Unified framework for choice-based facility location problem

Yun Hui Lin
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, liny@ihpc.a-star.edu.sg

Qingyun Tian
School of Civil and Environmental Engineering, Nanyang Technological University, qytian@ntu.edu.sg

Yanlu Zhao
Durham University Business School, Durham University, yanlu.zhao@durham.ac.uk

Abstract: Choice-based facility location (CBFL) problem arises in various industrial and business contexts. The problem stands on a decentralized perspective: companies set up chains of facilities, and customers determine from which chain or facility to seek service according to their own preferences. Essentially, customer preferences or choices play a key role in characterizing various CBFL problems, which differ themselves mainly in the models or rules employed to characterize the choice. Consequently, a large number of formulations appear and are oftentimes solved by dedicatedly designed approaches in the literature. Such a situation significantly complicates practitioners' decision-making process when they are facing practical problems but are unsure which ad-hoc model is suitable for their cases. In this article, we address this dilemma by providing a unified modeling framework based on the concept of preference dominance. Specifically, we conceptualize the choice behavior as a sequential two-step procedure: Given a set of open facilities, each customer first forms a non-dominated consideration set and then splits the buying power within the set. Such an interpretation renders practitioners high modeling flexibility as they can tailor how preference dominance is constructed according to their specific contexts. In particular, we show that our model can represent several streams of CBFL problems. To support our model's applicability, we design an efficient exact decomposition algorithm. Extensive computational studies reveal that although the algorithm is designed for a general purpose, it outperforms most approaches that are tailored for ad-hoc problems by a large margin, which justifies both the effectiveness and the efficiency of the unified framework.

Keywords: Choice-based facility location, Consideration set, Preference dominance, Decomposition algorithm

1 Introduction

Choice-Based Facility Location (CBFL) problem is a rapidly-evolving research area that investigates the strategic placement of service facilities by one or more companies, with the aim of enhancing market competitiveness and achieving strategic objectives. The problem is approached from a fully *decentralized* perspective, in which individual customers make their own decisions regarding which chain or facility to visit, based on their unique preferences. In this regard, companies are unable to dictate customer allocation to facilities in a centralized manner, but instead must rely on designing effective service profiles, such as the location and design of facilities, to attract and retain customers. The customer patronage behavior associated with CBFL is typically characterized by well-established discrete choice models or rules, such as the multinomial logit model and the gravity rule. Notably, CBFL is analogous to the competitive facility location problem, wherein both arise in a range of industrial and business contexts, including park-and-ride station planning (Aros-Vera et al. 2013), locker deployment for last-mile delivery (Lin et al. 2022a), preventive healthcare network design (Zhang et al. 2012, Krohn et al. 2021),

and retailer store location (Méndez-Vogel et al. 2022), among others. Departing from these applicable contexts, CBFL is more general than the competitive facility location problem as it could characterize various problems with market competition absence. For example, the p-median problem with user preferences also belongs to CBFL since it concerns a spatial monopoly that locates facilities to minimize its service provision costs, anticipating how customers choose facilities to seek service (Casas-Ramírez and Camacho-Vallejo 2017).

Given the number of companies under investigation, CBFL studies can be broadly classified into two streams. The first stream focuses on a *single* company, which is either a monopoly in the market or operates alongside other companies that do not change their service profiles during the planning horizon. Accordingly, the company does not face company-level interactive decision-making situations, and its main concern is how customers will react to the services it provides. This stream of research has been studied extensively in previous works (Drezner et al. 2018, Dan and Marcotte 2019, Lin et al. 2022a). By contrast, the second stream of CBFL research includes *multiple* companies, and each company is assumed to maximize its own objective. Therefore, there will be interactive decisions at the company level. The problem is characterized from a game-theoretical perspective, which leads to two modeling paradigms: Nash game (Gur et al. 2018) and Stackelberg game (Drezner et al. 2015). We suggest interested readers referring to Mallozzi and Daskin (2017) for comprehensive investigation, which provides an in-depth analysis of the problem, including the mathematical models, solution methods, and applications of CBFL in various industries.

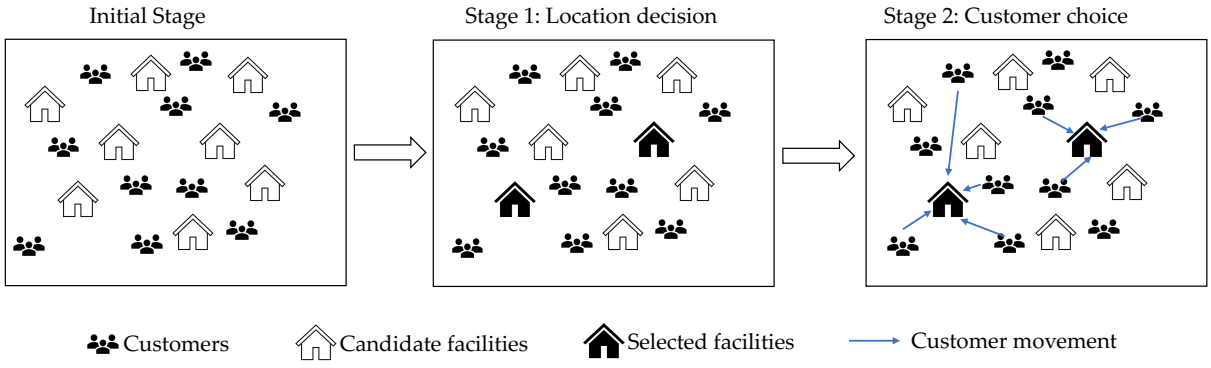


Figure 1 An example of the choice-based facility location problem decision procedure.

This article focuses on the single-company case. We briefly describe the CBFL decision procedure in Figure 1. Specifically, in the initial stage, customers and candidate facilities are known at discrete sites. In Stage 1, the company selects a subset of facilities to open and offer services. In the following Stage 2, each customer then determines from which facility to seek service. The objective of each company is to achieve a specific goal, such as maximum market share, maximum net profit from serving customers, or minimum service provision cost, while taking into account the choices of customers.

Essentially, the customer choice behavior plays a pivotal role in distinguishing different CBFL problems. The variations in these problems often stem from the models employed to characterize customer

choices in Stage 2. Currently, the most commonly used models in academia and industry are the multinomial logit model (MNL) and the Huff-based gravity model. Both models operate on a "proportional choice" principle, where the likelihood of a customer visiting a facility is directly proportional to the facility's attractiveness or utility (Aboolian et al. 2007, Aros-Vera et al. 2013, Ljubić and Moreno 2018, Mai and Lodi 2020). Typically, these problems are formulated as mixed-integer nonlinear programs and solved using outer-approximation based algorithms or mixed-integer linear program (MILP) approaches that leverage the inherent structural properties. Other notable models include the "binary choice" rule (BCR) and the "partially binary choice" rule (PBCR). The BCR assumes that each customer is solely attracted to the option with the highest utility, representing an *all-or-nothing* choice mode (Beresnev 2013, Fernández et al. 2017, Lančinskas et al. 2020). Employing such a choice rule in CBFL problems often leads to bilevel formulations, which require advanced solution approaches. The PBCR, akin to the binomial logit model, assumes that customers only consider the most attractive facility from the company. They may also choose not to seek service from this facility and instead opt out of the system altogether (Biesinger et al. 2016, Fernández et al. 2017). In recent years, more sophisticated models have emerged. One example is the threshold Luce model (TLM) introduced by Lin et al. (2022a), which permits effective conic reformulation approaches. Another notable model is the Pareto-Huff model (PHM) proposed by Fernández et al. (2018), which is solved by tailored MILP approaches or efficient heuristics. These advanced models strengthen the capabilities for addressing CBFL problems and expanding the range of potential solution techniques.

As we review above, the existing research on CBFL problems has primarily focused on utilizing ad-hoc models and solution approaches that leverage the specific characteristics of the problem at hand. However, these dedicated studies pose challenges when it comes to comparing and transferring existing models. Furthermore, the inherent complexity arises from the fact that decision makers are typically unaware of how customers make choices, and it has been demonstrated that employing different choice models can lead to substantially different solutions (Suárez-Vega et al. 2004, Fernández et al. 2018, Lin et al. 2022a). Consequently, the vagueness surrounding customer choices can overwhelm tailored solution approaches, rendering them ineffective when applied to other business contexts. Developing a comprehensive modeling and solution framework, derived from thorough surveys, thus becomes invaluable. Such a framework would provide valuable insights and guidance for researchers and practitioners working on CBFL, enabling them to overcome the challenges posed by diverse choice models and effectively adapt solutions to various intricate business scenarios.

To this end, we propose a unified modeling prescription in this article that can represent a wide range of problems, including but not limited to proportional choice rule (i.e., MNL), BCR, PBCR, TLM, PHM, and even nested logit models (NLM). In other words, these models are special outcomes of the unified modeling prescription. To effectively solve these problems for practical usage, we also develop a powerful exact solution approach that outperforms existing algorithms in several aspects.

1.1 Related literature

In this section, we aim to position our contributions in the literature by briefly reviewing relevant studies on single-company CBFL ad-hoc models and their associated solution approaches. To provide contextual structure, we adopt the classification scheme proposed by Suárez-Vega et al. (2004) and Biesinger et al. (2016), which categorizes studies based on demand models. Within this framework, we identify two main streams of CBFL research: those concentrating on *essential* demand and those focusing on *unessential* demand.

Within the context of essential demand, customers are certain to use the services of the company, meaning that demand loss can be completely avoided. After the facilities are opened, customers determine which facility to seek services from based on the BCR, resulting in the binary essential model according to Biesinger et al. (2016). Subsequently, the company serves customers based on their individual choices, which can either generate profits or incur service provision costs. One classical problem in this regard is the facility location (p-median) problem with user preferences, which is often formulated as a bilevel integer linear program (Camacho-Vallejo et al. 2014, Casas-Ramírez and Camacho-Vallejo 2017). Prevailing solution approaches are developed relying on single-level MILP reformulation. For instance, one can leverage the Karush-Kuhn-Tucker (KKT) conditions of the lower-level problem to derive an equivalent formulation with complementarity constraints. The non-convex complementarity constraints can then be linearized, resulting in a MILP that can be directly solved by off-the-shelf solvers (Cao and Chen 2006, Casas-Ramírez and Camacho-Vallejo 2017). Alternatively, the *closest assignment constraints* (CACs) can be utilized to subtly reformulate the lower-level problem into linear conditions without introducing dual variables. For a comprehensive discussion on the CACs, we refer readers to Espejo et al. (2012). It turns out that a general constraint in our modeling prescription exactly recovers an efficient type of CAC under essential demand.

Unessential demand refers to situations where customers may decide not to use the company's service because there is at least one competitor in the market. In these studies, competition is usually aggregated into a single "opt-out" option, and customer choices are interpreted probabilistically given discrete choice models/rules, leading to mathematical formulations with multi-ratio fractional objective functions. In this aspect, comparative summaries across different models are available in Suárez-Vega et al. (2004) and Méndez-Vogel et al. (2022). Among others, the MNL and Huff models are deemed as the most popular ones. On one hand, the literature employing the MNL is usually referred to as the maximum capture problem with random utilities, in which a company aims to maximize its market share through opening a fixed number of facilities (Benati and Hansen 2002). As the market share is a multi-ratio fractional linear 0-1 function, a standard solution approach is to reformulate the original problem into a MILP (Borrero et al. 2017). For example, several equivalent MILP formulations are presented for discussion in Haase and Müller (2014). However, the MILP complexity suffers from scalability issues for large-scale instances, rendering it unappealing as problem size expands. Fortunately, the market share under the MNL exhibits both submodularity and concavity with respect to the location variable (Benati and Hansen 2002), allowing for branch-and-cut algorithms based on outer-approximation

cuts and submodular cuts to achieve better computational performance on large-scale instances (Ljubić and Moreno 2018) . On the other hand, in the Huff model, the utility of a facility provided for a customer is a function of the instrinct facility attraction and the distance between them, with facility attraction assumed to depend on facility design. Many related problems thus consider both the location and the design decision of facilities. For example, Aboolian et al. (2007) assumed that each facility has a finite number of available design options, and only one option will be selected for each open facility. The problem is solved by an approximation approach (i.e., Tangent Line Approximation), which yields near-optimal solutions subject to adjustable errors. Recently, Lin and Tian (2021b) investigated a variant with facility-customer pairwise attraction, where an iterative Benders decomposition method was proposed and shown to significantly outperform both the Tangent Line Approximation under reasonable error bounds and the branch-and-cut out-approximation algorithm in Ljubić and Moreno (2018). Apart from the proportional choice rule, the PBCR also received considerable attention (Suárez-Vega et al. 2004, Biesinger et al. 2016, Fernández et al. 2017). However, exact solution approaches with PBCR are rather limited. To the best of our knowledge, all available exact approaches for the PBCR model are based on the MILP reformulation techniques (Fernández et al. 2017, Méndez-Vogel et al. 2022).

In fact, more and more efforts have been invested to generalize choice modeling in the CBFL literature, building on recent progress in discrete choice theory about dominance and consideration sets (Ahumada and Ülkü 2018, Echenique and Saito 2019). For example, based on the deterministic dominance in Echenique and Saito (2019), the TLM allows for flexibly restricting the size of the consideration set. Lin et al. (2022a) showed that both the proportional choice rule and PBCR are indeed special cases of the TLM, which is solved via a mixed-integer conic quadratic programming (MICQP) approach. Another example is the Pareto-Huff model, which assumes that customers are only willing to patronize facilities that are Pareto-optimal to both facility attraction and distance. This model is rather challenging and initially solved using a ranking-based heuristic (Fernández et al. 2018). In a very recent work, a specialized linearization scheme was developed to recast this model into a MILP (Fernández et al. 2021), which is considered the only exact approach in this subarea. However, such an approach has limited computational capability, which signalizes the need for developing more efficient exact algorithms.

As discussed earlier, the utilization of different choice models/rules often leads to distinct ad-hoc formulations, which, in turn, necessitate carefully tailored solution approaches for specific scenarios. Unfortunately, this restricts the general applicability of these approaches and complicates their implementation for practitioners. The lack of clarity regarding which choice model to employ further exacerbates the dilemma. To tackle this challenge, we propose a unified framework that can comprehensively capture the diverse characteristics of CBFL problems. Rather than solving each type of CBFL problem in an ad-hoc manner, our approach is derived from a single underlying unified framework. This framework allows for the systematic incorporation of various choice models and facilitates a more holistic understanding of the problem.

1.2 Our contributions: Unified framework for CBFL problem

In this article, our main contribution focuses on the development of a unified framework to model and solve the classical CBFL problem and its variants. Specifically, the unified framework consists of two key components:

(i) A unified modeling prescription. We first develop a unified modeling prescription, which generalizes the dominance concept in Echenique and Saito (2019) with our novel definitions of preference dominance types and valid conditions. Specifically, we conceptualize the choice behavior as a sequential two-step procedure: Given a set of open facilities, each customer first forms a *consideration set* that consists of all non-dominated open facilities. Subsequently, customers split their buying powers/demand among facilities inside the set (possibly also plus an outside option). Such a new interpretation renders users of the prescription high modeling flexibility as they can configure how preference dominance is constructed and how the consideration set is formed according to a specific problem setting. To this end, technically, we propose a *dominance inequality* to model the consideration set and derive a *directed path inequality* to strengthen the formulation by characterizing the dominance relation as a directed acyclic graph. We then demonstrate that two broad ranges of CBFL (i.e., essential and unessential demand) can be representable by the unified modeling prescription. Additionally, the prescription can be easily adapted to a new third class of problem where facilities are subject to "similarity", which does not obey the proportional rule on the consideration set, and instead, resembles the use of the nested logit model in a broader context. In fact, the unified modeling prescription is a sharp technique to characterize various existing CBFL problems, not limited to the aforementioned types.

(ii) A unified solution approach. To support the application of unified modeling prescription in decision-making contexts, we further design an exact decomposition algorithm, which projects out the decision variables related to the consideration set and maintains only the location variables during the searching process. The main algorithm includes an iterative procedure to generate initial cutting planes and a branch-and-cut process to search for improved solutions and proven optimality. We also propose acceleration techniques with the analytical support of theoretic results from the dominance and consideration set formation. Finally, the effectiveness of our unified solution approach is verified by comparing it with other state-of-the-art exact solution approaches in the literature. Specifically, we conduct extensive computational studies on featured CBFL problems. Numerical results reveal that, although the unified solution approach is designed for a general model, it still outperforms most approaches that are dedicatedly developed for ad-hoc problems by a large margin.

2 Unified modeling prescription

2.1 Problem description

A classical CBFL problem is described as follows. Consider a company that plans to open p facilities, selected from J candidate sites $\mathscr{N} = \{1,2,...,J\}$ to serve I customers $\mathscr{M} = \{1,2,...,I\}$. Customers act as independent decision makers. When facilities are open, each customer will decide which facility to

seek service from based on his/her own preference. It is generally assumed that the choice of a customer is affected by the facilities' utilities, so let $u_{ij} > 0$ denote the utility of an open facility $j \in \mathscr{N}$ provided for a customer $i \in \mathscr{M}$.

If customers patronize a company's facility, then the company earns "rewards" by serving them. The company aims to locate facilities such that the earned total rewards are maximized. To this end, let us define a binary decision variable x_j, $\forall j \in \mathscr{N}$, representing the location decision at site j, i.e., $x_j = 1$ indicates that facility j is open; otherwise, facility j is closed. We might drop the subscript j when it is not misleading.

To find the optimal sites locations, the company needs to anticipate how the customers will make patronizing choices given the open sites. Hereafter, we assume that customers may not consider all open facilities when making the final choices. Instead, given p open facilities, each customer $i \in \mathscr{M}$ first forms a *consideration set*, i.e., a subset of open facilities that this customer is willing to consider before making the final patronizing choice (Masatlioglu et al. 2012, Ahumada and Ülkü 2018, Echenique and Saito 2019, Lin and Tian 2021a), and all other facilities that are not in the consideration set will, by definition, have a zero probability of being selected. We introduce another binary decision variable y_{ij} to denote whether facility $j \in \mathscr{N}$ is in customer i's consideration set. Given the consideration set, customer i then selects a facility from the set to patronize based on a certain choice rules. Accordingly, the patronization allows the company to obtain an estimated reward $R_i(y_i)$ from customer i, which depends on the consideration set variable y_i (note that y_i is a vector of J binary variables defined for customer i). Hereafter, we drop the subscript of y_i and use $R_i(y)$ as the reward for notational convenience when it is not misleading.

The above description is able to capture the common features of most existing CBFL models. The key idea is to re-interpret customer behaviors in a sequential perspective: forming a consideration set at first and then selecting a facility (or facilities) to patronize. Such perception facilitates us to provide a unified modeling prescription for the general CBFL problem in the remainder of this section.

2.2 Dominance inequality

The first ingredient in our modeling prescription is *preference dominance*, which is motivated by recent advance in choice modeling and will be used to construct the consideration set. Formally, it is given by

DEFINITION 1 (PREFERENCE DOMINANCE). *For two alternatives j and k, we say that j dominates k if k will not be considered by customers in the presence of j.*

Let $j \succ k$ represent that j dominates k or in other words, customers prefer j over k.

In CBFL, each customer maintains an individual consideration set. Consequently, let $j_i \succ k_i$ denote that customer i prefers facility j over facility k. If facility j is open, then customer i will not keep facility k in the consideration set as it is dominated by j. Mathematically speaking, with the previous decision variables, this condition can be stated as $y_{ik} \leq 1 - x_j$, if $j_i \succ k_i, \forall i \in \mathscr{M}, j \in \mathscr{N}, k \in \mathscr{N}$, ensuring that $y_{ik} = 0$ holds if $x_j = 1$. Unfortunately, this representation requires $O(IJ^2)$ numbers of constraints,

which leads to an inefficient large-scale formulation. Therefore, we model the dominance in another way, which is beneficial to scale down the formulation size. Let us define

$$\Delta_{ij} = \{k \in \mathscr{N} \mid j_i \succ k_i\} \quad \forall i \in \mathscr{M}, j \in \mathscr{N} \tag{1}$$

as the set of facilities dominated by a facility j. Obviously, if the facility j is open, then all facilities in Δ_{ij} will not appear in the consideration set of the customer i. This interpretation gives rise to the following dominance inequality

$$\sum_{k \in \Delta_{ij}} y_{ik} \leq Q_{ij} \cdot (1 - x_j) \quad \forall i \in \mathscr{M}, j \in \mathscr{N} \tag{2}$$

where Q_{ij} is initially set as $Q_{ij} = \min\{p, |\Delta_{ij}|\}$ because when p facilities are open, $\sum_{k \in \Delta_{ij}} y_{ik} \leq Q_{ij}$ always hold. Note that, it is possible to have even smaller values of Q_{ij}, which can lead to a tighter formulation.

Two types of preference dominance. In our framework, we define two general rules of *utility-related preference dominance*. Note that since the dominance is determined specific to each customer, for ease of exposition, we drop the subscript i tentatively in the definition.

In real-world contexts, the utility of a facility is attributed to multiple features (or factors), and it can be measured as a function of the relevant features. Here, we consider T available features to reflect the utility values. More precisely, let $\pi_j = \{\pi_j^t\}$ be the feature vector with $\pi_j^t > 0$ being the value of the t^{th} feature, $\forall t = 1, ..., T$. Then, we express the utility of facility j as a function of π_j, which is stated as

$$u_j = \mathbf{u}(\pi_j) = \mathbf{u}(\pi_j^1, \pi_j^2, ..., \pi_j^T) \tag{3}$$

Without loss of generality, we assume that u_j increases with each argument π_j^t. For example, in the Huff model (Huff 1964), the utility of facility j is affected by the intrinsic facility attraction $a_j > 0$ and the distance between the facility and the customer $d_j > 0$, i.e., $u_j = a_j^1 \cdot d_j^{-2}$. Given this, the features are defined as $\pi_j^1 = a_j$ and $\pi_j^2 = d_j^{-2}$ such that $u_j = \pi_j^1 \cdot \pi_j^2$. To connect the utility and dominance, we impose the following assumption.

ASSUMPTION 1. *For two facilities j and k, if $u_j \leq u_k$, then $j \not\succ k$.*

This assumption states that a facility will not be dominated by any facilities that cannot generate a higher utility. Based on the above utility characterization, we proceed to discussing two types of dominance.

DEFINITION 2 (SINGLE-CRITERION DOMINANCE INDICATOR, SDI). *For facilities j and k with utilities u_j and u_k, a function $\mathscr{D}^s(u_j, u_k)$ is called single-criterion dominance indicator function if (i) $j \succ k \iff \mathscr{D}^s(u_j, u_k) = 1$, (ii) $j \not\succ k \iff \mathscr{D}^s(u_j, u_k) = 0$, and (iii) $\mathscr{D}^s(u_j, u_k) = 0$ if $u_j \leq u_k$.*

In Definition 2, the dominance is constructed based on a single criterion, i.e., the magnitude of the utility. As we shall see later, many existing CBFL models fall into this category.

Note that, apart from the single-criterion case, it is possible that the utility values are determined by multiple features, such as the Pareto-Huff model (Peeters and Plastria 1998, Fernández et al. 2021) where customers only consider facilities that are "Pareto-optimal" with respect to the features. These situations motivate us to develop another definition of multi-criteria dominance.

DEFINITION 3 (MULTI-CRITERIA DOMINANCE INDICATOR, MDI). *For facilities j and k with utility feature vector π_j and π_k, a function $\mathscr{D}^m(\pi_j, \pi_k)$ is called multi-criteria dominance indicator function if (i) $j \succ k \iff \mathscr{D}^m(\pi_j, \pi_k) = 1$, (ii) $j \not\succ k \iff \mathscr{D}^m(\pi_j, \pi_k) = 0$, and (iii) $\mathscr{D}^m(\pi_j, \pi_k) = 0$ if $\mathbf{u}(\pi_j) \leq \mathbf{u}(\pi_k)$.*

2.3 Directed path inequality

The second ingredient in the unified modeling prescription is *directed path inequality* (DPI), which is defined exclusively using the consideration set variable y and imposed to strengthen the formulation.

Essentially, the preference dominance in most choice models is a *partial order*, which shows the following properties: (i) *Irreflexivity*: $j \not\succ j$, i.e., an alternative cannot dominate itself; (ii) *Asymmetry*: if $j \succ k$, then $k \not\succ j$; and (iii) *Transitivity*: if $j \succ k$ and $k \succ l$, then $j \succ l$.

Given these properties, we can recast the dominance relation of facilities as a Directed Acyclic Graph (DAG). Specifically, the facilities are represented by vertices and the pairwise dominance $j \succ k$ is described as a directed edge (j,k) linking from vertix j to vertix k. For a customer $i \in \mathscr{M}$, the graph $\mathscr{G}_i(\mathscr{N},\mathscr{E}_i)$ denotes the associated DAG, where $\mathscr{N}$ is the vertex set (which is exactly the set of candidate facilities) and $\mathscr{E}_i$ is the set of direct edges (which includes all pairs of dominance relation defined for the customer). For example, Figure 2 shows a DAG with 5 candidate facilities for customer i, where facility 1 and facility 2 are in the consideration set as they are not dominated by any other facilities (i.e., no entering edge). Moreover, we have $\Delta_{i1} = \{3,4,5\}$ (because facility 1 dominates facilities 3, 4, and 5); $\Delta_{i2} = \{4,5\}$; $\Delta_{i3} = \{4,5\}$; $\Delta_{i4} = \{5\}$; and $\Delta_{i5} = \emptyset$.

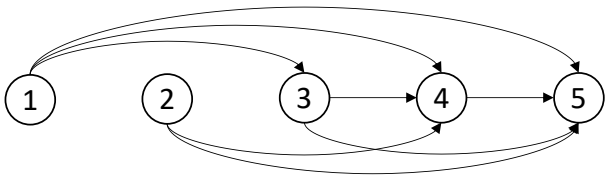


Figure 2 A directed acyclic graph with 5 vertices.

With the recast representation, the DPI is formally defined as follows.

DEFINITION 4 (DIRECTED PATH INEQUALITY, DPI). *For customer i, let ρ_i be a subset of $\mathscr{N}$. If there exists a permutation σ of ρ_i, where σ_n denotes the $n^{th}(1 \le n \le |\rho_i|)$ element in the σ, such that $\sigma_1 \succ \sigma_2 \succ \cdots \succ \sigma_{|\rho_i|-1} \succ \sigma_{|\rho_i|}$, then*

$$\sum_{j \in \rho_i} y_{ij} \le 1 \tag{4}$$

is a directed path inequality, and ρ_i is a path set generated by customer i's consideration set.

Given Definition 4, ρ_i is a set of vertices, which can be arranged into sequence σ such that there is a path in the associated DAG, starting from vertex σ_1, transversing through the vertices in σ, and ending at vertex $\sigma_{|\rho_i|}$, i.e., a sequence $\sigma_1 \to \sigma_2 \to \cdots \to \sigma_{|\rho_i|-1} \to \sigma_{|\rho_i|}$. Such a path interpretation leads to inequality (4) and is denoted as DPI.

LEMMA 1. *For the consideration set of customer i, inequality (4) is valid, and it enforces $|\rho_i|(|\rho_i|-1)/2$ pairwise dominance relations holding simultaneously.*

The proof is presented in E-Companion EC.2.1. Taking an example, consider the DAG in Figure 2. $\{1,3,4,5\}$ is a path set as the path $1 \to 3 \to 4 \to 5$ exists in the DAG. The associated DPI is $y_{i1} + y_{i3} + y_{i4} + y_{i5} \le 1$. Since there are 4 vertices in the path set, the number of pairwise dominance relations

imposed is 6. They are $1 \succ \{3,4,5\}$, $3 \succ \{4,5\}$, and $4 \succ 5$, which correspond to the edges in the subgraph consisting of vertices 1, 3, 4, and 5. Similarly, $\{2,4,5\}$ is also a path set, and it embeds 3 pairwise dominance relations.

Obviously, for customer i, there may exist more than one path set. From the computational perspective, the path set with more vertices is preferable because such type of path imposes more dominance relations and generates tighter relaxation bound. We thus define the *maximal path set* as follows.

DEFINITION 5 (MAXIMAL PATH SET, MPS). *In $\mathscr{G}_i(\mathscr{N},\mathscr{E}_i)$, a path set ρ_i^* is maximal if there does not exist another path set $\bar{\rho}_i$ from $\mathscr{G}_i(\mathscr{N},\mathscr{E}_i)$ such that $\rho_i^* \subsetneq \bar{\rho}_i$.*

Informally, ρ_i^* is maximal if we cannot further extend its associated path in $\mathscr{G}_i$. Recall the above example in Figure 2, we claim that $\{1,3,4,5\}$ is a maximal path set; whereas $\{3,4,5\}$ and $\{4,5\}$ are not maximal because their associated paths can be extended by adding the vertex 1 (i.e., path $1 \to 3 \to 4 \to 5$ and path $1 \to 4 \to 5$ indeed exist in $\mathscr{G}_i$).

In the following sections, the DPI development is based on the maximal path set ρ_i^*. It is also worth noting that there may be more than one maximal path in a DAG. Theoretically, it is possible to identify all maximal paths. However, the exhaustive enumeration demands significant computational efforts. Fortunately, in some CBFL models, the maximal path is unique (as we will see in Section 3); therefore, for consistency, we only generate one MPS for each customer throughout our modeling prescription. Algorithm 1 presents an efficient approach to generate the MPS. In line 1, we initialize the iteration

Algorithm 1 Maximal path set generation.

1: Set iteration counter $n = 1$. Select $\sigma_1 = \arg\max_{j \in \mathscr{N}} |\Delta_{ij}|$ and initialize $\rho_i^* = \{\sigma_1\}$.
2: **while** $|\Delta_{i\sigma_n}| > 0$ **do**
3: $\quad \sigma_{n+1} = \arg\max_{j \in \Delta_{i\sigma_n}} |\Delta_{ij}|$
4: $\quad \rho_i^* = \rho_i^* \cup \{\sigma_{n+1}\}$
5: $\quad n = n+1$
6: Return ρ_i^*.

counter $n = 1$. Then we select facility σ_1 with the most number of dominated facilities as the initial vertex of the path (i.e., the starting vertex of the path). Note that due to the transitivity of dominance, facility σ_1 will not be dominated by any other facilities (otherwise, there exists a facility k such that $|\Delta_{ik}| > |\Delta_{i\sigma_1}|$). Therefore, vertex σ_1 is the source vertex in the DAG. Line 3 finds the successor vertex σ_{n+1} that the path will visit. We select σ_{n+1} from $\Delta_{i\sigma_t}$ (so that there is a edge linking from σ_n to σ_{n+1}) such that the number of its dominated facilities is the maximum, i.e., $\sigma_{n+1} = \arg\max_{j \in \Delta_{i\sigma_n}} |\Delta_{ij}|$. This selection strategy ensures that we cannot further expand the path by inserting a vertex between σ_n and σ_{n+1} (otherwise, there exists a facility k in $\Delta_{i\sigma_n}$ leading to $|\Delta_{ik}| > |\Delta_{i\sigma_{n+1}}|$). Line 4 adds σ_{n+1} into the path. Then we update the counter n in Line 5 and proceed to next iteration. The while loop is terminated upon $|\Delta_{i\sigma_n}| = 0$, which signals that the current vertex σ_n is a sink vertex, and it is unable to further expand the current path. As a result, the following result arises immediately.

COROLLARY 1. *Algorithm 1 generates a maximal path set ρ_i^* for customer i, $\forall i \in \mathscr{M}$.*

2.4 General model for choiced-based facility location (G-CBFL)

With the above ingredients, we obtain the following general model for the CBFL problem.

$$\max \quad \sum_{i\in\mathscr{M}} R_i(y) \tag{5a}$$

$$\text{st.} \quad \sum_{k\in\Delta_{ij}} y_{ik} \le Q_{ij}(1-x_j) \quad \forall i\in\mathscr{M}, j\in\mathscr{N} \tag{5b}$$

[G-CBFL]

$$\sum_{j\in\rho_i^*} y_{ij} \le 1 \qquad \forall i\in\mathscr{M} \tag{5c}$$

$$\sum_{j\in\mathscr{N}} x_j = p \tag{5d}$$

$$y_{ij} \le x_j \qquad \forall i\in\mathscr{M}, j\in\mathscr{N} \tag{5e}$$

$$x_j \in \{0,1\} \qquad \forall j\in\mathscr{N} \tag{5f}$$

$$y_{ij} \in \{0,1\} \qquad \forall i\in\mathscr{M}, j\in\mathscr{N} \tag{5g}$$

where the objective (5a) is to maximize the total rewards of the company. More precisely, $R_i(y)$ is the reward that the company obtains from customer i, which could be revenue, net profit, or market share, depending on the specific problems that the model describes. The reward is dependent on the consideration set and the variable y. Constraint (5b) is the dominance inequality, which specifies how the consideration set of each customer is formed. Constraint (5c) is the DPI, which is independent of the location decision x and introduced to strengthen the formulation. Such a constraint will be tailored when applying G-CBFL to specific problems. Constraint (5d) ensures that the company will open p facilities. Constraint (5e) imposes that if a facility is not open, then it will not be in the consideration set of any customer. Finally, Constraints (5f) and (5g) defines the binary variables x and y.

3 Representable models

In this section, we present several existing CBFL models that can be represented by the framework G-CBFL. Consistent with prior research (Suárez-Vega et al. 2004, Biesinger et al. 2016), we utilize a classification scheme that categorizes demand types into essential and unessential, and then combines them pairwisely with different choice rules. Next, we proceed to instantiate the G-CBFL with appropriate combination of demand and choice rule. A summary of the classification and acronym of G-CBFL is given in E-Companion EC.1.

To simplify the formulation in later discussion, we define the following two sets

$$\Omega = \{x \mid (5d), (5f)\} \qquad \Xi = \{(x,y) \mid (5d) - (5g)\}$$

where Ω is the feasible space of x, and Ξ is the restricted feasible space of x and y.

3.1 Essenential demand

We begin our analysis by considering a model that assumes *essential demand*. This means that there is only one company operating in the market, and customers will definitely seek out its services. As a result, there is no possibility of demand loss from the company's perspective. In this model, customer choices are characterized by BCR, which stipulates that a customer will select the facility that provides them with the highest utility. Once customers have made their individual choices, the company proceeds

to serve them accordingly. This model allows us to gain insights into the behavior of customers in a market with no competition and provides a baseline for comparison when considering more complex market structures. To this end, the following assumption is necessary for the BCR.

ASSUMPTION 2. *In the BCR, there is a strict preference of facilities for every customer, i.e.,* $u_{ij} \neq u_{ik}, \forall i \in \mathscr{M}, j \in \mathscr{N}, k \in \mathscr{N}, k \neq j$.

In other words, for any two facilities j and k, their utilities to customer i are absolutely different (and not tied), which ensures that only one facility appears in the consideration set of customer i.

Now, let b_{ij} be the net profit if customer i is served by facility j. Then the CBFL problem characterizing by binary choice rule and essential demand is formulated as a mixed-integer bilevel linear program.

$$\max_{x,y} \quad \sum_{i\in\mathscr{M}} \sum_{j\in\mathscr{N}} b_{ij} y_{ij} \tag{7a}$$

$$\text{st.} \quad x \in \Omega \tag{7b}$$

$$\max_{y} \quad \sum_{i\in\mathscr{M}} \sum_{j\in\mathscr{N}} u_{ij} y_{ij} \tag{7c}$$

$$\text{st.} \quad \sum_{j\in\mathscr{N}} y_{ij} \leq 1 \qquad \forall i \in \mathscr{M} \tag{7d}$$

$$y_{ij} \leq x_j \qquad \forall i \in \mathscr{M}, j \in \mathscr{N} \tag{7e}$$

$$y_{ij} \in \{0,1\} \qquad \forall i \in \mathscr{M}, j \in \mathscr{N} \tag{7f}$$

where the upper level problem states that the company aims to maximize the total profits by locating p facilities; whereas in the lower level problem, each customer will patronize service from the open facility with the highest utility. Given the Assumption 2, the lower-level problem has a unique solution, as it resembles the p-median facility location with user preferences (Camacho-Vallejo et al. 2014, Casas-Ramírez and Camacho-Vallejo 2017).

In the above bilevel model, it is straightforward that customer i prefers facility j over k if $u_{ij} > u_{ik}$, i.e., $j_i \succ k_i$. The associated SDI function is $\mathscr{D}^s(u_{ij}, u_{ik}) = \mathbb{I}_{\{u_{ij}>u_{ik}\}}$. Accordingly, we define Δ_{ij} as

$$\Delta_{ij} = \{k \in \mathscr{N} \mid u_{ij} > u_{ik}\} \quad \forall i \in \mathscr{M}, j \in \mathscr{N} \tag{8}$$

As a result, the bilevel model can be restated as a single-level MILP given by

$$\max \quad \sum_{i\in\mathscr{M}} \sum_{j\in\mathscr{N}} b_{ij} y_{ij} \tag{9a}$$

$$\text{[EBCR-CBFL]} \qquad \text{st.} \quad \sum_{k\in\Delta_{ij}} y_{ik} \leq 1 - x_j \quad \forall i \in \mathscr{M}, j \in \mathscr{N} \tag{9b}$$

$$\sum_{j\in\mathscr{N}} y_{ij} \leq 1 \qquad \forall i \in \mathscr{M} \tag{9c}$$

$$(x,y) \in \Xi \tag{9d}$$

which matches the generic formulation of G-CBFL with $R_i(y) = \sum_{j\in\mathscr{N}} b_{ij} y_{ij}$, $Q_{ij} = 1$, and $\rho_i^* = \mathscr{N}$. Interestingly, Constraint (9b) appears to be a generalization of a closest assignment constraint, named as *WF* in Espejo et al. (2012), which has been demonstrated as a very effective way of representing "all-or-nothing" choices. Furthermore, the maximal path set for all customers is the same and equals to the set of all candidate facilities. Clearly, the following result holds.

COROLLARY 2. *In EBCR-CBFL, set* $\mathscr{N}$ *is the unique maximal path set.*

Therefore, we formulate the CBFL problem employing *binary choice rule with essential demand* as EBCR-CBFL[1].

3.2 Unessential demand

Expanding our analysis beyond the essential demand assumption, we investigate another stream of CBFL models—the *unessential demand* ones, where customers may not necessarily choose to patronize the services offered by the company. Given this, apart from the utilities provided by the facilities, $u_{ij}, \forall i \in \mathscr{M}, j \in \mathscr{N}$, there exists an *outside option* with a utility of $\tilde{u}_i, \forall i \in \mathscr{M}$. Virtually speaking, this outside option can be perceived as a "competitor" that competes with the company for customers. In this scenario, customers must decide whether to seek service from the company or from the outside option, leading to potential demand losses from the company's perspective. More specifically, customers split their buying power/demand proportionally among the facilities in the consideration set plus the outside option. As a result, customer choices are interpreted probabilistically, and the probability of customer i seeking service from the company is estimated in proportion to the total utility of the facilities in the consideration set and the representative utility of the outside option (Aboolian et al. 2007, Lin et al. 2022b), that is,

$$p_i(y) = \frac{\sum_{j\in\mathscr{N}} u_{ij} y_{ij}}{\sum_{j\in\mathscr{N}} u_{ij} y_{ij} + \tilde{u}_i} \quad \forall i \in \mathscr{M} \tag{10}$$

This model provides a more realistic representation of customer behavior in markets where there is competition and an outside option available. Then, the reward of the company acquired from customer i is described as $R_i(y) = b_i p_i$, where $b_i > 0$ is the buying power of customer i. We thus have the following formulation to describe the G-CBFL model under unessential demand and the proportional rule:

$$\max \quad \sum_{i\in\mathscr{M}} b_i \frac{\sum_{j\in\mathscr{N}} u_{ij} y_{ij}}{\sum_{j\in\mathscr{N}} u_{ij} y_{ij} + \tilde{u}_i} \tag{11a}$$

$$\text{st.} \quad \sum_{k\in\Delta_{ij}} y_{ik} \le Q_{ij}(1-x_j) \quad \forall i \in \mathscr{M}, j \in \mathscr{N} \tag{11b}$$

$$\sum_{j\in\rho_i^*} y_{ij} \le 1 \qquad \forall i \in \mathscr{M} \tag{11c}$$

$$(x,y) \in \Xi \tag{11d}$$

In the following subsection, we will present two particular models that fall within the generic framework outlined above and we will elaborate on their specific features and characteristics in detail. Specifically, they mainly differ in (i) how the consideration set Δ_{ij} is formed, and (ii) how the associated DPI ρ_i^* is defined. Note that, when Δ_{ij} is available, the DPI can be generated as the Algorithm 1. Therefore, in the rest of this subsection, we will focus on the discussing the Δ_{ij} construction.

3.2.1 Threshold Luce model

The threshold Luce model (TLM) developed by Echenique and Saito (2019), employs SDI to characterize the customer choice behavior given competitive market environment. In this regard, an alternative j dominates an alternative k if j's utility is strictly higher than $1+\gamma$ times of k's. The nonnegative parameter γ captures a threshold, beyond which alternative k is dominated by a more attractive alternative j.

Applying this concept to the G-CBFL model, we know $j_i \succ k_i$ if $u_{ij} > (1+\gamma)u_{ik}$ for a customer i. That is, $\mathscr{D}^s(u_{ij}, u_{ik}) = \mathbb{I}_{\{u_{ij}>(1+\gamma)u_{ik}\}}$, and the dominance set is given by

$$\Delta_{ij} = \{k \in \mathscr{N} \mid u_{ij} > (1+\gamma)u_{ik}\} \quad \forall i \in \mathscr{M}, j \in \mathscr{N} \tag{12}$$

Thus, we formulate the CBFL problem employing *threshold Luce model with unessential demand* as UTLM-CBFL.

As discussed in Lin et al. (2022a), several existing models are *de facto* special cases of the UTLM-CBFL. For example, on the one hand, when $\gamma = \infty$, Δ_{ij} and ρ_i^* become empty sets, indicating that all open facilities are in the consideration set. The TLM thus is degenerated to the standard Luce model and resembles the *multinominal logit model* (MNL) (Benati and Hansen 2002, Ljubić and Moreno 2018), where the probability of a customer visiting a facility is directly proportional to the facility's attractiveness or utility. In this case, (11b) and (11c) can be eliminated, and variable y can be removed by substituting $y_{ij} = x_j, \forall i \in \mathscr{M}, j \in \mathscr{N}$. On the other hand, when $\gamma = 0$, by introducing an additional assumption that facilities have distinct utilities, we know $j_i \succ k_i$ if $u_{ij} > u_{ik}$. This generates a consideration set with only one facility (i.e., a facility with the highest utility), and according to equation (10), the demand is then split proportionally between this facility and the outside option. Such a choice rule is exactly the well-known *partially binary choice rule* (PBCR) in the literature (Suárez-Vega et al. 2004, Biesinger et al. 2016, Fernández et al. 2017). Similar to the EBCR-CBFL, set $\mathscr{N}$ is the unique maximal path set for the PBCR, i.e., $\rho_i^* = \mathscr{N}, \forall i \in \mathscr{M}$. Therefore, our framework also can be applied to characterize the CBFL models featuring the classical MNL and PBCR.

3.2.2 Pareto-Huff model

In previous subsection, we discuss the possibility to adopt our modeling framework for the CBFL problems with SDI condition. Now, we further discuss its applicability on other models by imposing MDI into the choice rules, called Pareto-Huff model (PHM) (Peeters and Plastria 1998). In this regard, the utility of facility j to customer i usually consists of two features, i.e., the facility attraction a_j and the distance d_{ij}. The utility is typically estimated as

$$u_{ij} = \frac{a_j}{d_{ij}^{\psi}} \quad \forall i \in \mathscr{M}, j \in \mathscr{N} \tag{13}$$

where $\psi > 0$ is a decay parameter, reflecting how fast the utility decreases with the distance.

In the PHM, a customer will visit more distant facility only if that facility has higher attraction than any other closer facility; therefore, a distant facility will be considered by a customer only if there does not exist a closer open facility at least with the same attraction. In other words, customers only consider those facilities that are Pareto optimal to them with respect to attraction and distance (Peeters and Plastria 1998, Fernández et al. 2018, 2021). These facilities are "Pareto-Huff" ranked, and are used to form the consideration set in the unified framework.

To construct the set Δ_{ij}, we say that facility j dominates facility k if $a_j \geq a_k$ and $d_{ij} < d_{ik}$ hold for customer i, i.e., $j_i \succ k_i$. This condition ensures that the customer will eliminate the facilities that is not

Pareto-Huff from the consideration set. In other words, we have the feature vector $\pi_{ij} = \{a_j, d_{ij}^{-\psi}\}, \forall i \in \mathscr{M}, j \in \mathscr{N}$, and $\mathscr{D}^m(\pi_{ij}, \pi_{ik}) = \mathbb{I}_{\{a_j \geq a_k \text{ and } d_{ij}^{-\psi} > d_{ik}^{-\psi}\}}$. Then, the set Δ_{ij} is given by

$$\Delta_{ij} = \{k \in \mathscr{N} \mid a_j \geq a_k \text{ and } d_{ij}^{-\psi} > d_{ik}^{-\psi}\} \quad \forall i \in \mathscr{M}, j \in \mathscr{N} \tag{14}$$

Thus, we formulate the CBFL problem employing *Pareto-Huff model with unessential demand* as UPHM-CBFL.

3.3 Unessential demand with two-nest logit model under facility similarity

In the above unessential demand models, when estimating the probability of a customer seeking the service from the company, the total utility of the chained facilities is computed linearly by summing up the utilities of all facilities in the consideration set. Implicitly, this assumes that the utility is additive and that each facility is independent in terms of the utility value. However, since the facilities are provided by the same company, they may share similarities in terms of the service or products they offer (e.g., a common brand cannibalization effect or equivalent service), which can result in a non-linear relationship between the total utility and the number of facilities considered. For example, in the context of chained business operations, the total utility may not be additive in a linear fashion. Instead, the inclusion of new facilities to the consideration set may result in a diminishing marginal utility, leading to an overestimated reward in some scenarios if implementing the above mentioned models.

Inspired by the *nested logit model* (NLM), we generalize the basic unessential demand models by incorporating the concept of facility similarity. In particular, we focus on a two-nest scenario where we treat the facilities in the consideration set as a virtual nest and the outside option as the other. Based on this approach, customers are expected to first eliminate inferior options from the consideration set and then make a selection following the nested logit model.

According to the NLM, we introduce a parameter $\beta \in (0,1]$ representing the *dissimilarity* factor, and measuring the degree of difference among facilities. A lower value of β indicates that facilities are more similar to each other, and a higher value indicates the opposite. Moreover, we define $U_{ij} = \ln u_{ij}, \forall i \in \mathscr{M}, j \in \mathscr{N}$ and $\tilde{U}_i = \ln \tilde{u}_i, \forall i \in \mathscr{M}$. Thus, the probability of customer i seeking the service from the company is estimated as

$$p_i(y) = \frac{\left(\sum_{j \in \mathscr{N}} e^{U_{ij}/\beta} \cdot y_{ij}\right)^{\beta}}{\left(\sum_{j \in \mathscr{N}} e^{U_{ij}/\beta} \cdot y_{ij}\right)^{\beta} + e^{\tilde{U}_i}} \quad \forall i \in \mathscr{M} \tag{15}$$

Correspondingly, the utility of the company to customer i is represented as a utility of the nest, i.e.,

$$z_i = \left(\sum_{j \in \mathscr{N}} e^{U_{ij}/\beta} \cdot y_{ij}\right)^{\beta} \quad \forall i \in \mathscr{M} \tag{16}$$

When $\beta = 1$, facilities are totally dissimilar, and the utility becomes linearly additive. In this case, the equation (15) reduces to (10), and it degenerates to the standard unessential demand model in previous subsection. However, as the β decreases, z_i decreases accordingly, indicating that if facilities show higher similarity, then the company will attract less customers given the diminishing utilities of open similar facilities. Note that the resulting reward function $R_i(y) = b_i p_i$ is concave, which can be easily verified by composition rules. Our proposed unified modeling framework thus remains applicable to this extension. Consequently, we formulate the CBFL problem featuring *two-nest logit model with unessential demand* as UNLM-CBFL.

3.4 General discussion on the modeling framework

In this subsection, we present the following examples to facilitate the understanding of how dominance and directed path come into effect in the G-CBFL representable models. For customer i, let us define three subsets of $\mathscr{N}$ according to the possible scenarios of any feasible solution (x,y), i.e.,

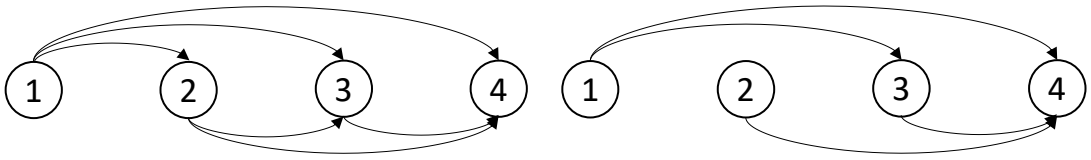
$$\mathbf{J}_i^{11} = \{j \in \mathscr{N} \mid x_j = y_{ij} = 1\} \tag{17a}$$

$$\mathbf{J}_i^{10} = \{j \in \mathscr{N} \mid x_j = 1, y_{ij} = 0\} \tag{17b}$$

$$\mathbf{J}_i^{00} = \{j \in \mathscr{N} \mid x_j = 0, y_{ij} = 0\} \tag{17c}$$

Specifically, $\mathbf{J}_i^{11}$ is the set of open facilities that are in the consideration set of customer i; $\mathbf{J}_i^{10}$ includes those open facilities that are dominated and thus does not present in the consideration set; and $\mathbf{J}_i^{00}$ is the set of facilities that are not open in the current location solution.

EXAMPLE 1. *Consider SDI and suppose that there are 4 open facilities. For a customer, the utilities of these facilities are $u_1 = 4.5$, $u_2 = 3.5$, $u_3 = 2.2$, and $u_4 = 1.0$.*
(i) For EBCR-CBFL, we have $1 \succ \{2,3,4\}$, $2 \succ \{3,4\}$, $3 \succ \{4\}$. Only facility 1 is in the consideration set. Therefore, $\mathbf{J}_i^{11} = \{1\}$ and $\mathbf{J}_i^{10} = \{2,3,4\}$. We can visualize the dominance as Figure 3(a). Accordingly, the maximal path set $\rho^* = \{1,2,3,4\}$.
(ii) For UTLM-CBFL, suppose $\gamma = 1$. We have $1 \succ \{3,4\}$, $2 \succ \{4\}$, $3 \succ \{4\}$. Facility 1 and facility 2 are in the consideration set. Therefore, $\mathbf{J}_i^{11} = \{1,2\}$ and $\mathbf{J}_i^{10} = \{3,4\}$. We can visualize the dominance as Figure 3(b). Accordingly, the maximal path set $\rho^* = \{1,3,4\}$. □



(a) BCR (b) TLM

Figure 3 Visualization of dominance as DAGs. Facilities are represented as vertices. A directed edge pointing from vertex j to vertex k indicates that facility j dominates facility k.

One interesting observation is that in both cases, facility 1 dominates all open facilities that are not in the consideration set. In fact, we generalize this result into a formal lemma.

LEMMA 2. *Under the SDI, for each customer i, there exists at least one facility in the consideration set $\mathbf{J}_i^{11}$ that dominants all facilities in $\mathbf{J}_i^{10}$.*

The proof is presented in E-Companion EC.2.2. Note that the validity of this result cannot be generalized to the MDI case, as explained in the below example.

EXAMPLE 2. *Under the MDI, suppose that there are two features. Consider 9 open facilities whose features are given as follows*
Assume that the indicator function is $\mathscr{D}^m(\pi_j,\pi_k) = \mathbb{I}_{\{\pi_j^1 > \pi_k^2 \text{ and } \pi_j^2 > \pi_k^2\}}$. We have $\mathbf{J}^{11} = \{1,2,3,4\}$ and $\mathbf{J}^{10} = \{5,6,7,8,9\}$. Since $1 \not\succ \{8,9\}, 2 \not\succ \{5,8,9\}, 3 \not\succ \{5,6,9\}, 4 \not\succ \{5,6,7\}$, there does not exist any facility in $\mathbf{J}^{11}$ that can dominate all facilities in $\mathbf{J}^{10}$. □

j	1	2	3	4	5	6	7	8	9
π_j^1	8	9	5	4	6	7	5	4	3
π_j^2	3	2	4	6	3	1	2	4	5

The above discussions provide insights into the structure of the DAGs generated by the SDI and MDI. In particular, for SDI-based models, the existence of "dominance facility" that dominates all open facilities outside the consideration set, according to Lemma 2, will greatly facilitate development of a general solution algorithm for the G-CBFL. For the MDI-based models, the absence of "dominance facility" will require additional algorithm implementation efforts, as we will see later.

4 Unified solution approach

In this section, we introduce an efficiently exact method for solving the G-CBFL and its representable models. We begin by presenting an overview of the designed algorithm, followed by a description of the necessary components, including the Benders separation and implementation procedures.

4.1 Overview

To facilitate the following discussion, we define two parameters δ_{ijk} and α_{ij} given by

$$\delta_{ijk} = \begin{cases} 1 & \text{if } k \in \Delta_{ij} \\ 0 & \text{otherwise} \end{cases} \quad \forall i \in \mathscr{M}, j \in \mathscr{N}, k \in \mathscr{N} \tag{18}$$

$$\alpha_{ij} = \begin{cases} 1 & \forall j \in \rho_i^* \\ 0 & \forall j \notin \rho_i^* \end{cases} \quad \forall i \in \mathscr{M} \tag{19}$$

With these parameters, the unified solution approach is described as follows. Note that for models that can be recast into the G-CBFL, the binary decision variable y can be relaxed to $y \geq 0$. Moreover, as the reward $R_i(y)$ is a concave function, consequently, the G-CBFL formulation is appealing for Benders decomposition because we can decompose it into I convex optimization subproblems for any fixed value $\bar{x}$ of the location variable x. Meanwhile, these subproblems are always feasible and bounded. They only contain the consideration set related to the variable y, and solving them allows to generate effective cutting planes to eliminate non-optimal location solutions. Specifically, we project out the variable y and consider a master problem, defined on the x-space,

$$\text{[MP]} \qquad \max_{x \in \Omega} \sum_{i \in \mathscr{M}} \Phi_i(x) \tag{20a}$$

in which $\Phi_i(x)$ is obtained by solving an individual subproblem for each customer zone i,

$$\Phi_i(x) = \max_{y} \; R_i(y) \tag{21a}$$

$$\text{st.} \;\; y_{ij} \leq x_j \qquad \forall j \in \mathscr{N} \tag{21b}$$

$$[\text{SP}_i(x)] \qquad \sum_{k \in \mathscr{N}} \delta_{ijk} y_{ik} \leq Q_{ij}(1 - x_j) \quad \forall j \in \mathscr{N} \tag{21c}$$

$$\sum_{j \in \mathscr{N}} \alpha_{ij} y_{ij} \leq 1 \tag{21d}$$

$$y_{ij} \geq 0 \qquad \forall j \in \mathscr{N} \tag{21e}$$

where constraints (21c) and (21d) utilizes the new defined parameters δ_{ijk} and α_{ij}. Essentially, $\Phi_i(x)$ represents "the reward of the company obtained from customer zone i", which is expressed as an implicit function of the location variable x. Obviously, $\Phi_i(x)$ is a concave function due to the concavity of $R_i(y)$. Therefore, [MP] is a mixed-integer convex optimization problem (MICP).

To solve this MICP, we linearize $\Phi_i(x)$ by its first-order approximator. That is, for a fixed $\bar{x}$, we build a linear cut at $\bar{x}$ as

$$w_i \leq \Phi_i(\bar{x}) + s_i(\bar{x})^T(x-\bar{x}) \quad \forall i \in \mathscr{M} \tag{22}$$

where $s_i(\bar{x}) \in \mathbb{R}^J$ is the subgradient of $\Phi_i(x)$ at value $\bar{x}$. Such a cut is often called the *generalized Benders cut* (GBC). For [MP], the cut is globally valid because it approximates $\Phi_i(x)$ from above and thus cannot cut off any feasible region. Consequently, [MP] can be approximated and solved by the following MILP:

$$\text{[rMP]} \qquad \max \sum_{i\in I} w_i \tag{23a}$$

$$\text{st. } w_i \leq \Phi_i(\bar{x}) + s_i(\bar{x})^T(x-\bar{x}) \quad \forall i \in \mathscr{M}, \bar{x} \in \mathscr{T} \tag{23b}$$

$$x \in \Omega \tag{23c}$$

where $\mathscr{T}$ is the set of x solutions that are visited previously and will be used to define GBCs for $\Phi_i(x)$ approximation. Such a MILP, defined on the projected decision space x, is called the *relaxed master problem* (rMP). In modern Benders decomposition, [rMP] is solved within the branch-and-cut framework using a single B&C tree, wherein the most critical task is to generate the GBC on-the-fly at nodes of the search tree. This process is commonly referred as *Benders separation*.

4.2 Benders separation

The key element is the Benders subgradient of $\Phi_i(\bar{x})$ at $\bar{x}$, i.e., $s_i(\bar{x})$. In fact, the subgradient value depends on the optimal dual variables of [SP$_i(\bar{x})$]. Note that there is no duality gap for [SP$_i(\bar{x})$] because all related constraints are linear. Let $\lambda_i \in \mathbb{R}^J_+$, $\mu_i \in \mathbb{R}^J_+$, $v_i \in \mathbb{R}_+$, and $\gamma_i \in \mathbb{R}^J_+$ be the dual variables associated with constraints (21b), (21c), (21d), and (21e), respectively. The Lagrangian function of [SP$_i(\bar{x})$] at $\bar{x}$ reads as

$$\ell_i = R_i(y) + \sum_{j\in\mathscr{J}} \lambda_{ij}(\bar{x}_j - y_{ij}) + \sum_{j\in\mathscr{J}} \mu_{ij}\left[Q_{ij}(1-\bar{x}_j) - \sum_{k\in\mathscr{J}} \delta_{ijk}y_{ik}\right] + v_i(1 - \sum_{j\in\mathscr{J}} \alpha_{ij}y_{ij}) + \sum_{j\in\mathscr{J}} \gamma_{ij}y_{ij} \tag{24}$$

The closed form of the Benders subgradient is given by

$$s_i(\bar{x}) = \frac{\partial \ell_i}{\partial x}\Big|_{x=\bar{x}} = \lambda_i - Q_i\mu_i \tag{25}$$

where an optimal dual solution $(\lambda_i, \mu_i) \in \mathbb{R}^J_+ \times \mathbb{R}^J_+$ is required to finalize the subgradient evaluation.

In modern convex optimization computer packages and software (e.g., MOSEK), after solving a primal problem (i.e., [SP$_i(\bar{x})$]), an optimal dual solution associated with the constraints can be automatically provided, alongside the optimal primal solution. A lazy and straightforward way of generating GBCs in the form of inequality (22) is to directly use MOSEK to solve [SP$_i(\bar{x})$]. Then, we obtain $\Phi_i(\bar{x})$ as the optimal objective of [SP$_i(\bar{x})$], retrieve (λ_i, μ_i), and evaluate $s_i(\bar{x})$ as equation (25).

4.3 Implementation

Our solution approach is named as *Two-phases Generalized Benders Decomposition Algorithm*, which is motivated by Bodur and Luedtke (2017). Figure 4 depict the implementation details.

Phase I involves solving the continuous relaxation of [rMP] using the traditional iterative Benders algorithm. The relaxed [rMP] is a linear program, and thus, we denote it by [rMP-LP]. At each iteration,

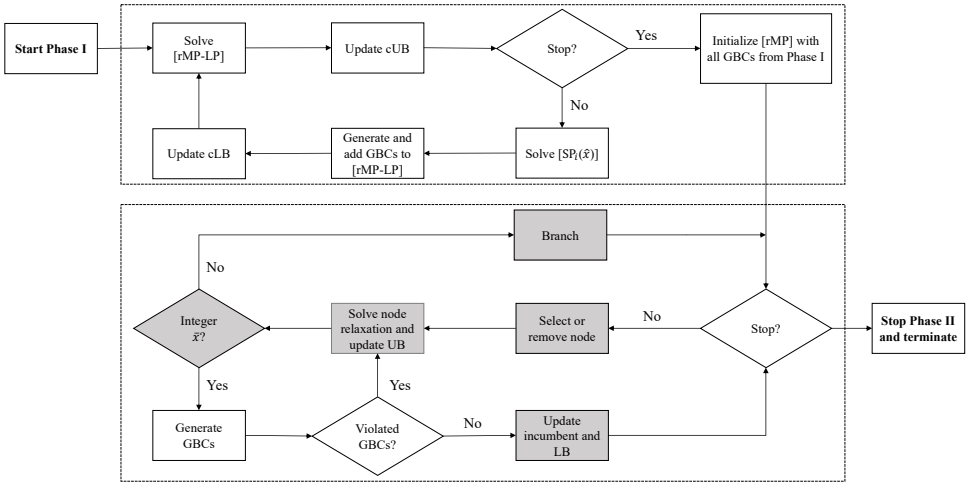


Figure 4 Two-phase generalized Benders decomposition algorithm.

we solve the [rMP-LP] to obtain a continuous upper bound (cUB) and a relaxed solution $\hat{x}$ (either fractional or integer). Then we proceed to generating GBCs at $\hat{x}$. This includes (i) solving [SP$_i(\hat{x})$] to obtain $\Phi_i(\hat{x})$ and (ii) evaluating the subgradient $s_i(\hat{x})$ in the form of equation (25). Here, we use the toolbox CVXPY to model [SP$_i(\hat{x})$] and subsequently call MOSEK solver to solve the problem. CVXPY can automatically generate the associated dual variables; therefore, the GBCs can be derived in a straightforward manner and are added to the [rMP-LP] to tighten the cUB. Then, the continuous lower bound (cLB) is updated according to the $\Phi_i(\hat{x})$. This loop repeats until $|cUB - cLB|/cUB < 10^{-4}$. We store all generated GBCs at the end of Phase I.

Phase II is the standard branch-and-cut (B&C) process, where we initialize [rMP] with the GBCs generated in the Phase I. The B&C is conducted within the commerical solver Gurobi 9.5.2. To enhance the numerical stability, we adopt conservative numerical tolerances in Gurobi, i.e., *IntFeasTol*, *FeasibilityTol*, and *OptimalityTol* are set to the minimum value 10^{-9}. In Figure 4, the 5 procedures (i.e., blocks) marked with dark color mean that they are conducted automatically by Gurobi's internal functions. We only intervene the solution process when Gurobi visits integer nodes in the B&C searching tree. That is, when the current node relaxation solution produces an integer $\bar{x}$, we generate the corresponding GBCs at $\bar{x}$ by solving [SP$_i(\bar{x})$] to obtain $\Phi_i(\bar{x})$ and evaluating $s_i(\bar{x})$. In other words, GBCs are only separated at integer nodes. This step is achieved within the *lazy callback* module of Gurobi. GBCs are added to the current node if the current solution violates a minimum threshold of 10^{-5}.

To summarize, the G-CBFL model shows separable convex structure when the location decision is fixed. Leveraging this property, we design a two-phase Benders decomposition algorithm that projects out the consideration set variable. However, during the B&C process in Phase II, the separation of GBCs is performed using external MOSEK. Such separation is indeed straightforward to implement, particularly owing to the merits that the dual variables of [SP$_i(\bar{x})$] can be generated automatically, and thus, $s_i(\bar{x})$ can be easily evaluated. However, we typically have to activate the separation for a large number of times. As a result, the solution process could be significantly slow down because solving [SP$_i(\bar{x})$] by an external solver is not trivial, especially for large-scale instances.

5 Algorithm acceleration

Perceiving that the Benders separation could be a bottleneck of the algorithm, this section is dedicated to designing a more efficient separation function. In particular, our goal is to obtain $\Phi_i(\bar{x})$ and the associated dual variables $\lambda_i \in \mathbb{R}_+^J$ and $\mu_i \in \mathbb{R}_+^J$ analytically.

5.1 Primal solution and KKT condition

Algorithm 2 Computing primal solution of $[\mathrm{SP}_i(\bar{x})]$

1: Initialize $\bar{y}_{ij} = 1, \forall j \in \mathscr{N}$.
2: **for** each $j \in \mathscr{N}$ **do**
3: **if** $\bar{x}_j = 0$ **then**
4: Set $\bar{y}_{ij} = 0$ ▷ Remove closed facilities from the consideration set
5: **else** ▷ $\bar{x}_j = 1$
6: Set $\bar{y}_{ik} = 0, \forall k \in \Delta_{ij}$ ▷ Remove facilities that are dominated by j
7: Return $\bar{y}_{ij}, \forall j \in \mathscr{N}$ ▷ Return the consideration set of customer i

To start with, note that given an integer $\bar{x}$, we can obtain the consideration set for customer i and thus solve $[\mathrm{SP}_i(\bar{x})]$ by Algorithm 2. Intuitively, the algorithm initializes all facilities in the consideration set (line 1) and then sequentially remove facilities that are either closed (line 4) or dominated (line 6). The resultant solution $\bar{y}$ is apparently optimal to $[\mathrm{SP}_i(\bar{x})]$ and is obtained in polynomial time.

Given the primal solution pair $(\bar{x}, \bar{y}_i)$, the reduced KKT system of $[\mathrm{SP}_i(\bar{x})]$ becomes

$$[\mathrm{KKT}_i(\bar{x},\bar{y})] \quad \begin{aligned}
&\frac{\partial R_i(\bar{y})}{\partial y_{ij}} - \lambda_{ij} - \sum_{k\in\mathscr{N}} \delta_{ikj}\mu_{ik} - \alpha_{ij}v_i + \gamma_{ij} = 0 && \forall j \in \mathscr{N} && \text{(26a)}\\
&\lambda_{ij}(\bar{x}_j - \bar{y}_{ij}) = 0 && \forall j \in \mathscr{N} && \text{(26b)}\\
&\mu_{ij}\left[Q_{ij}(1-\bar{x}_j) - \sum_{k\in\mathscr{N}} \delta_{ijk}\bar{y}_{ik}\right] = 0 && \forall j \in \mathscr{N} && \text{(26c)}\\
&v_i\left(\beta_i - \sum_{j\in\mathscr{N}} \alpha_{ij}\bar{y}_{ij}\right) = 0 && && \text{(26d)}\\
&\gamma_{ij}\bar{y}_{ij} = 0 && \forall j \in \mathscr{N} && \text{(26e)}\\
&(\lambda_i, \mu_i, \gamma_i, v_i) \geq 0 && && \text{(26f)}
\end{aligned}$$

where (26a) is the stationary condition, (26b)-(26e) state complementary slackness, and (26f) ensures dual feasibility. We omit the primal feasibility because $\bar{y}_i$ is feasible and optimal. The remaining task is to obtain a dual solution $(\lambda_i, \mu_i, \gamma_i, v_i)$ that satisfies $[\mathrm{KKT}_i(\bar{x},\bar{y})]$. The feasibility of $(\lambda_i, \mu_i, \gamma_i, v_i)$ to $[\mathrm{KKT}_i(\bar{x},\bar{y})]$ guarantees that $(\lambda_i, \mu_i, \gamma_i, v_i)$ is an optimal dual solution because the Lagrangian function (24) will be reduced to $R_i(\bar{y})$, which is exactly the primal objective.

Now, for customer i, we express the sets defined in (17) as $\bar{\mathbf{J}}_i^{11}$, $\bar{\mathbf{J}}_i^{10}$, and $\bar{\mathbf{J}}_i^{00}$ to emphasize their dependence on $(\bar{x}, \bar{y}_i)$. The following result arises.

LEMMA 3. *Given primal solution $(\bar{x}, \bar{y}_i)$, for each $i \in \mathscr{M}$, an optimal dual solution satisfies*

$$v_i = \mathbb{I}_{\left\{\sum_{j\in\mathscr{N}} \alpha_{ij}\bar{y}_{ij}=1\right\}} \cdot \min_{j\in\bar{\mathbf{J}}_i^{11}:\alpha_{ij}=1} \frac{\partial R_i(\bar{y})}{\partial y_{ij}} \tag{27}$$

$$\sum_{k\in\bar{\mathbf{J}}_i^{11}} \delta_{ikj}\mu_{ik} \geq \left[\frac{\partial R_i(\bar{y})}{\partial y_{ij}} - \alpha_{ij}v_i\right]^+ \qquad \forall j \in \bar{\mathbf{J}}_i^{10} \tag{28}$$

$$\lambda_{ij} = \begin{cases} \frac{\partial R_i(\bar{y})}{\partial y_{ij}} - \alpha_{ij} v_i & \forall j \in \bar{\mathbf{J}}_i^{11} \\ 0 & \forall j \in \bar{\mathbf{J}}_i^{10} \\ \left[\frac{\partial R_i(\bar{y})}{\partial y_{ij}} - \sum_{k \in \bar{\mathbf{J}}_i^{11}} \delta_{ikj} \mu_{ik} - \alpha_{ij} v_i \right]^+ & \forall j \in \bar{\mathbf{J}}_i^{00} \end{cases} \tag{29}$$

where $\mathbb{I}_{\{\cdot\}}$ *is the indicator function and* $[z]^+ = \max\{z, 0\}$. *Moreover,* $\mu_{ij} = 0, \forall j \in \bar{\mathbf{J}}_i^{00} \cup \bar{\mathbf{J}}_i^{10}$.

We present how we derive these conditions in E-Companion EC.2.3. According to the lemma, given $(\bar{x}, \bar{y}_i)$, v_i is immediately available. Unfortunately, $|\bar{\mathbf{J}}_i^{11}|$ elements in μ_i are still unknown and, in fact, constrained by $|\bar{\mathbf{J}}_i^{10}|$ linear inequalities in (28). However, once we know all values of μ_i, λ_i can be directly computed from equation (29). Therefore, the remaining task is to find out the missing pieces of μ_i.

The RHS of (28) can be evaluated when the primal solution is available. By inspection, setting μ_{ik}, $\forall k \in \bar{\mathbf{J}}_i^{11}$, to a large positive number is straightforwardly feasible for the inequality (28). However, this will lead to rather weak and ineffective GBCs. Furthermore, our Benders subgradient is defined as equation (25), where each μ_{ij} is multiplied by Q_{ij} (where Q_{ij} serves as a "big-M"). Therefore, we prefer to keep the vector μ_i as "sparse" as possible (i.e., to have fewest number of nonzero elements) for the sake of generating tight cuts.

The following two subsections illustrate how a sparse μ_i is generated to finalize the design of analytical Benders separation. It turns out that the approaches for the SDI and MDI models should be separately developed.

5.2 Dual solution under SDI

We first discuss the case with single-criterion dominance. According to Lemma 2, for each customer i, it is possible to find out one facility from set $\bar{\mathbf{J}}_i^{11}$ that dominants all facilities in $\bar{\mathbf{J}}_i^{10}$. As the dominance is based on the value of utility, the facility with the highest utility in $\bar{\mathbf{J}}^{11}$ must dominate all facilities in $\bar{\mathbf{J}}^{10}$. This observation leads to the following dual solution, where exactly one element in vector $\mu_{i\cdot}$ is nonzero.

COROLLARY 3. *Under the SDI, let* $m_i = \arg\max_{j \in \bar{\mathbf{J}}_i^{11}} u_{ij}$ *for customer i. A feasible and sparse* $\mu_i \in \mathbb{R}_+^J$ *is given by*

$$\mu_{ij} = \begin{cases} \max_{k \in \bar{\mathbf{J}}_i^{10}} \left[\frac{\partial R_i(\bar{y})}{\partial y_{ik}} - \alpha_{ik} v_i \right]^+ & \text{if } j = m_i \\ 0 & \forall j \in \mathcal{N} \setminus \{m_i\} \end{cases} \tag{30}$$

With the Corollary 3 and an integer $\bar{x}$ from the B&C searching tree of [rMP], we summarize the procedures to generate GBCs as follows. Firstly, we appy Algorithm 2 to obtain $\bar{y}_i$. For customer i, we construct sets $\bar{\mathbf{J}}_i^{11}$, $\bar{\mathbf{J}}_i^{10}$, and $\bar{\mathbf{J}}_i^{00}$. Then, we compute v_i and μ_i according to equations (27) and (30) and, subsequently, λ_i according to (29). Finally, with these primal and dual values, we generate GBCs in the form of (22), where the Benders subgradient is computed by equation (25).

Note that for EBCR-CBFL, the consideration set consists of one facility, i.e., $|\bar{\mathbf{J}}_i^{11}| = 1$. Naturally, this facility should dominate all other open facilities. For UTLM-CBFL, the distinct preference assumption is not required; therefore, it is possible that two or more facilities in the consideration set have the same and maximum utility. In this case, we select the one with the smallest facility index.

5.3 Dual solution under MDI

Different from the SDI case, under multi-criteria dominance, there may not exist a facility from the consideration set $\bar{\mathbf{J}}_i^{11}$ that dominants all facilities in $\bar{\mathbf{J}}_i^{10}$. Consequently, there could be more than one nonzero elements in the μ_i. In the worst case, the number of nonzero elements can go up to $\left|\bar{\mathbf{J}}_i^{11}\right|$.

For customer i, given a primal solution pair $(\bar{x}, \bar{y}_i)$, we define a $\bar{\mathbf{J}}_i^d \subseteq \bar{\mathbf{J}}_i^{11}$ as the "dominant set", such that each facility in $\bar{\mathbf{J}}_i^{10}$ is dominated by at least one facility in $\bar{\mathbf{J}}_i^d$. That is, for each $k \in \bar{\mathbf{J}}_i^{10}$, there exists a $m \in \bar{\mathbf{J}}_i^d$ such that $m \succ k$. Given the $\bar{\mathbf{J}}_i^d$, we can restrict that the set of nonzero elements in μ_i is only relevant to $\bar{\mathbf{J}}_i^d$.

Now, we need to derive a $\bar{\mathbf{J}}_i^d$ with as few elements as possible. To this end, we propose a greedy heuristic in Algorithm 3 to generate the $\bar{\mathbf{J}}_i^d$. Specifically, in line 1, we initialize $\bar{\mathbf{J}}_i^d$ as an empty set. In the while loop (line 2-7), we select the facility m from the consideration set $\bar{\mathbf{J}}_i^{11}$ that dominates the most open facilities (line 3) and include it to $\bar{\mathbf{J}}_i^d$ (line 4). Then, we remove this facility from $\bar{\mathbf{J}}_i^{11}$ to prevent it from being selected again later (line 5). Finally, we remove all open facilities that are dominated by the facility m (line 6). The while loop continues until $\bar{\mathbf{J}}_i^{10}$ becomes an empty set, signalizing that all open facilities that are not in the consideration set are dominated by facilities in $\bar{\mathbf{J}}_i^d$. Note that in line 3, if more than one facility has the same and maximum number of dominated facilities, the one with the smallest index is selected.

Algorithm 3 Generating dominant set.

1: Given $\bar{\mathbf{J}}_i^{10}$ and $\bar{\mathbf{J}}_i^{11}$, initialize $\bar{\mathbf{J}}_i^d = \emptyset$.
2: **while** $\bar{\mathbf{J}}_i^{10} \neq \emptyset$ **do**
3: $\quad m = \arg\max_{j \in \bar{\mathbf{J}}_i^{11}} \sum_{k \in \bar{\mathbf{J}}_i^{10}} \delta_{ijk}$
4: $\quad \bar{\mathbf{J}}_i^d = \bar{\mathbf{J}}_i^d \cup \{m\}$ ▷ Append m to dominant set
5: $\quad \bar{\mathbf{J}}_i^{11} = \bar{\mathbf{J}}_i^{11} \setminus \{m\}$
6: $\quad \bar{\mathbf{J}}_i^{10} = \bar{\mathbf{J}}_i^{10} \setminus \left(\Delta_{im} \cap \bar{\mathbf{J}}_i^{10}\right)$ ▷ Remove open facilities dominated by m
7: Return $\bar{\mathbf{J}}_i^d$.

Figure 5 shows an example of Algorithm 3. In iteration 1, facility 1 is selected and added to $\bar{\mathbf{J}}_i^d$. Accordingly, the facilities 5, 6, and 7 are removed because they are dominated by the facility 1. In iteration 2, facility 4 is selected and added to $\bar{\mathbf{J}}_i^d$. $\bar{\mathbf{J}}_i^{10}$ becomes an empty set since we remove facilities 8 and 9. Then, we know that with the facilities 1 and 4, all facilities in the original $\bar{\mathbf{J}}_i^{10}$ are dominated.

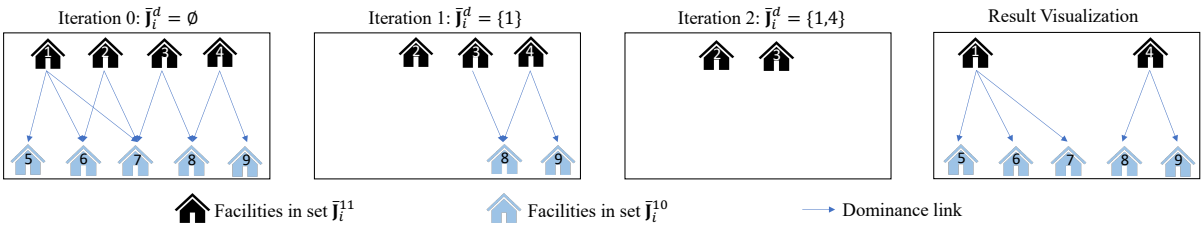


Figure 5 Example of finding a dominant set for multi-dimensional preference ordering using Algorithm 3.

With the $\bar{\mathbf{J}}_i^d$ generated by Algorithm 3, the following result arises immediately.

COROLLARY 4. *Under the MDI, let $\bar{\mathbf{J}}_i^d$ represent the dominant set for customer i. A feasible and sparse μ is given by*

$$\mu_{ij} = \begin{cases} \max\limits_{k \in \bar{\mathbf{J}}_i^{10} \cap \Delta_{ij}} \left[\frac{\partial R_i(\bar{y})}{\partial y_{ik}} - \alpha_{ik} v_i \right]^+ & \forall j \in \bar{\mathbf{J}}_i^d \\ 0 & \forall j \in \mathcal{N} \setminus \bar{\mathbf{J}}_i^d \end{cases} \tag{31}$$

The above corollary ensures that the inequality (28) holds, thereby allowing us to complete all missing pieces of μ_i necessary for Lemma 3. The rest of the Benders separation for multi-criteria dominance can be achieved in the same way as that for the SDI.

This section develops acceleration techniques for the unified solution approach by exploring the general problem structures under SDI and MDI respectively. The acceleration is achieved through replacing the use of external solvers by analytical formulas to handle Benders separation at integer nodes of B&C searching tree, which turns out to be rather effective in our preliminary computational test. Therefore, in what follows, when mentioning our solution algorithm, we explicitly refer to the two-phase generalized Benders decomposition in which acceleration techniques are default executed.

6 Featured computational study

In this section, we present computational experiments and results analysis. We refer to our generalized Benders decomposition approach as *GBD*, and primarily compare *GBD* with existing exact methods. To further demonstrate the efficiency of *GBD*, we also conduct extensive experiments using heuristics or approximation benchmarks. For more detailed discussions, interested readers are directed to EC.4.4.

To implement the algorithm, we code it by Python 3.10.6 on a 32 GB RAM Macbook with M1 processor and adopt Gurobi 9.5.2 as the solver. Unless explicitly mentioned below, Gurobi runs in the single-thread mode. We refer interested readers to Lin et al. (2024) for all codes that we used. For later discussion, we define several performance measurements: $t[s]$ denotes the computational time in seconds. By default, we set a 2-hour time limit (i.e., 7200 seconds). $rg[\%]$ stands for the relative exit gap in percentages when the solution process terminates, computed as $|zbb\text{-}zopt|/|zbb| \times 100\%$, where *zopt* is the value of the current best solution and *zbb* is the best bound. An instance is solved to optimality if $rg < 0.01\%$. '# branch nodes' is the number of branch-and-cut nodes explored by Gurobi, and '# instances unsolved/tested' indicates the number of unsolved instances among tested instances.

6.1 Experiment on Binary Choice Rule with Essential Demand

Our first experiment is concerned about the CBFL featuring binary choice rule with essential demand. According to the unfined framework, the traditional CBFL models can be reformulated into the EBCR-CBFL and further solved following the *GBD* procedure. Furthermore, we consider two benchmarks: (i) directly solving the reformulated MILP in Equation (9), which is called `EBCR-CBFL`^{2}, and (ii) reformulating the bilevel CBFL in Equation (7) into a MILP using the KKT-based approach (Casas-Ramírez and Camacho-Vallejo 2017), as EBCR-KKT, then solve it directly. We present the KKT-based reformulation details in E-Companion EC.3.1 and name such a benchmark as `EBCR-KKT`.

6.1.1 Rnd-EBCR instances

We test randomly generated instances with 100 candidate facilities, where the coordinates of customers and facilities are generated from $\mathscr{U}[1,100] \times \mathscr{U}[1,100]$. Distance d_{ij} between facility j and customer i is computed by the Euclidean distance. The utility is defined as $u_{ij} = 1/D_{ij}$, where $D_{ij} \sim \mathscr{U}[0.4 \cdot d_{ij}, 1.6 \cdot d_{ij}]$. Additionally, let us set $b_{ij} = 150 - d_{ij}$. All data are generated with fixed random seeds. These instances are referred as *Rnd-EBCR* instances. In total, we consider 30 Rnd-EBCR instances with the following combination of instance scale and parameter: number of customers $I = \{100, 150, 200, 300, 400\}$ and $p = \{3, 5, 7, 10, 15, 20\}$. The complete computational results are given in E-Companion EC.4.1.

Table 1 Computational results summary of Rnd-EBCR instances for EBCR-CBFL.

I	EBCR-KKT				EBCR-CBFL				*GBD*			
	t[s]	rg[%]	# branch nodes	# instances unsolved/tested	t[s]	rg[%]	# branch nodes	# instances unsolved/tested	t[s]	rg[%]	# branch nodes	# instances unsolved/tested
100	39.1	0.00	722	0/6	28.2	0.00	37	0/6	5.9	0.00	138	0/6
150	772.1	0.00	5359	0/6	50.3	0.00	151	0/6	11.4	0.00	729	0/6
200	1838.3	0.00	13312	0/6	81.7	0.00	494	0/6	23.5	0.00	2363	0/6
300	6515.5	0.16	14270	4/6	333.6	0.00	2362	0/6	111.1	0.00	13085	0/6
400	6956.6	0.39	7030	5/6	1220.3	0.00	6417	0/6	524.8	0.00	55480	0/6

Table 1 provides the summary of the computational results. In each row, 't[s]', 'rg[%]' and '# branch nodes' are average values over 6 instances. We mainly have two observations. First of all, `EBCR-CBFL` is more efficient than `EBCR-KKT` in general. For example, when $I = 150$, the average computational times of these two approaches are 50.3 seconds versus 772.1 seconds. Moreover, `EBCR-CBFL` successfully solves all instances to optimal, whereas `EBCR-KKT` fails in 9 instances. In fact, by comparing the formulation of EBCR-CBFL and EBCR-KKT, we know that the EBCR-KKT significantly expands the formulation size (i.e., introducing more variables and constraints). Furthermore, the EBCR-KKT seems weaker in the continuous relaxation tightness because it typically requires exploring substantially more branching nodes to solve an instance optimally. Therefore, it is clear that the advantage of EBCR-CBFL over EBCR-KKT is primarily owing to its smaller formulation size and tighter relaxation.

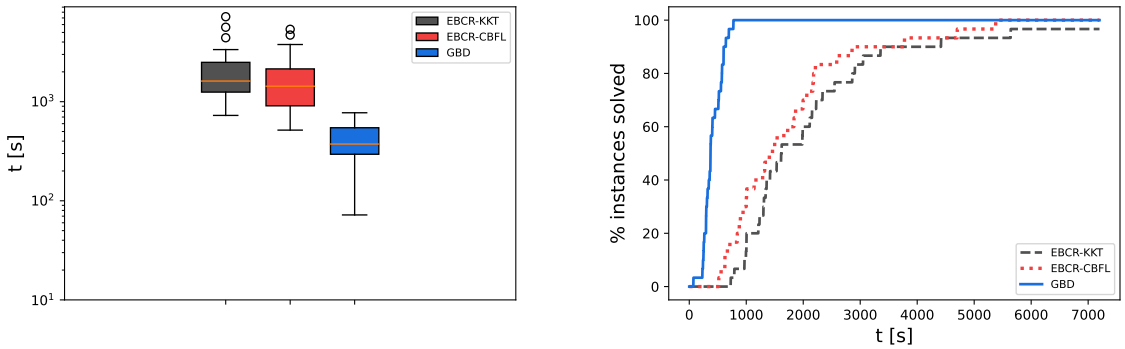
Secondly, *GBD* is the most efficient approach as it requires the lowest computational time to solve all instances to optimality. The advantage of *GBD* is that only the location variable x is maintained in the algorithm, and the consideration set related variable y is projected out through the decomposition scheme. Compared to `EBCR-CBFL`, the searching space is further reduced in *GBD*, which, together with the effectiveness of the proposed Benders separation technique in Section 5, substantially improves the efficiency to solve the Rnd-BCR instances.

6.1.2 PMPUP instances

We then tested 30 structured problem data *PMPUP*, which is a standard testbed for *p-Median Problem with Users Preferences* from benchmark library *Discrete Location Problems*[3]. For ease of exposition, the detailed description and testing results of 30 PMPUP instances are presented in E-Companion EC.4.2.

We show the graphical insights in Figure 6. Figure 6(a) reports the boxplot of the computational time (in log-scale). Apparently, *GBD* is the fastest approach to successfully solve all instances. In particular, it outperforms `EBCR-KKT` by almost an order of magnitude. Figure 6(b) shows the percentage of instances

solved optimally (% instances solved) within any given computational time. Clearly, *GBD* is the best one as the blue solid-line consistently dominates the others with solving all instances within time limit. In fact, for `EBCR-CBFL` and `EBCR-KKT`, their average computational times are 1736.6 seconds versus 2139.8 seconds, and their average branching nodes are 97881 versus 238250 respectively. Such an observation is conformed with that in the Rnd-BCR experiments.



(a) (b)

Figure 6 Computational results of PMPUP instances for EBCR-CBFL: (a) run time (b) performance profile.

Based on the computational results of the above two datasets, we conclude that the unified modeling framework could yield a tighter MILP formulation than the KKT-based approach. Furthermore, the *GBD* is comparatively powerful to solve the reformulated MILP model as it can reduce the run time for a siginificant order of magnitude.

6.2 Experiment on Threshold Luce Model with Unessential Demand

The next experiment is conducted on the CBFL problem characterized by the threshold Luce model with unessential demand. As discussed in Section 3.2.1, under the assumption of distinct utility, the TLM reduces to the PBCR when $\gamma = 0$. Therefore, we separately conduct experiments on the general cases with $\gamma > 0$ and the special case with $\gamma = 0$ which is equivalent to PBCR.

6.2.1 General case

The state-of-the-art exact solution approach for such a problem is proposed by Lin et al. (2022a). The authors took advantage of the path-based inequality, provided a model similar to the UTLM-CBFL in this article, reformulated the model as a mixed-integer conic quadratic program (MICQP) and then solved it by a commercial solver. This method is demonstrated to outperform several approaches in the literature and we thus take it as a benchmark for *GBD*. Specifically, we follow the reformulation procedure in E-Companion EC.3.3 and obtain a benchmark denoted as `MICQP`.

For a fair comparison, we adopt the dataset in Lin et al. (2022a) as our testbed[4]. The dataset is named as *LLPTL* and consists of two classes of instances, one with 200 customers and 100 facilities and the other with 400 customers and 150 facilities. The data also provides the coordinates of customers and facilities in plane space, and the distance matrix d is measured by Euclidean distance. According to Lin et al. (2022a), we set the utility as $u_{ij} = e^{-d_{ij}}, \forall i \in \mathscr{M}, j \in \mathscr{N}$, and $\tilde{u}_i = e^{-1}, \forall i \in \mathscr{M}$.

Table 2 Computational results of LLPTL instances for UTLM-CBFL.

p	γ	$(I,J)=(200,100)$				$(I,J)=(400,150)$			
		MICQP		*GBD*		MICQP		*GBD*	
		t[s]	rg[%]	t[s]	rg[%]	t[s]	rg[%]	t[s]	rg[%]
10	1	170.1	0.00	24.9	0.00	686.5	0.00	52.8	0.00
10	3	862.5	0.00	23.5	0.00	724.5	0.00	58.6	0.00
10	5	591.3	0.00	24.0	0.00	1392.4	0.00	65.5	0.00
10	7	657.7	0.00	21.4	0.00	867.9	0.00	58.8	0.00
10	10	687.3	0.00	18.8	0.00	748.5	0.00	54.8	0.00
10	20	610.8	0.00	16.4	0.00	691.5	0.00	61.0	0.00
20	1	733.4	0.00	250.7	0.00	1423.4	0.00	74.3	0.00
20	3	1318.1	0.00	140.1	0.00	2624.0	0.00	67.1	0.00
20	5	711.0	0.00	83.3	0.00	2332.3	0.00	71.8	0.00
20	7	890.4	0.00	41.2	0.00	2213.7	0.00	63.0	0.00
20	10	589.2	0.00	26.9	0.00	2243.8	0.00	56.1	0.00
20	20	615.8	0.00	25.8	0.00	2135.7	0.00	50.8	0.00
30	1	680.6	0.00	385.5	0.00	7200.0	0.67	1317.9	0.00
30	3	749.0	0.00	217.5	0.00	7200.0	0.53	1195.5	0.00
30	5	564.6	0.00	87.1	0.00	7200.0	0.64	717.8	0.00
30	7	578.3	0.00	60.5	0.00	7200.0	0.44	384.6	0.00
30	10	732.5	0.00	34.1	0.00	7200.0	0.37	192.5	0.00
30	20	562.9	0.00	26.0	0.00	5433.4	0.00	128.4	0.00

Here, we consider instances with $p = \{10,20,30\}$ and threshold $\gamma = \{1,3,5,7,10,20\}$. Computational results in Table 2 show that the *GBD* outperforms MICQP. In general, *GBD* is faster by more than one-order of magnitude. More importantly, it successfully solves all instances (the maximum time is only 1317.9 seconds), while the MICQP struggles in 5 instances with size $(I,J) = (400,150)$ and terminates with non-trivial gaps. Therefore, we argue that the proposed modeling prescription and corresponding solution approach perform efficiently and effectively on the CBFL problem employing Theshold Luce model with unessential demand.

6.2.2 Special case: partially binary choice rule with unessential demand

Now, we focus on a special case where $\gamma = 0$ and the utility is distinct. The underlying choice rule thus degenerates to the PBCR. In Lin and Tian (2021a), a Benders approach was proposed and shown to be efficient for the CBFL problem under PBCR and unessential demand. We thus take this method, denoted by $\texttt{Benders}_{\texttt{LT}}$, as the benchmark for our *GBD*. We also adopt the original dataset and the same solver configuration as Lin and Tian (2021a) for a fair comparison.

The computational results are summarized in Table 3. In general, when dealing with relatively easier instances (e.g., $J = 100$), $\texttt{Benders}_{\texttt{LT}}$ exhibits faster performance. However, for more challenging instances, *GBD* outperforms $\texttt{Benders}_{\texttt{LT}}$ in terms of efficiency and solution quality. Overall, *GBD* achieves optimality for all except one instance, and even in that unsolved instance, the termination gap is smaller than $\texttt{Benders}_{\texttt{LT}}$. Consequently, we claim that *GBD* remains competitive when compared to the state-of-the-art Benders approach, particularly when solving large-scale challenging instances.

Table 3 Computational results of state-of-the-art $\texttt{Benders}_{\texttt{LT}}$ approach and *GBD* for the PBCR case.

I	J	$\texttt{Benders}_{\texttt{LT}}$			*GBD*		
		t[s]	rg[%]	# branch nodes	t[s]	rg[%]	# branch nodes
1500	100	105.6	0.00	1953	159.4	0.00	215
1500	200	400.4	0.00	3527	916.4	0.00	8781
1500	300	2866.7	0.00	94361	2150.8	0.00	2624
2000	100	213.6	0.00	2246	305.7	0.00	525
2000	200	1435.7	0.00	14105	1357.6	0.00	8661
2000	300	7200.0	1.56	73433	4199.6	0.00	6868
3000	100	765.4	0.00	2048	545.5	0.00	371
3000	200	7200.0	0.85	32955	3708.3	0.00	2997
3000	300	7200.0	3.95	23540	7200.0	1.53	3082

6.3 Experiment on Pareto-Huff Model with Unessential Demand

We proceed to discuss the computational experiment on the CBFL problem characterized by Pareto-Huff model with unessential demand given three benchmarks. The first one is a MILP reformulation approach, which, to the best of our knowledge, is the only exact method dedicated to solving the CBFL model under PHM in the literature (Fernández et al. 2021). The reformulation procedure is illustrated in E-Companion EC.3.2, which utilizes a set of inequalities to construct the Pareto consideration set. We refer to this benchmark as $\texttt{MILP}_\texttt{F}$. Moreover, the MICQP reformulation approach is also applicable to this problem. Therefore, our second benchmark is a MICQP approach (also see EC.3.2), referred to as $\texttt{MICQP}_\texttt{F}$, which leverages the consideration set inequalities in Fernández et al. (2021). Finally, we obtain a MICQP formulation based on the model UPHM-CBFL and denote this benchmark as MICQP.

The instances are generated as follows. In PHM, the utility consists of two dimensions, i.e., the attraction a_j and the distance d_{ij}. We first generate $a_j \sim \mathscr{U}[1,100]$. Then, we generate the coordinates of customers and facilities as $\mathscr{U}[0,100] \times \mathscr{U}[0,100]$ and compute the d_{ij} by Euclidean distance. Accordingly, utility u_{ij} is defined in the standard way as $u_{ij} = a_j / d_{ij}^{\psi}$. Finally, we randomly draw $b_i \sim \mathscr{U}[1,1000]$.

Table 4 Computational results of random instances for UPHM-CBFL.

I	J	p	$\psi=2$								$\psi=3$							
			$\texttt{MILP}_\texttt{F}$		$\texttt{MICQP}_\texttt{F}$		MICQP		*GBD*		$\texttt{MILP}_\texttt{F}$		$\texttt{MICQP}_\texttt{F}$		MICQP		*GBD*	
			t[s]	rg[%]	t[s]	rg[%]	t[s]	rg[%]	t[s]	rg[%]	t[s]	rg[%]	t[s]	rg[%]	t[s]	rg[%]	t[s]	rg[%]
100	50	5	446.6	0.00	295.9	0.00	26.1	0.00	3.9	0.00	16.8	0.00	10.6	0.00	7.2	0.00	2.4	0.00
100	50	7	639.5	0.00	94.0	0.00	28.3	0.00	6.4	0.00	19.7	0.00	13.3	0.00	6.7	0.00	2.0	0.00
100	50	10	1055.8	0.00	596.6	0.00	42.1	0.00	17.8	0.00	27.2	0.00	15.3	0.00	7.7	0.00	2.4	0.00
100	80	5	567.1	0.00	135.3	0.00	49.8	0.00	8.8	0.00	47.3	0.00	22.6	0.00	8.5	0.00	3.9	0.00
100	80	7	1450.2	0.00	189.3	0.00	54.8	0.00	11.6	0.00	81.2	0.00	44.5	0.00	27.2	0.00	3.8	0.00
100	80	10	7200.0	18.01	3036.1	0.00	344.1	0.00	41.7	0.00	81.1	0.00	31.0	0.00	25.0	0.00	4.6	0.00
100	100	5	770.5	0.00	3095.4	0.00	73.1	0.00	11.6	0.00	60.9	0.00	65.3	0.00	18.8	0.00	5.1	0.00
100	100	7	7200.0	15.05	3734.5	0.00	371.5	0.00	25.9	0.00	136.8	0.00	79.8	0.00	26.4	0.00	5.1	0.00
100	100	10	7200.0	29.95	7200.0	3.02	554.0	0.00	86.7	0.00	350.6	0.00	104.0	0.00	41.0	0.00	5.8	0.00
200	100	5	7200.0	13.97	7200.0	11.95	1264.6	0.00	65.4	0.00	329.3	0.00	1064.8	0.00	152.3	0.00	16.2	0.00
200	100	7	7200.0	31.39	7200.0	12.18	2283.4	0.00	240.5	0.00	452.6	0.00	1384.4	0.00	349.4	0.00	14.3	0.00
200	100	10	7200.0	51.81	7200.0	11.15	4248.5	0.00	1643.3	0.00	826.8	0.00	1144.9	0.00	320.5	0.00	15.9	0.00
300	100	5	7200.0	20.53	7200.0	13.56	4662.8	0.00	106.8	0.00	1117.8	0.00	1805.9	0.00	187.2	0.00	25.7	0.00
300	100	7	7200.0	44.13	7200.0	12.79	4533.2	0.00	246.1	0.00	1529.2	0.00	6392.0	0.00	362.5	0.00	25.1	0.00
300	100	10	7200.0	53.58	7200.0	13.52	7200.0	10.14	2569.0	0.00	1694.8	0.00	6547.9	0.00	431.8	0.00	26.8	0.00
400	100	5	7200.0	6.92	7200.0	11.14	7200.0	5.39	58.7	0.00	1358.9	0.00	5873.2	0.00	555.8	0.00	30.2	0.00
400	100	7	7200.0	43.83	7200.0	14.65	7200.0	5.08	214.2	0.00	2525.3	0.00	7200.0	4.42	881.0	0.00	32.9	0.00
400	100	10	7200.0	56.20	7200.0	17.36	7200.0	10.45	4112.9	0.00	4010.3	0.00	7200.0	6.64	1002.0	0.00	33.3	0.00

Table 4 shows the computational results of three solution approaches under different instance scales (i.e., number of customers and candidate facilities), numbers of facilities to open p, and distance decay parameter ψ. Clearly, instances under $\psi = 2$ are substantially more challenging than those under $\psi = 3$. We also find that the benchmarks $\texttt{MILP}_\texttt{F}$ and $\texttt{MICQP}_\texttt{F}$ are not as powerful as the MICQP and *GBD*. Although $\texttt{MILP}_\texttt{F}$ can solve all instances under $\psi = 3$, it requires significantly longer computational time than MICQP and *GBD*. Besides, $\texttt{MICQP}_\texttt{F}$ even fails to solve 2 instances. Furthermore, under $\psi = 2$, $\texttt{MILP}_\texttt{F}$ and $\texttt{MICQP}_\texttt{F}$ can only handle instances with limited scales. As the number of candidate facilities exceeding 100, most instances are unsolved and terminated with large gaps. Note that $\texttt{MILP}_\texttt{F}$ and $\texttt{MICQP}_\texttt{F}$ are formulated based on the consideration set inequalities in Fernández et al. (2021), whereas MICQP and *GBD* are based on our unified modeling framework. Therefore, we conclude that our formulation is more effective in addressing the CBFL problem featuring Pareto-Huff model with unessential demand.

For MICQP, although its overall performance seems satisfactory, it still shows scalable weakness under $\psi = 2$ because there are 4 large-scale instances unsolved with considerable large gaps ($\geq 5\%$). Clearly, *GBD* is the most powerful approach as it solves all instances with the shortest computational time (on average 270.2 seconds). In particular, all instances under $\psi = 3$ are solved within 34 seconds.

6.4 Experiment on Two-nest Logit Model with Unessential Demand

In the final experiment, we discuss the computational performance of *GBD* under different facility similarity levels. Here, we do not test any MICQP or MILP reformulation approaches for benchmark comparison because they are not applicable to this case.

Specifically, we select the CBFL problem with Pareto-Huff model and unessential demand as the baseline to generate the consideration set but replace the reward function by the one with facility similarity. Moreover, we directly borrow the challenging (i.e., $\psi = 2$) PHM instances from the last subsection. For those 18 combinations of different (I, J, p), we consider 4 scenarios on the dissimilarity factor, i.e., $\beta = \{0.3, 0.7, 0.9, 1.0\}$. The complete computational results are presented in E-Companion EC.4.3, and we show a performance overview in Figure 7 .

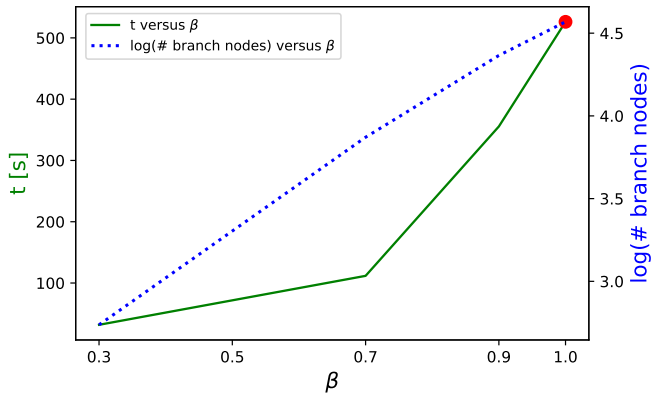


Figure 7 Computational results for UNLM-CBFL with different facility dissimilarity levels β.

Each data point in Figure 7 represents the average values over 18 instances and the number of B&C nodes is reported in log-scale, i.e., log(# branch nodes). Note that when $\beta = 1$, the instances are exactly the same as those investigated in the UPHM-CBFL when $\psi = 2$, marked as a red dot in the figure. As β increases, solving instances becomes computationally challenging. This can be observed through significant increases in both the computational time and the number of required B&C nodes. Specifically, the number of branch nodes grows almost exponentially with respect to β. Considering that *GBD* has been shown to be effective and efficient in solving the UPHM-CBFL, which represents the most challenging scenario of the UNLM-CBFL (with $\beta = 1$), it is reasonable to infer that *GBD* would perform well for the UNLM-CBFL in general.

7 Conclusion and Future Research

In this article, we introduced a unified framework for classes of CBFL problems arising from various industrial and business contexts. The unified framework consists of two key ingredients, a unified modeling prescription and a unified solution approach. Specifically, the modeling technique leverages the dominance consideration set and flexibly recasts common CBFL problems into unified representable formulations G-CBFL. Specifically, considering the essential or unessential demand, we investigated the reformulation of CBFL problems employing the BCR, TLM, and PHM models to capture customer choice behaviors respectively. We also showed that G-CBFL can effectively characterize unessential

demand models concerning facility similarity. In fact, the unified modeling prescription is capable of representing more CBFL problems beyond the aforementioned scenarios. To support its application in decision-making contexts, we also design an exact two-phase generalized Benders decomposition algorithm and accelerate it with analytical separation functions given different dominance indicators. Extensive computational studies demonstrate the effectiveness of the proposed unified framework.

We note that our framework permits the possibility of using a mixture of choice models to represent the varying behaviors of customers (for example, in the same problem, some customers may follow the TLM, whereas others may follow the PHM), which can be easily achieved by applying the corresponding rules, described in Section 3, to generate a dedicated consideration set for each customer. The validity lies in the fact that each individual's behavior is assumed to be independent in the existing literature and in our framework. This observation, in return, leads us to a question, i.e., *would it be possible to consider the behavior correlation between customers in the CBFL problem?* Indeed, the choice of a customer could be affected by the choices of others in some scenarios (e.g., the effect of word-of-mouth). Thus, developing a model that can characterize such a correlative effect could be meaningful.

Endnotes

[1]Please note that in this particular model, we replace the 'G' with 'EBCR' to signify the specific demand type and choice rule combination. To maintain consistency, we will also reacronym the G-CBFL in the subsequent models.

[2]Note that, we apply the font 'TEX Gyre Cursor' on a model name to indicate that this model is solved by a commerical solver directly. For example, `EBCR-CBFL` represents the model EBCR-CBFL is solved by the commerical solver directly.

[3]see `http://old.math.nsc.ru/AP/benchmarks/Bilevel/bilevel-eng.html`

[4]available at `https://doi.org/10.13140/RG.2.2.34645.55527`

Acknowledgments

The authors thank the area editor Pascal Van Hentenryck, the associate editor and two anonymous reviewers for their valuable comments that help improve this article significantly.

References

Aboolian, R., Berman, O., and Krass, D. (2007). Competitive facility location and design problem. *European Journal of Operational Research*, 182(1):40–62.

Ahumada, A. and Ülkü, L. (2018). Luce rule with limited consideration. *Mathematical Social Sciences*, 93:52–56.

Aros-Vera, F., Marianov, V., and Mitchell, J. E. (2013). p-hub approach for the optimal park-and-ride facility location problem. *European Journal of Operational Research*, 226(2):277–285.

Benati, S. and Hansen, P. (2002). The maximum capture problem with random utilities: Problem formulation and algorithms. *European Journal of Operational Research*, 143(3):518–530.

Beresnev, V. (2013). Branch-and-bound algorithm for a competitive facility location problem. *Computers & Operations Research*, 40(8):2062–2070.

Biesinger, B., Hu, B., and Raidl, G. (2016). Models and algorithms for competitive facility location problems with different customer behavior. *Annals of Mathematics and Artificial Intelligence*, 76(1-2):93–119.

Bodur, M. and Luedtke, J. R. (2017). Mixed-integer rounding enhanced benders decomposition for multiclass service-system staffing and scheduling with arrival rate uncertainty. *Management Science*, 63(7):2073–2091.

Borrero, J. S., Gillen, C., and Prokopyev, O. A. (2017). Fractional 0–1 programming: applications and algorithms. *Journal of Global Optimization*, 69(1):255–282.

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

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.

Dan, T. and Marcotte, P. (2019). Competitive facility location with selfish users and queues. *Operations Research*, 67(2):479–497.

Drezner, T., Drezner, Z., and Kalczynski, P. (2015). A leader–follower model for discrete competitive facility location. *Computers & Operations Research*, 64:51–59.

Drezner, T., Drezner, Z., and Zerom, D. (2018). Competitive facility location with random attractiveness. *Operations Research Letters*, 46(3):312–317.

Echenique, F. and Saito, K. (2019). General luce model. *Economic Theory*, 68(4):811–826.

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.

Fernández, P., Pelegrín, B., Lančinskas, A., and Žilinskas, J. (2017). New heuristic algorithms for discrete competitive location problems with binary and partially binary customer behavior. *Computers & Operations Research*, 79:12–18.

Fernández, P., Pelegrín, B., Lančinskas, A., and Žilinskas, J. (2018). The huff versus the pareto-huff customer choice rules in a discrete competitive location model. In *International Conference on Computational Science and Its Applications*, pages 583–592. Springer.

Fernández, P., Pelegrín, B., Lančinskas, A., and Žilinskas, J. (2021). Exact and heuristic solutions of a discrete competitive location model with pareto-huff customer choice rule. *Journal of Computational and Applied Mathematics*, 385:113200.

Gur, Y., Saban, D., and Stier-Moses, N. E. (2018). The competitive facility location problem in a duopoly: advances beyond trees. *Operations Research*, 66(4):1058–1067.

Haase, K. and Müller, S. (2014). A comparison of linear reformulations for multinomial logit choice probabilities in facility location models. *European Journal of Operational Research*, 232(3):689–691.

Huff, D. L. (1964). Defining and estimating a trading area. *Journal of Marketing*, 28(3):34–38.

Krohn, R., Müller, S., and Haase, K. (2021). Preventive healthcare facility location planning with quality-conscious clients. *OR Spectrum*, 43(1):59–87.

Lančinskas, A., Žilinskas, J., Fernández, P., and Pelegrín, B. (2020). Solution of asymmetric discrete competitive facility location problems using ranking of candidate locations. *Soft Computing*, 24(23):17705–17713.

Lin, Y., Wang, Y., Lee, L. H., and Chew, E. P. (2022a). Profit-maximizing parcel locker location problem under threshold luce model. *Transportation Research Part E: Logistics and Transportation Review*, 157:102541.

Lin, Y. H. and Tian, Q. (2021a). 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. (2021b). Generalized benders decomposition for competitive facility location with concave demand and zone-specialized variable attractiveness. *Computers & Operations Research*, 130:105236.

Lin, Y. H., Tian, Q., and Zhao, Y. (2022b). Locating facilities under competition and market expansion: Formulation, optimization, and implications. *Production and Operations Management*, 31(7):3021–3042.

Lin, Y. H., Tian, Q., and Zhao, Y. (2024). Unified framework for choice-based facility location problem. http://dx.doi.org/10.1287/ijoc.2022.0366.cd, https://github.com/INFORMSJoC/2022.0366.

Ljubić, I. and Moreno, E. (2018). Outer approximation and submodular cuts for maximum capture facility location problems with random utilities. *European Journal of Operational Research*, 266(1):46–56.

Mai, T. and Lodi, A. (2020). A multicut outer-approximation approach for competitive facility location under random utilities. *European Journal of Operational Research*, 284(3):874–881.

Mallozzi, L., D'Amato, E., and Pardalos, P. M. (2017). *Spatial Interaction Models*. Springer.

Masatlioglu, Y., Nakajima, D., and Ozbay, E. Y. (2012). Revealed attention. *American Economic Review*, 102(5):2183–2205.

Méndez-Vogel, G., Marianov, V., Lüer-Villagra, A., and Eiselt, H. (2022). Store location with multipurpose shopping trips and a new random utility customers' choice model. *European Journal of Operational Research*.

Peeters, P. H. and Plastria, F. (1998). Discretization results for the huff and pareto-huff competitive location models on networks. *Top*, 6(2):247–260.

Suárez-Vega, R., Santos-Peñate, D. R., and Dorta-González, P. (2004). Competitive multifacility location on networks: the (r|xp)-medianoid problem. *Journal of Regional Science*, 44(3):569–588.

Zhang, Y., Berman, O., and Verter, V. (2012). The impact of client choice on preventive healthcare facility network design. *OR spectrum*, 34(2):349–370.

E-Companion for "Unified framework for choice-based facility location problem"

EC.1 Classification and acronym of choice-based facility location problems

Figure EC.1 gives the summary of classification and acronym of the CBFL literature where only one company is involved in the decision making process.

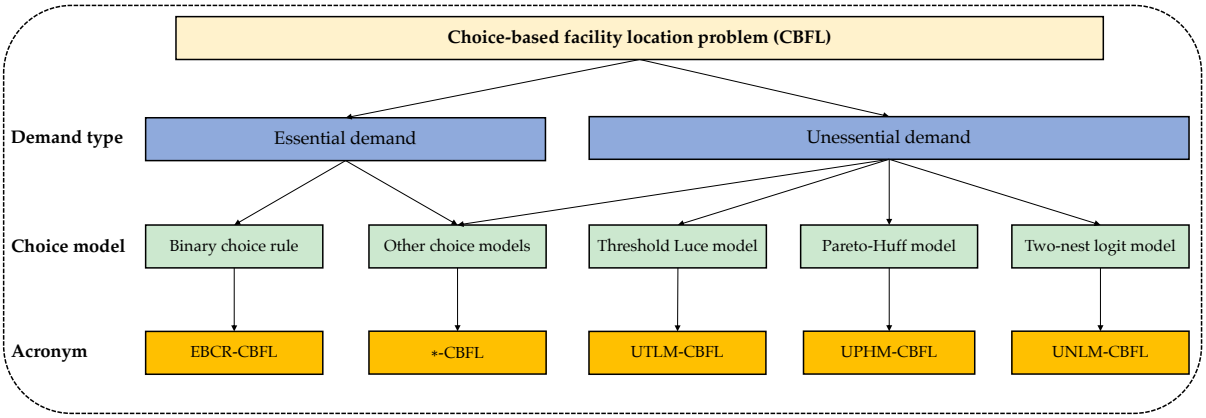


Figure EC.1 Classification and acronym of choice-based facility location problems.

The literature is first classified according to the demand type and then categorized by the choice model employed to characterize customer choices. Then, the acronyms of different CBFL problems are derived by combining the demand type (with 'E' denoting *essential demand* and 'U' denoting *unessential demand*) and the choice model. For example, EBCR-CBFL refers to the CFLB problem under essential demand and binary choice rule.

EC.2 Proofs

EC.2.1 Lemma 1

We prove the validity of the path strengthening inequality by induction. (i) If facility σ_1 is open, then we have $\sum_{j\in\rho_i\setminus\{\sigma_1\}} y_{ij} = 0$ since facility σ_1 dominates all facilities in $\rho_i \setminus \{\sigma_1\}$. Therefore, $\sum_{j\in\rho_i} y_{ij} = y_{i\sigma_1} \leq 1$. (ii) If facility σ_1 is not open and facility σ_2 is open, then we have $\sum_{j\in\rho_i\setminus\{\sigma_2\}} y_{ij} = 0$ since $x_{\sigma_1} = 0$ and facility σ_2 dominates all facilities in $\rho_i \setminus \{\sigma_1, \sigma_2\}$. Therefore, $\sum_{j\in\rho_i} y_{ij} = y_{i\sigma_2} \leq 1$. (iii) If facility σ_1 and facility σ_2 are not open and facility σ_3 is open, then $\sum_{j\in\rho_i\setminus\{\sigma_3\}} y_{ij} = 0$ since $x_{\sigma_1} = x_{\sigma_2} = 0$ and σ_3 dominates all facilities in $\rho_i \setminus \{\sigma_1, \sigma_2, \sigma_3\}$. Therefore, $\sum_{j\in\rho_i} y_{ij} = y_{i\sigma_3} \leq 1$. Continuing this procedure leads us to $\sum_{j\in\rho_i} y_{ij} \leq 1$. Therefore, the path strengthening inequality is valid.

Moreover, σ_1 dominates $|\rho_i| - 1$ facilities; σ_2 dominates $|\rho_i| - 2$ facilities; σ_3 dominates $|\rho_i| - 3$ facilities; and so on. In total, $\sum_{j\in\rho_i} y_{ij} \leq 1$ imposes $|\rho_i|(|\rho_i| - 1)/2$ pariwise dominance.

EC.2.2 Lemma 2

For the ease of exposition, we drop subscript i. For $j \in \mathbf{J}^{11}$, sort the preference u_j in nonincreasing order, i.e.,

$$u_{[1]} \geq u_{[2]} \geq \cdots \geq u_{[|\mathbf{J}^{11}|]}$$

Let set $\Pi_{[j]}$ be the set of facilities in $\mathbf{J}^{10}$ that are dominated by the facility with rank j , i.e., $\Pi_{[j]} = \Delta_{i[j]} \cap \mathbf{J}^{10}$. For two facilities with ranks j and k, if $j < k$, then $\Pi_{[j]} \supseteq \Pi_{[k]}$ (which leverages Assumption 1). As a result, we have

$$\mathbf{J}^{10} \supseteq \Pi_{[1]} \supseteq \Pi_{[2]} \supseteq \cdots \supseteq \Pi_{[|\mathbf{J}^{11}|]}$$

Now, suppose that $\Pi_{[1]} \subsetneq \mathbf{J}^{10}$ (i.e., $\Pi_{[1]}$ is a strict subset of $\mathbf{J}^{10}$). Then, at least one facility from $\mathbf{J}^{10}$ is not dominated by any open facility and thus should appear in the consideration set. This, however, contradicts to the definition that all facilities in $\mathbf{J}^{10}$ are not in the consideration set. Therefore, we must have $\Pi_{[1]} = \mathbf{J}^{10}$.

EC.2.3 Lemma 3: Recasting the KKT system

To begin with, note that if facility j is not open, then we safely have $\mu_{ij} = 0$ since Q_{ij} acts like a "Big-M", which presents the constraint from being active when $\bar{x}_j = 0$ (i.e., $j \in \bar{\mathbf{J}}_i^{00}$). Furthermore, for $j \in \bar{\mathbf{J}}_i^{10}$, facility j is not in the consideration set and thus must be dominated by at least one facility in $\bar{\mathbf{J}}_i^{11}$. Let m be a facility in $\bar{\mathbf{J}}_i^{11}$ that dominates facility j, i.e., $m_i \succ j_i$. We can also set μ_{ij} to 0. This is because (21c) becomes $\sum_{k\in\mathcal{N}} \delta_{ijk} y_{ik} \leq 0$. Facilities dominate by facility j are also dominated by facility m due to the transitivity of dominance relation. As a result, perturbing the right hand side of (21c) will not change the solution of y since the existence of facility m still enforce $\sum_{k\in\mathcal{N}} \delta_{ijk} y_{ik} = 0$. Therefore, we set $\mu_{ij} = 0, \forall j \in \bar{\mathbf{J}}_i^{00} \cup \bar{\mathbf{J}}_i^{10}$.

We then discuss these three scenarios in detail.

Scenario 1: $j \in \bar{\mathbf{J}}_i^{11}$. Since $\bar{x}_j = \bar{y}_{ij} = 1$, $\gamma_{ij} = 0$ due to (26e). Facilitiy j is in the consideration set and thus must be not dominated by any open facility; therefore, $\sum_{k\in\mathcal{N}} \delta_{ikj}\mu_{ik} = \sum_{k\in\bar{\mathbf{J}}_i^{11}} \delta_{ikj}\mu_{ik} = 0$, giving rise to following result

$$\lambda_{ij} = \frac{\partial R_i(\bar{\mathbf{y}})}{\partial y_{ij}} - \alpha_{ij} v_i \quad \forall j \in \bar{\mathbf{J}}_i^{11} \tag{EC.1a}$$

Note that $\lambda_{ij} \geq 0$. We must have

$$\frac{\partial R_i(\bar{\mathbf{y}})}{\partial y_{ij}} \geq \alpha_{ij} v_i \quad \forall j \in \bar{\mathbf{J}}_i^{11} \tag{EC.2a}$$

If $\alpha_{ij} = 0$, then the above condition holds naturally. Therefore, we can set v_i as

$$v_i = \mathbb{I}_{\left\{\sum_{j\in\mathcal{N}} \alpha_{ij}\bar{y}_{ij}=1\right\}} \cdot \min_{j\in\bar{\mathbf{J}}_i^{11}:\alpha_{ij}=1} \frac{\partial R_i(\bar{\mathbf{y}})}{\partial y_{ij}} \tag{EC.3}$$

where the indicator function $\mathbb{I}_{\{\cdot\}}$ is introduced to enforce the complementary condition (26d).

Scenario 2: $j \in \bar{\mathbf{J}}_i^{10}$. Since $\bar{x}_j > \bar{y}_{ij}$, $\lambda_{ij} = 0$ due to (26b). Facility j is dominated by at least one opened facility. As a result, we have

$$\frac{\partial R_i(\bar{\mathbf{y}})}{\partial y_{ij}} - \sum_{k\in\bar{\mathbf{J}}_i^{11}} \delta_{ikj}\mu_{ik} - \alpha_{ij} v_i + \gamma_{ij} = 0 \quad \forall j \in \bar{\mathbf{J}}_i^{10} \tag{EC.4a}$$

Rearranging the equation leads us to

$$\sum_{k\in\bar{\mathbf{J}}_i^{11}} \delta_{ikj}\mu_{ik} = \frac{\partial R_i(\bar{\mathbf{y}})}{\partial y_{ij}} - \alpha_{ij}v_i + \gamma_{ij} \geq \max\left\{0, \frac{\partial R_i(\bar{\mathbf{y}})}{\partial y_{ij}} - \alpha_{ij}v_i\right\} \quad \forall j \in \bar{\mathbf{J}}_i^{10} \tag{EC.5}$$

which can be further restated as

$$\sum_{k\in\bar{\mathbf{J}}_i^{11}} \delta_{ikj}\mu_{ik} \geq \left[\frac{\partial R_i(\bar{\mathbf{y}})}{\partial y_{ij}} - \alpha_{ij}v_i\right]_+ \quad \forall j \in \bar{\mathbf{J}}_i^{10} \tag{EC.6}$$

where $[z] = \max\{0, z\}$.

Scenario 3: $j \in \bar{\mathbf{J}}_i^{00}$. Since $\bar{x}_j = \bar{y}_{ij} = 0$, we have $\mu_{ij} = 0$ and

$$\frac{\partial R_i(\bar{\mathbf{y}})}{\partial y_{ij}} - \lambda_{ij} - \sum_{k\in\bar{\mathbf{J}}_i^{11}} \delta_{ikj}\mu_{ik} - \alpha_{ij}v_i + \gamma_{ij} = 0 \quad \forall j \in \bar{\mathbf{J}}_i^{00} \tag{EC.7a}$$

We can set

$$\lambda_{ij} = \left[\frac{\partial R_i(\bar{\mathbf{y}})}{\partial y_{ij}} - \sum_{k\in\bar{\mathbf{J}}_i^{11}} \delta_{ikj}\mu_{ik} - \alpha_{ij}v_i\right]_+ \quad \forall j \in \bar{\mathbf{J}}_i^{00} \tag{EC.8}$$

Summarizing the above results give rise to Lemma 3.

EC.2.4 Corollary 3

Based on the definition of m_i, we have $m_i \succ \bar{\mathbf{J}}_i^{10}$, or equivalently, $\delta_{i,m_i,j} = 1, \forall j \in \bar{\mathbf{J}}_i^{10}$. We rewrite (28) as

$$\mu_{im_i} + \sum_{k\in\bar{\mathbf{J}}_i^{11}\setminus\{m_i\}} \delta_{ikj}\mu_{ik} \geq \left[\frac{\partial R_i(\bar{\mathbf{y}})}{\partial y_{ij}} - \alpha_{ij}v_i\right]_+ \quad \forall j \in \bar{\mathbf{J}}_i^{10} \tag{EC.9}$$

To have a sparse μ, we set $\mu_{ik} = 0, \forall k \in \bar{\mathbf{J}}_i^{11} \setminus \{m_i\}$, leading to (30).

EC.3 Benchmark approaches and reformulation models

EC.3.1 KKT-based MILP reformulation technique

Here, we will show how leverage the KKT-based reformulation technique to obtain a MILP for the mixed-integer bilevel linear program relevant to CBFL problem characterizing by binary choice rule and essential demand in the manuscript Section 3.1. Let g_i and h_{ij} be the dual variables associated with Constraints (7d) and (7e). We have the KKT conditions for the lower-level problem as

$$\sum_{j\in\mathscr{N}} y_{ij} \leq 1 \qquad \forall i \in \mathscr{M} \tag{EC.10a}$$

$$y_{ij} \leq x_j \qquad \forall i \in \mathscr{M}, j \in \mathscr{N} \tag{EC.10b}$$

$$y_{ij} \geq 0 \qquad \forall i \in \mathscr{M}, j \in \mathscr{N} \tag{EC.10c}$$

$$h_{ij} \geq 0 \qquad \forall i \in \mathscr{M}, j \in \mathscr{N} \tag{EC.10d}$$

$$g_i + h_{ij} - u_{ij} \geq 0 \qquad \forall i \in \mathscr{M}, j \in \mathscr{N} \tag{EC.10e}$$

$$y_{ij}(g_i + h_{ij} - u_{ij}) = 0 \quad \forall i \in \mathscr{M}, j \in \mathscr{N} \tag{EC.10f}$$

$$h_{ij}(x_j - y_{ij}) = 0 \qquad \forall i \in \mathscr{M}, j \in \mathscr{N} \tag{EC.10g}$$

(EC.10f) and (EC.10g) are bilinear but can be exactly linearized as

$$g_i + h_{ij} - u_{ij} \leq M_{ij}^1(1 - y_{ij}) \quad \forall i \in \mathscr{M}, j \in \mathscr{N} \tag{EC.11a}$$

$$h_{ij} \leq M_{ij}^2(x_j - y_{ij}) \qquad \forall i \in \mathscr{M}, j \in \mathscr{N} \tag{EC.11b}$$

where M^1_{ij} and M^1_{ij} are sufficiently large numbers. To achieve tight relaxation bounds, they can be set as $M^1_{ij} = \max_{j\in\mathcal{N}} u_{ij}$ and $M^2_{ij} = u_{ij}$. We then replace the lower-level problem by the linear constraints: (EC.10a)-(EC.10e), (EC.11a)-(EC.11b). In our experiment, we found that restricting y_{ij} to binary variable can enhance the numerical stability and efficiency of the approach.

EC.3.2 MILP formulation for CBFL problem employing Pareto-Huff model with unessential demand

This appendix shows a MILP reformulation approach for CBFL problem employing Pareto-Huff model with unessential demand. leveraging the techniques presented by Fernández et al. (2021). We first define set $\mathbf{D}_{ij}$ as the set of facilities that dominates facility j for customer i, that is,

$$\mathbf{D}_{ij} = \{k \in \mathcal{N} \mid A_k \geq A_j \text{ and } d_{ik} \leq d_{ij} \text{ and } u_{ik} > u_{ij}\} \tag{EC.12}$$

Now, $y_{ij} = 1$ if and only if $x_j = 1$ (facility j is open) and $x_k = 0, \forall k \in \mathbf{D}_{ij}$ (all facilities that dominate facility j are not open). This logic can be expressed as the following constraints

$$y_{ij} \leq x_j \qquad \forall i \in \mathcal{M}, j \in \mathcal{N} \tag{EC.13a}$$

$$\sum_{k\in D_{ij}} x_k + y_{ij} \geq x_j \qquad \forall i \in \mathcal{M}, j \in \mathcal{N} \tag{EC.13b}$$

$$\sum_{k\in D_{ij}} x_k \leq |\mathbf{D}_{ij}| \cdot (1 - y_{ij}) \quad \forall i \in \mathcal{M}, j \in \mathcal{N} \tag{EC.13c}$$

Therefore, this problem can be restated as

$$\max \quad \sum_{i\in\mathcal{M}} b_i \frac{\sum_{j\in\mathcal{N}} u_{ij} y_{ij}}{\sum_{j\in\mathcal{N}} u_{ij} y_{ij} + \tilde{u}_i} \tag{EC.14a}$$

$$\text{st.} \quad \sum_{j\in\mathcal{N}} x_j = p \tag{EC.14b}$$

$$y_{ij} \leq x_j \qquad \forall i \in \mathcal{M}, j \in \mathcal{N} \tag{EC.14c}$$

$$\sum_{k\in D_{ij}} x_k + y_{ij} \geq x_j \qquad \forall i \in \mathcal{M}, j \in \mathcal{N} \tag{EC.14d}$$

$$\sum_{k\in D_{ij}} x_k \leq |\mathbf{D}_{ij}| \cdot (1 - y_{ij}) \quad \forall i \in \mathcal{M}, j \in \mathcal{N} \tag{EC.14e}$$

$$x_j \in \{0,1\} \qquad \forall j \in \mathcal{N} \tag{EC.14f}$$

$$y_{ij} \in \{0,1\} \qquad \forall i \in \mathcal{M}, j \in \mathcal{N} \tag{EC.14g}$$

Note that in this model, y_{ij} cannot be relaxed to $y_{ij} \geq 0$.

Now, we linearize the objective function. Let $z_i = 1/(\sum_{j\in\mathcal{N}} u_{ij} y_{ij} + \tilde{u}_i)$. We have equation $\sum_{j\in\mathcal{N}} u_{ij} y_{ij} z_i + \tilde{u}_i z_i = 1$, and the objective function becomes $\sum_{i\in\mathcal{M}} \sum_{j\in\mathcal{N}} b_i u_{ij} y_{ij} z_i$ with a bilinear term $y_{ij} z_i$.

To proceed, define an additional variable t_{ij} such that $t_{ij} = y_{ij} z_i$. Note that, a valid upper bound of z_i is $1/\tilde{u}_i$, we can thus rewrite $t_{ij} = y_{ij} z_i$ as

$$0 \leq t_{ij} \leq y_{ij}/\tilde{u}_i \qquad \forall i \in \mathcal{M}, j \in \mathcal{N} \tag{EC.15a}$$

$$z_i - (1 - y_{ij})/\tilde{u}_i \leq t_{ij} \leq z_i \quad \forall i \in \mathcal{M}, j \in \mathcal{N} \tag{EC.15b}$$

which ensure that when $y_{ij} = 0$, we have $t_{ij} = 0$ and that when $y_{ij} = 1$, we have $t_{ij} = z_i$.

Altogether, this problem can be equivalently restated as

$$\max \sum_{i\in\mathscr{M}}\sum_{j\in\mathscr{N}} b_i u_{ij} t_{ij} \tag{EC.16a}$$

$$\text{st. } (EC.13), (EC.15) \tag{EC.16b}$$

$$\sum_{j\in\mathscr{N}} u_{ij} t_{ij} + \tilde{u}_i z_i = 1 \qquad \forall i \in \mathscr{M} \tag{EC.16c}$$

$$y_{ij} \in \{0,1\} \qquad \forall i \in \mathscr{M}, j \in \mathscr{N} \tag{EC.16d}$$

$$\mathbf{x} \in \Omega^p \tag{EC.16e}$$

which is a MILP model and is ready to be solved by off-the-shelf solvers. For notational convenience, we use the MILP_F to represent this MILP model in the manuscript.

In fact, the objective function (EC.14a) can be reformulated as a rotated conic inequality, leading to a MICQP formulation (for more details about the reformulation, refer to E-Companion EC.3.3). This MICQP, leveraging the dominance condition (EC.13), is referred to as MICQP_F, and it is used as a benchmark approach in Section 6.3.

EC.3.3 General MICQP reformulation for G-CBFL under unessential demand

When demand loss or competition exists, the reward is often modeled as a linear fractional function, which is second-order conic representable. Here, we show that G-CBFL can be reformulated as a mixed-integer conic quadratic program (MICQP), to which effective algorithms are available in modern commercial solvers.

Define variable n_i such that $n_i = \sum_{j\in\mathscr{N}} u_{ij} y_{ij} + \tilde{u}_i$. We can rewrite the objective function as

$$\max \sum_{i\in\mathscr{M}} b_i \left(1 - \frac{\tilde{u}_i}{n_i}\right) \tag{EC.17}$$

We then introduce variable q_i such that $q_i \geq \tilde{u}_i / n_i$ holds. Noting that $\tilde{u}_i$ and n_i are positive by definition and $0 \leq q_i \leq 1$, the following quadratic rotated conic inequality arises immediately

$$q_i n_i \geq \tilde{u}_i \quad \forall i \in \mathscr{M} \tag{EC.18}$$

Therefore, G-CBFL can be equivalently restated as the following program

$$\max \sum_{i\in\mathscr{M}} b_i (1 - q_i) \tag{EC.19a}$$

$$\text{st. } n_i = \sum_{j\in\mathscr{N}} u_{ij} y_{ij} + \tilde{u}_i \qquad \forall i \in \mathscr{M} \tag{EC.19b}$$

$$q_i n_i \geq \tilde{u}_i \qquad \forall i \in \mathscr{M} \tag{EC.19c}$$

[MICQP]

$$0 \leq q_i \leq 1 \qquad \forall i \in \mathscr{M} \tag{EC.19d}$$

$$\sum_{k\in\Delta_{ij}} y_{ik} \leq Q_{ij}(1 - x_j) \quad \forall i \in \mathscr{M}, j \in \mathscr{N} \tag{EC.19e}$$

$$\sum_{j\in\mathscr{N}} \alpha_{ij} y_i \leq \beta_i \qquad \forall i \in \mathscr{M} \tag{EC.19f}$$

$$(\mathbf{x}, \mathbf{y}) \in \Xi \tag{EC.19g}$$

which is a MICQP since, except for the rotated conic inequality, all other constraints and the objective function are linear.

EC.4 Supplementary computational result

EC.4.1 Complete computational results of Rnd-EBCR instances for EBCR-CBFL

See Table EC.1.

Table EC.1 Complete computational results of Rnd-EBCR instances.

I	p	EBCR-KKT			EBCR-CBFL			*GBD*		
		t[s]	rg[%]	# branch nodes	t[s]	rg[%]	# branch nodes	t[s]	rg[%]	# branch nodes
100	3	113.7	0.00	886	36.1	0.00	55	6.1	0.00	211
	5	30.3	0.00	1188	28.6	0.00	60	6.4	0.00	232
	7	42.3	0.00	2256	28.9	0.00	101	6.9	0.00	338
	10	22.6	0.00	1	26.0	0.00	1	5.6	0.00	7
	15	13.4	0.00	1	26.2	0.00	1	5.4	0.00	24
	20	12.4	0.00	1	23.2	0.00	1	5.3	0.00	15
150	3	895.3	0.00	5532	64.7	0.00	196	13.1	0.00	679
	5	2581.2	0.00	17549	60.3	0.00	319	13.0	0.00	1195
	7	168.0	0.00	2891	49.1	0.00	124	11.9	0.00	795
	10	473.6	0.00	4478	43.7	0.00	177	12.0	0.00	960
	15	445.2	0.00	1586	43.2	0.00	54	9.9	0.00	578
	20	69.4	0.00	118	40.7	0.00	38	8.5	0.00	168
200	3	1806.3	0.00	11282	97.6	0.00	375	23.0	0.00	1054
	5	2632.9	0.00	14912	90.8	0.00	441	21.9	0.00	1616
	7	818.1	0.00	5476	72.4	0.00	210	19.1	0.00	1606
	10	3983.2	0.00	37068	98.1	0.00	1192	41.7	0.00	5947
	15	1235.2	0.00	8285	70.9	0.00	540	20.3	0.00	2393
	20	553.8	0.00	2847	60.5	0.00	205	15.1	0.00	1559
300	3	3488.9	0.00	7737	171.0	0.00	305	31.3	0.00	902
	5	7200.0	0.40	12789	231.5	0.00	835	91.0	0.00	5175
	7	7200.0	0.23	16829	409.2	0.00	1533	106.0	0.00	6430
	10	7200.0	0.12	19640	423.5	0.00	1828	99.0	0.00	7579
	15	7200.0	0.23	10322	577.5	0.00	7797	264.5	0.00	45497
	20	6794.6	0.00	18304	188.9	0.00	1874	74.8	0.00	12926
400	3	5739.5	0.00	7993	262.1	0.00	289	46.6	0.00	908
	5	7200.0	0.34	8453	418.2	0.00	829	129.6	0.00	4318
	7	7200.0	0.71	4658	1117.5	0.00	2747	424.0	0.00	17598
	10	7200.0	0.76	3189	1985.5	0.00	8377	706.7	0.00	51739
	15	7200.0	0.39	7645	3083.6	0.00	23762	1651.7	0.00	232355
	20	7200.0	0.16	10243	454.7	0.00	2497	190.2	0.00	25961

EC.4.2 Complete computational results of PMPUP instances

PMPUP is a standard testbed for *p-Median Problem with Users Preferences* from benchmark library *Discrete Location Problems* (see `http://old.math.nsc.ru/AP/benchmarks/Bilevel/bilevel-eng.html`). There are 30 structured problem instances. The instance number identifiers are $333, 433, 533, ..., 3122, 3233$. Note that in the dataset, for each customer, only a subset of facilities are accessible. We thus define a parameter $\mathscr{A}_{ij}$, which is 1 if facility j is accessible to customer i and 0 otherwise. We then add constraints $\sum_{j\in\mathscr{N}} \mathscr{A}_{ij} x_j \geq 1$. For the case of $\mathscr{A}_{ij} = 1$, the original data provides the cost c_{ij} by serving customer i from facility j and the disutility g_{ij} of facility j to customer i. To adapt the data set for our problem, if $\mathscr{A}_{ij} = 1$, we set $b_{ij} = 6 - c_{ij}$ and define utility $u_{ij} = 12 - g_{ij}$; and if $\mathscr{A}_{ij} = 0$, we set $b_{ij} = u_{ij} = 0$.

The complete computational results of these 30 instances are given in Table EC.2.

EC.4.3 *GBD* complete computational results under different values of dissimilarity factor

See Table EC.3.

Table EC.2 Complete computational results of PMPUP instances. For the unsolved EBCR-KKT, the associated relative gap is 5.61%.

inst.	EBCR-KKT		EBCR-CBFL		*GBD*	
	t[s]	# branch nodes	t[s]	# branch nodes	t[s]	# branch nodes
333	3355.4	478462	2212.6	118139	555.2	410757
433	1356.6	149256	557.5	30751	239.6	181913
533	1002.3	92269	876.9	49356	72.0	48806
633	2905.8	398720	1864.9	96586	601.8	457050
733	1416.3	163283	1460.6	97712	450.5	344386
833	2153.5	266050	2061.2	130103	510.3	380221
933	1300.4	161715	1540.9	76651	377.6	277112
1033	1989.8	168224	1137.8	67199	357.1	286651
1133	1532.5	188952	1323.9	60954	297.6	240192
1233	2226.5	299522	2180.8	130400	601.2	459875
1333	3049.5	400954	2855.4	169213	693.3	502185
1433	4417.1	603615	5376.1	343720	773.2	561274
1533	789.8	85668	943.2	61802	293.3	233319
1633	5639.6	439625	1726.5	125623	518.3	386361
1733	1211.2	142690	1000.4	53957	641.4	398606
1833	1309.3	129806	1005.9	42101	262.4	224922
1933	1620.5	196945	2587.7	158856	406.6	319641
2033	993.3	113250	1404.6	70596	373.1	305335
2133	2113.1	250156	1995.1	111075	402.5	315131
2233	1004.9	93535	841.5	43975	310.0	249943
2333	1232.2	139956	711.6	43121	320.3	263137
2433	726.9	72235	624.1	34954	229.9	183124
2533	2859.4	231344	516.6	22945	252.9	201687
2633	1983.7	282047	1836.8	105071	293.9	230517
2733	2548.1	352723	2169.4	110713	341.1	272842
2833	7200.0*	505088	4701.9	268624	570.9	446524
2933	967.9	89717	625.2	31027	248.1	211054
3033	1340	158659	1287.4	108972	374.3	300182
3133	2337.2	316049	896.6	50668	370.4	288360
3233	1610.3	176983	3775.4	121553	574.9	439770
Avg	**2139.8**	**238250**	**1736.6**	**97881**	**410.5**	**314029**

Table EC.3 *GBD* computational results under different values of dissimilarity factor β.

I	J	p	$\beta = 0.3$			$\beta = 0.7$		
			t[s]	# branch nodes	Profit	t[s]	# branch nodes	Profit
100	50	5	4.4	211	14256.3	5.9	74	14863.7
100	50	7	6.9	448	16426.7	6.5	683	17349.3
100	50	10	4.8	213	18731.7	16.7	4066	19937.4
100	80	5	6.0	351	14989.3	8.4	744	15573.5
100	80	7	8.1	1	17719.4	9.6	1156	18605.0
100	80	10	7.6	236	20902.9	29.8	5367	21792.0
100	100	5	13.6	535	15433.5	11.6	983	15942.0
100	100	7	9.9	761	18009.7	21.2	1772	18655.7
100	100	10	18.6	1209	21050.4	46.8	8586	21993.5
200	100	5	43.0	709	29965.5	61.8	3402	30947.0
200	100	7	37.4	604	35870.9	97.5	6900	37430.2
200	100	10	35.5	576	42606.4	393.4	44788	44755.0
300	100	5	60.3	659	44395.4	80.8	2263	46062.8
300	100	7	62.7	600	53559.8	165.0	4126	55992.0
300	100	10	54.1	622	63137.5	328.6	23167	66006.4
400	100	5	44.1	544	61948.5	88.0	852	64328.6
400	100	7	68.1	679	73485.1	127.5	2471	76534.5
400	100	10	86.9	873	84508.3	564.2	22333	88574.8

I	J	p	$\beta = 0.9$			$\beta = 1.0$		
			t[s]	# branch nodes	Profit	t[s]	# branch nodes	Profit
100	50	5	4.2	193	15438.9	3.9	230	15794.9
100	50	7	7.4	1026	18198.5	6.4	758	18753.8
100	50	10	16.9	3651	21064.7	17.8	5154	21712.9
100	80	5	8.5	1082	16127.4	8.8	977	16472.2
100	80	7	13.7	2365	19437.6	11.6	1276	19931.7
100	80	10	38.2	6376	22712.0	41.7	8330	23260.1
100	100	5	11.4	1204	16432.7	11.6	1318	16742.5
100	100	7	31.5	5046	19455.6	25.9	5613	19966.5
100	100	10	139.1	30363	22972.1	86.7	21468	23558.7
200	100	5	63.0	4018	32234.1	65.4	4321	33168.3
200	100	7	181.3	13302	38972.6	240.5	21879	39905.1
200	100	10	1322.8	172026	46659.2	1643.3	264490	47823.5
300	100	5	102.7	3463	47799.8	106.8	2925	48872.0
300	100	7	270.4	13951	58166.3	246.1	7974	59487.9
300	100	10	1356.7	87956	68694.5	2569.0	127775	70302.7
400	100	5	70.5	1688	66566.8	58.7	1557	67956.6
400	100	7	252.5	7257	79328.7	214.2	5611	81116.2
400	100	10	2508.0	62447	92170.0	4112.9	184946	94328.3

EC.4.4 Comparing *GBD* and Rank-based Discrete Optimization Algorithm

The Rank-based Discrete Optimization Algorithm (RDOA) might be perceived as one of the most widely used and effective heuristic/approximation algorithms for choice-based facility location prob-

lems and competitive facility location problems in recent years (Fernández et al. 2017, Lančinskas et al. 2020). In light of this, we implement the RDOA presented in Fernández et al. (2017) for solving the unified formulation and take it as a heuristic benchmark for our proposed exact algorithm *GBD*.

To facilitate the subsequent discussion, we refer to the Rank-based Discrete Optimization Algorithm with n numbers of function evaluations as `RDOA-n`. For example, if the algorithm is executed with 3000 function evaluations, it will be denoted as `RDOA-3000`. In this context, a function evaluation entails utilizing Algorithm 2 to calculate the optimal y given a location decision x, followed by evaluating the associated profit under x.

The `RDOA-n` employs a randomized subroutine to search for improved x solutions. Due to its inherent randomness, the best solution obtained by `RDOA-n` upon completion may vary across different runs. Consistent with Fernández et al. (2017), we conduct 10 independent runs of `RDOA-n` (using fixed and independent random seeds) for each problem instance. All results pertaining to `RDOA-n` are then averaged over these 10 independent runs. Furthermore, to facilitate solution quality comparison, we define the relative profit difference as follows:

$$\text{diff} = \frac{\text{Profit}(\textit{GBD}) - \text{Profit}(\texttt{RDOA-n})}{\text{Profit}(\textit{GBD})} \times 100\%$$

A *positive* value of diff indicates that *GBD* finds a better solution. When *GBD* achieves an optimal solution for a given instance, it guarantees that the global best solution has been reached. In such cases, the value of diff is nonnegative and can be regarded as an indicator of the "optimality gap" associated with `RDOA-n`.

We mainly test the *GBD* and `RDOA-n` on three categories of CBFL problems and analyze their computational performances as follows.

Computational results under UPHM. `RDOA-n` has been used to solve the CBFL under PHM. Therefore, we first compare its performance with GBD on UPHM-CBFL. We use the same random instances as presented in Section 6.3. Table EC.4 reports the computational results obtained by *GBD*, `RDOA-3000`, and `RDOA-5000` for these instances. We observe that: (i) `RDOA-3000` exhibits the capability to generate high-quality solutions, with several instances even achieving optimality. The maximum value of diff is 0.56%, which is an acceptable optimality gap. (ii) The computational time of `RDOA-n` seems to be more stable across instances. Since the function evaluation within `RDOA-n` can be executed in polynomial time using Algorithm 2, the computational complexity of `RDOA-n` increases only polynomially with respect to the problem size and near-linearly with respect to the number of function evaluations. However, as the model itself is inherently NP-hard, attempting to solve it exactly may encounter an exponential increase in computational complexity as the problem size grows. This accounts for the longer computational times observed for a few instances using *GBD* (mainly when $\psi = 2$ and $p = 10$). Despite this, *GBD* remains considerably efficient. As an exact algorithm, it successfully achieves optimal solutions for all instances (and it has been shown to outperform other exact approaches by a large margin in Section 6.3). In particular, when the instances are relatively less challenging (i.e., $\psi = 3$), *GBD* often takes a shorter computation time than `RDOA-3000`.

Table EC.4 Computational results of *GBD* and RDOA−n on the random instances of UPHM-CBFL.

ψ	I	J	p	*GBD*			RDOA−3000			RDOA−5000		
				t[s]	rg[%]	Profit	t[s]	Profit	diff[%]	t[s]	Profit	diff[%]
2	100	50	5	3.9	0.00	15794.9	4.2	15794.9	0.00	6.7	15794.9	0.00
	100	50	7	6.4	0.00	18753.8	6.0	18753.8	0.00	9.4	18753.8	0.00
	100	50	10	17.8	0.00	21712.9	8.3	21706.5	0.03	13.3	21712.9	0.00
	100	80	5	8.8	0.00	16472.2	5.3	16472.2	0.00	8.3	16472.2	0.00
	100	80	7	11.6	0.00	19931.7	7.6	19914.5	0.09	11.7	19931.7	0.00
	100	80	10	41.7	0.00	23260.1	10.6	23233.5	0.11	16.8	23246.5	0.06
	100	100	5	11.6	0.00	16742.5	6.0	16742.5	0.00	9.6	16742.5	0.00
	100	100	7	25.9	0.00	19966.5	8.4	19932.6	0.17	13.1	19952.2	0.07
	100	100	10	86.7	0.00	23558.7	12.2	23500.5	0.25	19.2	23558.7	0.00
	200	100	5	65.4	0.00	33168.3	12.1	32984.0	0.56	19.8	32984.0	0.56
	200	100	7	240.5	0.00	39905.1	16.5	39834.4	0.18	25.7	39876.7	0.07
	200	100	10	1643.3	0.00	47823.5	23.4	47685.5	0.29	37.2	47775.0	0.10
	300	100	5	106.8	0.00	48872.0	18.0	48621.1	0.51	28.4	48696.5	0.36
	300	100	7	246.1	0.00	59487.9	24.7	59316.9	0.29	38.6	59461.9	0.04
	300	100	10	2569.0	0.00	70302.7	35.1	70195.1	0.15	54.8	70195.1	0.15
	400	100	5	58.7	0.00	67956.6	23.2	67891.4	0.10	40.2	67956.6	0.00
	400	100	7	214.2	0.00	81116.2	32.2	81031.1	0.10	53.3	81116.2	0.00
	400	100	10	4112.9	0.00	94328.3	45.8	93846.2	0.51	73.9	94031.3	0.31
3	100	50	5	2.4	0.00	4730.9	4.0	4730.9	0.00	6.6	4730.9	0.00
	100	50	7	2.0	0.00	5863.5	5.5	5863.5	0.00	9.0	5863.5	0.00
	100	50	10	2.4	0.00	7217.7	7.4	7217.7	0.00	12.4	7217.7	0.00
	100	80	5	3.9	0.00	5538.9	5.1	5538.9	0.00	8.4	5538.9	0.00
	100	80	7	3.8	0.00	6849.6	6.9	6849.6	0.00	11.2	6849.6	0.00
	100	80	10	4.6	0.00	8532.3	9.6	8503.5	0.34	14.8	8532.3	0.00
	100	100	5	5.1	0.00	5595.4	5.4	5595.4	0.00	9.2	5595.4	0.00
	100	100	7	5.1	0.00	7012.0	7.5	7012.0	0.00	12.2	7012.0	0.00
	100	100	10	5.8	0.00	8796.6	10.5	8796.3	0.00	17.7	8796.6	0.00
	200	100	5	16.2	0.00	9330.3	11.9	9330.3	0.00	19.5	9330.3	0.00
	200	100	7	14.3	0.00	12331.3	15.8	12331.3	0.00	25.3	12331.3	0.00
	200	100	10	15.9	0.00	16148.9	22.2	16058.5	0.56	35.5	16148.9	0.00
	300	100	5	25.7	0.00	13837.6	17.6	13835.7	0.01	29.8	13837.6	0.00
	300	100	7	25.1	0.00	18035.0	23.9	18031.4	0.02	39.2	18035.0	0.00
	300	100	10	26.8	0.00	23941.6	33.0	23919.9	0.09	52.8	23931.8	0.04
	400	100	5	30.2	0.00	17740.0	23.6	17740.0	0.00	38.7	17740.0	0.00
	400	100	7	32.9	0.00	22975.8	32.9	22968.7	0.03	52.7	22968.7	0.03
	400	100	10	33.3	0.00	30025.3	45.3	29991.1	0.11	72.3	30025.3	0.00

Computational results under UPBCR. RDOA−n has also been utilized in the context of partially binary choice rule (PBCR), which is a special case for UTLM with $\gamma = 0$. Therefore, we conduct experiments on this type of problem. We use a large-scale dataset for PBCR from Lin and Tian (2021a) and set $p = 10$. The computational results of *GBD*, RDOA−5000 and RDOA−10000 are summarized in Table EC.5. In terms of solution quality, the performance of RDOA−5000 is not satisfactory as the maximum diff can exceed 1%. This indicates the need for an increase in the number of function evaluations. On the other hand, RDOA−10000 produces better solutions, albeit at the cost of doubling the computational time. Note that for the most challenging instance with $I = 3000$ and $J = 300$, *GBD* fails to solve it optimally and terminates with a relative exit gap of 1.53%. Consequently, the resulting solution obtained by *GBD* is not optimal. Interestingly, this instance is the only case where RDOA−10000 outperforms *GBD* by finding a better solution. Regarding computational time, we once again observe that for relatively less complex problem instances (i.e., $J = 100$ and 200), *GBD* generally presents faster computation times compared to RDOA−10000.

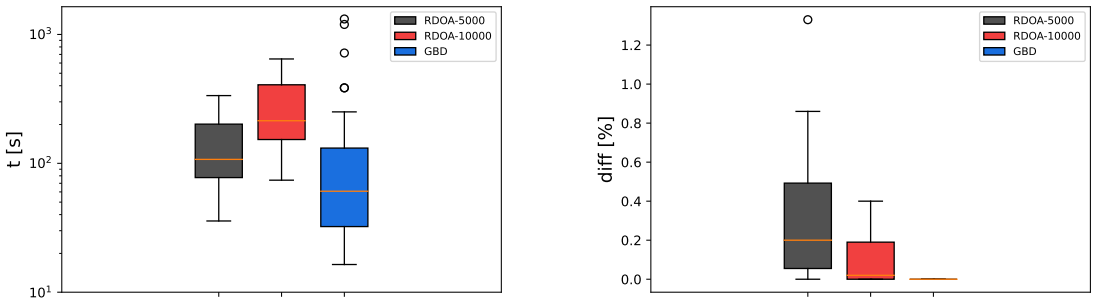
Table EC.5 Computational results of *GBD* and RDOA−n under PBCR using the dataset from Lin and Tian (2021a).

I	J	*GBD*			RDOA−5000			RDOA−10000		
		t[s]	rg[%]	Profit	t[s]	Profit	diff[%]	t[s]	Profit	diff[%]
1500	100	159.4	0.00	165283.3	314.4	165178.0	0.06	617.5	165283.3	0.00
1500	200	916.4	0.00	168425.8	537.2	167594.7	0.49	1063.8	168353.7	0.04
1500	300	2150.8	0.00	171838.0	761.3	170033.6	1.05	1471.9	171485.4	0.21
2000	100	305.7	0.00	217028.8	457.8	216560.3	0.22	940.6	216703.4	0.15
2000	200	1357.6	0.00	220568.6	713.2	218423.4	0.97	1430.1	219238.0	0.60
2000	300	4199.6	0.00	221516.6	1029.2	220292.7	0.55	1989.5	221204.6	0.14
3000	100	545.5	0.00	316193.8	671.2	313731.9	0.78	1310.2	314209.9	0.63
3000	200	3708.3	0.00	319984.4	1090.8	318738.2	0.39	2227.6	319545.8	0.14
3000	300	7200.0	1.53	316436.8	1562.3	316170.0	0.08	3108.8	317336.9	-0.28

Computational results under UTLM. In the final experiment, we conduct experiments on the UPHM-CBFL using the LLPTL instances described in Section 6.2. The computational results are presented in Table EC.6 and Figure EC.2. The key observations from this experiment align with the findings of the previous two experiments.

Table EC.6 Computational results of *GBD* and RDOA−n on the LLPTL instances of UPHM-CBFL.

			GBD			RDOA−5000			RDOA−10000		
I	p	γ	t[s]	rg[%]	Profit	t[s]	Profit	diff[%]	t[s]	Profit	diff[%]
200	10	1	24.9	0.00	16066.7	41.3	16066.2	0.00	82.8	16066.7	0.00
	10	3	23.5	0.00	16154.4	40.2	16147.3	0.04	80.6	16154.4	0.00
	10	5	24.0	0.00	16185.4	37.7	16185.4	0.00	78.8	16185.4	0.00
	10	7	21.4	0.00	16243.9	37.5	16242.5	0.01	78.2	16243.9	0.00
	10	10	18.8	0.00	16299.7	36.8	16299.5	0.00	76.4	16299.7	0.00
	10	20	16.4	0.00	16358.6	35.7	16358.6	0.00	74.0	16358.6	0.00
	20	1	250.7	0.00	25159.3	83.8	25126.7	0.13	163.8	25142.4	0.07
	20	3	140.1	0.00	25726.0	79.7	25688.7	0.14	155.9	25717.0	0.03
	20	5	83.3	0.00	26157.1	77.9	26139.9	0.07	153.9	26157.1	0.00
	20	7	41.2	0.00	26464.2	76.3	26449.2	0.06	150.2	26464.2	0.00
	20	10	26.9	0.00	26617.1	73.1	26611.4	0.02	146.1	26617.1	0.00
	20	20	25.8	0.00	26861.4	72.2	26826.5	0.13	143.8	26861.4	0.00
	30	1	385.5	0.00	31790.5	123.6	31726.7	0.20	240.3	31758.9	0.10
	30	3	217.5	0.00	32992.1	115.5	32899.8	0.28	234.2	32960.4	0.10
	30	5	87.1	0.00	33577.4	111.8	33479.2	0.29	230.9	33559.5	0.05
	30	7	60.5	0.00	33917.0	110.4	33850.5	0.20	224.5	33912.9	0.01
	30	10	34.1	0.00	34173.3	108.3	34093.9	0.23	222.3	34160.4	0.04
	30	20	26.0	0.00	34552.8	109.3	34538.1	0.04	215.9	34549.4	0.01
400	10	1	52.8	0.00	21625.8	106.3	21609.0	0.08	212.1	21625.8	0.00
	10	3	58.6	0.00	21703.9	103.1	21627.5	0.35	202.7	21701.5	0.01
	10	5	65.5	0.00	21732.6	101.9	21703.7	0.13	205.8	21732.6	0.00
	10	7	58.8	0.00	21773.1	101.1	21769.1	0.02	203.4	21773.1	0.00
	10	10	54.8	0.00	21801.4	99.6	21767.5	0.16	200.9	21801.4	0.00
	10	20	61.0	0.00	21849.7	98.2	21787.5	0.28	194.0	21849.7	0.00
	20	1	74.3	0.00	37284.3	214.1	36789.3	1.33	427.8	37135.6	0.40
	20	3	67.1	0.00	37512.5	219.9	37333.6	0.48	419.7	37411.9	0.27
	20	5	71.8	0.00	37624.2	207.8	37363.1	0.69	410.0	37555.3	0.18
	20	7	63.0	0.00	37879.2	199.3	37651.4	0.60	405.8	37858.5	0.05
	20	10	56.1	0.00	37987.0	198.8	37800.7	0.49	400.1	37961.7	0.07
	20	20	50.8	0.00	38142.2	196.3	37964.8	0.47	383.4	38054.6	0.23
	30	1	1317.9	0.00	48144.0	335.4	47728.5	0.86	645.3	47971.1	0.36
	30	3	1195.5	0.00	48836.2	305.5	48487.6	0.71	621.8	48696.7	0.29
	30	5	717.8	0.00	49193.8	314.1	48890.3	0.62	617.7	49086.1	0.22
	30	7	384.6	0.00	49608.4	300.2	49324.6	0.57	608.7	49484.0	0.25
	30	10	192.5	0.00	49889.9	297.5	49622.4	0.54	607.1	49729.4	0.32
	30	20	128.4	0.00	50293.1	287.9	50039.2	0.50	582.5	50158.4	0.27



(a) (b)

Figure EC.2 Computational results of LLPTL instances for UPHM-CBFL: (a) run time (b) optimality gap.

Through the above experiments, we have demonstrated that the computational time of *GBD* is competitive when compared to the heuristic RDOA−n under acceptable optimality gaps. This provides further validation of the efficiency of *GBD* as an exact algorithm.