problem with its KKT conditions, i.e.,

$$\sum_{j \in J} y_{ij} = 1 \qquad \forall i \in I \tag{A.2a}$$

$$y_{ij} \leq x_j \qquad \forall i \in I, j \in J \tag{A.2b}$$

$$\alpha_i - \beta_{ij} \leq \pi_{ij} \qquad \forall i \in I, j \in J \tag{A.2c}$$

$$y_{ij}(\alpha_i - \beta_{ij} - \pi_{ij}) = 0 \qquad \forall i \in I, j \in J \tag{A.2d}$$

$$\beta_{ij}(y_{ij} - x_j) = 0 \qquad \forall i \in I, j \in J \tag{A.2e}$$

$$\beta_{ij} \leq 0 \qquad \forall i \in I, j \in J \tag{A.2f}$$

where (A.2d) and (A.2e) are bilinear functions. Noting both $y_{ij}$ and $x_j$ will take $0/1$ in the optimal solution and the maximum value of the matrix $\pi$ is $n = |J|$ by definition, these bilinear functions can exactly linearized by

$$\alpha_i - \beta_{ij} - \pi_{ij} \geq n(y_{ij} - 1) \qquad \forall i \in I, j \in J \tag{A.3a}$$

$$\beta_{ij} \leq n(y_{ij} - x_j + 1) \qquad \forall i \in I, j \in J \tag{A.3b}$$

It is worth noting that, when solving the reformulated model by modern MILP solvers, setting $y_{ij}$ as a binary variable will improve the continuous relaxation and enhance the numerical stability.

## Appendix B. Branch-and-cut enhancement for single-level reformulation models

In this appendix, we demonstrate the effectiveness of branch-and-cut approaches for solving single-level reformulation models. Specifically, for both `SRM-BDTW` and `SRM-WF` , we implemented the associated branch-and-cut approaches (referred to as `BC-BDTW` and `BC-BDTW` ).

Table B.5 shows the computational result on the ORLIB instances. It turns out that the branch-and-cut approaches can largely improve the computational efficiency, as they reduce the overall computational time by a large margin.

Table B.5: ORLIB instances. Comparison of (i) directly solving the single-level reformulation models (i.e., SRM−BDTW and SRM−WF) with (ii) solving them within branch-and-cut framework (i.e., BC−BDTW and BC−WF).

| inst. | $\delta$ | $p$ | SRM−BDTW | | | BC−BDTW | | | SRM−WF | | | BC−WF | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | CPU[s] | t-gp[%] | #N | CPU[s] | t-gp[%] | #N | CPU[s] | t-gp[%] | #N | CPU[s] | t-gp[%] | #N |
| capa | 0.3 | 5 | 1783.5 | 0 | 41 | 147.5 | 0 | 876 | 1598.7 | 0 | 51 | 118.8 | 0 | 96 |
| | 0.3 | 7 | 2040.4 | 0 | 101 | 296.3 | 0 | 1307 | 1975.4 | 0 | 34 | 397.4 | 0 | 361 |
| | 0.3 | 10 | 2428.3 | 0 | 85 | 333.1 | 0 | 2670 | 1474.5 | 0 | 25 | 249.9 | 0 | 501 |
| | 0.5 | 5 | 4789.8 | 0 | 1834 | 693.7 | 0 | 7911 | 5238.5 | 0 | 1152 | 702.8 | 0 | 2334 |
| | 0.5 | 7 | 5050.9 | 0 | 4493 | 3434.7 | 0 | 45642 | 4495.5 | 0 | 2659 | 3183.8 | 0 | 12006 |
| | 0.5 | 10 | 7200.0 | 2.31 | 1827 | 4111.5 | 0 | 48052 | 4108.1 | 0 | 3969 | 2071.3 | 0 | 10923 |
| capb | 0.3 | 5 | 3139.7 | 0 | 1 | 97.1 | 0 | 156 | 1531.5 | 0 | 1 | 101.2 | 0 | 97 |
| | 0.3 | 7 | 1692.2 | 0 | 160 | 264.6 | 0 | 6595 | 2778.3 | 0 | 77 | 142.7 | 0 | 274 |
| | 0.3 | 10 | 2180.5 | 0 | 180 | 231 | 0 | 2110 | 1448.3 | 0 | 11 | 141.1 | 0 | 508 |
| | 0.5 | 5 | 4863.6 | 0 | 1175 | 1542.7 | 0 | 13308 | 5803.9 | 0 | 1478 | 1004.3 | 0 | 2702 |
| | 0.5 | 7 | 4663.9 | 0 | 3801 | 2994.7 | 0 | 48402 | 4524.1 | 0 | 3207 | 2376.3 | 0 | 8274 |
| | 0.5 | 10 | 3998.6 | 0 | 4676 | 2548.3 | 0 | 13174 | 2482.3 | 0 | 1848 | 1139.8 | 0 | 3903 |
| capc | 0.3 | 5 | 2878.7 | 0 | 95 | 248.1 | 0 | 1815 | 2780.6 | 0 | 45 | 285.9 | 0 | 269 |
| | 0.3 | 7 | 2058.1 | 0 | 1 | 97.6 | 0 | 297 | 3276.2 | 0 | 10 | 314.3 | 0 | 157 |
| | 0.3 | 10 | 1625.8 | 0 | 1 | 248.3 | 0 | 636 | 802.1 | 0 | 1 | 361.0 | 0 | 533 |
| | 0.5 | 5 | 7077.2 | 0 | 3996 | 3072.8 | 0 | 29321 | 7200.0 | 4.27 | 998 | 1397.1 | 0 | 3553 |
| | 0.5 | 7 | 5223.2 | 0 | 1437 | 1324.8 | 0 | 11960 | 4623.5 | 0 | 2023 | 2319.5 | 0 | 5108 |
| | 0.5 | 10 | 7200.0 | 2.89 | 1760 | 7200 | 1.16 | 101683 | 5539.4 | 0 | 7772 | 7132.2 | 0 | 56561 |
| | | **Avg** | **3883.0** | **0.29** | **1426** | **1604.8** | **0.06** | **18662** | **3426.7** | **0.24** | **1409** | **1302.2** | **0.00** | **6009** |

## References


Cabezas, X. and García, S. (2018). A lagrangean relaxation algorithm for the simple plant location problem with preferences. *arXiv preprint arXiv:1805.03945*.

Cabezas, X., García, S., Martin-Barreiro, C., Delgado, E., and Leiva, V. (2021). A two-stage location problem with order solved using a lagrangian algorithm and stochastic programming for a potential use in covid-19 vaccination based on sensor-related data. *Sensors*, 21(16):5352.

Calvete, H. I., Galé, C., Iranzo, J. A., Camacho-Vallejo, J.-F., and Casas-Ramírez, M.-S. (2020). A matheuristic for solving the bilevel approach of the facility location problem with cardinality constraints and preferences. *Computers & Operations Research*, 124:105066.

Camacho-Vallejo, J.-F., Casas-Ramírez, M., and Miranda, P. (2014a). The p-median bilevel problem under preferences of the customers. *Recent Advances in Theory, Methods and Practice of Operations Research*, pages 121–127.

Camacho-Vallejo, J.-F., Cordero-Franco, Á. E., and González-Ramírez, R. G. (2014b). Solving the bilevel facility location problem under preferences by a stackelberg-evolutionary algorithm. *Mathematical Problems in Engineering*, 2014.

Cánovas, L., García, S., Labbé, M., and Marín, A. (2007). A strengthened formulation for the simple plant location problem with order. *Operations Research Letters*, 35(2):141–150.

Cao, D. and Chen, M. (2006). Capacitated plant selection in a decentralized manufacturing environment: a bilevel optimization approach. *European Journal of Operational Research*, 169(1):97–110.

Casas-Ramírez, M.-S. and Camacho-Vallejo, J.-F. (2017). Solving the p-median bilevel problem with order through a hybrid heuristic. *Applied Soft Computing*, 60:73–86.

Casas-Ramírez, M.-S., Camacho-Vallejo, J.-F., and Martínez-Salazar, I.-A. (2018). Approximating solutions to a bilevel capacitated facility location problem with customer's patronization toward a list of preferences. *Applied Mathematics and Computation*, 319:369–386.

Daskin, M. S. (2011). *Network and discrete location: models, algorithms, and applications*. John Wiley & Sons.

Díaz, J. A., Luna, D. E., Camacho-Vallejo, J.-F., and Casas-Ramírez, M.-S. (2017). Grasp and hybrid grasp-tabu heuristics to solve a maximal covering location problem with customer preference ordering. *Expert Systems with Applications*, 82:67–76.

Espejo, I., Marín, A., and Rodríguez-Chía, A. M. (2012). Closest assignment constraints in discrete location problems. *European Journal of Operational Research*, 219(1):49–58.

Fischetti, M., Ljubić, I., and Sinnl, M. (2016). Benders decomposition without separability: A computational study for capacitated facility location problems. *European Journal of Operational Research*, 253(3):557–569.

Hanjoul, P. and Peeters, D. (1987). A facility location problem with clients' preference orderings. *Regional Science and Urban Economics*, 17(3):451–473.

Hansen, P., Kochetov, Y., and Mladenovic, N. (2004). *Lower bounds for the uncapacitated facility location problem with user preferences*. Groupe d'études et de recherche en analyse des décisions, HEC Montréal.

Lee, J. M. and Lee, Y. H. (2012). Facility location and scale decision problem with customer preference. *Computers & Industrial Engineering*, 63(1):184–191.

Lin, Y. H. and Tian, Q. (2021). Branch-and-cut approach based on generalized benders decomposition for facility location with limited choice rule. *European Journal of Operational Research*, 293(1):109–119.

Lin, Y. H. and Tian, Q. (2023). Facility location and pricing problem: Discretized mill price and exact algorithms. *European Journal of Operational Research*, 308(2):568–580.

Lotfi, R., Mardani, N., and Weber, G.-W. (2021). Robust bi-level programming for renewable energy location. *International Journal of Energy Research*, 45(5):7521–7534.

Marić, M., Stanimirović, Z., and Milenković, N. (2012). Metaheuristic methods for solving the bilevel uncapacitated facility location problem with clients' preferences. *Electronic Notes in Discrete Mathematics*, 39:43–50.

Melo, M. T., Nickel, S., and Saldanha-Da-Gama, F. (2009). Facility location and supply chain management–a review. *European Journal of Operational Research*, 196(2):401–412.

Mrkela, L. and Stanimirović, Z. (2021). A variable neighborhood search for the budget-constrained maximal covering location problem with customer preference ordering. *Operational Research*, pages 1–39.

Taherkhani, G., Alumur, S. A., and Hosseini, M. (2020). Benders decomposition for the profit maximizing capacitated hub location problem with multiple demand classes. *Transportation Science*, 54(6):1446–1470.

Vasilev, I., Klimentova, K., and Kochetov, Y. A. (2009). New lower bounds for the facility location problem with clients' preferences. *Computational Mathematics and Mathematical Physics*, 49(6):1010–1020.

Vasilyev, I. and Klimentova, K. (2010). The branch and cut method for the facility location problem with client's preferences. *Journal of Applied and Industrial Mathematics*, 4(3):441–454.