Cross-entropy Method for the Maximal Covering Location Problem

Hongtao Wang[a], Jian Zhou[b,c,*]

[a] *Grupo de Sistemas de Optimización Aplicada, Universitat Politècnica de València, Camino de Vera s/n, 46021 València, Spain.*
[b] *School of Management, Shanghai University, Shanghai 200444, China.*
[c] *Confucius Institute, University of Bahrain, Manama, Kingdom of Bahrain.*

Abstract

The maximal covering location problem (MCLP) involves identifying optimal locations to maximize the covered demand with constraints from the number of facilities or budget limitations. This paper introduces a new MCLP formulation and a metaheuristic, the cross-entropy method to solve the problem. The method refers to a sampling-based solution construction from statistically tractable distribution models with iterative updates via inclusion probabilities in which a Pareto order sampling and a new local search are introduced. Extensive experiments are carried out on, to our knowledge, the most complete 8 benchmark data of 3 network types and 2 MCLP settings with 100 to 100,000 demand nodes. It demonstrates that (*i*) the proposed model is more compact with the number of variables and constraints; (*ii*) the cross-entropy method is highly effective in finding optimal solutions and competitive with other proposals and state of the art CPLEX 20.1 considering the involved large or massive instances.

Keywords: Maximal covering location problem, Cross-entropy method, Metaheuristic, Pareto sampling.

1. Introduction

The maximal covering location problem (MCLP), initially presented by Church and ReVelle (1974), is a classical and well-known combinatorial optimization problem. It aims to select the most appropriate locations of facilities to maximize the total demand covered within a predefined service distance or time. Once a configuration of the facilities' locations is determined, a demand node is said to be

*Corresponding author.
Email addresses: hwang8@doctor.upv.es (Hongtao Wang), zhou_jian@shu.edu.cn (Jian Zhou)

covered if it is within the service distance or time of at least one of these facilities. Due to resources or budget constraints, it is unnecessary for the MCLP to find out solutions satisfying the requirement that all demand nodes are covered, which distinguishes itself from the location set covering problem that intends to cover all demand nodes with a minimum number of facilities.

The MCLP has been addressing strategic decisions to determine facility locations. It has been intensively studied and frequently used in applications related to the location analyses of public resources such as medical supplies, police stations, and various types of emergency services. Some research has extended it to dynamic covering networks beyond the location-allocation modeling framework, like wireless networks (Lee and Murray 2010) and social networks (Güney et al. 2021). Decisions can also be made at the operational level. For instance, news feeds in a media network may be distributed at each hour to cater for the maximal number of interested readers. In these applications, the efficiency of solving the MCLP remains challenging. Some reviews and discussions can be found in Revelle et al. (2008), Berman et al. (2010), Lee and Murray (2010), Farahani et al. (2012), Murray (2016), Güney et al. (2021) and Blanco et al. (2023).

Of natural forms, the MCLP is commonly formulated as a 0-1 integer linear programming and tackled directly by a number of exact and heuristic methods. Although it may be structured as an equivalent p-median problem as illustrated by Church and ReVelle (1976) and Hillsman (1984). An exact solution method routinely uses the branch-and-bound technique with explicit or implicit linear programming relaxation, which can be seen in Church and ReVelle (1974) and Dwyer and Evans (1981), for instance. Although it is NP-hard in general, the MCLP behaves integer-friendly, as observed by Church and ReVelle (1974) and ReVelle (1993), in the sense that the corresponding linear programming relaxation often admits an all-integer optimal solution. This implies that the MCLP on the instances of medium or even large size might be solved within a reasonable amount of time by calling general purpose integer programming solvers. However, such a procedure would suffer from both primal and dual degeneracy in linear programming relaxation and would be likely to perform poorly on large-size instances. To deal with the degeneracy issues, the Lagrangian relaxation has been proposed by taking advantage of the structural features in its dual formulation (Downs 1991, Downs and Camm 1996).

Table 1 presents an overview of the literature on the original MCLP (Church and ReVelle 1974), highlighting the key features. Denoting the demand nodes set I and possible facility locations J, we summarize the literature with two MCLP settings and problem dimensions on I, J and p (budget on facilities), benchmark data (BD), instances (BI), types of networks (Net) and approaches. Some of the dimensions are detailed in Table 2. The MCLP setting involves how to set

possible facility sites utilized to choose p facilities, selecting from all demand nodes (denoted as $I = J$ for simplicity) or not ($I \neq J$). Even though not mandatory, the first is generally adopted in their experimental settings while the latter applies for more realistic applications with large or massive demand nodes (Resende 1998, Cordeau et al. 2019). Both settings are entailed in the current work with extensive collections of the most popular benchmark sets and covering networks.

Table 1: Existing literature and features of the MCLP (Church and ReVelle 1974)

Literature	Setting	Max.$\|I\|/\|J\|/p$	BD	BI	Net	Approach
Galvão and ReVelle (1996)	$I = J$	150/150/12	1	45	1	Lagrangean heuristic
Resende (1998)	$I \neq J$	10000/1000/900	1	90	1	Greedy randomized adaptive search procedure (GRASP)
Galvão et al. (2000)	$I = J$	900/900/28	2	341	2	Lagrangean and surrogate relaxations
Lorena and Pereira (2002)	$I = J$	900/900/28	3	133	2	Lagrangean/surrogate heuristic (LRS)
ReVelle et al. (2008)	$I = J$	900/900/20	2	28	1	Heuristic concentration
Senne et al. (2010)	$I = J$	3038/3380/22	2	89	1	Lagrangean relaxation with clusters
Zarandi et al. (2011)	$I = J$	2500/2500/25	1	18	1	Genetic algorithm (GA)
Rodriguez et al. (2012)	$I = J$	3038/3380/22	2	89	1	Population-based iterated greedy with large neighbourhood search (PBIGLNS)
Máximo et al. (2017)	$I = J$	7730/7730/100	1	24	1	Intelligent-guided adaptive search (IGAS)
Atta et al. (2018)	$I = J$	2500/2500/25	4	151	2	GA with local refinement (GALR)
Máximo and Nascimento (2019)	$I = J$	7730/7730/100	1	24	1	IGAS with path-relinking
Cordeau et al. (2019)	$I \neq J$	1.5×10^7/100/20	2	840	1	Benders decomposition
Atta (2023)	$I = J$	818/818/14	1	83	1	Harmony search
Islam and Islam (2023)	$I = J$	2500/2500/25	4	151	2	Chemical reaction optimization
Máximo et al. (2023)	$I = J$	7730/7730/100	1	24	1	Adaptive iterated local search
Current paper	$I = J$ & $I \neq J$	7730/7730/100 & 100,000/1000/20	8	601	3	Cross-entropy method (CEM)

I: Set of demand nodes, J: Set of possible facility sites, p: Number of chosen facilities. Max.$|I|/|J|/p$: Maximum dimension of I, J, p tested. BD: Number of benchmark data set, BI: Number of benchmark instances, Net: Number of network types involved.

Due to its essential nature of NP-hardness, much effort has been devoted to devising effective and efficient heuristics and metaheuristics for high quality solutions. The simple greedy search introduced by Church and ReVelle (1974) is by far the most successful heuristic approach to the MCLP and has been incorporated into more sophisticated forms. The performance has been analyzed by Nemhauser et al. (1978), Fisher et al. (1978), and Feige (1998) from a worst-case viewpoint and by Vohra and Hall (1993) from a probabilistic perspective. Although the simple greedy search possesses the best possible worst-case performance in terms of approximation ratios, the swap local search, originally introduced by Church and ReVelle (1974) as an improvement of the simple greedy search, is also a frequently used approach with its approximation ratio, proved by Kerkkamp and Aardal (2016). Moreover, it reliably outperforms the simple greedy search on both randomly generated and real-world instances of the MCLP. The heuristic methods based on Lagrangian and surrogate relaxations and dual formulations have also been developed by Galvão and ReVelle (1996), Galvão et al. (2000), and Jia et al. (2007) and tested on a large number of instances available in the literature.

The interest in heuristics for the MCLP has decreased in the past decades. The main reason is the advancement of commercial solvers, which have become increasingly proficient across a wide range of problem instances (Máximo et al.

2017). However, for large-scale applications, MCLP remains challenging. Various metaheuristic methods have been devised during the last several decades, among which the genetic algorithm (GA) is one of the most popular. GA has been customized in different versions by Arakaki and Lorena (2001), Jaramillo et al. (2002), Jia et al. (2007), Xia et al. (2009a), Zarandi et al. (2011) and Atta et al. (2018). The other metaheuristic methods that have been implemented for the MCLP include simulated annealing (Murray and Church 1996, Xia et al. 2009a,b), tabu search (Adenso-Díaz and Rodriguez 1997, Xia et al. 2009a), GRASP by Resende (1998), Rodriguez et al. (2012), and Máximo et al. (2017), and heuristic concentration by ReVelle et al. (2008) and Senne et al. (2010). According to the computational experiments conducted by Xia et al. (2009a), the simulated annealing method performs robustly and effectively regardless of the instance size of the MCLP.

To the best of our knowledge, only a few works have focused on this problem in recent years. A new metaheuristic, IGAS (Máximo et al. 2017), is developed with the artificial neural network and then updated with Path-relinking techniques by Máximo and Nascimento (2019) and Máximo et al. (2023). The IGAS uses an adaptive memory to guide the search with promising solutions obtained under a GRASP construction framework. Since the learning process is employed, a series of adaptive training is required making the algorithm expensive to be capacitive for more general applications. So far, the most representative GA is given by Atta et al. (2018). The method is designed with a local search strategy which substantially improves the empirical performance compared with previous approaches, especially for large-scale instances. A recent Benders decomposition (Cordeau et al. 2019) has shown strong efficiency on instances with millions of demand nodes. While it draws competitive advantages from highly asymmetric networks where the size of possible facility sets J ($|J| = 100$) are far smaller than demand sets I ($|I| \geq 10,000$), making relatively simple selections within only 100 locations. This partially explains why Benders decomposition can tackle massive instances ($|I| \geq 500,000$) in seconds, yet cannot solve instances (Máximo et al. 2017) with less than 8000 demand nodes within 10 minutes. More recently, presolving methods for covering location problems are presented by Chen et al. (2023). The application to the MCLP shows effectiveness by embedding with branch and cut methods.

In this paper, we present a new formulation of the MCLP and a metaheuristic, cross-entropy method (CEM) to tackle the MCLP. The CEM, a simulation-based metaheuristic, uses an adaptive calibration scheme on a probabilistic model by iteratively generating promising candidate solutions to a problem concerned. It is effective for various combinatorial and continuous optimization problems (Rubinstein 1999, Rubinstein and Kroese 2004, De Boer et al. 2005, Laguna et al. 2009, Caserta and Quiñonez Rico 2009, Ning and Li 2018)). However, the probabilistic

model that yields solutions in a sampling way can be problem-based which may limit applications of the method. To apply it to the MCLP, we designed a Pareto order sampling to generate MCLP solutions under a probabilistic sampling scheme. Moreover, a simple swap local search procedure is proposed in a continuing improvement framework, which is incorporated within the proposed method as a solution fine-tuning procedure. Such a hybrid method has also been applied to the max-cut problem (Laguna et al. 2009), the capacitated facility location problem (Caserta and Quiñonez Rico 2009), and the single row facility layout problem (Ning and Li 2018), respectively.

The remainder of the paper is organized as follows. Section 2 first presents a new formulation of the MCLP using geographic covering properties. Next, a cross-entropy approach is introduced in Section 3 with the new constructive method and local search. In Section 4, the method is calibrated with a ParamILS algorithm (Hutter et al. 2009) and evaluated with the most complete benchmark data sets. Finally, conclusions and discussions are addressed in Section 5.

2. Formulation of the MCLP

The MCLP is defined on a network $G(V, E)$ consisting of a location set $V = I \cup J$ and set E of edges connecting demand nodes of set I and possible facility locations of set J. Each node $i \in I$ is associated with a nonnegative demand w_i that may represent, for example, the population residing at this node. Without loss of generality, coverage radius $r \geq 0$ is predefined beyond which, i is not covered by a facility $j \in J$ with a traveling distance/time d_{ij}. Consequently, for each demand node $i \in I$, a facility set $N_i = \{j \in J : d_{ij} \leq r, (i, j) \in E\}$ is defined with eligible facilities that can serve this demand node. On a budget, the number of facilities chosen from J is p ($p > 0$). The MCLP is, therefore, to select p facilities from J to cover as many demands in I as possible within the service radius r. Under these settings, Church and ReVelle (1974) has formulated the MCLP as a 0-1 integer linear programming:

$$\max \quad \sum_{i \in I} w_i y_i \tag{1}$$

$$\text{s.t.} \quad \sum_{j \in N_i} x_j \geq y_i, \qquad \forall i \in I \tag{2}$$

$$\sum_{j \in J} x_j = p \tag{3}$$

$$x_j, y_i \in \{0, 1\}, \qquad \forall i \in I, j \in J \tag{4}$$

where $x_j = 1$ if a facility is sited at position j, and $x_j = 0$ otherwise. Moreover, $y_i = 1$ if demand node i is covered by at least one facility, or not when $y_i = 0$. Such a formulation benefits from relaxation techniques, like linear programming or Lagrangian relaxations (Downs 1991, Downs and Camm 1996).

Even though not compulsory, selecting p facilities entirely from demand nodes, has been a popular setting in the MCLP literature. Indeed, this setting enhances the potential for identifying better facilities while it may lead to exponentially complex problems when the number of possible facility sites is quite large or even massive. An alternative MCLP setting have scaled down the size of potential sites to chose p facilities within an independent set J with a smaller size comparing with demand set I, i.e., $|I| \gg |J|$ ($J \notin I$). This setting applies to more realistic and complex MCLP applications, making it possible for an extreme case with 20 million demand nodes in Cordeau et al. (2019). Technically, this setting is not limited to $J \notin I$ where possible sites in J can be regarded as zero-demand nodes of I, leading to $J \in I$. Such features also benefit from encoding into and retrieving from the network structure and the sets of eligible facility sites for demand nodes. This issue is discussed in Berman et al. (2010) and references therein.

Let us consider such a common and useful case where massive demand nodes in I need to be maximally covered by p facilities selected from a relatively small set of possible facility J. Firstly, we divide demand nodes $i \in I$ into $|J| + 1$ subset $I^R = \{i \in I : R = |N_i|\}$ where rank R, ranging from 0 to $|J|$ indicates the number of possible sites that can cover demand node i. Consequently, $I = \bigcup_{R=0}^{|J|} I^R$ and $I^a \cap I^b = \emptyset$ for any a, b $(a \neq b) \in \{0, \cdots, |J|\}$. Subsequently, we analyze subsets I^R for more insights: 1) For any demand node $i \in I^0$, i cannot be serviced, as no potential sites in J are accessible; 2) For $i \in I^1$, i can be serviced if and only if its corresponding facility site is chosen as one of the facility; 3) For $i \in I^{R \geq 2}$, the demand has no unique correspondence to a single facility site. It is intuitive to probe the second case where I^1 may include more demand nodes than higher-rank subsets, $I^{R \geq 2}$, due to the limited service distance and sparse facility sites. In particular, as node $i \in I^1$ can be uniquely covered by a single facility $j \in J$, its demand can be summarized to facility site j according to Definition 1. Thereby, the cumulative demand w'_j of each possible facility site $j \in J$ is:

$$w'_j = \sum_{i \in I^1, d_{ij} \leq r} w_i, \quad \forall j \in J \tag{5}$$

and the demand w'_j can be serviced if and only if site j is chosen as a facility ($x_j = 1$).

Definition 1. *The cumulative demand for a facility $j \in J$ is the summary demand of all nodes $i \in I$ that can be only covered by a single facility j.*

For ease of notation, we denote the union of sets $I^{R\geq 2}$ to be I'. Using new parameters w'_j and set I', a linear MCLP model aMCLP is formulated as follows:

$$\max \quad \sum_{j\in J} w'_j x_j + \sum_{i\in I'} w_i y_i \tag{6}$$

$$\text{s.t.} \quad \sum_{j\in N_i} x_j \geq y_i, \qquad \forall i \in I' \tag{7}$$

$$\sum_{j\in J} x_j = p \tag{8}$$

$$x_j, y_i \in \{0,1\}, \qquad \forall i \in I', j \in J. \tag{9}$$

This model is equivalent to the original proposal of Church and ReVelle (1974). Especially, if facilities are selected entirely from the demand nodes, the new formulation simplifies to the model of Church and ReVelle (1974). Here, each possible site covers at least one node, especially the node itself, leading to $I^0 = \emptyset$ and $I^1 = \{i \in I : N_i = \{i\}\}$. By summarizing all cumulative serviced w'_j, it obtains

$$\sum_{j\in J} w'_j x_j = \sum_{i\in I^1} w_i y_i. \tag{10}$$

Consequently, it meets $\sum_{j\in J} w'_j x_j + \sum_{i\in I'} w_i y_i = \sum_{i\in I} w_i y_i$. Note that aMCLP considers a simplified covering network $G'(I' \cup J, E')$ where E' is the set of edges between new set I' and J. Technically, the elimination of nodes has been accepted as a pre-solving method prior to applying the solving algorithm, e.g., the cross-entropy method in this paper.

3. Cross-entropy method for the MCLP

The cross-entropy method (CEM), originally proposed for rare event estimation, has been shown by Rubinstein (1997, 1999) to be a generic simulation-based approach to combinatorial and continuous optimization. Roughly speaking, the CEM for an optimization problem first constructs an associated rare event estimation problem and then applies an importance sampling-based adaptive procedure iteratively where each iteration consists of two essential steps: (1) solution generation according to a random mechanism specified by a parameterized probability distribution, (2) distribution parameter updating for the next iteration based on the generated solution sample. Since the most important features of the CEM have been addressed in Rubinstein and Kroese (2004) and De Boer et al. (2005), the technical details are introduced in this section only for those relevant to the MCLP.

3.1. Transformation of the MCLP to a rare event estimation

Let the binary vector $\boldsymbol{x} = (x_1, x_2, \cdots, x_n)^T$ denote a configuration of facility locations where $n = |J|$ and $x_j = 1$ if a facility is sited at node j, and $x_j = 0$ otherwise. To be feasible solutions, the vector $\boldsymbol{x}$ should belong to the solution set $C_{n,p}$, defined as

$$C_{n,p} = \{\boldsymbol{x} = (x_1, \cdots, x_n)^T \in \{0,1\}^n : \sum_{j=1}^n x_j = p\}. \tag{11}$$

Instead of directly calculating the total covered demand, we opt to minimize the uncovered demand, expressed as $S(\boldsymbol{x}) = \sum_{i\in I} w_i - \sum_{i\in I} w_i y_i$. This is a common practice in the MCLP literature (Church and ReVelle 1974, Lorena and Pereira 2002) as there are typically far fewer uncovered demand nodes than covered ones. Consequently, we consider such a problem, $\min\{S(\boldsymbol{x}) : \boldsymbol{x} \in C_{n,p}\}$ and $S(\boldsymbol{x})$ can be calculated by

$$S(\boldsymbol{x}) = \sum_{i\in I} (1 - \bigvee_{j\in N_i} x_j) w_j \tag{12}$$

where $\bigvee$ is the maximum operation, or equivalently, the logical disjunction, with the convention that $\bigvee \emptyset = 0$. Suppose that $\boldsymbol{x}^*$ is an optimal solution and $s^* = S(\boldsymbol{x}^*)$. Let be given a random vector $\boldsymbol{X} = (X_1, X_2, \cdots, X_n)^T$ on the finite solution space $\boldsymbol{x} \in C_{n,p}$. It transforms the MCLP to an associated rare event estimation problem

$$\ell(\boldsymbol{\pi}) = P_{\boldsymbol{\pi}}(S(\boldsymbol{X}) \leq s^*) \tag{13}$$

where $P_{\boldsymbol{\pi}}$ is a probability measure and $\boldsymbol{\pi}$ is the inclusion probabilities of $\boldsymbol{X}$ in

$$\Pi_{n,p} = \{\boldsymbol{\pi} = (\pi_1, \ldots, \pi_n) \in [0,1]^n : \sum_{j=1}^n \pi_j = p\}. \tag{14}$$

It exists a probability mass function (PMF) $f(\boldsymbol{x}, \boldsymbol{\pi}), \boldsymbol{x} \in C_{n,p}, \boldsymbol{\pi} \in \Pi_{n,p}$ on $C_{n,p}$ that its inclusion probabilities of variables X_j are equal to π_j for $j = 1, \cdots, n$. In the case of MCLP, the PMF involves the first order inclusion probability of a possible facility site being one of the p chosen facilities indicating $f(\boldsymbol{x}; \boldsymbol{\pi}) = P(X_1 = x_1, X_2 = x_2, \cdots, X_n = x_n)$ for each $\boldsymbol{x} = (x_1, x_2, \cdots, x_n)^T \in C_{n,p}$. Therefore, the inclusion probability π_j of site $j \in J$ is actually the sum of the point probability over solution $\boldsymbol{x} \in C_{n,p}$:

$$\pi_j = \sum_{\boldsymbol{x}\in C_{n,p}: x_j = 1} f(\boldsymbol{x}, \boldsymbol{\pi}), \quad \forall j = 1, \cdots, n. \tag{15}$$

In general, the first order inclusion probabilities meet $\sum_{j=1}^n \pi_j = p$ with the average probability being p/n.

The event $S(\boldsymbol{X}) \leq s^*$, with s^* being the optimal objective value, is typically

assumed to be rare. An iterative importance sampling strategy may be applied to estimate its probability and simultaneously obtain the optimal or near-optimal configuration of facility locations.

3.2. Pareto order sampling for solution generation

To launch such a simulation-based procedure, it is crucial to design an appropriate sampling method to draw solutions to the MCLP from the set $C_{n,p}$, which is exactly the problem known as weighted sampling or unequal probability sampling without replacement. More specifically, given $\boldsymbol{\pi} = (\pi_1, \pi_2, \cdots, \pi_n)^T$ as the vector of prescribed first order inclusion probabilities defined in (15), such an unequal probability sampling without replacement requires a particular sampling scheme to construct a random vector $\boldsymbol{X} = (X_1, X_2, \cdots, X_n)^T$ over $\boldsymbol{x} \in C_{n,p}$ such that for each $j \in J$, it obtains

$$P(X_j = 1) = \pi_j \quad \text{and} \quad P(X_j = 0) = 1 - \pi_j. \tag{16}$$

So far, various methods have been developed for unequal probability sampling; see, e.g., Tillé (2006) for a detailed overview. However, an exact sampling scheme is not a requisite in this particular context since an optimal solution is ultimately desired but not a solution sample to solve the MCLP. Considering that the sampling scheme will be repeatedly called in this method with iteratively modified first-order inclusion probabilities, an approximate but rapid method is preferred. Pareto order sampling or Pareto sampling developed by Rosén (1997a,b), serves this purpose.

Pareto order sampling is an approximation method of the exact sampling scheme with unequal probabilities. It satisfies three principles of good approximation (Rosén 1997b): simplicity, estimation precision, and variance estimation. Algorithm 1 presents the basic procedure. It first generates n independent uniformly distributed random numbers $U_j \sim U(0,1)$, $j \in J$, and then calculate the corresponding ranking values

$$Q_j = \frac{U_j/(1-U_j)}{\pi_j/(1-\pi_j)}, \quad \forall j \in J. \tag{17}$$

Subsequently, the exact p random variables X_j's with the smallest ranking values are assigned the value 1. Note that the name Pareto order sampling follows from the fact that $U_j/(1-U_j)$ is a Pareto distributed random variable when $U_j \sim U(0,1)$. The quality of Pareto approximation is affected by $\sum_{j\in J} \pi_j(1-\pi_j)$, where the larger the value is, the better the approximation (Aires 1999). While the Pareto order sampling is proven to be quite robust even if this value is rather small (Aires 2000, Bondesson et al. 2006), and advanced techniques that improve the Pareto sampling can be found therein.

Algorithm 1 : Pareto order sampling for unequal probability sampling without replacement

Input: a vector of first order inclusion probabilities $\boldsymbol{\pi} = (\pi_1, \pi_2, \cdots, \pi_n)^T$
Output: a random binary vector $\boldsymbol{x} = (x_1, x_2, \cdots, x_n)^T \in C_{n,p}$ with $x_1 + x_2 + \cdots + x_n = p$

1: Generate independently $U_j \sim U(0,1)$, $j = 1, 2, \cdots, n$.
2: Calculate $Q_j = \frac{U_j/(1-U_j)}{\pi_j/(1-\pi_j)}$ for $j = 1, 2, \cdots, n$.
3: Rank the values of Q_j's in an increasing order such that $Q_{(1)} \leq Q_{(2)} \leq \cdots \leq Q_{(n)}$.
4: Initiate $\boldsymbol{x} = (x_1, x_2, \cdots, x_n)^T$ as a zero vector.
5: Set the element $x_{(j)} = 1$ corresponding to $Q_{(j)}$ for $j = 1, 2, \cdots, p$.
6: **return** $\boldsymbol{x} = (x_1, x_2, \cdots, x_n)^T$

3.3. Parameters tuning for first order inclusion probabilities

By the importance sampling method, the probability $\ell(\boldsymbol{\pi})$ can be estimated by drawing a solution sample $\boldsymbol{X}_1, \boldsymbol{X}_2, \cdots, \boldsymbol{X}_N$, of size N with respect to a different probability mass function $f(\boldsymbol{x}; \boldsymbol{u})$ so that

$$\widehat{\ell}(\boldsymbol{\pi}) = \frac{1}{N} \sum_{k=1}^{N} I_{\{S(\boldsymbol{X}_k) \leq s^*\}} \frac{f(\boldsymbol{X}_k; \boldsymbol{\pi})}{f(\boldsymbol{X}_k; \boldsymbol{u})} \tag{18}$$

where $I_{\{\cdot\}}$ is the indicator function and $\widehat{\ell}(\boldsymbol{\pi})$ is called the importance sampling estimator or alternatively the likelihood ratio estimator. The vector $\boldsymbol{u}$, called the reference parameter corresponding to s^*, may be chosen to minimize the cross-entropy between $f(\boldsymbol{x}; \boldsymbol{\pi})$ and $f(\boldsymbol{x}; \boldsymbol{u})$ by solving

$$\max_{\boldsymbol{u}} \quad \frac{1}{N} \sum_{k=1}^{N} I_{\{S(\boldsymbol{X}_k) \leq s^*\}} \ln f(\boldsymbol{X}_k; \boldsymbol{u}). \tag{19}$$

According to De Boer et al. (2005), the empirically optimal rule may be used to set the reference parameter as

$$\boldsymbol{u}^* = \frac{\sum_{k=1}^{N} I_{\{S(\boldsymbol{X}_k) \leq s^*\}} \boldsymbol{X}_k}{\sum_{k=1}^{N} I_{\{S(\boldsymbol{X}_k) \leq s^*\}}}. \tag{20}$$

It is clear that $\boldsymbol{u}^*$ is a legitimate approximation of first-order inclusion probabilities $\boldsymbol{\pi}$ where the elements of each $\boldsymbol{X}_k$ sum up to p. However, it cannot be implemented directly because the value s^* is unknown as a priori.

Alternatively, starting with an initial solution sample, a sequence of decreasing

threshold values $\{s^t\}_{t\geq 1}$ may be constructed iteratively using sample quantiles, which converges either to the value or close to s^*. To be specific, once a solution sample $\boldsymbol{X}_1^t, \boldsymbol{X}_2^t, \cdots, \boldsymbol{X}_N^t$, is available at the t-th iteration via Pareto order sampling, the corresponding objective values can be ranked in an increasing order such that $S_{(1)} \leq S_{(2)} \leq \cdots \leq S_{(N)}$. The threshold value s^t is then set as

$$s^t = S_{(\lceil \rho N \rceil)} \tag{21}$$

where $\rho \in (0, 1)$ is the rarity parameter used to determine the ρ-th quantile of the sampled objective values and $\lceil \rho N \rceil$ denotes the integer rounding up on ρN. Subsequently, the empirically optimal rule is used to calculate

$$\widehat{\boldsymbol{u}}^t = \frac{\sum_{k=1}^N I_{\{S(\boldsymbol{X}_k^t) \leq s^t\}} \boldsymbol{X}_k^t}{\sum_{k=1}^N I_{\{S(\boldsymbol{X}_k^t) \leq s^t\}}} \tag{22}$$

by which it extracts from the current solution sample the features of the best $\lceil \rho N \rceil$ ones, called the elite solutions. Instead of using $\widehat{\boldsymbol{u}}^t$ directly as the reference parameter corresponding to s^t, the smoothed version

$$\boldsymbol{u}^t \ = \alpha \widehat{\boldsymbol{u}}^t + (1 - \alpha) \boldsymbol{u}^{t-1} \tag{23}$$

is applied for the next iteration, with a predetermined smoothing parameter $\alpha \in (0, 1)$, and $\boldsymbol{u}^0 = \boldsymbol{\pi}^0$ is predefined. In this manner, $\boldsymbol{u}^t$ balances the information learned from the current solution sample, and that inherited from the previous iteration and is a legitimate vector of first-order inclusion probabilities. Whereas, such a procedure may lead to premature convergence in later iterations, where the vector $\boldsymbol{u}^t$ could regress to $\boldsymbol{u}^{t-1}$, especially when the variation among elite solutions is small. To mitigate this, we introduce a perturbation mechanism into the CEM. It places a random unchosen site as a facility before the sampling on the condition that the best solution remains unchanged for $Cmax/2$ iterations. Here, $Cmax$ is the maximal iteration that the best solution is retained, and $p - 1$ facilities are selected from Algorithm 1. Consequently, the procedure of Pareto order sampling is ready to be called with the input $\boldsymbol{u}^t$ to generate a new solution sample for the next iteration of the CEM. Such an adaptive procedure, along with the elitist strategy introduced in Section 3.4, continues until certain stopping criteria are activated and the best solution ever found is output as the solution to the MCLP.

3.4. Local search heuristic

As described in Section 3.3, elite solutions play an essential role in the method. To ensure the overall quality of the generated solution sample, the incumbent elite solutions may be retained to replace the worst $\lceil \rho N \rceil$ ones in the newly generated

solution sample. Such an elitist strategy may further be boosted by implementing a local search procedure on the elite solutions, which helps to identify the optimal or near-optimal solutions, particularly for large-size instances. However, given the elite solutions are of good quality and some may be regenerated and labeled as elite solutions, exhaustive local search schemes on these elite solutions could consume significantly more CPU time than solution sampling, potentially leading to an iterated local search.

Instead, we propose a simple swap neighborhood local search in Algorithm 2. The method is inspired by the coverage set substitute of the alternating algorithm (Maranzana 1964) for the p-median problem. More specifically, each chosen facility $c \in J$ is swapped with an unchosen site $j \in U_c$ that covers the maximum unserviced demand after c is removed (setting $x_c = 0$). To narrow down the swap space, we restrict j to the unchosen sites that are distant from other selected facilities, defined as $U_c = \{j(\neq c) \in J : x_j = 0, d_{c'j} > r, \forall c' \in S \backslash c\}$ where S is the set of chosen facilities. Facility sites U_c represent the unserviced nodes, especially if facilities have totally come from the demand nodes. For each $j \in U_c$, the uncovered demand unique to j, denoted as Δj, is then calculated as follows:

$$\Delta j = \sum_{i \in I: d_{ij} \leq r} w_i (1 - y_i^c), \quad \forall j \in U_c \tag{24}$$

where y_i^c indicates whether i remains serviced without facility c, defined as $y_i^c = \bigvee_{j \in N_i: j \neq c} x_j, \forall i \in I$. The local search, incorporating global improvements, continues after one run of neighborhood search until no further gains can be achieved throughout the whole procedure.

Note that such a local search procedure is at a similar complicity of Berman et al. (2009), Atta et al. (2018), Máximo et al. (2017), Máximo and Nascimento (2019). Differently, the adaptive swapping space is narrowed into U_c, resulting in a faster local search than an entire neighborhood swapping. Taking into account the elitist strategy and the local search scheme, the parameters specified in Section 3.3 are updated accordingly to incorporate the adaptive tuning for probabilities from the solution samples.

Algorithm 2 : Swap neighborhood local search

Input: solution $\boldsymbol{x}$ and current objective value $S(\boldsymbol{x})$
Output: the updated $\boldsymbol{x}$ and $S(\boldsymbol{x})$

```
Improvement = true;
while Improvement == true do
    Improvement = false;
    for each chosen facility c do
        x* = swap values of c with arg_{j*∈U_c} max Δj*;
        if S(x) is strictly improved then
            Improvement = true;
            x = x*, S(x) = S(x*);
```

As illustrated in Algorithm 3, the main procedure of the proposed CEM is simple. To launch the method, the parameters involved should be initialized first. The initial internal probabilities in the vector of first order inclusion probability $\boldsymbol{\pi^0} = (\pi_1, \pi_2, \cdots, \pi_n)^T$ where $\pi_j = p/n, \forall j = 1, \cdots, n$. The remaining parameters that matter in the sampling and elitist policy are carefully calibrated in Section 4.2. The iteration starts with sampling N solutions population using Pareto order sampling in the main loop. The top ρN elites are adopted to update the inclusive probability. It is worth noting that the memory of solutions after the local search will be retained in $\boldsymbol{\pi}$, as elite solutions have been promoted and updated before the probability updating. These procedures iterate until the maximum absolute change in π between the two successive iterations is smaller than 10^6 or the threshold s^t remains unimproved for $Cmax$ successive iterations.

Algorithm 3 : Cross-entropy method

Input: Initialized data: π, N, ρ, nLS, α, $Cmax$
Output: the best solution s^*

```
while the stopping criterion is not met do
    Population ← Pareto order sampling (π, N, ρ);
    Elites ← Top ρN solutions of Population;
    s*, Elites ← Local search on the best nLS solutions of Elites;
    π ← Inclusion probabilities updating (Elites, π, ρ, Cmax);
```

4. Computational study

Two computational experiments are executed on the new MCLP formulation and the proposed CEM. We first compare the numbers of variables and constraints of the new model with that of Church and ReVelle (1974). Next, the proposed CEM is calibrated with a ParamILS algorithm (Hutter et al. 2009) evaluated with

benchmark instance. The CEM is coded in C# using Microsoft Visual Studio 2019 and .Net 6.0. All computations are randomly distributed to the servers running on PROXMOX Virtualization Environment with 4 cores and 16 GBytes memory powered by four AMD Opteron Abu Dhabi 6344 processors running at 2.6 GHz and with 256 GBytes of RAM without any parallel coding or computing.

4.1. Benchmark instances

Table 2 summarises the main attributes of all popular data sets involved for the MCLP. It includes random benchmark data $G\&R$ (Galvão and ReVelle 1996, Galvão et al. 2000), *Beasley* (Beasley 1990), ZDS (Zarandi et al. 2011), PCB (Reinelt 1994), and real-world benchmark data SJC (Lorena and Pereira 2002), *Maximo* (Máximo et al. 2017), as well as massive benchmark data BDS (Cordeau et al. 2019) and data $BDS1000$ extended in this research.

Table 2: The most popular benchmark instances of the MCLP

Data	Data sources	*Sets*	*BI*	*Nodes*	Location	Demand	Network
MCLP with $I = J$:							
$G\&R$	Galvão and ReVelle (1996) and Galvão et al. (2000)	2	32	$\{100, 150\}$	$U(20, 30)$ $U(40, 60)$	$N(80, 15)$	Floyd
Beasley	Beasley (1990)	2	18	$\{700, 900\}$	Network	$N(80, 15)$	Floyd
SJC	Lorena and Pereira (2002)	5	83	$\{324, 402, 500, \}$ and $\{708, 818\}$	Network	Given	Euclidean
ZDS	Zarandi et al. (2011)	2	18	$\{1800, 2500\}$	$U(0, 30)$	$U(0, 100)$	Euclidean
PCB	Reinelt (1994)	1	6	$\{3038\}$	Network	$U(0, 100)$	Euclidean
Maximo	Máximo et al. (2017)	4	24	$\{2713, 3839, \}$ and $\{5137, 7730\}$	Network	Given	Geographic
MCLP with $I \neq J$	:						
BDS	Cordeau et al. (2019)	3x5	15x14	$\{10000, 50000, 100000\}$	$U(0, 30)$	$U(0, 100)$	Euclidean
$BDS1000$	This work	3x5	15x14	$\{10000, 50000, 100000\}$	$U(0, 30)$	$U(0, 100)$	Euclidean

U: uniform distribution, N: normal distribution, *Sets*: Number of sets, *BI*: Number of benchmark instances
Nodes: Number of demand nodes, Floyd: Floyd's distance.

In detail, the random sets $G\&R100$ and $G\&R150$ firstly tested by Galvão and ReVelle (1996) and Galvão et al. (2000) are pre-established with network densities δ, the ratio of arc numbers between the network and its complete network, of 0.1 and 0.05. The arc length of $G\&R$ sets is randomly generated from the uniform distribution $U(40, 60)$. Whereas the demand follows $U(20, 30)$ for $G\&R100$ and normal distribution $N(80, 15)$ for $G\&R150$. Besides, the locations of *pmed*32 and *pmed*39 of *Beasley* data are used for sets $B700$ and $B900$ where the demand meets $N(80, 15)$ as well. Note that Floyd's algorithm (Shier 1981) is utilized to calculate the distance for $G\&R$ and *Beasley* data. The Euclidean coordinates of nodes in the large-scale data sets $ZDS1800$ and $ZDS2500$ are randomly generated by two-dimensional Cartesian coordinates $(x, y) \in U(0, 30)$ with the demand of $U(0, 100)$, the same as $PCB3038$ obtained from the $TSPLIB$ (Reinelt (1994)).

Data SJC and $Maximo$ come from real-world geo-referenced databases. All plane coordinates of sets $SJC324$, $SJC402$, $SJC500$, $SJC708$, and $SJC818$ represent the blocks of São José dos Campos city, Brazil. Their demands are determined by the number of public facilities taken from the geo-referenced database available on the website http://lac.inpe.br/∼lorena/instancias.html. Particularly, data of $Manhattan2713$, $Bronx3839$, $SanFrancisco5137$ and $Kings7730$ are extracted from block streets of four regions of the United States. Using latitude (lat), longitude (lon) the involved distance between node i and j is calculated by:

$$d_{ij} = arccos(cos(lat_i)cos(lat_j)cos(lon_j - lon_i) + sin(lat_i)sin(lat_j)) \times R_e \quad (25)$$

where R_e is the ratio of the earth, adopted as 6,378,100. The data is available at https://sites.google.com/site/ nascimentomcv/downloads/mclp.

A massive MCLP set BDS is first presented in Cordeau et al. (2019). In which 3 sizes of instances reaching 100,000 demand nodes were generated for 5 replicates with each containing 14 instances following the same rule as ZDS. While this dataset is highly realistic, its application is limited to facility options within only 100 possible sites across all of Cordeau et al. (2019)'s sets. In this study, we have extended the BDS to $BDS1000$ for more applications with 10 times of possible facility sites. Original BDS can be found in Cordeau et al. (2019) are available from https://github.com/fabiofurini/LocationCovering.

In summary, we collected 8 benchmark data of 3 types covering networks and 2 MCLP settings with 181 small and large instances, as well as 420 massive instances. Each instance is executed for 30 runs for a grand total of $(181 + 420) \times 30 = 18030$ experiments. With this many results, we are able to provide an accurate picture of the effectiveness of the proposed CEM. All the benchmark data of this study are open at https://github.com/HWangUPV.

4.2. Calibration for the method

One feature of the presented CEM is that it has flexible parameters, i.e., sample size N, elite quantile ρ, runs of local search nLS, smoothing parameter α and terminating number $Cmax$. Since the sampling efficiency is highly related to the instance size (De Boer et al. 2005), which ranges from 100 nodes to 100,000 nodes, 2 configurations of parameters are suggested in this study for simple and complex cases. To meet the instance nature, groups of instances are reconstructed 10 times with the formulation in Table 2, based on simple instances GR150-12-85($p = 12, r = 85$), B900-24-13, SJC708-7-800, ZDS2500-20-3.75, PCB3038-19-400, and the other complex instances Manhattan2713-80-400, SanFrancisco5137-70-600, BDS10000-10-5.5, BDS50000-15-4 and BDS100000-20-3.5. In which the demand of real-world instants is given from $U(0, 100)$, similar to (Zarandi et al. 2011). Finally, a total of 100 replicates of instances were generated for the parameter calibration.

Building upon previous studies (Rubinstein 1999, Maher et al. 2013, Ning and Li 2018), following intervals are adopted:

- $N \in \{200, 400, 600, \cdots, 2000\}$
- $\rho \in \{0.005, 0.01, 0.02, \cdots, 0.1\}$
- $nLS \in \{1, 2, 4, \cdots, 20\}$
- $\alpha \in \{0, 0.1, 0.2, \cdots, 1\}$
- $Cmax \in \{5, 10, 15, \cdots, 50\}$

The parameters tuning employs the ParamILS algorithm (Hutter et al. 2009) as an independent calibration approach beyond the proposed CEM. An Iterated Local Search (ILS) metaheuristic is used in ParamILS to explore the configuration space combined with full options of parameters. Starting from an initial configuration, ParamILS employs a one-exchange neighborhood search strategy to find the next configuration. This strategy works by varying the value of this fixed parameter within interval sets. ParamILS also introduces perturbation to a fixed number of parameters (three, as suggested by Hutter et al. (2009)) to escape the local optima. Both the parameters and their values are selected randomly. Each calibration instance runs 5 times with 10s and 20s time limits for 2 groups of instances respectively, and the average coverage proportion is adopted to evaluate the quality of the parameter configuration. As a result of the ParamILS algorithm, parameters configuration $N = 2000$, $\rho = 0.05$, $nLS = 20$, $\alpha = 0.8$, $Cmax = 30$ is adopted for *G&R*, *Beasley*, SJC, ZDS, PCB data sets, and $N = 200$, $\rho = 0.01$, $nLS = 4$, $\alpha = 0.4$, $Cmax = 20$ for the rest benchmark instances.

4.3. Comparison of the proposed MCLP model

This section compares the proposed mathematical formulation aMCLP in Section 2 with the original model (Church and ReVelle 1974). In regard to the complexity of mathematical formulations, Table 3 exhibits the numbers of variables and constraints of the aMCLP model and Church's model. Both formulations were examined on IBM ILOG CPLEX 20.1 with BDS instances Cordeau et al. (2019) with a CPU time limit of 600 seconds. It turns out that the aMCLP results in fewer variables and constraints than Church's model. At the same radius level, the aMCLP reduced a maximum of 17.27% variables for BDS instances. However, also note that this reduction effect gradually weakens with a bigger radius. On average, the proposed model accounts for 4.72% simplification than the original model of Church and ReVelle (1974) on variables and constraints for all BDS instances.

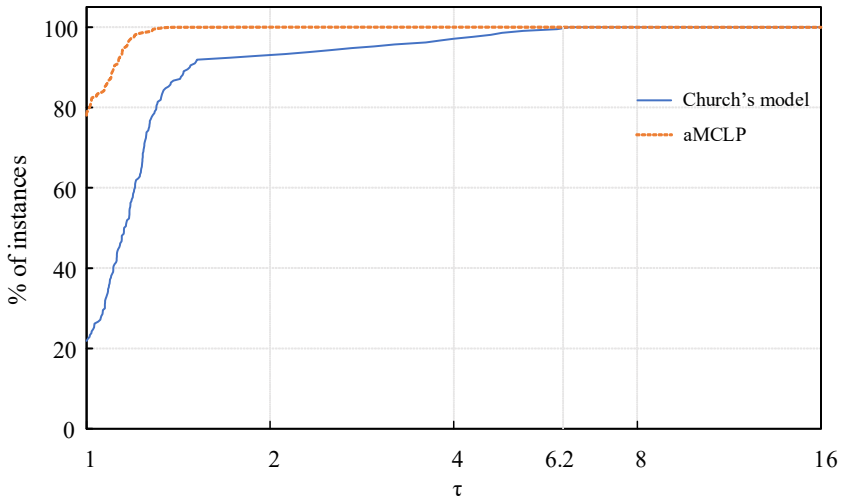

Figure 1: Performance profile for the MCLP formulations on BDS instances with $|J| \in \{10,000; 50,000; 100,000\}$.

Figure 1 displays a graphical representation of the relative performance profiles of two formulations above on all BDS instances. For each instance, we compute a normalized time τ in the horizontal axis (Cordeau et al. 2019, Chen et al. 2023), as the ratio of the computing time of the considered formulation to the best one using the minimum time. The graphic illustrates how the percentage of instances varies by allowing τ times more CPU time than the fastest formulation where $\tau = 1$ reports exactly the percentages of considered formulation outperforming the other and the ends of curves present percentages of instances solved optimally. The new formulation outperforms the Church's model in 78.10% of the corresponding instances, and in reverse, the number is only 21.90%. The new formulation achieves 100% optimal solutions much faster, although the other formulation can also solve all instances if given 6.2 times more CPU time. The new formulation reaches 100% optimal solutions much quicker, even though the other is also capable of all instances, which is even better CPLEX results than Cordeau et al. (2019), when 6.2 more times of CPU time is allowed to Church's formulation. We explain that the new formulation benefits from the reduction of variables and constraints, while improvements are bounded by the coverage radius as aforementioned.

Interestingly, the aMCLP in Section 2 also enlightens a clustering reduction procedure (CRP). The process involves simplifying the covering network by clustering techniques where the less-connected demand has to be accumulated into possible facilitate nodes. The involved demand nodes can be removed from the network. This procedure integrated into the aMCLP also works as the data pre-processing method applied before metaheuristics, e.g., CEM in this study. The CRP is easy to

apply by removing a set of demand nodes of rank $R = 0$ and diminishing demand nodes of rank 1 by accumulating demand to the corresponding facilities sites. To illustrate this procedure, Figure 2 shows covering maps with a massive instance of 10,000 demand nodes with $|J| = 100$ and $r = 2.5$. Eventually, the clustering strategy simplifies the covering network to 5970 demand nodes in which sizes of enlarged possible site locations have their radius scaled up with accumulated demands. Recently, similar procedures have been applied as presovling methods in Chen et al. (2023).

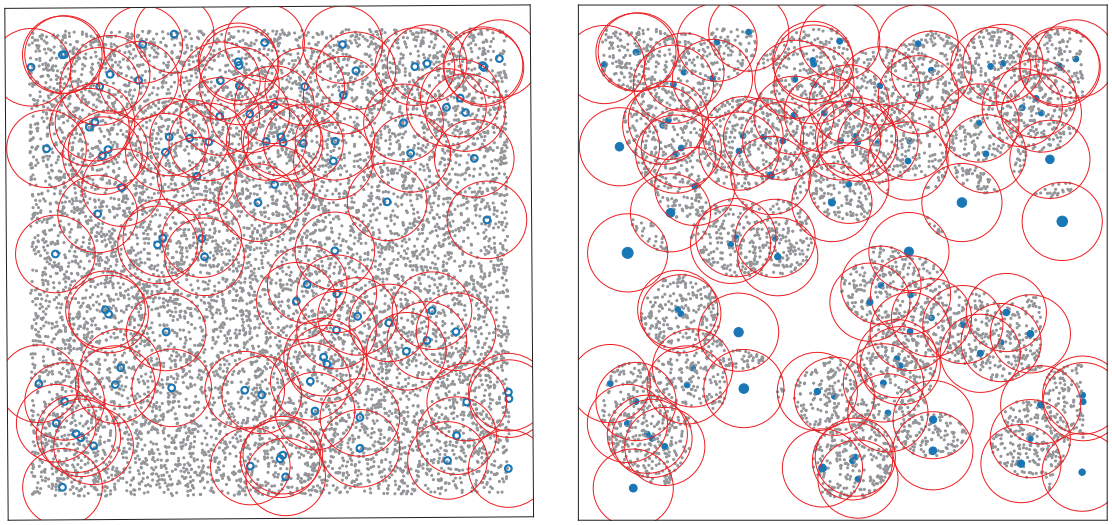

Figure 2: Scatter plots of MCLP with $|I| \gg |J|$ before (left, $\#nodes : 10000$) and after (right, $\#nodes : 5970 + 100$) clustering reduction procedures on a BDS instance (Cordeau et al. 2019) with $|I| = 10000, |J| = 100, r = 2.5$.

Table 3: The number of variables ($\#Var$) and constraints ($\#Cons$) of the aMCLP in Section 2 and the model of (Church and ReVelle 1974) on BDS instances (Cordeau et al. 2019)

	n=10000				n=50000				n=100000			
	Church's Model		aMCLP		Church's Model		aMCLP		Church's Model		aMCLP	
r	#Var	#Cons	#Var	#Cons	#Var	#Cons	#Var	#Cons	#Var	#Cons	#Var	#Cons
3.25	10101	10002	8404.6	8305.6	50101	50002	41497.2	41398.2	100101	100002	82830.4	82731.4
3.5	10101	10002	8900.8	8801.8	50101	50002	43993	43894	100101	100002	87879.6	87780.6
3.75	10101	10002	9264	9165	50101	50002	45806	45707	100101	100002	91493.6	91394.6
4	10101	10002	9504.8	9405.8	50101	50002	47055.8	46956.8	100101	100002	93980.6	93881.6
4.25	10101	10002	9679	9580	50101	50002	47933.8	47834.8	100101	100002	95727.4	95628.4
4.5	10101	10002	9800.2	9701.2	50101	50002	48546.6	48447.6	100101	100002	96965.6	96866.6
4.75	10101	10002	9885.2	9786.2	50101	50002	49000	48901	100101	100002	97886.4	97787.4
5	10101	10002	9949.6	9850.6	50101	50002	49340.2	49241.2	100101	100002	98573.8	98474.8
5.5	10101	10002	10029.8	9930.8	50101	50002	49750.8	49651.8	100101	100002	99402.8	99303.8
5.75	10101	10002	10054.4	9955.4	50101	50002	49872.2	49773.2	100101	100002	99644.4	99545.4
6	10101	10002	10073.6	9974.6	50101	50002	49955	49856	100101	100002	99807	99708
6.25	10101	10002	10082.6	9983.6	50101	50002	50010	49911	100101	100002	99915.2	99816.2
Average	10101	10002	9635.717	9536.717	50101	50002	47730.05	47631.05	100101	100002	95342.23	95243.23

4.4. Computing results

Comprehensive experiments for MCLP evaluate the performance of the proposed metaheuristic, CEM. All code and benchmark data used in the experiments are available in the GitHub repository (Wang and Zhou 2025). Detailed computational results can be found in the online materials, specifically in Tables A1 to A9. The best, worst, and median results of 30 executions, and the average time (AT) to obtain final solutions, are presented in the last block. In other blocks, results from other referred proposals, as well as the latest CPLEX 20.1 solver are presented. Namely, LRS (Lorena and Pereira 2002), GA (Zarandi et al. 2011), PBIGLNS (Rodriguez et al. 2012), IGAS (Máximo et al. 2017) and GALR (Atta et al. 2018)). Note that the results of CPLEX 20.1 were obtained in the same computing environment with time limits of 20,000s for the best potential. Besides, only results from identical instances are compared in which non-inferior results of the best column are marked in bold, and results with optimal guarantees are labeled with '*'.

4.4.1. Results of small data sets

Table **A1** illustrates the optimal objectives (Optimal) and maximal coverage (Cov.%) of CPLEX and the best results in coverage (Best) and the average elapsed time reported (AT) for achieving them. The last block represents the best, worst and median coverage (%) and the average time to obtain final solutions within the time limit of 300s. Notably, the CEM has obtained all 9 $G\&R100$ and all 23 $G\&R150$ instances at the time cost of less than 1s in most cases. In Table **A2**, a similar conclusion can also be drawn from the comparative results of the Beasley data sets. The proposed CEM yields 8 and 7 optimal solutions of 9 *Beasley* instances with gaps less than 0.1%. Note that results given by Lorena and Pereira (2002), Atta et al. (2018) are unsuitable for comparison as these instances are randomly generated with the methodology in Section 4.1. Experiments have attested to the superiority of CEM over LRS and GALR on the real-world data SJC with the most MCLP instances. In Tables **A3** to **A5**, CEM presented a median solution concerning 30 executions equal to optimal solutions for 81 of the 83 instances, better than the best 72 solutions from LRS and 70 from GALR in 20 runs Atta et al. (2018), separately. Even for the worst case of $SJC818$, the CEM still found more 15.79% optimal solutions than the best results of Lorena and Pereira (2002) and Atta et al. (2018). The gap between the worst case and the optimal solution remains below 0.07% across all small instances. This demonstrates that the proposed method is quite robust. Besides, due to the different computational environments, no proofs show the priority of solving. Even so, it can be observed that the running times of the proposed CEM are always the lowest values in which 92% of the average times are less than 10s.

Regarding small sets of $G\&R$, *Beasley* and SJC, the proposed CEM has achieved 129 optimality out of all 133 instances. The gaps for the rest four

instances between CEM and CPLEX 20.1 are not bigger than 0.1%. It is worth mentioning the robustness of CEM on all small sets of instances. For all of them, the best, the worst and the median solutions concerning the algorithm executions found in the independent executions were almost the same. Moreover, the CEM costs relatively lower CPU time than the CPLEX 20.1, especially on $Beasley$ and SJC data sets. These results enforce the capability of the metaheuristics to solve large instances of the MCLP.

4.4.2. Results of large data sets

Three large data sets ZDS, PCB and $Maximo$ are tested in this section. The first attempt for large-scale instances of the MCLP is given by Zarandi et al. (2011) on ZDS data sets. Yet, the ZDS data sets may not make fair comparisons since origin instances are not given (Zarandi et al. 2011). Pre-experiments have shown that the gaps in coverage rates between two regenerated instances can exceed 2%, even more significant than that from different approaches.

It is mainly because the results of ZDS data sets, $ZDS1800$ and $ZDS2500$, suffer from dual randomness on both networks and demands. The diversity increases with the instances scales, even though generated with the same methodology in Section 4.1. For these reasons, all MCLP optimal solutions are presented for all random instances of this study. In Table **A6**, the proposed method is able to achieve 14 best solutions which are proved to be optimal in all 18 instances. Even in the worst case, the CEM still produces solutions with small gaps, remaining within 0.7% of optimality, while the best solution deviates by only 0.05%.

In Table **A7**, we report average performance on PCB set. The CEM outperforms for 5 solutions in all 6 instances obtaining all optimal solutions of CPLEX 20.1. However, CPLEX 20.1 fails to offer all optimal solutions for instance with $p = 22$ with the time limit of 20,000s. Additionally, the Gap is introduced to evaluate the relative percentage between this solution $This_{obj}$ and the linear programming (LP) solutions.

$$Gap = \frac{This_{obj} - LP}{LP} \times 100 \tag{26}$$

The gap of median results concerning 30 algorithm executions still presents 4 solutions equal to the best lower bound found by CPLEX for all 6 instances. On the other hand, the CEM saves almost half of the CUP time used in CPLEX 20.1 in the same computing environment.

Most of $Maximao$ instances are challenging. Previous methods like intelligent base metaheuristic (Máximo et al. 2017) and branch-and-Benders-cut (Cordeau et al. 2019) failed in finding optimal solutions within 10 minutes of computing time (Cordeau et al. 2019). In Table **A8**, the proposed method outperforms GRASP and CPLX12.6 (Máximo et al. 2017) with a lower average gap of 30 executions. However, it failed to find better solutions within 10 minutes of computing time.

This is not surprising, as the proposed CEM is developed for general use for 8 kinds of data sets investigated with different properties, like instance scales and structures on network and demands (see Section 4.1). No calibration has been adopted on a particular type of data sets for further potential. On the other hand, the proposed metaheuristic is not driven by the intelligent learning process which requires computation from GPU. This may partially explain the lower time cost reported in Máximo et al. (2017).

In general, the CEM is demonstrated to be efficient in optimal solution searching for MCLP data sets. In the best execution of the algorithm, a total of 148 of 157 MCLP instances of 5 benchmark data sets, $G\&R$, $Beasley$, SJC, ZDS, and PCB, have been proven optimality within lower average times. This is quite competitive with the performance of the latest CPLEX 20.1 within 20,000s time limits. For sets of $Maximo$, the proposed method outperforms the CPLEX and GRASP of Máximo et al. (2017), while slightly weaker than (Máximo et al. 2017)'s machine learning algorithm, IGAS. It is worth pointing out that the CEM is very robust among small and large instances of ZDS and PCB with less than 0.7% gaps.

4.4.3. Results of the massive data sets

A series of massive instances has been well tested on BDS and $BDS1000$ sets of Section 4.1. In Table **A9**, the left columns report the average number of optimal solutions (Opt) and CPU time on BDS. The instances are grouped by the three different demand nodes, facilities, and radius numbers. Column # reports the total number of instances per row. The proposed CEM outperforms 23.64% solutions than the CPLEX and only fails to prove optimality for 6 instances out of 210. However, it is still worse than Ben B2 reported by Cordeau et al. (2019). This is not surprising. As an exact approach, Ben B2 takes advantage of the desired characteristics, $|I| \gg |J|$ and $|J|$ is relatively small (Cordeau et al. 2019). Also, note that the demand node reported in Cordeau et al. (2019) reached up to 20 million, which is not involved in this study due to expensive IO operations and possible memory issues caused by the data scale, especially when it reaches a million level. Instead, we have enlarged Cordeau et al. (2019)'s BDS to a $BDS1000$ with 10 times of available facility nodes, $|J| = 1000$. Compared with BDS where all facilities are restricted to 100 nodes, $BDS1000$ meets more general applications as well, thus to be adopted for the proposed method.

Table **A9** proves that increasing the value of $|J|$ makes the instances extremely hard to solve. Most $BDS1000$ instances on CPLEX 20.1 have exceeded the CPU time limit of 20,000s. There is no surprise in the fact that the growth of facility locations ends in geometric solution spaces, making it always more complicated than the boost of demand nodes. To evaluate their performance, solutions ($This_{obj}$) have been transformed to the Relative percentage deviation (RPD) with the related

best one ($Best_{Obj}$) ever found.

$$RPD = \frac{This_{obj} - Best_{Obj}}{Best_{Obj}} \times 100 \tag{27}$$

The average RPD of the CPLEX 20.1 and the proposed algorithm from 5 replicant instances with the same attributes, are presented in the right part of Table **A9**. It proves that the CEM has attested to superior performance on average RPD of 0.014 in the best case, within a far less average time cost of 307.38s. Even for the worst case of 30 algorithm excitation, the RPD has decreased 70 times from that of CPLEX 20.1. Given the most difficult 210 instances in this work, CPLEX 20.1 attests to the optimality for 85 instances. However, CPLEX 20.1 fails to find 189 better solutions of the proposed metaheuristic within 20,000s.

Using the largest benchmark sets in this work, the proposed CEM attests to superior performance over CPLEX on the massive instances for MCLP. Generally, CEM outperformed CPLEX by finding 97% optimal solutions for BDS, even though Ben B2 (Cordeau et al. 2019) reported the best performance on BDS sets. For the most sophisticated instances of $BDS1000$, the algorithm outperforms CPLEX with 90% non-inferior solutions and thus to be efficient with lower RPD and less time cost.

5. Concluding remarks

In this paper, we propose a novel formulation for the maximal covering location problem (MCLP) and a new sampling-based metaheuristic, the cross-entropy method. The new proposed model is more compact than the initial version given by Church and ReVelle (1974) with the tested MCLP instances where the proposed model accounts for an extra decrement of variables and constraints in our instances. The new formulation outperforms the Church's model in 78.10% of the tested instances requiring up to 6.2 times fewer CPU efforts. For the proposed method, a sampling-based solution construction is designed with the Pareto order sampling and incorporated with the inclusion probabilities renewing. A new swap local search is employed for further improvement. The method is carefully calibrated for parameters tuning with a ParamILS algorithm and evaluated with small, large and massive MCLP instances from 8 popular benchmark data sets with up to 100,000 demand nodes.

Comprehensive experiments show the cross-entropy method is efficient. It proves 149 optimality out of 181 small and large MCLP benchmark instances, outperforming Lagrangean surrogate heuristic (Lorena and Pereira 2002) and genetic algorithm (Atta et al. 2018) on involved instances. The proposed approach yields 90% non-inferior solutions for the most complex massive set with a much

lower CPU time and average relative percentage deviation (RPD) than the state of the art IBM CPLEX running within 20,000s. Small gaps are reported with IGAS (Máximo et al. 2017) and Ben B2 (Cordeau et al. 2019) on certain instances. Therefore, extending investigations is recommended with tailor-made parameters tuning and combining techniques like path-relinking (Máximo and Nascimento 2019) or more variety of solution sampling plans (De Boer et al. 2005).

Acknowledgements

The authors thank the anonymous referees for careful and insightful comments, which led to a stronger manuscript; Pingke Li from Tsinghua University; Instituto Tecnológico de Informática for the computational support; and all the relevant editors, reviewers, and other contributors during publication. Detailed computational results within the paper are available as online supplements, and data and codes are accessible at the accompanying GitHub repository (Wang and Zhou 2025) and https://github.com/HWangUPV/MCLP.

References

Adenso-Díaz B, Rodriguez F (1997) A simple search heuristic for the mclp: Application to the location of ambulance bases in a rural region. *Omega* 25(2):181–187, URL http://dx.doi.org/10.1016/0966-8349(97)90127-3.

Aires N (1999) Algorithms to find exact inclusion probabilities for conditional poisson sampling and pareto πps sampling designs. *Methodology and Computing in Applied Probability* 1:457–469, URL http://dx.doi.org/10.1023/A:1010091628740.

Aires N (2000) Comparisons between conditional poisson sampling and pareto πps sampling designs. *Journal of Statistical Planning and Inference* 88(1):133–147, URL http://dx.doi.org/10.1016/S0378-3758(99)00205-0.

Arakaki RGI, Lorena LAN (2001) A constructive genetic algorithm for the maximal covering location problem. *Proceedings of the Fourth Metaheuristics International Conference*, 13–17 (Porto), URL http://dx.doi.org/10.1016/S0966-8349(97)83342-6.

Atta S (2023) An improved harmony search algorithm using opposition-based learning and local search for solving the maximal covering location problem. *Engineering Optimization* 56(8):1298–1317, URL http://dx.doi.org/10.1080/0305215X.2023.2244907.

Atta S, Mahapatra PRS, Mukhopadhyay A (2018) Solving maximal covering location problem using genetic algorithm with local refinement. *Soft Computing* 22(12):3891–3906, URL http://dx.doi.org/10.1007/s00500-017-2598-3.

Beasley JE (1990) Or-library: Distributing test problems by electronic mail. *Journal of the Operational Research Society* 41(11):1069–1072, URL http://dx.doi.org/10.2307/2582903.

Berman O, Drezner Z, Krass D (2010) Generalized coverage: New developments in covering location models. *Computers & Operations Research* 37(10):1675–1687, URL http://dx.doi.org/10.1016/j.cor.2009.11.003.

Berman O, Drezner Z, Wesolowsky GO (2009) The maximal covering problem with some negative weights. *Geographical Analysis* 41(1):30–42, URL http://dx.doi.org/10.1111/j.1538-4632.2009.00746.x.

Blanco V, Gázquez R, Saldanha-da Gama F (2023) Multi-type maximal covering location problems: Hybridizing discrete and continuous problems. *European Journal of Operational Research* 307(3):1040–1054, URL http://dx.doi.org/10.1016/j.ejor.2022.10.037.

Bondesson L, Traat I, Lundqvist A (2006) Pareto sampling versus sampford and conditional poisson sampling. *Scandinavian Journal of Statistics* 33(4):699–720, URL http://dx.doi.org/10.1111/j.1467-9469.2006.00497.x.

Caserta M, Quiñonez Rico E (2009) A cross entropy-based metaheuristic algorithm for large-scale capacitated facility location problems. *Journal of the Operational Research Society* 60(10):1439–1448, URL http://dx.doi.org/10.1057/jors.2008.77.

Chen L, Chen SJ, Chen WK, Dai YH, Quan T, Chen J (2023) Efficient presolving methods for solving maximal covering and partial set covering location problems. *European Journal of Operational Research* 311(1):73–87, URL http://dx.doi.org/10.1016/j.ejor.2023.04.044.

Church RL, ReVelle CS (1974) The maximal covering location problem. *Papers in Regional Science* 32(1):101–118, URL http://dx.doi.org/10.1287/trsc.23.4.277.

Church RL, ReVelle CS (1976) Theoretical and computational links between the p-median, location set-covering, and the maximal covering location problem. *Geographical Analysis* 8(4):406–415, URL http://dx.doi.org/10.1111/j.1538-4632.1976.tb00547.x.

Cordeau JF, Furini F, Ljubić I (2019) Benders decomposition for very large scale partial set covering and maximal covering location problems. *European Journal of Operational Research* 275(3):882–896, URL http://dx.doi.org/10.1016/j.ejor.2018.12.021.

De Boer PT, Kroese DP, Mannor S, Rubinstein RY (2005) A tutorial on the cross-entropy method. *Annals of Operations Research* 134(1):19–67, URL http://dx.doi.org/10.23919/ACC.1986.4789131.

Downs BT (1991) A robust dual-based solution algorithm for the maximal covering problem. *PhD Dissertation, University of Cincinnati,* Cincinnati, URL http://dx.doi.org/10.1016/j.endm.2018.01.016.

Downs BT, Camm JD (1996) An exact algorithm for the maximal covering problem. *Naval Research Logistics* 43(3):435–461, URL http://dx.doi.org/10.1016/S0966-8349(97)83448-1.

Dwyer FR, Evans JR (1981) A branch and bound algorithm for the list selection problem in direct mail advertising. *Management Science* 27(6):658–667, URL http:

//dx.doi.org/10.1287/mnsc.27.6.658.

Farahani RZ, Asgari N, Heidari N, Hosseininia M, Goh M (2012) Covering problems in facility location: A review. *Computers & Industrial Engineering* 62(1):368–407, URL http://dx.doi.org/10.1016/j.cie.2011.08.020.

Feige U (1998) A threshold of ln n for approximating set cover. *Journal of the ACM* 45(4):634–652, URL http://dx.doi.org/10.1145/237814.237977.

Fisher ML, Nemhauser GL, Wolsey LA (1978) An analysis of approximations for maximizing submodular set functions—ii. *Polyhedral Combinatorics* 8:73–87, URL http://dx.doi.org/10.1007/BFb0121195.

Galvão RD, Espejo LGA, Boffey B (2000) A comparison of lagrangean and surrogate relaxations for the maximal covering location problem. *European Journal of Operational Research* 124(2):377–389, URL http://dx.doi.org/10.1016/S0377-2217(99)00171-X.

Galvão RD, ReVelle CS (1996) A lagrangean heuristic for the maximal covering location problem. *European Journal of Operational Research* 88(1):114–123, URL http://dx.doi.org/10.1016/S0966-8349(97)83342-6.

Güney E, Leitner M, Ruthmair M, Sinnl M (2021) Large-scale influence maximization via maximal covering location. *European Journal of Operational Research* 289(1):144–164, URL http://dx.doi.org/10.1016/j.ejor.2020.06.028.

Hillsman EL (1984) The p-median structure as a unified linear model for location-allocation analysis. *Environment and Planning A* 16(3):305–318, URL http://dx.doi.org/10.1068/a160305.

Hutter F, Hoos HH, Leyton-Brown K, Stützle T (2009) Paramils: An automatic algorithm configuration framework. *Journal of Artificial Intelligence Research* 36:267–306, URL http://dx.doi.org/10.1613/jair.2861.

Islam MS, Islam MR (2023) A solution method to maximal covering location problem based on chemical reaction optimization (cro) algorithm. *Soft Computing* 27(11):7337–7361, URL http://dx.doi.org/10.1007/s00500-023-07972-w.

Jaramillo JH, Bhadury J, Batta R (2002) On the use of genetic algorithms to solve location problems. *Computers & Operations Research* 29(6):761–779, URL http://dx.doi.org/10.1016/S0305-0548(01)00021-1.

Jia H, Ordóñez F, Dessouky MM (2007) Solution approaches for facility location of medical supplies for large-scale emergencies. *Computers & Industrial Engineering* 52(2):257–276, URL http://dx.doi.org/10.1016/j.cie.2006.12.007.

Kerkkamp R, Aardal K (2016) A constructive proof of swap local search worst-case instances for the maximum coverage problem. *Operations Research Letters* 44(3):329–335, URL http://dx.doi.org/10.1016/j.orl.2016.03.001.

Laguna M, Duarte A, Martí R (2009) Hybridizing the cross-entropy method: An application to the max-cut problem. *Computers & Operations Research* 36(2):487–498, URL http://dx.doi.org/10.1016/j.cor.2007.10.001.

Lee G, Murray AT (2010) Maximal covering with network survivability requirements in wireless mesh networks. *Computers, Environment and Urban Systems* 34(1):49–57, URL http://dx.doi.org/10.1016/j.compenvurbsys.2009.05.004.

Lorena LA, Pereira MA (2002) A lagrangean/surrogate heuristic for the maximal covering location problem using hillman's edition. *International Journal of Industrial Engineering* 9:57–67, URL http://dx.doi.org/10.1016/j.cor.2006.03.007.

Maher M, Liu R, Ngoduy D (2013) Signal optimisation using the cross entropy method. *Transportation Research Part C: Emerging Technologies* 27:76–88, URL http://dx.doi.org/10.1016/j.trc.2011.05.018.

Maranzana F (1964) On the location of supply points to minimize transport costs. *Journal of the Operational Research Society* 15(3):261–270, URL http://dx.doi.org/10.2307/3007214.

Máximo VR, Cordeau JF, Nascimento MC (2023) A hybrid adaptive iterated local search heuristic for the maximal covering location problem. *International Transactions in Operational Research* 32(1):176–193, URL http://dx.doi.org/10.1111/itor.13387.

Máximo VR, Nascimento MC (2019) Intensification, learning and diversification in a hybrid metaheuristic: An efficient unification. *Journal of Heuristics* 25(4):539–564, URL http://dx.doi.org/10.1007/s10732-018-9373-1.

Máximo VR, Nascimento MC, Carvalho AC (2017) Intelligent-guided adaptive search for the maximum covering location problem. *Computers & Operations Research* 78:129–137, URL http://dx.doi.org/10.1016/j.cor.2016.08.018.

Murray AT (2016) Maximal coverage location problem: Impacts, significance, and evolution. *International Regional Science Review* 39(1):5–27, URL http://dx.doi.org/10.1177/0160017615600222.

Murray AT, Church RL (1996) Applying simulated annealing to location-planning models. *Journal of Heuristics* 2(1):31–53, URL http://dx.doi.org/10.1007/BF00226292.

Nemhauser GL, Wolsey LA, Fisher ML (1978) An analysis of approximations for maximizing submodular set functions—i. *Mathematical Programming* 14(1):265–294, URL http://dx.doi.org/10.1007/BF01588971.

Ning X, Li P (2018) A cross-entropy approach to the single row facility layout problem. *International Journal of Production Research* 56(11):3781–3794, URL http://dx.doi.org/10.1080/00207543.2017.1399221.

Reinelt G (1994) *The traveling salesman: Computational solutions for TSP applications* (Springer: Verlag), URL http://dx.doi.org/10.5772/5583.

Resende MG (1998) Computing approximate solutions of the maximum covering problem with grasp. *Journal of Heuristics* 4(2):161–177, URL http://dx.doi.org/10.1023/A:1009677613792.

ReVelle C, Scholssberg M, Williams J (2008) Solving the maximal covering location problem with heuristic concentration. *Computers & Operations Research* 35(2):427–

435, URL http://dx.doi.org/10.1016/j.cor.2006.03.007.

ReVelle CS (1993) Facility siting and integer-friendly programming. *European Journal of Operational Research* 65(2):147–158, URL http://dx.doi.org/10.2307/j.ctt1p9wrp6.4.

Revelle CS, Eiselt HA, Daskin MS (2008) A bibliography for some fundamental problem categories in discrete location science. *European Journal of Operational Research* 184(3):817–848, URL http://dx.doi.org/10.1016/j.ejor.2006.12.044.

Rodriguez FJ, Blum C, Lozano M, García-Martínez C (2012) Iterated greedy algorithms for the maximal covering location problem. *European Conference on Evolutionary Computation in Combinatorial Optimization*, 172–181 (Springer: Berlin), URL http://dx.doi.org/10.1007/978-3-642-29124-1_15.

Rosén B (1997a) Asymptotic theory for order sampling. *Journal of Statistical Planning and Inference* 62(2):135–158, URL http://dx.doi.org/10.1016/S0378-3758(96)00185-1.

Rosén B (1997b) On sampling with probability proportional to size. *Journal of Statistical Planning and Inference* 62(2):159–191, URL http://dx.doi.org/10.1201/9781482273465-17.

Rubinstein RY (1997) Optimization of computer simulation models with rare events. *European Journal of Operational Research* 99(1):89–112, URL http://dx.doi.org/10.1016/S0377-2217(96)00385-2.

Rubinstein RY (1999) The cross-entropy method for combinatorial and continuous optimization. *Methodology and Computing in Applied Probability* 1(2):127–190, URL http://dx.doi.org/10.1007/s11464-009-0044-2.

Rubinstein RY, Kroese DP (2004) *The cross-entropy method: A unified approach to combinatorial optimization, monte-carlo simulation and machine learning* (Springer: New York), URL http://dx.doi.org/10.1198/tech.2006.s353.

Senne ELF, Pereira MA, Lorena LAN (2010) A decomposition heuristic for the maximal covering location problem. *Advances in Operations Research* 2010, ArticleID:120756, URL http://dx.doi.org/10.1155/2010/120756.

Shier DR (1981) A computational study of floyd's algorithm. *Computers & Operations Research* 8(4):275–293, URL http://dx.doi.org/10.1007/s10287-009-0094-7.

Tillé Y (2006) *Sampling algorithms* (Springer: New York), URL http://dx.doi.org/10.1007/0-387-34240-0_3.

Vohra RV, Hall NG (1993) A probabilistic analysis of the maximal covering location problem. *Discrete Applied Mathematics* 43(2):175–183, URL http://dx.doi.org/10.1016/0166-218X(93)90006-A.

Wang H, Zhou J (2025) Cross-entropy method for the maximal covering location problem. URL http://dx.doi.org/10.1287/ijoc.2024.0611.cd, available for download at https://github.com/INFORMSJoC/2024.0611.

Xia L, Xie M, Xu W, Shao J, Yin W, Dong J (2009a) An empirical comparison of five

efficient heuristics for maximal covering location problems. *2009 IEEE/INFORMS International Conference on Service Operations, Logistics and Informatics, Chicago*, 747–753 (IEEE), URL http://dx.doi.org/10.1109/SOLI.2009.5204032.

Xia L, Yin W, Xu W, Xie M, Shao J (2009b) Efficient optimization of maximal covering location problems using extreme value estimation. *Winter Simulation Conference, Austin*, 2395–2404 (Winter Simulation Conference), URL http://dx.doi.org/10.1109/WSC.2009.5429731.

Zarandi MHF, Davari S, Sisakht SH (2011) The large scale maximal covering location problem. *Scientia Iranica* 18(6):1564–1570, URL http://dx.doi.org/10.1016/j.scient.2011.11.008.