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.

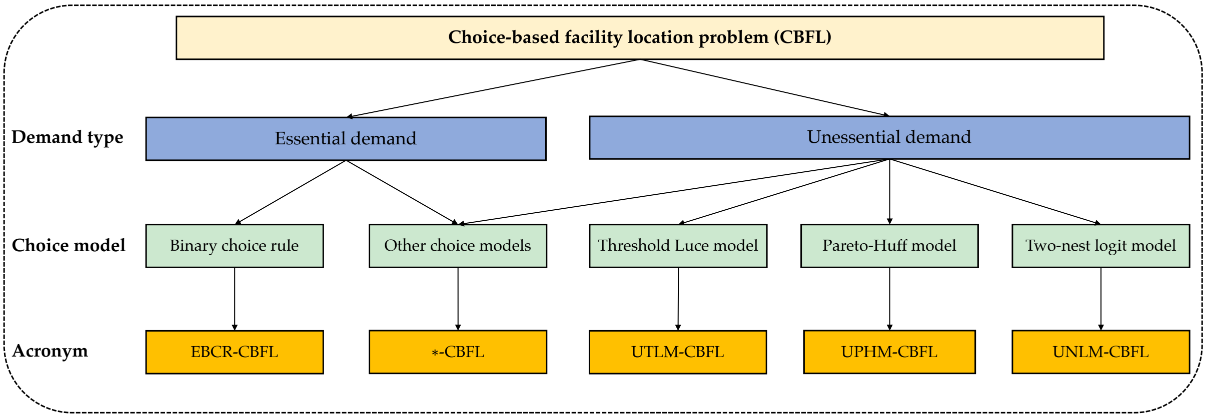


Figure EC.1 Classification and acronym of choice-based facility location problems.

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 $\sum_{k \in \mathscr{N}} \delta_{ijk} y_{ik} \leq 0$. 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 $\sum_{k \in \mathscr{N}} \delta_{ijk} y_{ik} = 0$. 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, $\sum_{k \in \mathscr{N}} \delta_{ikj} \mu_{ik} = \sum_{k \in \bar{\mathbf{J}}_i^{11}} \delta_{ikj} \mu_{ik} = 0$, giving rise to following result

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

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

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

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

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

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

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

Rearranging the equation leads us to

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

which can be further restated as

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

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

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

We can set

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

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

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

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

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

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

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

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

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

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

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

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

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

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

where M_{ij}^1 and M_{ij}^1 are sufficiently large numbers. To achieve tight relaxation bounds, they can be set as $M_{ij}^1 = \max_{j\in\mathcal{N}} u_{ij}$ and $M_{ij}^2 = u_{ij}$. 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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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 problems and competitive facility location problems in recent years (Fernández et al. 2017, Lančinskas et al. 2020, Yu 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:

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

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.

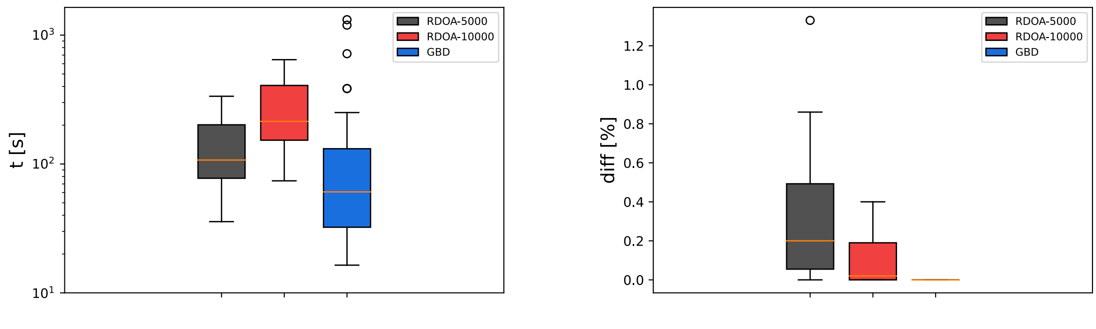
Table EC.5 Computational results of *GBD* and RDOA−n under PBCR using the dataset from Lin and Tian (2021a).

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.

I	p	γ	*GBD* t[s]	*GBD* rg[%]	*GBD* Profit	RDOA-5000 t[s]	RDOA-5000 Profit	RDOA-5000 diff[%]	RDOA-10000 t[s]	RDOA-10000 Profit	RDOA-10000 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.