



Discrete Optimization

A mean-shift algorithm for large-scale planar maximal covering location problems

Zhou He^a, Bo Fan^{b,*}, T. C. E. Cheng^c, Shou-Yang Wang^d, Chin-Hon Tan^a^a Department of Industrial & Systems Engineering, National University of Singapore, Singapore^b School of International and Public Affairs, Shanghai Jiao Tong University, 1954 Huashan Road, Shanghai 200030, China^c Department of Logistics and Maritime Studies, The Hong Kong Polytechnic University, Kowloon, Hong Kong, China^d Academy of Mathematics and Systems Science, Chinese Academy of Sciences, Haidian District, Beijing 100190, China

ARTICLE INFO

Article history:

Received 19 March 2015

Accepted 2 September 2015

Available online 11 September 2015

Keywords:

Location

Large scale optimization

Planar maximal covering location problem

Mean shift

ABSTRACT

The planar maximal covering location problem (PMCLP) concerns the placement of a given number of facilities anywhere on a plane to maximize coverage. Solving PMCLP requires identifying a candidate locations set (CLS) on the plane before reducing it to the relatively simple maximal covering location problem (MCLP). The techniques for identifying the CLS have been mostly dominated by the well-known circle intersect points set (CIPS) method. In this paper we first review PMCLP, and then discuss the advantages and weaknesses of the CIPS approach. We then present a mean-shift based algorithm for treating large-scale PMCLPs, i.e., MSMC. We test the performance of MSMC against the CIPS approach on randomly generated data sets that vary in size and distribution pattern. The experimental results illustrate MSMC's outstanding performance in tackling large-scale PMCLPs.

© 2015 Elsevier B.V. and Association of European Operational Research Societies (EURO) within the International Federation of Operational Research Societies (IFORS). All rights reserved.

1. Introduction

Covering problems in facility location have received considerable research interest due to its applicability in the real world (Farahani, Asgari, Heidari, Hosseini, & Goh, 2012). Each facility is able to provide services within a given critical distance, i.e., the coverage radius. A customer is considered served from a facility if the distance between them is less than or equal to the facility's coverage radius. In reality, however, budget limits often constrain the number of service facilities to be located. This gives rise to the maximal covering location problem (MCLP) (Church & Velle, 1974), which seeks to maximize the coverage of customer demands by siting a given number of new facilities. In the last 40 years, MCLP and its extensions have been widely applied to study various location issues such as planning emergency facilities (e.g., Schilling, Reville, Cohon, & Elzinga, 1980; Eaton, Daskin, Simmons, Bulloch, & Jansma, 1985; Murray & Tong, 2007), siting telecommunications equipment (e.g., Akella, Delmelle, Batta, Rogerson, & Blatt, 2010; Oztekin, Pajouh, Delen, & Swim, 2010; Shillington & Tong, 2011), location for business

(e.g., Jones & Simmons, 1993; Pastor, 1994), and public services (e.g., Hougland & Stephens, 1976; Otto & Boysen, 2014).

Most MCLP models have been built under the assumption that the candidate locations of the new facilities are known in advance. In other words, facilities can only be installed in discrete nodes. Some researchers (e.g., Mehrez, 1983; Mehrez & Stulman, 1982, 1984; Church, 1984) have relaxed this constraint and extended the discrete version of MCLP to consider facility location in a continuous space, i.e., facilities are allowed to be placed anywhere on a plane. This problem is known as the planar maximal covering location problem (PMCLP), originally defined in Church (1984). For PMCLP, it is possible to attain a greater demand coverage because many more desired locations are available for selection when making strategic facility location decisions such as infrastructure investment (Murray & Tong, 2007; Wei, 2008). Murray and Tong (2007) suggested that more general representations (points, lines or polygons) of demand can also be optimally served in a region. They introduced the extended planar maximal covering location problem-Euclidean (EPMCE) and applied it to emergency warning sirens siting in Dublin. Matisziw and Murray (2009) formulated the 1-facility continuous maximal covering problem (CMCP-1), where demand is considered as continuously distributed within a whole region (convex or non-convex). They addressed CMCP-1 by generating the medial axis, which can be viewed as a geometrical representation of the region. Later in Wei (2008), the

* Corresponding author. Tel./fax: +86 21 6293 3095.

E-mail addresses: isehz@nus.edu.sg (Z. He), fanbo@sjtu.edu.cn (B. Fan), edwin.cheng@polyu.edu.hk (T.C.E. Cheng), sywang@amss.ac.cn (S.-Y. Wang), isetch@nus.edu.sg (C.-H. Tan).

multi-facility case of CMCP was solved based on Voronoi diagrams and the geometric properties of the region. Recently, several applications of PMCLP have been reported by geographical researchers (see for example Liu & Hodgson, 2013; Wei, Murray, & Batta, 2014; Wei & Murray, 2014b).

To solve PMCLP, it is natural to reduce it to MCLP by finding a finite number of potential sites on the plane. With this set of discrete sites, MCLP can be solved by either exact or heuristic approaches. In other words, PMCLP can be addressed in two phases: I – identify a candidate locations set (CLS) and II – use exact or non-exact methods to address the degraded PMCLP for coverage maximization. Therefore, it is critical to identify a *good* CLS for solving PMCLP. We can analyze a CLS from various angles: (1) Coverage. The objective of the MCLP is to find a solution with maximal coverage. Hence, the coverage of the candidates in a CLS is an important consideration. (2) Size of the CLS. MCLP is NP-hard (Downs & Camm, 1996), whose size is determined by the numbers of demand nodes, potential locations, and facilities to be located. For a very large-scale MCLP, we have to apply heuristic algorithms to search for the optimal solution. However, using such non-exact methods may cause a loss of coverage (obtaining a local optimal solution only), or even fail to produce a feasible solution. Therefore, given a large number of demand points, it is highly desirable to reduce the number of potential sites in Phase I before solving MCLP. In addition to the above two essential principles, decision-makers in reality may also be concerned with the following objectives. (3) Time to generate the CLS. Is there an efficient way to generate the CLS in Phase I? This research question has largely been neglected in the literature, yet it is a significant issue to address in real life. For example, when a severe disaster (e.g., a storm, an earthquake etc.) takes place, it is an urgent task for the government to decide where to site search and rescue (SAR) stations in order to provide medical and rehabilitation services for as many victims as possible in a large area. The scale of such a problem can be remarkably large as the numbers of SAR stations and victims may be numerous. In the context of such emergency situations, the decision-maker must consider not only the above features of the candidate locations, but also the efficiency of solving such large-scale PMCLPs in real time. (4) Average distance to demand. Given that serving faraway demands will incur a high cost, planners may also be concerned about the distances between the covered demand points to the closest facility.

A three-step procedure, often referred to as the circle intersect points set (CIPS) method in the literature, has been the dominating technique for creating the CLS since it was introduced by Church (1984). First, this method produces a demand and intersection points set (DIPS¹) by exploiting the geometric properties of coverage. The circles are centered to cover demand locations with predefined coverage radii under the Euclidean distance measure. The second step in Phase I, which is optional if the DIPS only has few members, is to remove all the dominated points from the DIPS in order to reduce the size of the CLS. It has been shown that the reduced DIPS, i.e., the final CLS, contains at least one optimal solution to the PMCLP (Church, 1984). Given this CLS, the last step is to solve MCLP in Phase II. The CIPS method greatly facilitates coverage maximization and contributes significantly to size reduction. Consequently, it has been widely applied and extended in many studies (e.g., Younies & Wesolowsky, 2004; Murray & Tong, 2007; Canbolat & Massow, 2009; Yildiz, Akkaya, Sisikoglu, & Sir, 2011) as a standard approach to address PMCLP. However, the CIPS method has high time complexity and is generally unable to handle large-scale PMCLPs (see Section 2.2 for the details).

¹ Note that this set was also called CIPS in Church (1984) and thus the term CIPS had two meanings: the whole method and the points set. To avoid confusion, hereafter in this paper we use the term CIPS to denote the method, and DIPS to call the demand and intersection points set generated by the CIPS method at the first step. We thank an anonymous referee for this helpful suggestion.

In this study we propose a mean-shift based algorithm for treating large-scale PMCLPs, i.e., MSMC. In MSMC, we introduce a revised mean-shift procedure that is less time consuming, and hence more suitable for solving large PMCLPs, than the traditional CIPS method. The mean-shift procedure has been successfully adapted to various application domains, such as cluster analysis in computer vision and image processing (Comaniciu & Meer, 2002). To the best of our knowledge, this work is the first attempt to solve location problems using the mean-shift procedure. The advantages for choosing the mean-shift procedure to identify the CLS for PMCLP are discussed in detail in Section 3.2.

The remainder of the paper is organized as follows: in Section 2 we review PMCLP, and discuss the pros and cons of the traditional CIPS approach. In Section 3 we briefly introduce the mean-shift procedure and the core of the proposed MSMC algorithm. We present each step of MSMC in detail. In Section 4 we compare the performance of MSMC against the CIPS method on randomly generated data sets that vary in size and distribution pattern, and discuss the experimental results to reveal the various performance aspects of the MSMC and CIPS approaches. In Section 5 we conclude the paper, discuss the research limitations, and suggest topics for future research.

2. Problem formulation and the CIPS method

PMCLP seeks to maximize the demand coverage on a plane. Unlike MCLP which locates facilities on a network or discrete locations, PMCLP concerns the siting of a given number of facilities anywhere on a plane; in other words, the number of potential sites for location is infinite in PMCLP. To address this issue, there is a need to generate a set of discrete potential locations from the continuous space, essentially reducing PMCLP to MCLP. To recap, solving PMCLP can be executed in two phases, namely I – identify a CLS and II – solve the MCLP given the CLS.

In this section we first present the MCLP formulation, from which PMCLP naturally arises. We then detail Phase I of the CIPS approach to solve PMCLP proposed by Church (1984). Finally, we discuss the advantages and shortcomings of the CIPS method.

2.1. The maximal covering location problem

MCLP has been mathematically formulated by Church and Velle (1974) as follows:

$$\text{Maximize } Z = \sum_{i \in I} w_i Y_i. \quad (1)$$

subject to

$$\sum_{j \in \Theta_i} X_j \geq Y_i, \quad \text{for all } i \in I \quad (2)$$

$$\sum_j X_j = p, \quad (3)$$

$$X_j = \{0, 1\}, \quad \text{for all } j \in J \quad (4)$$

$$Y_i = \{0, 1\}, \quad \text{for all } i \in I \quad (5)$$

where

i = index of demand points (entire set I);

j = index of facility locations (entire set J);

w_i = weight of demand node i ;

$\Theta_i = \{j \in J \mid d_{ij} \leq R\}$;

d_{ij} = the shortest distance from i to j ;

R = predefined coverage radius of facility;

p = number of facilities to be located;

$$X_j = \begin{cases} 1, & \text{if a facility is located at } j, \\ 0, & \text{otherwise.} \end{cases}$$

$$Y_i = \begin{cases} 1, & \text{if demand } i \text{ is covered by any sited facility,} \\ 0, & \text{otherwise.} \end{cases}$$

The objective function (1) maximizes the total weighted demand served by the facilities. Constraint (2) allows Y_i to equal 1 only when demand point i is served by at least one facility. The number of facilities to be sited is restricted to equal p in constraint (3). Constraints (4) and (5) impose the binary integer restriction on the decision variables.

Given a CLS with size $f = |\text{CLS}|$, the number of possibilities of facility location is $\binom{f}{p} = f! / (p!(f - p)!)$. OR-based methods like mixed integer programming can be applied to find the best solution for small-sized MCLPs. However, when the size of the CLS is sufficiently large, exploring the huge solution space is quite time consuming and may be prohibitive, so heuristic algorithms are required to search for acceptable solutions within a reasonable time. There are both exact and non-exact approaches for various types of MCLPs in the literature. However, given that MCLP is NP-hard, it is desirable to find ways to reduce the size of the CLS in Phase I as a smaller f can save considerable computing time in Phase II.

2.2. The CIPS method

According to Church (1984), the CIPS approach can be itemized as follows²:

1. Generating DIPS. Draw the covering boundary for each demand point (DP) and identify the intersection points (IPs) of the boundaries, and combine the DPs and the IPs as the DIPS.
2. Refining DIPS (optional for small-sized DIPS). Drop all the dominated points from the DIPS as final CLS.
3. Solving MCLP given the CLS.

Suppose that there are n demand points to be covered on the plane. Our interest is to generate a CLS in an efficient way to help address a large-scale PMCLP. Fig. 1 gives a simple example to show the basic process of CLS generation using the CIPS method. There are four DPs placed on the plane according to their coordinates and the coverage radius $R = 1$ in this instance. In Step 1, four circles centered at the blue DPs are drawn as shown in Fig. 1(a). As a result, five red IPs are recognized and their coverage boundaries are also plotted in Fig. 1(b). The next step is to recognize the dominating points from the nine nodes. According to Church (1984), point A is considered to dominate point B if point A covers the same, or more than, all the demand points that point B can serve. According to this rule, all the four DPs are considered to be dominated by the four corner IPs since a facility located at any one of the DPs can only serve one DP, i.e., itself, while each corner IP covers two DPs. At the end, however, the only member of the DIPS, or the CLS, is IP_5 located at the origin because it has all the four DPs covered, dominating the other IPs.

Admittedly, the CIPS is excellent in creating a CLS in terms of coverage maximization since the generated DIPS is guaranteed to contain at least one optimal solution to the PMCLP (Church, 1984). However, finding the dominating points can be time consuming, especially for large-sized PMCLPs. The total amount of time required to refine the DIPS depends on not only the number of DPs, but also the pattern of the demand distribution. Given n demand points on the plane, the DIPS may have at least n members when the DPs are too far away from one another, so no IP can be created. In this case, it is optional to recognize the dominating points. In the worst case where all the DPs

gather inside a circle of radius $(2 * R)$, the DIPS has $2 * \binom{n}{2} + n = n^2$ members, making it necessary to remove the dominated candidate locations from the DIPS. In other words, the size of the DIPS is $c \in [n, n^2]$. Before incorporating the reduction technique in Step 2, we know that the final size f of the CLS will be less than or equal to c . However, it is difficult to predict the precise f since we need to compute regardless of whether these c sets containing the covered DPs are subsets of one another. In the worst case, the complexity will tend towards $O(\binom{c}{2} * n) = O(n^5)$, as detailed in Appendix A. Therefore, the CIPS method may fail to produce the CLS for a large-sized PMCLP with point-based demands in a reasonable time, as suggested by Murray and Tong (2007) and Wei (2008).

Motivated by the above observations, we set out to propose an algorithm that is more efficient in creating the CLS for solving a large-scale PMCLP.

3. The mean-shift algorithm

The mean-shift algorithm is an iterative mode-seeking method introduced by Fukunaga and Hostetler (1975). It presents a simple but efficient way to converge to the local density maxima (LDM) in any probability distribution. During the last decade, the mean-shift algorithm has been extensively studied and successfully applied to many applications in the computer vision domain, such as real-time object tracking (Comaniciu, Ramesh, & Meer, 2000), image segmentation (Tao, Jin, & Zhang, 2007), cluster analysis (Wu & Yang, 2007) etc.

In this section we present a brief review of the original mean-shift procedure. We then propose a new algorithm MSMC based on a revised mean-shift procedure to generate the CLS for solving PMCLPs.

3.1. A brief introduction to the meanshift algorithm

Suppose that there are n data points $\{\mathbf{y}_i\}_{i=1}^n$ in a d -dimensional Euclidean space R^d . The mean-shift algorithm assumes that they are independent, identically distributed samples drawn from a population with an unknown density function $f(\mathbf{y})$. The multivariate kernel density estimator obtained with kernel $K(\mathbf{y})$ and bandwidth h is defined as Silverman (1986):

$$\hat{f}(\mathbf{y}) = \frac{1}{nh^d} \sum_{i=1}^n K\left(\frac{\mathbf{y} - \mathbf{y}_i}{h}\right), \tag{6}$$

where the kernel $K(\mathbf{y})$ is a scalar function under the conditions given in Fukunaga and Hostetler (1975). Among the various valid kernels, the radially symmetric kernel is often selected since it is more suited for the mean-shift algorithm (Comaniciu & Meer, 2002), satisfying:

$$K(\mathbf{y}) = c_{k,d} k(\|\mathbf{y}\|^2), \tag{7}$$

where $c_{k,d}$ is a positive normalization constant that makes $K(\mathbf{y})$ integrate to one. The function $k(x)$, only for $x \geq 0$, is called the profile of the kernel (Comaniciu & Meer, 2002).

In order to locate the modes of the density function, the gradient of the density estimator is formulated by exploiting the linearity of Eq. (6) as follows:

$$\begin{aligned} \hat{\nabla} f_{h,K}(\mathbf{y}) &\equiv \nabla \hat{f}_{h,K}(\mathbf{y}) \\ &= \frac{2c_{k,d}}{nh^{d+2}} \sum_{i=1}^n (\mathbf{y}_i - \mathbf{y}) g\left(\left\|\frac{\mathbf{y} - \mathbf{y}_i}{h}\right\|^2\right) \\ &= \frac{2c_{k,d}}{nh^{d+2}} \left[\sum_{i=1}^n g\left(\left\|\frac{\mathbf{y} - \mathbf{y}_i}{h}\right\|^2\right) \right] \mathbf{m}_h(\mathbf{y}), \end{aligned} \tag{8}$$

where

$$g(x) = -k'(x), \tag{9}$$

² Appendix A provides the pseudo codes of the key procedures discussed in this paper, including both the CIPS and MSMC approaches. Comprehensive discussion of the time complexity of each procedure is also given.

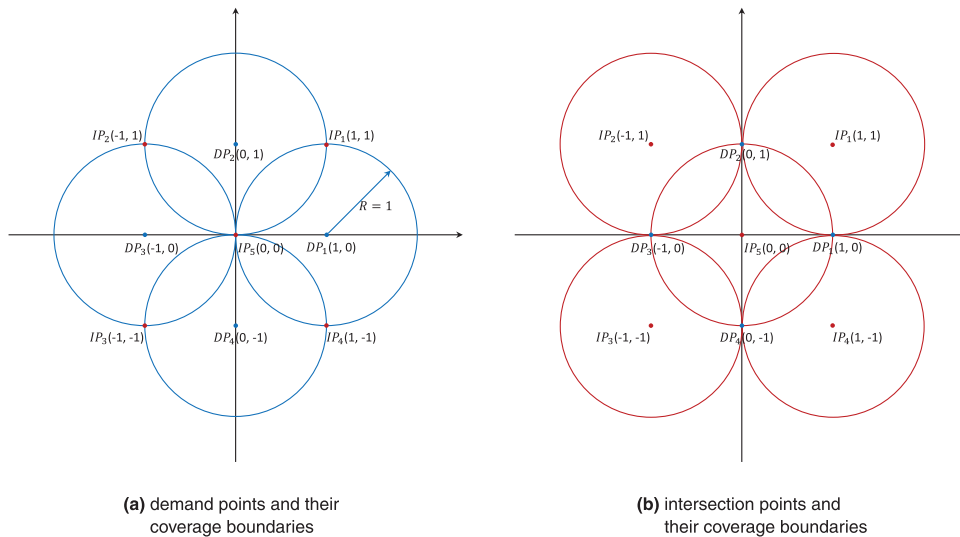


Fig. 1. The CLS generation process given in Church (1984). Blue nodes represent demand points (DPs) and red nodes are intersection points (IPs) where circles intersect. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article).

$$\mathbf{m}_h(\mathbf{y}) = \frac{\sum_{i=1}^n \mathbf{y}_i g\left(\left\|\frac{\mathbf{y} - \mathbf{y}_i}{h}\right\|^2\right)}{\sum_{i=1}^n g\left(\left\|\frac{\mathbf{y} - \mathbf{y}_i}{h}\right\|^2\right)} - \mathbf{y}. \quad (10)$$

The first term of the product in (8) is proportional to the density estimate at \mathbf{y} computed with the kernel $G(\mathbf{y})$:

$$G(\mathbf{y}) = c_{g,d} g(\|\mathbf{y}\|^2). \quad (11)$$

Eq. (10), i.e., the second term in Eq. (8), is the *mean shift*, which denotes the vector from the centre of the kernel \mathbf{y} to the weighted mean computed with the kernel G . In other words, it always points towards the direction of maximum increase in density (Comaniciu & Meer, 2002). In the event that $\mathbf{m}_h(\mathbf{y}) = 0$, the mean-shift algorithm converges to the LDM where the density estimator $\hat{f}_{h,K}(\mathbf{y})$ has a zero gradient. Hence, given the bandwidth h , also known as the window size, and the initial starting points (SPs), the mean-shift procedure comprises the following steps:

1. fix a kernel (window) around each SP;
2. compute the mean-shift vector $\mathbf{m}_h(\mathbf{y}^t)$ within the kernel (window);
3. shift the kernel (window) to the mean: $\mathbf{y}^{t+1} = \mathbf{y}^t + \mathbf{m}_h(\mathbf{y}^t)$;
4. go to Step 2 until convergence.

The above procedure has been shown to converge at the LDM where the *multivariate kernel density estimator* has zero gradient (see Comaniciu & Meer, 2002; Li, Hu, & Wu, 2007). Therefore, the mean-shift algorithm is actually a gradient ascent method which has good potential to search for LDM.

Fig. 2 is a demonstration of applying the mean-shift algorithm on the planar where the kernel $g(\mathbf{y}) = 1$. As a result, the mean-shift vector $\mathbf{m}_{h,G}(\mathbf{y}) = \frac{1}{n} \sum_{i=1}^n (\mathbf{y}_i - \mathbf{y})$, which means all the data points have identical weights. In Fig. 2(a), a window (circle) is centered at one data point and the mean-shift vector can be calculated after searching for the covered data points. Next, the centre of the disk keeps moving to the new mean location along the shift vector until reaching a LDM, as shown in Fig. 2(b), (c), and (d). These shift vectors compose a path leading to the stationary point, which is a desirable feature of our MSMC algorithm as we will see later in this section.

3.2. Why generate the CLS using the mean-shift algorithm?

As this work is the first attempt to solve location problems using the mean-shift procedure, there is a need to explain the rationale for it.³ We summarize the reasons for choosing the mean-shift procedure to identify the CLS for PMCLP are as follows.

- The PMCLP model meets all the requirements of the mean-shift method. A plane is a two-dimensional space and the demand points on it can be viewed as samples from an unknown distribution. Furthermore, the mean-shift algorithm requires no prior knowledge of the distribution because the mean-shift vector always points in the direction of maximum increase in the density. Therefore, it can be directly applied to identify the CLS for PMCLP.
- The aim of the mean-shift procedure is consistent with the first criterion of the CLS: coverage. Given a coverage region, a candidate location should cover as many demand points as possible. The identified LDM could be such ideal potential sites since they are located at the centres of the densest regions and should thus, cover the majority of demand points.
- The number of LDM could be relatively small, which is the second principle of a CLS. Comaniciu and Meer (2002) defined the term *basin of attraction* of a mode as the set of all the locations that get sufficiently close to the mode and will definitely converge to it eventually. Fig. 2 is an example to illustrate how the starting point in Fig. 2(a) is captured and shifted from the initial location to the mode in Fig. 2(d). This feature allows modes to attract nearby mean locations, so reducing the size of the set of unique stationary points significantly.
- The mean-shift procedure consumes less time to yield the CLS, which is another theoretical advantage. It requires multiple nearest neighbor searches during the execution of Step 2, so its time complexity is $O(\tau * s * n)$, where s and n are the numbers of starting points and data points, respectively, and τ stands for the number of iterations before convergence. For the original mean-shift procedure that choose the data points as initial starting points, it takes $O(\tau * n^2)$ time. Admittedly, this algorithm is not highly scalable; however, it is still much more efficient than the CIPS method, which runs in $O(n^5)$ time in the worst case. Therefore, one would expect that the larger the n is, the faster is MSMC than the traditional CIPS method.

³ We thank an anonymous referee for this helpful suggestion.

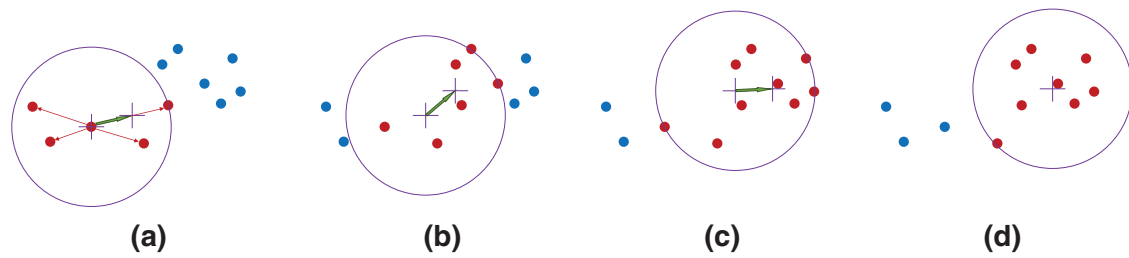


Fig. 2. The mean-shift procedure. The green arrow is the mean-shift vector computed with the red points within the window. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article).

- A LDM is the *mean*, i.e., the centre of covered demand locations. Therefore, the distances to these served nodes are minimized, which has a practical advantage over the CIPS method in terms of distance reduction, even though distance reduction is not an objective in the original PMCLP formulation. This is because siting facilities at intersection points produced by the CIPS method may not be optimal in practice if the distance factor is taken into account, as noted by Wei and Murray (2014a). For example, in Fig. 3, the four DPs are closer to one another than twice of the given radius. Therefore, 12 red IPs can be found in Fig. 3(a). After removing the dominated nodes, there are eight candidate locations left and each one is able to cover all the DPs. In other words, any one of them, i.e., four red IPs and the original DPs themselves as illustrated in Fig. 3(b), can be chosen as the only member of the CLS. If one of the blue DPs, e.g., DP_3 is selected, there is an overlap between the locations of the demand nodes and facility locations; if one of the red IPs, e.g., IP_1 is chosen, it is certain that two DPs will lie on their coverage boundaries. In real-world applications, such as distribution of medical supplies, the decision-maker may still prefer locating a facility on the origin of the plane since it is generally considered that a facility close to the demand points can provide a better quality of coverage (Dessouky, Ordez, Jia, & Shen, 2006; Jia, Ordez, & Dessouky, 2007). However, this kind of optimal locations in practice cannot be produced using the CIPS method.

In view of the above discussion, we consider that the mean-shift algorithm is a promising technique for identifying a *good* CLS for solving PMCLP.

3.3. The MSMC method

Similarly, our MSMC comprises the following three steps:

1. Generating DIPS.
2. Run the revised mean-shift procedure to identify LDM as final CLS.
3. Solving MCLP given the CLS.

The original mean-shift procedure was not designed for the PM-CLP. Therefore, we need to configure and revise it in order to improve its performance.

The original mean-shift approach has three main parameters: starting points (SPs), the bandwidth h and the kernel function $g(\mathbf{y})$. The mean-shift procedure usually launches from the data points themselves. As mentioned in previous subsection, demand points (DPs) in the PMCLP are the data points for the mean-shift procedure, thus $SPs = DPs$. However, we suggest that both DPs and IPs should be used as the SPs to identify the CLS, i.e., $SPs = DPs + IPs = DIPS$. The reason for this modification is that starting only with the DPs makes the mean-shift procedure fail to discover the critical candidate locations such as the IPs. For example, in Fig. 1(a), if we simply choose the DPs as the initial SPs, then the algorithm will stop immediately and offer four DPs as the LDM. As shown in Fig. 4, MSMC starting with the DIPS is able to recognize five more LDM, including the origin. However, the side effect of this adaptation is that it increases the time complexity. Although determining the DIPS is a computationally affordable operation with a time complexity of $O(n^2)$, the size of the DIPS raises the time complexity of the mean-shift procedure from $O(\tau * n^2)$ to $O(\tau * n^3)$ in the worst case. However, on weighing the performance tradeoff, we suggest that the DIPS, rather than the DPs, be used as the

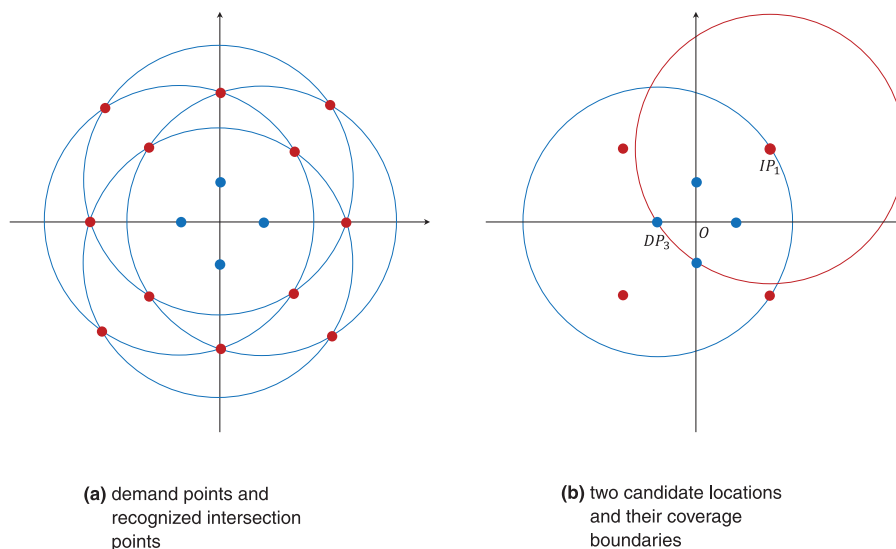


Fig. 3. CLS generation process given in Church (1984). Blue nodes represent demand points (DP), red nodes are intersection points (IP) where circles intersect. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article).

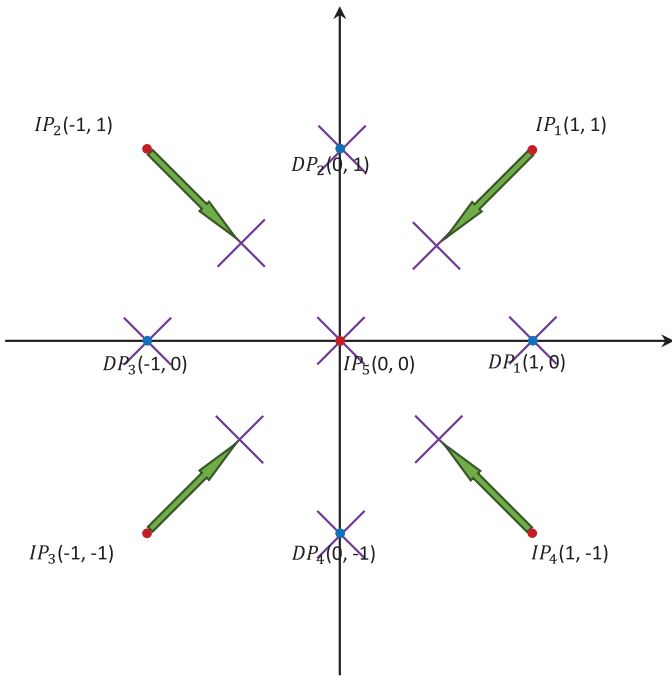


Fig. 4. Nine stationary modes, marked as purple signs, are computed when the DIPS is selected as the initial starting points for the MSMC algorithm. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article).

initial starting points for MSMC, with a view to achieving maximal coverage.

Bandwidth selection is an important topic in the research community since it directly affects the performance of density estimation, especially for the tracking of size-changing objects. Many approaches have been developed to address this issue and the bandwidth of the mean-shift procedure can be automatically selected in various practical applications (see for example Comaniciu, Ramesh, & Meer, 2001; Ming, Ci, Cai, Li, Qiao, & Du, 2012). In the context of PMCLP, it is natural to set the facility's covering distance as the bandwidth in MSMC, i.e., $h = R$.

The kernel function in the mean-shift process essentially imposes additional weights on the data points according to their distances to the shifting mean (Comaniciu & Meer, 2002). Cheng (1995) summarized four types of kernel functions frequently used in mean-shift research, namely flat, Gaussian, Epanechnikov, and quartic kernel. The experimental results presented in Appendix B demonstrate that the flat kernel performs best in terms of coverage. Therefore, we suggest that a flat kernel be chosen where all the data points are assumed to have the same distance weight when identifying LDM, i.e., $g(\mathbf{y}) = 1$ as in Fig. 2. Moreover, if the demand points are associated with different weights w_i in the PMCLP model, they should be taken into consideration when solving MCLP in Phase II, rather than in Phase I. The reason for forcing weights to be equal in computing the means and shift vectors is that we are able to recognize some saddle-like LDM without considering the weights. For instance, the origin found in Fig. 4 can be viewed as a saddle point. Suppose that one of data points, say, DP_1 has a larger weight, it will pull the centered LDM to the right. In that case, we will lose this candidate location on the origin.

The sharp-eyed readers may find in Fig. 4 that the origin dominates the other LDM, raising the question about the necessity of removing the dominated LDM using the same reduction technique employed by the second step of the CIPS method. The main motivation to add this step after identifying the LDM is that dropping the dominated LDM will reduce the size of the CLS, consequently saving time for solving MCLP. However, this operation is actually quite

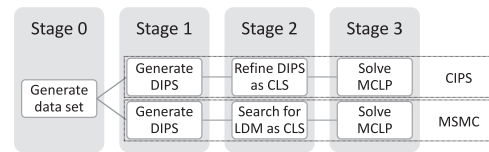


Fig. 5. Re-organized time-lines of the CIPS and MSMC approaches applied in the experiments.

Table 1
Four types of data set generated.

Name	Equal weight?	# of facilities (p)	Coverage radius (R)
Random	No	4	0.50
Circles	Yes	3	0.61
Moons	No	3	0.62
Blobs	Yes	2	0.73

computationally demanding, as analyzed in Appendix A. In other words, using the reduction procedure requires much more time than what it saves because running the solver for MCLP is usually not the bottleneck of the algorithm's whole performance, as we will see later in the experimental results.

In conclusion, the MSMC algorithm incorporates and modifies the original mean-shift procedure in order to generate CLS more efficiently for large-sized PMCLPs. Furthermore, it also borrows helpful ideas from the traditional CIPS method, such as creating the DIPS as the SPs for MSMC.

In the next section we generate a host of data sets to test the performance of the MSMC algorithm against the CIPS method.

4. Computational experiments

4.1. Implementation and experimental design

For comparison purposes, we re-organize the steps of the two algorithms as two time-lines illustrated in Fig. 5. We coded all the stages in Python 2.7, combining with the CPLEX® API⁴ for solving MCLP in Stage 3. We conducted all the experiments on an Intel Xeon computer with two processors (2.30 gigahertz) running 64-bit Windows 7 with 32 gigabyte of RAM.

At the beginning, we generated four types of test data set using the scikit-learn module, which is a well-known Python package for data mining and data analysis. Although we added the standard deviation of the Gaussian noise to the data, the generated data sets were repeated exactly for the two algorithms since the random seed was fixed. We also standardized all the data sets by removing the mean and scaling to unit variance in order to specify their x and y coordinates over the range of $(-2, 2)$. Most of their parameters are listed in Table 1, whose values are chosen arbitrarily. Random weights between 0 and 10 were assigned to the data points of the two data sets, namely *Circles* and *Blobs*. Therefore, the only parameter we needed to control was the number of data points, which was assigned as $n = \{20, 60, 100, 200, 300, 400, 500\}$. Additional large-scale instances where $n = \{1000, 2000, 3000\}$ are created to test the MSMC method only, since the CIPS is unable to solve them in a reasonable time. In other words, we generated 28 data sets to evaluate and compare the performance of two algorithms, and 12 larger-scale instances to assess the performance of MSMC in solving large-sized PMCLPs. Note that as n grows, a dramatically increasing number (up to $n(n-1)$) of the IPs will be produced in the cramped region with a higher density, causing stress-growing tests for both algorithms.

⁴ The Python API of CPLEX is part of IBM's ILOG CPLEX optimization studio (Version: 12.5).

Table 2
Experimental results for the data sets *Random* and *Circles* ($n \leq 500$).

Index	Random							Circles						
	20	60	100	200	300	400	500	20	60	100	200	300	400	500
z_1	0.85	0.6	0.53	0.44	0.41	0.42	0.41	0.6	0.45	0.43	0.41	0.39	0.4	0.38
z_2	0.85	0.46	0.51	0.4	0.39	0.4	0.37	0.45	0.42	0.38	0.38	0.37	0.37	0.37
wd_1	0.32	0.39	0.35	0.34	0.34	0.35	0.36	0.53	0.52	0.47	0.46	0.41	0.42	0.38
wd_2	0.26	0.33	0.32	0.31	0.34	0.34	0.34	0.46	0.43	0.35	0.36	0.33	0.35	0.35
c	88	734	2054	8308	18450	32854	51056	124	1000	2800	11244	25108	44780	69998
f_1	10	63	159	539	1243	2055	2953	24	89	191	599	1323	2234	3367
f_2	20	78	105	262	453	505	415	44	113	142	214	275	330	349
t_1^1	0.002	0.017	0.048	0.205	0.432	0.732	1.109	0.004	0.017	0.059	0.227	0.516	0.841	1.279
t_2^1	0.002	0.014	0.049	0.194	0.433	0.702	1.03	0.002	0.017	0.048	0.224	0.484	0.806	1.23
t_3^1	0.007	0.415	3.018	48.246	261.194	895.843	2140.779	0.023	0.709	5.313	86.159	456.7	1510.373	3799.186
t_1^2	0.019	0.199	0.755	3.654	9.841	20.27	36.8	0.024	0.272	0.958	4.713	13.199	23.456	38.459
t_2^2	0.019	0.024	0.059	0.157	0.67	1.749	4.235	0.018	0.027	0.048	0.228	0.926	2.542	6.566
t_3^2	0.018	0.031	0.039	0.075	0.178	0.25	0.24	0.019	0.03	0.035	0.078	0.138	0.176	0.227
T_1	0.028	0.456	3.125	48.608	262.296	898.324	2146.123	0.045	0.753	5.42	86.614	458.142	1513.756	3807.031
T_2	0.039	0.244	0.843	3.923	10.452	21.222	38.07	0.045	0.319	1.041	5.015	13.821	24.438	39.916

Table 3
Experimental results for the data sets *Moons* and *Blobs* ($n \leq 500$).

Index	Moons							Blobs						
	20	60	100	200	300	400	500	20	60	100	200	300	400	500
z_1	0.71	0.61	0.57	0.55	0.52	0.52	0.51	0.9	0.92	0.93	0.93	0.94	0.93	0.93
z_2	0.64	0.5	0.5	0.5	0.47	0.49	0.47	0.8	0.9	0.9	0.87	0.88	0.92	0.91
wd_1	0.55	0.54	0.5	0.49	0.49	0.43	0.43	0.45	0.44	0.44	0.43	0.43	0.43	0.44
wd_2	0.42	0.31	0.35	0.37	0.34	0.35	0.34	0.32	0.42	0.41	0.41	0.41	0.41	0.41
c	126	1126	3138	12572	28288	49646	77644	224	1942	5470	21754	49136	87532	137012
f_1	17	83	164	541	1195	1888	2862	5	8	15	45	79	129	176
f_2	48	49	67	69	61	69	52	5	10	7	4	4	6	11
t_1^1	0.002	0.019	0.069	0.25	0.529	0.894	1.364	0.003	0.025	0.096	0.348	0.733	1.252	1.939
t_2^1	0.004	0.018	0.069	0.259	0.517	0.91	1.306	0.003	0.036	0.101	0.34	0.722	1.223	1.912
t_3^1	0.016	0.802	5.844	96.989	509.774	1579.29	4016.619	0.032	1.796	13.306	208.691	1046.624	3289.905	8098.226
t_1^2	0.037	0.335	1.19	6.883	17.599	43.066	83.956	0.054	0.65	2.086	8.92	22.782	44.219	71.562
t_2^2	0.017	0.027	0.064	0.213	0.83	2.254	5.985	0.019	0.025	0.029	0.06	0.106	0.21	0.333
t_3^2	0.025	0.023	0.04	0.061	0.073	0.097	0.097	0.016	0.019	0.024	0.038	0.045	0.058	0.076
T_1	0.035	0.848	5.977	97.452	511.133	1582.438	4023.968	0.054	1.846	13.431	209.099	1047.463	3291.367	8100.498
T_2	0.066	0.376	1.299	7.203	18.189	44.073	85.359	0.073	0.705	2.211	9.298	23.549	45.5	73.55

4.2. Performance measures

In order to obtain a wealth of information about the experimental results, we define the following metrics based on the four criteria discussed above, where the subscript α is used to denote the two algorithms (CIPS: $\alpha = 1$; MSMC: $\alpha = 2$):

1. The ratio of covered demands to total demand: $z_\alpha \in [0, 1]$, and the gap between them: $\Delta z = z_1 - z_2$. Since it has been shown that the CIPS method is capable of yielding the optimal solution, the coverage ratio of the CIPS z_1 is always larger than or equal to z_2 .
2. The average weighted distance between all the covered data points to the nearest facility: $wd_\alpha = \frac{\sum_{k \in \Omega} w_k d'_k}{\sum_{k \in \Omega} w_k} > 0$, where $\Omega = \{\mathbf{y}_k\}_{k=1}^K$ is the set of the K covered demand points and d'_k denotes the distance from the covered demand point k to the nearest facility located. This indicator can be viewed as the average distance to unit demand, which is essential for the facility owner to minimize the transportation cost or time in practice. For example, in locating emergency services, it is paramount to achieve coverage as fast as possible. In the business context, a smaller wd usually means greater cost saving under the condition of same coverage.
3. The size of the generated locations c is the number of candidate locations after Stage 1, i.e., the number of elements in the DIPS. f_α is that in the CLS after Stage 2. A smaller f_α stands for better performance in reducing the size of the DIPS as it directly reduces the possible solution space for solving MCLP in Phase II.

4. The total time consumed: $T_\alpha = t_\alpha^1 + t_\alpha^2 + t_\alpha^3$. The three terms on the right hand side of this equation represent the time cost in the three stages, respectively. Note that the unit of these metrics is in seconds.

In sum, the four groups of indicators for evaluating these two approaches can be divided into two categories, namely solution quality and solution efficiency.

4.3. Results and discussion

We present all the results of the computational experiments in Tables 2–4. We also plot the plane and experimental results when $n = 100$ in Fig. 6, including the demand points, the final selected sites, and the coverage. The *Random* data set distributes the DPs uniformly, while the *Blobs* locates the DPs in three clusters. We generate the other two data sets in order to simulate demand regions that can be represented as an arc or a non-convex polygon.

4.3.1. Solution quality

The first group of metrics in the result tables measures the proportion of covered demands, where $z_1 \geq z_2$ as expected. Sometimes $\Delta z = z_1 - z_2 = 0$, showing that MSMC has good potential to reach the optimal solution as the CIPS method. However, we also see that $\Delta z = 0.15$ in the worst case of the *Circles* data set with $n = 20$. This result exposes the main weakness of the mean-shift procedure in covering the DPs: it could ignore some points located at the low density regions, especially relatively orphaned ones. For our objective

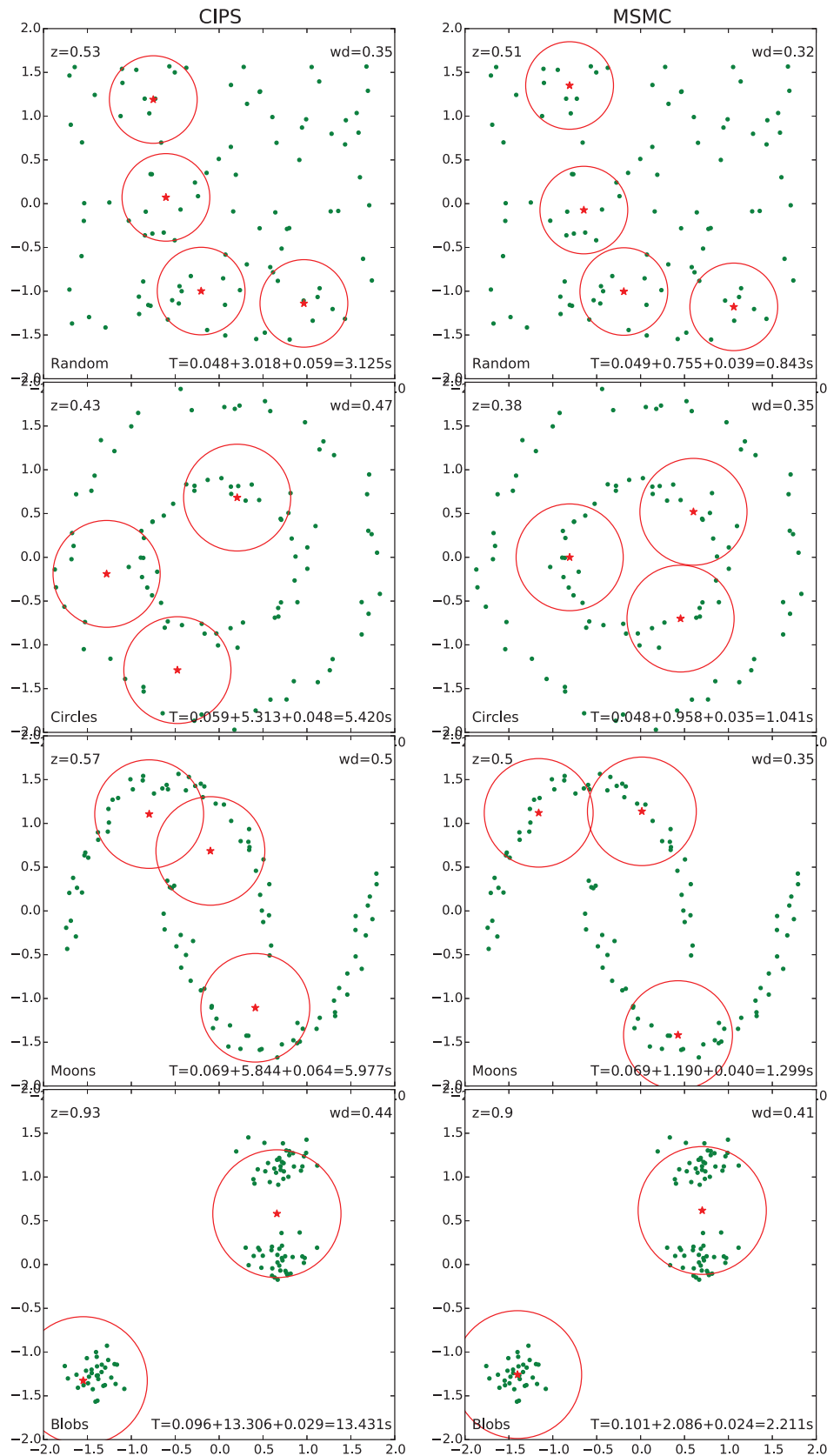


Fig. 6. Experimental results when $n = 100$. The red stars are the final selected locations produced by the two algorithms. The red circles indicate their coverage of the green demand points. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article).

Table 4
Experimental results for larger data sets ($n > 500$).

Index	Random			Circles			Moons			Blobs		
	1000	2000	3000	1000	2000	3000	1000	2000	3000	1000	2000	3000
z_2	0.32	0.31	0.30	0.36	0.36	0.35	0.46	0.45	0.44	0.91	0.92	0.91
wd_2	0.32	0.34	0.33	0.34	0.34	0.33	0.33	0.33	0.38	0.41	0.42	0.42
c	201,404	802,202	1,803,946	279,658	1,118,866	2,516,610	310,384	1,248,094	2,793,132	557,514	2,194,274	4,923,776
f_2	1029	2112	3466	510	640	938	35	30	38	12	10	11
t_2^1	4.287	17.107	39.643	5.055	20.492	46.447	5.233	23.445	49.020	7.640	36.268	71.696
t_2^2	236.067	1471.970	5364.804	245.100	1780.306	5448.018	747.432	5572.793	18408.357	482.530	2636.919	6731.579
t_2^3	1.339	9.213	29.819	0.621	1.626	4.325	0.183	0.392	0.496	0.103	0.277	0.461
T_2	241.693	1498.290	5434.266	250.776	1802.424	5498.790	752.848	5596.630	18457.873	490.273	2673.464	6803.736

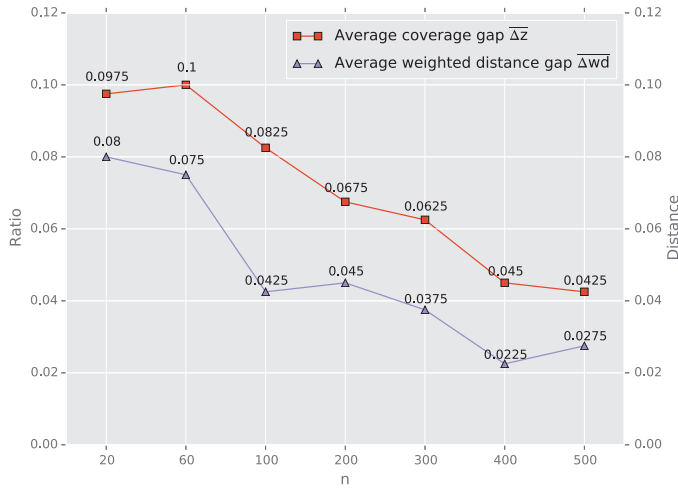


Fig. 7. The average coverage gap $\overline{\Delta z}$ and the average weighted distance $\overline{\Delta wd}$ of the four types of data set.

function seeking to maximize coverage, the reader may argue that a 15 percent loss of demand coverage is unacceptable. However, as n increases, the average coverage gap between MSMC and the CIPS method (i.e., $\overline{\Delta z}$) is diminishing as illustrated in Fig. 7, due to the higher density that promotes the advantage of the mean-shift procedure. Moreover, MSMC always outperforms the CIPS method in terms of service cost reduction, i.e., $\Delta wd = wd_1 - wd_2 \geq 0$. In fact, a linear regression equation with $R^2 = 0.88$ can be fitted to the obtained results in order to describe the significant positive relationship between $\overline{\Delta z}$ and $\overline{\Delta wd}$, as follows:

$$\overline{\Delta z} = 0.8975 \times \overline{\Delta wd} - 0.0166. \tag{12}$$

The gist of Eq. (12) is the second shortcoming of the CIPS method discussed in Section 2.2, i.e., at least two served DPs will lie on the facility’s coverage boundary if an IP is chosen, yielding the farthest distances between them. In other words, maximizing coverage is usually in conflict with shortening the service distance. Therefore, it is a challenge to simultaneously optimize these two conflicting objectives. Facing this issue, the CIPS method completely neglects the distance factor, while MSMC achieves a reasonable trade-off between coverage maximization and transportation cost saving. Based on the above findings, we believe that the MSMC method can be a promising approach to address small- and medium-sized PMCLPs.

4.3.2. Solution efficiency

For large-scale PMCLPs, however, MSMC may be the only viable approach to generate candidate locations. The fourth group of metrics indicates that the CIPS method experiences an exponential growth in computational time, e.g., 2.25 hours for the *Blobs* data set with $n = 500$. For identical data sets, MSMC only takes tens of seconds to produce the CLS, which can be one hundred times less than that

of the CIPS method as n increases. It is seen that Step 1 takes less than two seconds to create the DIPS for both algorithms, i.e. $t_1^1 \cong t_1^2$. The second step is the most computationally demanding operation whose time complexity is discussed in Appendix A. The final step widens the gap between the two algorithms’ performance since MSMC drops many more of the dominated candidate locations from the DIPS than the CIPS approach does as n increases. Both Tables 2 and 3 show that $f_1 > f_2$ when $n \geq 100$, which is the main reason for $t_1^3 > t_2^3$. Based on the above analysis, it is not surprising that MSMC is much more efficient than the CIPS method to address large-scale PMCLPs.

It is also worth noting that as the DPs gather into clusters, most of the performance metrics of MSMC improve. The main reason, as mentioned before, is that the increase in the local density helps MSMC cover more DPs within a fixed region. This factor also has a major impact on the size of the DIPS since more intersection points will be calculated if the radius of a cluster is relatively small. As demonstrated in Fig. 3(a), if the radius is less than $2 * R$, k DPs will generate $k(k - 1)$ IPs. This is the reason why the *Blobs* data set produces a DIPS of size 137,012 given $n = 500$, much more than the other data sets. Compared with the CIPS method, which has to perform the computationally demanding *isSubset* check one hundred thousand times, it is easy and fast for MSMC to locate the density mode just nearby. In other words, the iteration time before convergence is small, reducing the complexity from $O(\tau * c * n)$ to $O(c * n)$.

Table 4 reports experimental results of MSMC when solving large-scale PMCLPs. Comparing with Tables 2 and 3, the MSMC still performs well in terms of solution quality. Meanwhile, the increased execution time (T_2) is also acceptable given the large number of demand nodes, which could produce millions of intersection points in DIPS. There is an interesting finding that, although the *Blobs* data set is always able to create the largest DIPS, the most time-consuming instance is the *Moons*. This is because the mean-shift process, which is the bottleneck of MSMC, has to find more temporary means along the semi circumferences before locating the LDM, leading to higher time complexity. Yet despite the large size of data sets and increased time complexity, our proposed MSMC approach has demonstrated to be very successful in solving large-scale PMCLPs.

4.3.3. Summary

The CIPS algorithm has high time complexity and may not be appropriate for solving large-scale PMCLPs in practice. In contrast, the proposed MSMC, without the random factors, is a highly efficient tool for dealing with large-scale PMCLPs. The limitation of the MSMC algorithm is that it may not contain an optimal solution to the PMCLP. However, our experiments suggest that the optimality gap of the solution obtained using the CLS generated by the MSMC algorithm is likely to be small in practice. Furthermore, our experiments also suggest that the average distance to the demand points of the solution based on the MSMC algorithm is small. Table 5 summarizes the general performance of the two methods based on the four criteria discussed in the Introduction.

Table 5
Comparison of the CIPS method and MSMC by the four criteria.

Method	Coverage	Average distance to demand	Size of CLS	Time to generate CLS given n demand nodes
CIPS	Optimal	Two demand nodes may lie on coverage boundary	Less for small-scale PMCLP	$O(n^5)$ in the worst case
MSMC	Equal to or less than optimum	Center of covered demand nodes, local optimal	Much less for large-scale PMCLP	$O(\tau * n^3)$ in the worst case

5. Conclusion

PMCLP can be applied to make strategic facility location decisions, such as infrastructure investment, because it allows a given number of facilities to be placed anywhere on a plane for coverage maximization, so many more desired locations are available for selection. However, due to the analytical difficulty of tackling geometrical computation in a continuous plane, there is a need to identify potential sites for evaluation, reducing PMCLP to the relatively simple MCLP.

The technique for determining the CLS has been mostly dominated by the CIPS approach. This approach exploits the geometric properties of coverage by identifying all the demand nodes plus the intersection points as DIPS, and then drops the dominated points from the DIPS in order to generate a reduced CLS. Although it has been shown that the CLS created by the CIPS method contains at least one optimal solution to the PMCLP, the CIPS method has high time complexity and is generally unable to handle large-scale PMCLPs.

This paper presents a deterministic algorithm MSMC, based on a revised mean-shift procedure, to address large-scale PMCLPs. We make three revisions to the original mean-shift procedure to adapt it for PMCLP, and generate a number of data sets that vary in size and distribution pattern to test MSMC against the CIPS method. Unlike the CIPS method, we are able to generate the CLS for large PMCLPs in a reasonable amount of time. Besides computational time, we also discuss the experimental results in terms of different performance aspects of the algorithms, e.g., coverage ratio, weighted distance to customers, and CLS size. The experimental findings indicate that MSMC strikes a reasonable trade-off between coverage maximization and transportation cost saving. Hence, the MSMC approach is also potentially promising for solving small- and medium-sized PMCLPs.

Although the examined PMCLP assumes Euclidean distance as the coverage standard, the proposed MSMC algorithm remains applicable if other distance metrics are used. For example, if the metropolitan metric (rectilinear) distance measure is taken into account, MSMC only requires a minor revision of changing the distance computing function for the mean-shift procedure. Another difference concerns the development of the intersection points whereby the coverage area changes from a circle to a diamond, which may create an infinite diamond intersect points set when the intersection set becomes a line segment. Church (1984) commented on this case and suggested the use of the two endpoints of the line segment as the IPs. Therefore, the attributes of the diamond intersect points set are analogous to those of the DIPS. In other words, all the key procedures of the two algorithms can still be applied to solve PMCLP under the rectilinear distance measure.

This study is an initial attempt to solve the location problem using the mean-shift algorithm. Further research is possible such as developing a suitable mechanism to reconcile the solution features. Moreover, in our model, demand on the plane is represented as discrete points, rather than lines or polygons. Applying MSMC or new algorithms to optimally cover these shapes of demands is not only challenging but also interesting. Finally, the maximal service distance R for each demand is a constant. In reality, it may vary due to the facility's size, availability, or service standard. Therefore, relaxing this assumption makes PMCLP much more applicable in practice, but such a problem requires further in-depth research to address.

Acknowledgments

We thank the anonymous referees for their helpful comments on earlier versions of our paper. This research was supported in part by the Singapore National Research Foundation (grant no. E2S2) under the Campus for Research Excellence and Technological Enterprise (CREATE) programme, the National Science Foundation of China (Nos. 91024007 and 71371122), and National Social Science Foundation of China (No. 14ZDB152). Cheng was also supported in part by The Hong Kong Polytechnic University under the Fung Yiu King – Wing Hang Bank Endowed Professorship in Business Administration.

Appendix A. Pseudo codes and time complexity discussion

A.1. The DIPS generation procedure

Given the radii and centre coordinates of two circles, it is routine to programme the process of recognizing their intersection points. Algorithm 1 shows a piece of the pseudo code, whose main body consists of two nested loops. Each line of the codes within the inner loop is a constant-time operation, i.e., it runs in $O(1)$ time. As each loop needs to execute n times, the time complexity of the whole procedure is $O(n^2)$.

Algorithm 1: The DIPS generation procedure.

Data: A set Y containing n demand points with (x, y) coordinates and coverage radius R .
Result: A set C consisting of intersection points and demand points with their coordinates.

```

1 Create a temporary set copied from  $Y$ ;
2 for  $i$  in range( $n$ ) do loop  $Y$ 
3   for  $j$  in range( $n$ ) do loop  $Y$ 
4      $d_{i,j}$  = distance between  $Y_i$  and  $Y_j$ ;
5     if  $0 < d_{i,j} \leq 2R$  then
6       Calculate the coordinates of intersection points;
7       Add generated IPs to the temporary set;
8     end
9   end
10 end
11 Combine the temporary set and  $Y$ ;
12 Return the combined set as  $C$ ;
```

A.2. The DIPS reduction procedure

As noted by Church (1984), point A is considered to dominate point B if point A covers the same, or more than, all the demand points that point B can serve. Therefore, examining the coverage information of each potential location is necessary before comparison. As presented in Algorithm 2, we first construct an empty list with length c and fill it up with sublists of candidate node i 's covered data points. Note that this step runs in $O(c * n)$ time since the inside codes are independent of n and c . The second part of Algorithm 2 is much more computationally demanding because the two nested loops take subquadratic time $O(c^2)$, where c could be as large as $O(n^2)$, as mentioned before. It is even worse that the function *isSubset* in line 13 also has a time complexity of $O(n)$ as Ω_i could have n members.

Algorithm 2: The DIPS reduction procedure.

Data: Demand points set Y of size n , DIPS C of size c , and coverage radius R .

Result: An CLS contains candidate locations.

- 1 Create a temporary list Ψ to store coverage information;
- 2 **for** i in range(c) **do** loop C
- 3 Create a sublist Ω_i to catch the index of demand points covered by i ;
- 4 **for** j in range(n) **do** loop Y
- 5 $d_{i,j}$ = distance between i and j ;
- 6 **if** $d_{i,j} \leq R$ **then**
- 7 Add index j to Ω_i ;
- 8 **end**
- 9 **end**
- 10 **end**
- 11 **for** i in range(c) **do** loop Ψ
- 12 **for** j in range(c) **do** loop Ψ
- 13 **if** Ω_i is a subset of Ω_j **then**
- 14 Remove C_i from C ;
- 15 **end**
- 16 **end**
- 17 **end**
- 18 Return C as CLS;

Algorithm 3: The mean shift procedure.

Data: Demand points set Y of size n , DIPS C of size c , and coverage radius R .

Result: An CLS contains candidate locations.

- 1 Create a temporary set Ψ to receive obtained means (LDM);
- 2 **for** i in range(c) **do** loop C
- 3 NewMean = coordinates of C_i ;
- 4 **while** True **do** mean-shift iteration
- 5 Create a list Ω to catch the index of demand points covered by NewMean;
- 6 **for** j in range(n) **do** loop Y
- 7 d_j = distance between NewMean and j ;
- 8 **if** $d_j \leq R$ **then**
- 9 Add index j to Ω ;
- 10 **end**
- 11 **end**
- 12 OldMean = NewMean;
- 13 NewMean = the Mean of Ω ;
- 14 **if** NewMean=OldMean **then**
- 15 Add NewMean to Ψ ;
- 16 Break the While loop;
- 17 **end**
- 18 **end**
- 19 **end**
- 20 Return Ψ as CLS;

Therefore, this part is the bottleneck in running the DIPS reduction procedure, whose time complexity could be $O(c^2 * n)$ or $O(n^5)$ in the worst case.

A.3. The mean shift procedure

The mean shift procedure embedded in the MSMC is modified to share the same input and output parameters with the DIPS reduction process, as detailed in Algorithm 3. There are three nested loops, similar to the DIPS reduction as well. After initializing an empty set for storing the identified unique means, the algorithm runs an outer loop to pick each starting point from the DIPS of size c . The middle

loop is a do-while loop since the iteration time before convergence is unknown, denoted as τ . Inside the do-while loop are two functions. The first one is to compute the equal-weighted mean of the coverage window, requiring an inner loop to calculate the distances to all the n points. Therefore, the time complexity of this function is $O(c * \tau * n) = O(\tau * n^3)$. The second function is to check and break the do-while loop if the stop condition is met, without using a loop. To sum up, the time complexity of the mean shift procedure is $O(\tau * n^3)$ in the worst case, which is much less than that of the DIPS reduction procedure.

Appendix B. Kernel function selection

Given the same 28 data sets mentioned in Section 4.1, we have tested the MSMC using four different kernels as following:

- Flat kernel $g(\mathbf{y}) = 1$;
- Gaussian kernel $g(\mathbf{y}) = e^{-\|\mathbf{y}\|^2}$;
- Epanechnikov kernel $g(\mathbf{y}) = 1 - \|\mathbf{y}\|^2$, and
- Quartic kernel $g(\mathbf{y}) = (1 - \|\mathbf{y}\|^2)^2$.

The coverage ratios are presented in Table B.1, where the maximum ones are in bold.

Table B.1
The coverage ratios achieved by the MSMC using different kernels.

Data set	n	Flat	Gaussian	Epanechnikov	Quartic
Random	20	0.8458	0.8458	0.7831	0.7831
	60	0.4642	0.4288	0.3969	0.3969
	100	0.5067	0.4320	0.4079	0.4042
	200	0.3963	0.3787	0.3490	0.3640
	300	0.3882	0.3882	0.3233	0.3332
	400	0.3968	0.3773	0.3377	0.3400
Circles	500	0.3697	0.3691	0.3401	0.3404
	20	0.4500	0.4500	0.3000	0.3000
	60	0.4167	0.4000	0.3167	0.3333
	100	0.3800	0.3800	0.3600	0.3200
	200	0.3800	0.3700	0.3550	0.3350
	300	0.3707	0.3700	0.3533	0.3467
Moons	400	0.3675	0.3650	0.3550	0.3575
	500	0.3720	0.3660	0.3680	0.3560
	20	0.6411	0.5321	0.4370	0.4370
	60	0.5046	0.4935	0.4774	0.4599
	100	0.4987	0.4748	0.4673	0.4910
	200	0.5037	0.4796	0.4709	0.4576
Blobs	300	0.4740	0.4740	0.4489	0.4581
	400	0.4882	0.4853	0.4474	0.4573
	500	0.4711	0.4587	0.4382	0.4581
	20	0.8000	0.7000	0.7000	0.7000
	60	0.9000	0.6667	0.6667	0.6667
	100	0.9000	0.6700	0.6700	0.6700
	200	0.8650	0.6700	0.6700	0.6700
	300	0.8847	0.8800	0.8847	0.6667
	400	0.9150	0.6675	0.6675	0.6675
	500	0.9140	0.6680	0.9140	0.6680

References

- Akella, M. R., Delmelle, E., Batta, R., Rogerson, P., & Blatt, A. (2010). Adaptive cell tower location using geostatistics. *Geographical Analysis*, 42(3), 227–244.
- Canbolat, M. S., & Massow, M. v. (2009). Planar maximal covering with ellipses. *Computers & Industrial Engineering*, 57(1), 201–208.
- Cheng, Y. (1995). Mean shift, mode seeking, and clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(8), 790–799.
- Church, R., & Velle, C. R. (1974). The maximal covering location problem. *Papers in Regional Science*, 32(1), 101–118.
- Church, R. L. (1984). The planar maximal covering location problem. *Journal of Regional Science*, 24(2), 185–201.
- Comaniciu, D., & Meer, P. (2002). Mean shift: a robust approach toward feature space analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(5), 603–619.
- Comaniciu, D., Ramesh, V., & Meer, P. (2000). Real-time tracking of non-rigid objects using mean shift. In *Proceedings IEEE conference on computer vision and pattern recognition: vol. 2* (pp. 142–149).

- Comaniciu, D., Ramesh, V., & Meer, P. (2001). The variable bandwidth mean shift and data-driven scale selection. In *Proceedings eighth IEEE international conference on computer vision: vol. 1* (pp. 438–445).
- Dessouky, M., Ordez, F., Jia, H., & Shen, Z. (2006). Rapid distribution of medical supplies. In R. Hall (Ed.), *Patient flow: reducing delay in healthcare delivery*. In *International Series in Operations Research & Management Science: vol. 91* (pp. 309–338). Springer US.
- Downs, B. T., & Camm, J. D. (1996). An exact algorithm for the maximal covering problem. *Naval Research Logistics*, 43(3), 435–461.
- Eaton, D. J., Daskin, M. S., Simmons, D., Bulloch, B., & Jansma, G. (1985). Determining emergency medical service vehicle deployment in Austin, Texas. *Ultracentrifugation*, 15(1), 96–108.
- Farahani, R. Z., Asgari, N., Heidari, N., Hosseiniinia, M., & Goh, M. (2012). Covering problems in facility location: a review. *Computers & Industrial Engineering*, 62(1), 368–407.
- Fukunaga, K., & Hostetler, L. (1975). The estimation of the gradient of a density function, with applications in pattern recognition. *IEEE Transactions on Information Theory*, 21(1), 32–40.
- Houglund, E. S., & Stephens, N. T. (1976). Air pollutant monitor siting by analytical techniques. *Journal of the Air Pollution Control Association*, 26(1), 51–53.
- Jia, H., Ordez, F., & Dessouky, M. M. (2007). Solution approaches for facility location of medical supplies for large-scale emergencies. *Computers & Industrial Engineering*, 52(2), 257–276.
- Jones, K. G., & Simmons, J. W. (1993). *Location, location, location: analyzing the retail environment*. Canada: Nelson.
- Li, X., Hu, Z., & Wu, F. (2007). A note on the convergence of the mean shift. *Pattern Recognition*, 40(6), 1756–1762.
- Liu, S., & Hodgson, M. E. (2013). Optimizing large area coverage from multiple satellite-sensors. *Geoscience & Remote Sensing*, 50(6), 652–666.
- Matisziw, T. C., & Murray, A. T. (2009). Siting a facility in continuous space to maximize coverage of a region. *Socio-economic Planning Sciences*, 43(2), 131–139.
- Mehrez, A. (1983). A note on the linear integer formulation of the maximal covering location problem with facility placement on the entire plane. *Journal of Regional Science*, 23(4), 553–555.
- Mehrez, A., & Stulman, A. (1982). The maximal covering location problem with facility placement on the entire plane. *Journal of Regional Science*, 22(3), 361–365.
- Mehrez, A., & Stulman, A. (1984). An extended continuous maximal covering location problem with facility placement. *Computers & Operations Research*, 11(1), 19–23.
- Ming, D., Ci, T., Cai, H., Li, L., Qiao, C., & Du, J. (2012). Semivariogram-based spatial bandwidth selection for remote sensing image segmentation with mean-shift algorithm. *IEEE Geoscience and Remote Sensing Letters*, 9(5), 813–817.
- Murray, A. T., & Tong, D. (2007). Coverage optimization in continuous space facility siting. *International Journal of Geographical Information Science*, 21(7), 757–776.
- Otto, B., & Boysen, N. (2014). A dynamic programming based heuristic for locating stops in public transportation networks. *Computers & Industrial Engineering*, 78, 163–174.
- Oztekin, A., Pajouh, F. M., Delen, D., & Swim, L. K. (2010). An RFID network design methodology for asset tracking in healthcare. *Decision Support Systems*, 49(1), 100–109.
- Pastor, J. T. (1994). Bicriterion programs and managerial location decisions: application to the banking sector. *Journal of the Operational Research Society*, 45(12), 1351–1362.
- Schilling, D. A., Revelle, C., Cohon, J., & Elzinga, D. J. (1980). Some models for fire protection locational decisions. *European Journal of Operational Research*, 5(1), 1–7.
- Shillington, L., & Tong, D. (2011). Maximizing wireless mesh network coverage. *International Regional Science Review*, 34(4), 419–437.
- Silverman, B. W. (1986). *Density estimation for statistics and data analysis: vol. 26*. New York: Chapman & Hall.
- Tao, W., Jin, H., & Zhang, Y. (2007). Color image segmentation based on mean shift and normalized cuts. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, 37(5), 1382–1389.
- Wei, H. (2008). *Solving continuous space location problems* (Ph.D. thesis). The Ohio State University.
- Wei, R., & Murray, A. T. (2014a). Continuous space maximal coverage: insights, advances and challenges. *Computers & Operations Research*, 62, 325–336.
- Wei, R., & Murray, A. T. (2014b). Evaluating polygon overlay to support spatial optimization coverage modeling. *Geographical Analysis*, 46(3), 209–229.
- Wei, R., Murray, A. T., & Batta, R. (2014). A bounding-based solution approach for the continuous arc covering problem. *Journal of Geographical Systems*, 16(2), 161–182.
- Wu, K.-L., & Yang, M.-S. (2007). Mean shift-based clustering. *Pattern Recognition*, 40(11), 3035–3052.
- Yildiz, E., Akkaya, K., Sisikoglu, E., & Sir, M. (2011). An exact algorithm for providing multi-perspective event coverage in wireless multimedia sensor networks. In *Proceedings of the 7th international wireless communications and mobile computing conference (IWCMC)* (pp. 382–387).
- Younies, H., & Wesolowsky, G. O. (2004). A mixed integer formulation for maximal covering by inclined parallelograms. *European Journal of Operational Research*, 159(1), 83–94.