Hybrid Meta-heuristics with VNS and Exact Methods: Application to Large Unconditional and Conditional Vertex p-Centre Problems

Chandra Ade Irawan[1,2,*], Said Salhi[3] and Zvi Drezner[4]

[1]Centre for Operational Research and Logistics (CORL), Department of Mathematics, University of Portsmouth, Lion Gate Building, Lion Terrace, Portsmouth PO1 3HF, UK

[2]Department of Industrial Engineering, Institut Teknologi Nasional, Bandung, 40124, Indonesia

[3]Centre for Logistics & Heuristic Optimization (CLHO), Kent Business School, University of Kent, Canterbury, Kent CT2 7PE, UK

[4]Department of Information Systems and Decision Sciences, Steven G. Mihaylo College of Business and Economis, California State University-Fullerton, Fullerton, CA 92834, USA

Abstract Large-scale unconditional and conditional vertex p-centre problems are solved using two meta-heuristics. One is based on a three-stage approach whereas the other relies on a guided multi-start principle. Both methods incorporate Variable Neighbourhood Search, exact method, and aggregation techniques. The methods are assessed on the TSP dataset which consist of up to 71,009 demand points with p varying from 5 to 100. To the best of our knowledge, these are the largest instances solved for unconditional and conditional vertex p-centre problems. The two proposed meta-heuristics yield competitive results for both classes of problems.

Keywords Large unconditional and conditional vertex p-centre problems, aggregation, variable neighbourhood search, exact method.

1 Introduction

The vertex p-centre problem, also known as the minimax location problem, aims to optimally locate p facilities among n potential sites and to assign demand points to these facilities in order to minimise the maximum distance between demand points and their nearest facility. Applications include the location of facilities in emergency services such as police, fire, and ambulance stations. In the conditional p-centre problem some (say q) facilities already exist and the objective is to locate p new facilities in addition to the existing q facilities. A demand point can be served by the nearest facility whether it is new or existing.

* Corresponding author : Chandra Ade Irawan (chandra.irawan@port.ac.uk)

This problem is known as the (p, q) centre problem (see Drezner 1995). When q = 0, the problem becomes the unconditional p-centre problem (the p-centre problem for short) whose formulation is given as follows:

Minimise r (1)

Subject to

$$\sum_{j\in J} Y_{ij} = 1 \quad \forall i \in I \tag{2}$$

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

$$Y_{ij} - X_j \le 0, \quad \forall i \in I, j \in J \tag{4}$$

$$r \ge \sum_{j\in J} d(i, j)Y_{ij} \quad \forall i \in I \tag{5}$$

$$X_j \in \{0,1\} \quad \forall j \in J \tag{6}$$

$$Y_{ij} \in \{0,1\} \quad \forall i \in I, j \in J \tag{7}$$

Where

(I,J) : set of demand points/customers $(i \in I = \{1,...,n\})$ and set of potential sites $(j \in J = \{1,...,M\})$ (i.e. : $n = |I|$ and $M = |J|$), respectively

r : the maximum distance between a customer and its closest facility

$d(i, j)$: the distance between customer i and potential site j (Euclidian distance is used in our study);

p : the required number of facilities;

Y_{ij} = 1, if customer i is served by a facility at site j and = 0 otherwise;

X_j = 1, if a facility is opened at potential site j and = 0 otherwise;

The objective function (1) is to minimise the maximum distance between a customer and its nearest facility. Constraints (2) guarantee that each customer i is assigned to exactly one open facility whereas constraint (3) restricts the number of open facilities to be exactly p. Constraints (4) ensure that customer i can only be allocated to an open facility (i.e., $X_j = 1$). Constraints (5) define the maximum distance between customer i and its closest facility. Constraints (6) and (7) refer to the binary nature of the decision variables.

The p-centre problem is known to be NP-hard problem (Kariv and Hakimi 1979). Though this problem can be optimally solved for medium size instances ($n \le 4{,}000$, $p \le 100$), as will be shown in the computational result section (subsection 5.2), this problem is hard to solve when the size is relatively large. The p-centre problem may consist of a large number of

customers as well as potential facility sites. For example, a problem which includes individual private residences as customers may involve several thousands of demand points. One way to model such a problem is to aggregate customers from n to m points ($m << n$) so the reduced (approximated) problem becomes easier to solve. However, aggregation reduces the accuracy of the solution. In this paper, we propose two meta-heuristics. The first one consists of a three-stage approach, for solving large unconditional and conditional p-centre problems. The first stage uses aggregation and an exact method whereas the second utilises the information obtained in the first stage to define a problem which is then solved by a Variable Neighbourhood Search (VNS). The third and last stage utilises also a VNS to solve the original (disaggregated) problem using the best solution obtained so far as an initial solution. The second approach is based on a guided muti-start where VNS and exact method are incorporated. To the best of our knowledge, there is no published work for solving large p-centre problems though a few studies were conducted for its counterpart the p-median problem (see Hansen et al. 2009; Avella et al. 2012; Irawan and Salhi 2013; Irawan et al. 2013)

The contributions of this study include: (i) two powerful meta-heuristics that incorporate aggregation technique, a VNS, and an exact method for solving, for the first time, large unconditional and conditional p-centre problems, (ii) a new scheme for aggregating demand points for the unconditional and conditional p-centre problems, and (iii) a new distance calculation method for aggregated p-centre problems, and (iv) new best and optimal solutions for large instances for benchmarking purposes.

The paper is organized as follows. A brief review of the related literature is presented in Section 2. The ingredients that make up the two meta-heuristics as well as the overall respective algorithms are described in Section 3. This is followed by the detailed explanations of the main steps in Section 4. The computational results are presented and analysed in Section 5. The last section provides a summary of our findings and highlights some suggestions for future research.

2 Literature Review

A review on the unconditional and conditional discrete p-centre problems is first presented followed by highlights focussing on aggregation techniques for the p-centre problem in particular.

2.1 Related work on the p-centre problem

The p-centre problem was first proposed by Hakimi (1964) who investigated an absolute 1-centre problem on a graph. Minieka (1970) presented a method to solve the problem when $p > 1$. He suggested a basic algorithm based on solving a finite sequence of set covering problems. The weighted case of the p-centre problem was initially studied by Kariv and Hakimi (1979) who proved that the p-centre problem is NP-hard.

Tansel et al. (1982) proposed polynomially bounded procedures for solving p-centre and covering problems on a tree network. A review of network location problems including the p-centre problem is provided by Tansel et al. (1983a; 1983b). Drezner (1984) designed two heuristics and an optimal algorithm to solve the p-centre problem for a given value of p in polynomial time in n. For relatively small p, Jaeger and Kariv (1985) introduced algorithms for finding p-centres on a weighted tree.

Daskin (1995) suggested a useful and interesting recursive type algorithm using the Set Covering Problem (SCP) for obtaining an optimal solution for the problem. The algorithm is based on Minieka's method and uses the bisection technique that decreases the gap between upper and lower bounds. Bozkaya and Tansel (1998) proposed a spanning tree approach on cyclic networks. A unified limited column generation approach for facility problems including the p-centre problem on trees was presented by Shaw (1999).

Efficient exact algorithms for the vertex p-centre problem were later proposed by Daskin (2000) and Ilhan and Pinar (2001). The former formulated the problem as a maximum set covering sub-problem and then Lagrangean Relaxation is used to solve the problem. The latter proposed a method which consists of two phases namely the LP-Phase and the IP-Phase where in Phase 1 sub-problems with a certain covering distance are systematically discarded. Caruso et al. (2003) proposed an algorithm called Dominant whereas Mladenovic et al. (2003) implemented efficient meta-heuristics (tabu search and variable neighbourhood search) with excellent results. Elloumi et al. (2004) used Minieka's technique incorporating a greedy heuristic and the IP formulation of the sub-problem for solving the problem optimally.

Al-Khedhairi and Salhi (2005) introduced enhancements to the Daskin's method (1995) and Ilhan and Pinar (2001) with the aim in reducing the number of ILP iterations (calls to the SCPs). In the first approach, the gaps in the distance matrix are sorted and efficiently recorded whereas in the second approach, appropriate jumps in the covering distance are explored. Cheng et al. (2007) suggested an efficient algorithm by modelling the network as

an interval graph. Chen and Chen (2009) introduced relaxation algorithms for both the continuous and discrete p-centre problems by solving optimally smaller reduced problems which are then augmented gradually by adding 'k' customers at a time where k is a parameter that needs to be defined.

Salhi and Al-Khedhairi (2010) improved Daskin's approach (1995) even further by integrating heuristic information into exact methods. Tight upper bounds are obtained by a multilevel type meta-heuristic (Salhi and Sari, 1997) which are then used to derive promising lower bounds. Davidovic et al. (2011) introduced a bee colony optimization heuristic algorithm and a non-deterministic Voronoi diagram algorithm for the unconstrained and constrained p-centre problem respectively.

Calik and Tansel (2013) proposed a double bounded method based on two-element restrictions that obtain the optimal solution by solving a series of simple structured integer programs. Lu and Sheu (2013) recently introduced a robust vertex p-centre model for locating urgent relief distribution centres whereas Lu (2013) studied a generalized weighted vertex p-centre model that represents uncertain nodal weights and edge lengths.

Other studies related to the p-centre problem include Liu et al. (2010) who proposed a non-density-based approach related to spatial data analysis, Barua and Sander (2014) who devised a method to find dense co-located points, and Qu et al. (2014 who recently provided exact/approximate solutions to find a set of allied or alienated points.

2.2 Related work on the conditional (p, q)-centre problem

Minieka (1980) introduced the conditional location problem where conditional centres and medians on a graph were investigated. Drezner (1989) showed that conditional p-centre problems can be solved by solving O(log n) p-center problems, meaning that an effective algorithm for the p-centre problem can be adapted for the conditional problem. Berman and Simchi-Levi (1990) proposed an algorithm that requires the one-time solution of an unconditional $(p+1)$ center or $(p+1)$ median for solving the conditional $(p+1)$ center or $(p+1)$ median on networks. A method for solving minisum and minimax conditional location-allocation problems with $p \geq 1$ was developed by Chen (1990). Drezner (1995) introduced the term "(p, q) location problem".

A method for solving both the conditional p-median and p-center problems was studied by Berman and Drezner (2008). One-time solution of an unconditional p-median and p-center problem using the shortest distance matrix is used. Chen and Chen (2010) proposed a relaxation-based algorithm for solving the conditional discrete and continuous p-centre problem. Kaveh and Nasr (2011) investigated the conditional and unconditional p-centre problem using a modified harmony search algorithm.

2.3 Aggregation techniques for the p-centre problem

This subsection provides an overview of aggregation techniques focusing on p-centre problems. Hillsman and Rhoda (1978) classified aggregation errors into three types namely source A, B, and C errors. The use of the approximate distance between an Aggregate Spatial Unit (ASU) and a facility, instead of the true distance between a Basic Spatial Unit (BSU) and a facility, leads to the existence of those errors. Casillas (1987) introduced two measures to assess the accuracy of aggregated models namely the cost error and the optimality error.

Francis and Rayco (1996) and Rayco and Francis (1997) suggested aggregation schemes for the p-centre in the plane with rectilinear distances. Rayco et al. (1999) studied a grid-positioning aggregation procedure for the centre problem with rectilinear distance. Their procedure which consists of identical ‘diamonds’ of user-specified dimensions can also be utilised to estimate the maximum error, so letting the aggregation error to be kept within tolerable limits. Fortney et al. (2000) compared alternative measures of geographic access to health care providers using different levels of spatial aggregation and different cost calculations.

Francis et al. (2004a) investigated a demand point aggregation analysis for a class of constrained location models. Aggregation decomposition and aggregation guidelines for a class of minimax and covering location problems were studied by Francis et al. (2004b). They proposed a method to find an aggregation to attain a small error bound value. Later on, Francis et al. (2009) provided an excellent review of aggregation methods for location problems in general including the p-centre problem.

3 Methods for solving large p-centre problems

We propose two meta-heuristics for solving large p-centre problems namely a three-stage approach (TSA) and a guided multi-start based approach (GMA). Both methods incorporate Variable Neighbourhood Search (VNS), exact method and aggregation techniques. The former is an adaptation of the methods proposed by Irawan et al. (2014) and Irawan and Salhi (2014) initially designed for solving large-scale p-median problems whereas the latter is a new one.

3.1 A Three-stage Approach (TSA)

This method consists of three stages where the first stage is a learning process based on the aggregated problem. The second stage uses the information obtained from the previous stage namely the facility locations that act as the potential sites to solve aggregated problem by VNS. The last stage is a post-optimisation procedure where VNS is used to solve the original p-centre problem starting from the best solution obtained in the previous stage. In each stage, the problems are solved by either CPLEX (g, s, p) or VNS (g, s, p) where Method (g, s, p) refers to the procedure 'Method' for locating 'p' facilities, serving 'g' customers, and using 's' potential sites. Figure 1 presents the main stages of the Three-stage Approach (TSA). In this study, for the original problem, customer sites are used as potential facility sites (i.e. M= n).

The first stage is similar to Phase 1 of Irawan et al. (2013) except that a more efficient aggregation technique is used and an exact method is embedded into the search instead of VNS. In this stage, a number of aggregated problems are constructed. We aggregate n BSUs into m ASUs, with m << n. We define $\hat{d}(k, j)$ as the distance between the representative point of the k^{th} ASU and the j^{th} facility site. Consequently, each aggregated problem has m customers and m potential facility sites. Each aggregated problem is then solved by an exact method (m, m, p). The best way of solving the p-centre problem optimally is to utilise an auxiliary problem such as the Set Covering Problem which will be revisited in Subsection 4.3. As this approach requires initial upper and lower bounds, we incorporate VNS to generate such an input. The locations found by solving the aggregated problems are then stored in a list L.

Initialization

Determine the values of m and T. Set L = {Ø} where L denote a list of distinct facility locations obtained from the solutions of the aggregated problems.

Stage 1

Repeat the following steps T times (t = 1, …, T)

(i) Aggregate n BSUs into m ASUs and construct m clusters by allocating all BSUs to their nearest ASUs.

(ii) Calculate the distance between the k^{th} ASU and the j^{th} potential facility, $\hat{d}(k, j)$, k=1,..,m; *j=1,…,m*.

(iii) Solve the t^{th} aggregated p-centre problem using an exact method (m,m,p). Let $X_t = (\sigma_1^t, \sigma_2^t, ..., \sigma_p^t)$ be the obtained facility locations with σ_i^t denoting the i^{th} facility at iteration t and set $L = L \bigcup X_t$.

Stage 2

(i) Construct |L| clusters by allocating all BSUs to the closest point in L.

(ii) Compute the distance $\hat{d}(k, j)$, k=1,..,|L|; *j=1,…,*|L|.

(iii) Solve the aggregated p-centre problem by VNS (n, |L|, p) using the best obtained facility configuration from the previous stage as the initial solution.

Stage 3

Solve the disaggregated p-centre problem (i.e. the original problem) by VNS (n,n,p) using the solution obtained from Stage 2 as the initial solution.

Fig. 1 The main steps of the Three-stage Approach (TSA)

In Stage 2, the points in L are considered as the "promising" facility sites. This defines a p-centre problem which consists of n customers and |L| potential facility sites. This problem is solved with a VNS (n, |L|, p) using the best solution found in Stage 1 as the initial solution. The solution obtained in this stage is then used in Stage 3 as a starting solution.

In the final stage, the original (disaggregated) p-centre problem is solved by a VNS (n,n,p) starting from the solution obtained from the previous stage. At this point, the VNS is used as a post optimiser that is not expected to consume much extra computing time to solve the problem given its initial solution is of good quality.

3.2 A Guided Multi-start Approach (GMA)

The main idea behind this method is to provide flexibility in revisiting the aggregated problem so to produce a new solution configuration which is fed into a VNS. Similar to Stage 1 of TSA, n BSUs are aggregated into m ASUs, with m << n. The aggregated problem is solved by an exact method (m, m, p) producing an optimal facility configuration for the aggregated problem. This set of facility locations is then used as an initial solution for the original problem when applying the VNS. Figure 2 presents the main steps of GMA.

Initialization

Determine the values of m and N_{max}. Set f* = MAX_INT and X* = {Ø} where f* is the best objective function value and X* denote a list of the best facility configuration.

Main Steps

1. Set i = 1.
2. Generate the solution by solving the (m, m, p) aggregated problems
   - **(i)** Aggregate n BSUs into m ASUs.
   - **(ii)** Calculate the distance between the k^{th} ASU and the j^{th} potential facility, $\hat{d}(k, j)$, *k=1,..,m; j=1,...,m*.
   - **(iii)** Solve the aggregated p-centre problem using an exact method (m,m,p). Let f_0 and X_0 be the objective function value and the solution configuration respectively.
   - **(iv)** If $f_0 < f^*$ then set $f^* = f_0$ and $X^* = X_0$.
3. Apply VNS for the disaggregated (original) p-centre problem using X_0 as the initial solution. Let f_1 and X_1 be the objective function value and the solution configuration respectively.
4. Set i = i + 1 and set flag = false.
5. If ($f_1 < f^*$) then
   Set $f^* = f_1$, $X^* = X_1$, and $X_0 = X_1$.
   Set flag = true.
   End If
6. If ($i > N_{max}$) then stop.
7. If (flag = true) then go to Step 3.
   Else go to Step 2.

Fig. 2 The main steps of the Guided Multi-start Approach (GMA)

Firstly, the aggregated problem is constructed and solved by CPLEX (m, m, p). The obtained facility configuration is then used as an initial solution for the disaggregated problem when

using VNS. In our implementation of VNS, we call a number of times the VNS procedure (c_{max} in our study) where in each run VNS is performed until the k_{max}^{th} neighbourhood is explored without improvement. Once the process is completed, if there is an improvement we continue with the VNS, otherwise we diversify by solving again the aggregated problem leading to a new solution. This kind of multi-start is performed so to reduce the risk for the search from getting stuck. This process continues until a prescribed number of iterations (N_{max}) is performed.

Steps 2(i) to 2(iii) of GMA are similar to Stage 1(i) to 1(iii) of TSA whereas Step 3 of GMA is relatively similar to Stage 3 of TSA though the values of the parameters used are different. This will be presented in the computational results section.

4 Description of the Main Steps of both Approaches

In the next subsections, we explain our aggregation and the distance calculation methods. These are followed by the description of the exact method and the VNS. In the last subsection the adaptation of our approaches for the conditional (p, q) centre problem is presented.

4.1 The aggregation method

The procedure to aggregate n BSUs into m ASUs, used in Stage 1(i) of TSA and Step 2(i) of GMA, is described in this subsection. The procedure is an adaptation of the methods proposed by Irawan et al. (2014) and Irawan and Salhi (2014). The set of the m ASUs is obtained as follows: (i) ρ points are selected pseudo randomly where ρ will be set accordingly as it will be shown later; (ii) the remaining (m-ρ) are randomly chosen. This pseudo random scheme is based on the construction of the cells which is presented in Figure 3.

In the first step, we construct square cells that will cover all demand points with a side δ where we then delete empty cells. If the number of non-empty cells is not in the range of a prescribed number of ASUs then the value of δ is revised and the first step is repeated again. Once the specified number of the non-empty cells, ρ, is reached, a point is chosen randomly from each cell to represent the aggregated point (ASU) within that cell. Finally, to increase the diversity of the solutions, the remaining (m – ρ) ASUs are randomly generated. The main steps of the pseudo random scheme are given in Figure 4.

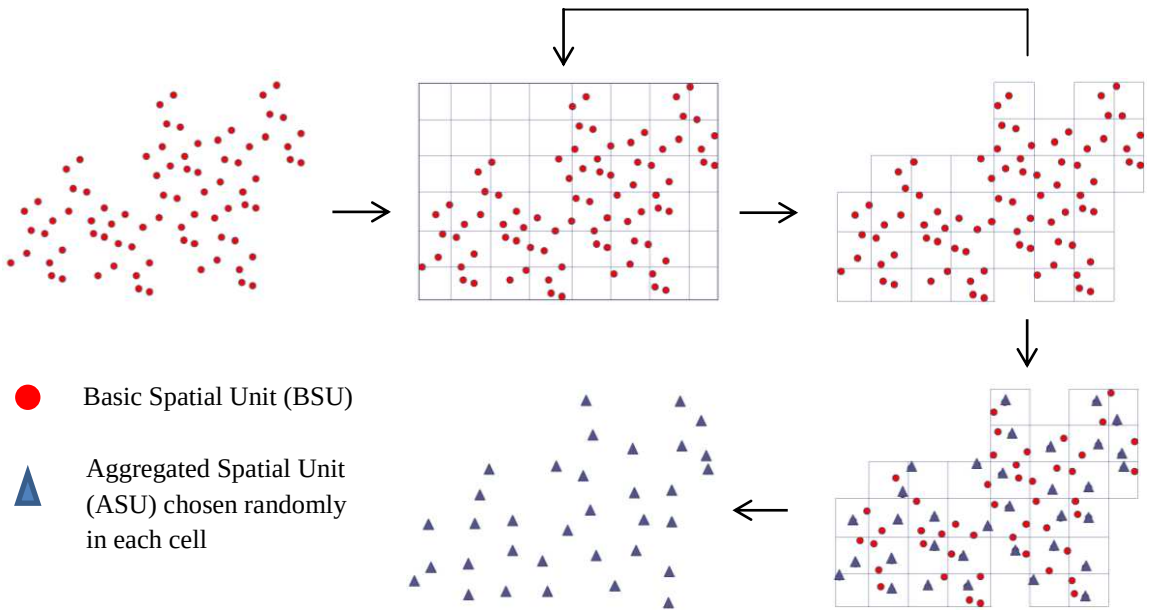


Fig. 3 The illustration of the pseudo random method (adapted from Irawan et al., 2014)

Step 1 Determine the values of m, γ, and λ.

Step 2 Initialise the length of the side of the cell δ as follows:

$$\delta = (x_{max} - x_{min}) / \sqrt{m\left(\frac{x_{max} - x_{min}}{y_{max} - y_{min}}\right)}$$

where x_{max} and x_{min} refer to the maximum and the minimum x coordinate of the points, respectively. Similarly, y_{max} and y_{min} refer to the maximum and the minimum y coordinates, respectively.

Step 2 Let ρ denote the number of non-empty cells, where $\rho \in [m\gamma(1-\lambda), m\gamma]$.

Step 3 Construct square cells of length δ which will cover all demand points where cell 1 has its bottom-left corner at (x_{min}, y_{min}). If $(\rho \in [m\gamma(1-\lambda), m\gamma]$ then go to Step 8.

Step 4 Let δ_L and δ_U be lower and upper bounds of the length of the side of the cell. Set $\delta_U = \delta$ and $\delta_L = \delta_U / 2$.

Step 5 Construct square cells of length δ_L. If $\rho < m\gamma(1-\lambda)$ then set $\delta_U = \delta_L$, $\delta_L = \delta_U / 2$ and repeat this step again, otherwise conduct the bisection method as follows.

Step 6 Calculate $\delta = (\delta_U + \delta_L) / 2$.

Step 7 Construct square cells of length δ. If $\rho \geq m\gamma(1-\lambda)$ and $\rho \leq m\gamma$ then go to Step 8, otherwise, if $\rho < m\gamma(1-\lambda)$ then $\delta_U = \delta$ else $\delta_L = \delta$. Go to Step 6.

Step 8 Allocate all demand points to their cells. Choose randomly a demand point in each cell which makes up ρ aggregated points.

Step 9 To complement m aggregated points, the remaining $(m-\rho)$ demand points are chosen randomly.

Fig. 4 The main steps of our aggregation method (Adapted from Irawan et al., 2014)

Another way of aggregating the demand points would be to first define initially n subsets of one demand point each, then combine the closest pair of subsets into one bigger subset and continue this way until the number of aggregated subsets is reduced to m. The 1-median point for each of the m clusters could then be used to represent the ASU for that cluster. This scheme, though interesting, took extremely long when tested on the large instances. For example, the determination of the closest pair of subset on its own consumed more than 170 seconds for n = 71,009 instead of around 10 seconds for our method to aggregate the same demand points. This new subset, if found in moderate time, could have been added to our random-based cells subsets which we generated.

Observation

In the p-centre problem, the optimal solution can be obtained by solving the aggregated problem. This occurs when all the 'critical' demand points are included in the aggregated points (ASUs). Figure 5 illustrates how the aggregated problem yields the optimal solution where the original (disaggregated) problem consists of 16 demand points and the number of facilities to be located is 2 (p=2). By solving it visually, it is clear that the facility locations will be in the middle and the objective function value is r_m. We aggregate these 16 points (n=16) to 8 points (m=8) where all the critical points are included in the aggregated problem. Figure 5 also displays that the facility locations and the objective function value for the aggregated problem are the same as the ones of the original problem. However, designing a method that identifies these 'critical' points is in itself a hard problem to solve.

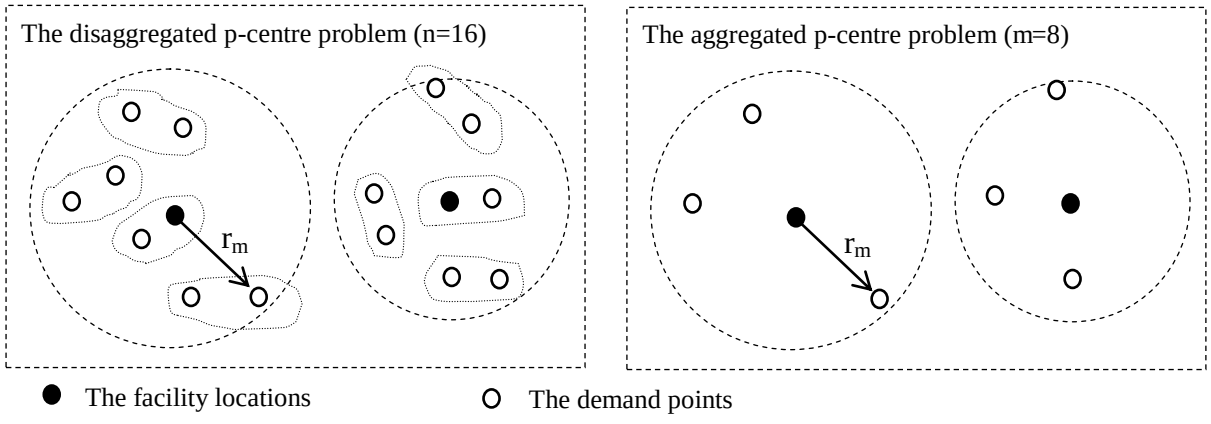


Fig. 5 Illustration of the aggregated problem yielding the optimal solution (p = 2)

4.2 A new distance calculation method for aggregated p-centre problems

Let *C'* denote the list of ASUs. To solve the aggregated p-centre problem with an exact method or a VNS, the distance matrix between points in C' needs to be calculated. For the p-median problem, Current and Schilling (1987) introduced a method for eliminating source A and source B errors. A distance between the k^{th} ASU and the j^{th} facility is set as $\hat{d}(k, j) = \sum_{i \in N_k} d(i, j)$ with N_k being the set of aggregated BSUs in the k^{th} ASU. We do not use their method as the objective function is the minimax instead of the minisum.

We propose another way which is more informative for the distance calculation. First, as in Current and Schilling's method, BSUs are aggregated into their nearest ASUs. The maximum distances (r_k , $k=1,...,|C'|$) between ASUs and their aggregated BSUs are then determined. Let $\tilde{d}(k, j)$ denote the true (real) distance between the k^{th} ASU and the j^{th} facility. The distance $\hat{d}(k, j)$ is set as $\hat{d}(k, j) = \tilde{d}(k, j) + r_k$. Figure 6 presents the illustration of our distance calculation method where it is assumed that the demand at BSU k, i, i+1 and i+2 has been aggregated as ASU k.

The reasoning behind this distance representation is to compute r_k once only and $\tilde{d}(k, j)$ when the location of facility j changes. This is much quicker than simply taking the maximum distance between the facility and all members of the ASU as this will need to be carried out every time the location of facility is changed which can be computationally excessive.

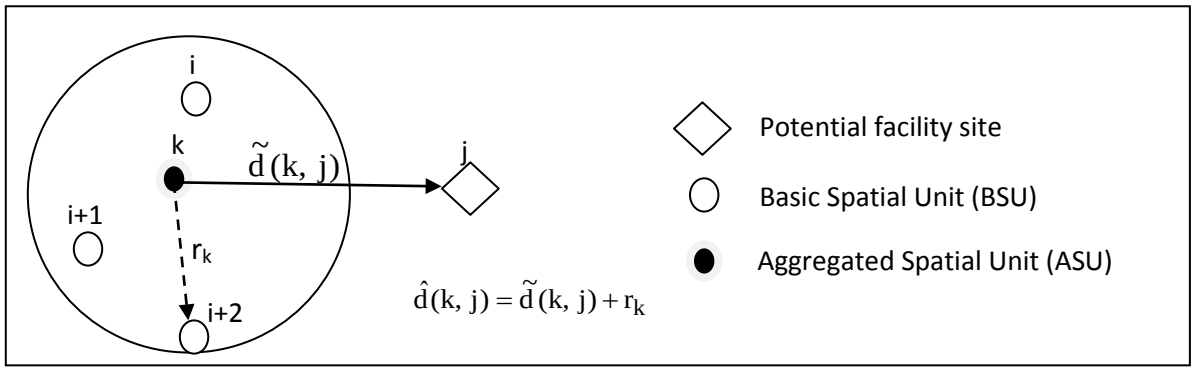


Fig. 6 The distance calculation method for the p-centre problem

A preliminary study was also carried out to compare these two calculation methods. The results showed that for the aggregated p-centre problem, the use of our proposed calculation method provides much better solutions than the average distance based on Current and

Schilling's method. In addition, the latter requires an excessive computational time due to the issues mentioned above.

4.3 An exact method for solving the aggregated vertex p-centre problem

The size of the aggregated problem is small enough to be solved optimally using the Set Covering Problem (SCP)-based approach as will be shown here. SCP aims to find the minimum number of facilities and their locations so that each customer is served by a facility within a given distance (or response time). Let D denote the given distance (covering distance), the matrix $A = (a_{ij})$ can be defined as follows:

$$a_{ij} = \begin{cases} 1 & \text{if customer } i \in I \text{ is covered by facility } j \text{ (ie. } d_{ij} \leq D) \\ 0 & \text{otherwise} \end{cases}$$

The SCP can be formulated as follows:

Minimise $$\sum_{j \in J} X_j \qquad (8)$$

Subject to

$$\sum_{j \in J} a_{ij} X_j \geq 1 \qquad \forall i \in I \qquad (9)$$

$$X_j \in \{0,1\} \qquad \forall j \in J \qquad (10)$$

The objective function (8) is to minimise the number of facilities. Constraints (9) ensure that each customer is served by at least one facility located within D whereas constraints (10) refer to the binary variables.

To solve the p-centre problem, the SPC is solved recursively using a binary search. Efficient exact algorithms for solving the p-centre problem include, for example, Daskin (1995; 2000), Ilhan and Pinar (2001), Elloumi et al. (2004), Al-Khedairi and Salhi (2005), and Salhi and Al-Khedairi (2010). Our algorithm is a hybrid of the last two where (i) a VNS is used to obtain tight upper bound and its corresponding lower bound, (ii) an ordered list of the distance matrix elements is constructed, (iii) a scheme that efficiently identifies the nearest value in the distance matrix to the new coverage value found by the binary search (i.e., value = (lower bound + upper bound)/2) is proposed, and (iv) a more effective stopping criterion is adopted. The latter is based on detecting the empty gap between the final lower and upper bounds. Figure 7 presents our SCP-based algorithm to solve the p-centre problem optimally.

In Step 1, to obtain a tight upper bound, a VNS is applied as it is also used in other phases of this study and its description will be given in Section 4.4. The idea of sorting the distance matrix in the vector G, containing distinct elements only, is quite simple but very effective in reducing the number of iterations needed to find the optimal solution as values of D in Step 6 that do not exist in G are not tried but their closest element in G is used instead. In addition, when there is no element in the vector G with distance value between L and U, there is no need to continue the binary search unnecessarily. In such a case (i.e., gap is empty), the optimal solution is exactly the upper bound value (U). Note that in other implementations, redundant iterations (i.e. solving more SCPs) could have been used till $U - L \leq 1$ if integer values were required and $U = L$ otherwise.

Step 1 Apply VNS to obtain the initial upper bound (U)

Step 2 Sort the distance matrix in ascending order in a vector G. Convert the distance values into integers (e.g. by multiplying by 1000 and then rounding the values) and remove any duplicates. Convert the value of the upper bound to an integer value in a similar way.

Step 3 Set the lower bound (L) to αU, where α is a parameter. Find a distance value in the vector G which is the closest to L and then update the value of L with the value found.

Step 4 Solve the SCP for the coverage distance L and let z be the number of facilities found.

If ($z \leq p$) then set $U = L$, $L = \alpha U$ and repeat this step again.

Step 5 If there is element in G between L and U then the optimal solution is U and the number of facilities found is z and then stop.

Step 6 Calculate $D = (U + L)/2$. Find a distance value in the vector G which is the closest to D and then update the value of D with the value found. Let z_u denote the number of facilities found for the upper bound.

Step 7 Solve the SCP for the coverage distance D.

If ($z > p$) then set $L = D$, otherwise set $U = D$ and $z_u = z$.

Step 8 If there is no element in G between L and U then

the optimal solution is U and the number of facilities found is z_u and then stop.

Else go to Step 6.

Fig. 7 Our proposed optimal method for solving the p-centre problem

The upper bound produced by the VNS in Step 1 can be a good solution as this may not be too far from the optimal in most cases. Steps 2-5 of Figure 7 aim to get a tight lower bound which can be obtained by setting the value of α close to 1 (for example, 0.8 – 0.9). Note that

the lower bound must also exist in the distance matrix, which is the closest value to αU. In other words, the more powerful the VNS is, the higher is α. If L=αU happens to generate a feasible solution when solving SCP, set U=L and L=αU again, and the process continues until we have a proper range [L,U] from which the binary search starts. Steps 6-8 of Figure 7 are the usual steps of the bisection (binary search) method. Similar to the lower bound generator, the coverage distance (D) has to be in the vector G, which is the closest to the average of L and U. This process stops when there is element in G between [L, U]. The optimal solution is then taken to be U and the number of facilities found is z_u ($z_u \leq p$). Solving the p-centre problem with the above method yields interesting results. It runs relatively much faster than the one using the classical p-centre formulation ((1)-(7)).

Observation

In special cases, it is worth noting that the optimal solution, U, might be obtained by locating a number of facilities $z_u < p$, though yielding the same objective function values as locating p facilities. This could occur in the following two cases.

(i) A facility with the largest radius (r_m) happens to serve all its customers with the same radius as presented in Figure 8. Besides serving customer i, the facility located at customer i serves the other three customers namely customer i+1, i+2, and i+3. The distance between this facility and those three customers is the same which is r_m. Figure 8 shows the p-centre problem with p = 2 and 3 which give the same optimal solution (r_m). In the case p = 3, we try to split the largest circle obtained by solving the p = 2 problem. A facility is inserted at customer i+1 and the facility located at customer i is moved to customer i+3. However, this failed to reduce the maximum distance (r_m) between a customer and its nearest facility. Therefore, the optimal solutions for p = 2 and 3 are the same. Note that this reasoning is not valid in the continuous space.

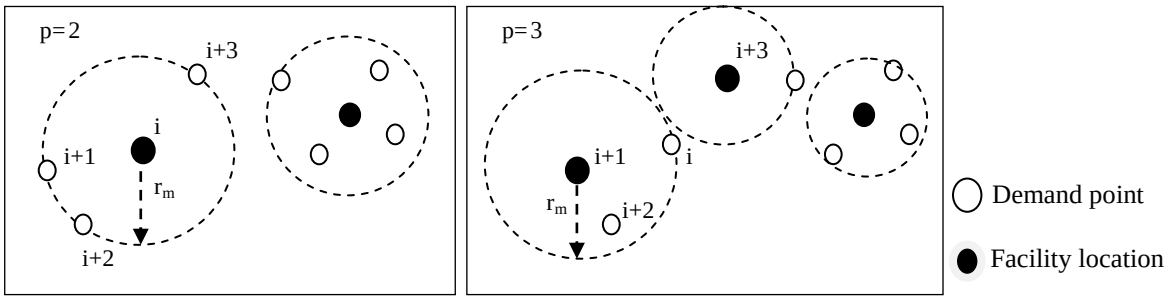


Fig. 8 The case where the distance between a facility and all its customers is the same

(ii) Let $z_u = p - s$, where s is the number of redundant (unneeded) facilities. If $s' \geq (s+1)$ facilities have the same maximum distance (r_m) (i.e. there are s' alternate optimal solutions).

Here, the optimal solution is obtained from those (s′) facilities which is given in Figure 9 where s′ = 3 for p = 7 and z_u = 5.

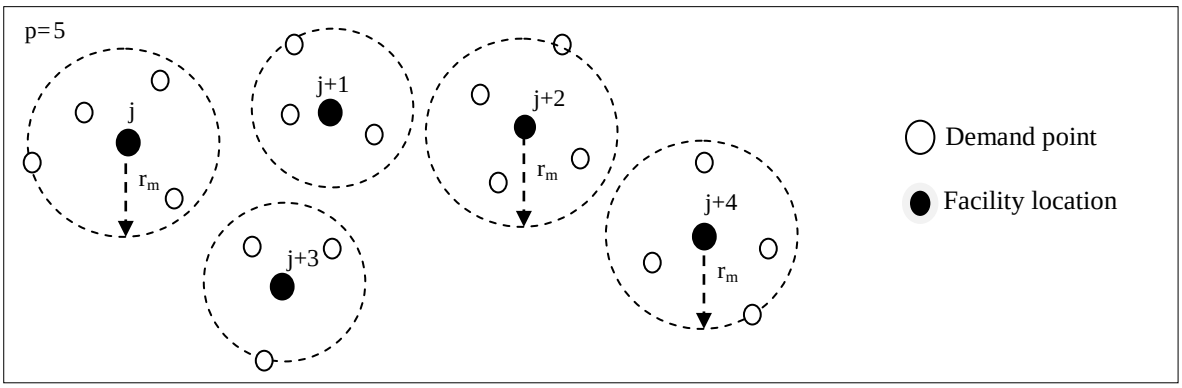


Fig. 9 The case where (s+1) facilities have the same maximum distance (r_m)

There are three facilities whose biggest radius (r_m) is the same which include facility j, j+2, and j+4. Inserting up to 2 facilities (s = 2) does not necessarily reduce the optimal solution when at least one furthest customer (from its facility) is still allocated to the same facility. In this case, the optimal solution for p = 5 and 7 is the same resulting in two redundant facilities.

4.4 The VNS algorithm

Variable Neighbourhood Search (VNS) was formally proposed by Hansen and Mladenovic (1997) for the solution of the p-median problem. VNS incorporates a local search which seeks local optima (intensification) and a systematic change of neighbourhood search (diversification) which intends to escape from local optima. VNS was implemented for the solution of the p-centre problem by Mladenovic et al. (2003) with good results. For more information and applications of VNS, see Hansen et al. (2010).

In this study, VNS is used to solve the (m, m, p), the (n, |L|, p), and the (n, n, p) centre problems in the TSA. In Stage 1 of TSA and Step 2 of GMA, VNS is utilised to solve the (m, m, p) centre problem to obtain the upper bound (UB) for the exact method. The (n, |L|, p) centre problem is solved by VNS in Stage 2 of TSA where the promising facilities found in the previous stage are considered as the potential sites. In the last stage (Stage 3) of TSA and Step 3 of GMA, VNS is applied on the original (disaggregated) p-centre problem (n, n, p).

Initial VNS implementation

Our VNS is based on the implementation proposed by Mladenovic et al. (2003) which is summarised in Figure 10. Let i_m refer to the customer whose largest distance to its nearest

facility while B denotes the list of customers which are located within r_m from customer i_m ($d(i, i_m) < r_m$). The set of neighbourhood structures (N_k), k = 1, 2, …, k_{max} is defined by swapping k times a randomly chosen facility location (say at customer in, where in $\in$ B) with one facility chosen randomly in the current solution.

In the local search, the vertex substitution heuristic was implemented. For each facility ($j=1,...,p$), its best substitution point (the point in B) is obtained by the procedure "Move" (see Mladenovic et al. 2003) using the best improvement strategy. Customers are then allocated to their nearest facility. This process is repeated until there is no improvement.

1. Choose randomly an initial solution (x_{best}), calculate r_m, determine i_m, and set k = 1, $x_{now} = x_{best}$, $r_{now} = r_m$ and $i_{now} = i_m$.
2. Repeat the following steps until $k = k_{max}$
   (i) Shaking process
   For j = 1 to k
   Choose randomly a facility (in) in B (i.e., $d(i_{now}, in) < f_{now}$), and swap it with a random one in x_{now} . Calculate r_{now} and determine i_{now}.
   (ii) Local search
   Apply the vertex substitution heuristic with x_{now} as an input. The heuristic returns the solution x_{now2}, r_{now2}, and i_{now2}. Set $x_{now} = x_{now2}$, $r_{now} = r_{now2}$ and $i_{now} = i_{now2}$.
   (iii) Move or Not
   If ($r_{now} < r_m$) set $x_{best} = x_{now}$, $r_m = r_{now}$, $i_m = i_{now}$ and k = 1
   Else set $x_{now} = x_{best}$, $r_{now} = r_m$, $i_{now} = i_m$ and k = k+1.

Fig. 10 A VNS implementation for the p-centre problem

An enhanced VNS implementation

We enhance the shaking process of the algorithm with the aim in reducing the computing time while enhancing the quality of the solution. Instead of choosing a facility randomly from the current solution (x_{now}), we choose a facility (say facility j) whose radius (the maximum distance between a facility and its customer) is the largest (r_m). We then move this facility to a customer site (customer in) served by facility j where $d(i_m, in) < r_m$. We also restrict the location of customer in not to be too close to customer i_m. This concept of using forbidden regions is shown to be effective when solving the multi-source Weber problem, see Gamal and Salhi (2001). Here, we set $d(i_m, in) > r_m/2$. Figure 11 illustrates our neighbourhood structure.

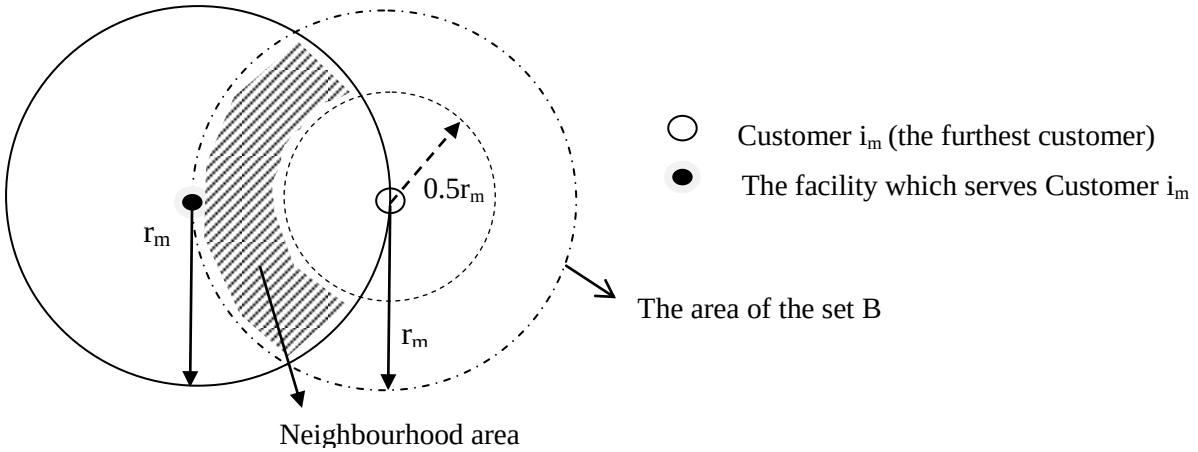


Fig. 11 The restricted but guided neighbourhood

In case there is no customer in the neighbourhood area, the shaking process is conducted by using the procedure of Step 2(i) of Figure 10. A preliminary study showed that our neighbourhood structure reduces the computing time and improves the quality of the solution. Figure 12 presents the enhancement of the VNS algorithm.

Initialization

Set the initial solution. Choose p points randomly for Stage 1, while for Stages 2 and 3 take the best solution from the previous steps.

Repeat c_{max} times the following steps:

Step 1 Set $k = 1$

Step 2 Shaking

Do the following step k times

- Move the facility which serves customer i_m to a customer site randomly in the neighbourhood. If there is no customer site in the neighbourhood apply Step 2(i) of Figure 10. Determine the objective function and identify the corresponding furthest customer.

Step 3 Local Search

Apply the vertex substitution heuristic using the best improvement strategy.

Step 4 Move or Not

If there is an improvement, update the solutions and set $k = 1$ else $k = k+1$.

Step 5 If $k \leq k_{max}$ then go to Step 2.

Fig. 12 The enhanced VNS for solving the p-centre problem

Let c_{max} denote the number of cycles (times) the VNS is executed. The value of c_{max} and k_{max} are set depending on the problem to be solved (i.e., the (m,m,p), the (n,|L|,p), or the (n,n,p) centre problem). The setting of the parameters will be presented in the computational

results section. The shaking process uses the neighbourhood structure described above while the local search remains the vertex substitution heuristic.

4.5 Adaptation of the methods for the (p, q) centre problem

Both TSA and GMA, which are developed for the p-centre problem, are easily adapted to solve the (p, q) centre problem (i.e., the conditional p-centre problem). The revised approaches which we refer to TSAq and GMAq consist of the following modifications.

a) The aggregation method (subsection 4.1)

The q existing facility locations are considered as the aggregated points (C'). The ρ aggregated points are added pseudo randomly to C' as described earlier while the remaining (m-ρ-q) points are chosen randomly.

b) The exact method (subsection 4.3)

Let Q be the set of existing facilities ($Q \subset J$).

To solve the (p, q) centre problem optimally, we add constraints (11) to equations (8) – (10) to ensure that the q existing facilities are always in the solution.

$$X_j = 1 \quad \forall j \in Q \tag{11}$$

The addition of constraints (11) into the p-centre formulation makes the problem relatively much easier to solve.

c) The VNS (subsection 4.4)

We fix the existing facilities in the solutions in both the shaking and the local search. In other words, the existing facilities cannot be removed from the solution.

- The shaking

  If customer i_m (the furthest customer) is not served by one of the existing facilities, we then use the enhancement procedure in the shaking process. Otherwise the shaking process is performed by the procedure of Step 2(i) of Figure 10 with the following additional rule: when a facility is randomly chosen from the current solution (say facility j), facility j cannot be one of the existing facilities (i.e., $j \notin Q$).

- The local search

  Because the existing facility locations are always part of the solution, the implementation of the best improvement strategy does not include these facilities.

5 Computational Results

We carried out a computational study to assess empirically the performance of our solution methods when solving both the unconditional and the conditional p-centre problems. The code was written in C++.Net 2010 and used the IBM ILOG CPLEX version 12.5 Concert Library. The code was executed on a PC with an Intel Core i5 CPU 650@ 3.20GHz processor, 4.00 GB of RAM and under Windows 7 (32bit).

The TSP dataset is used in our testing. These can be downloaded from http://www.tsp.gatech.edu/world/countries.html or http://www.kent.ac.uk/kbs/research/research-centres/clho/datasets.html. We classify this dataset into two types: small and large datasets. The small dataset consists of Oman Data (n = 1,979), Canada Data (n = 4,663), and Tanzania Data (n = 6,117) whereas the large one comprises Sweden Data (n = 24,978), Burma Data (n = 33,708), and China Data (n = 71,009). For most instances of the small dataset, the optimal solutions can be obtained for both the unconditional and conditional p-centre problems using the exact method described in Section 4.3. In other words, for these small instances we compare the performance of our methods against the optimal solution.

5.1 Parameter settings and notations

Following a preliminary study, the following parameters are selected as follows: m = 500 and 400 for TSA and m = 1,000 and 800 for GMA for small and large datasets respectively. The number of aggregated points was made dependent on the size of the original problem as it influences the quality of the solution. The higher this value is, the higher the chance of obtaining a better solution. However, the computing time required also increases with m.

We also set $\lambda = 0.05$ meaning that the number of aggregated points generated by the pseudo random method to be in the range [70, 75]% with $\gamma = 0.75$ for the large dataset and [95, 100]% with $\gamma = 1$ for the small dataset. The remaining points are generated randomly. In this study, the value of γ is the same for both TSA and GMA.

In Stage 1 of TSA, the number of iterations (T) affects the number of promising facilities which also affects the quality of the solution. The possibility of obtaining a good solution increases when T is high, but this requires a relatively longer computing time. Here, we set T=10. When solving the aggregated problem by the exact method, the parameter α needs to be determined for getting the lower bound which is based on the upper bound obtained by the VNS ($L = \alpha U$) as suggested by Salhi and Al-Khedhairi (2010). We set $\alpha = 0.5$ and 0.8 for the

unconditional and conditional problems, respectively. This means that the gap between the upper and lower bounds of the conditional problem is tighter than the one of the unconditional case. In the GMA, the value of N_{max} is set to 5.

In the VNS, we set $k_{max} = \min\{\max\{p,10\},20\}$ whereas the parameter setting of c_{max} is given in Table 1.

Table 1 Parameter setting of c_{max} for the VNS method

The type of the problem	TSA	GMA
(m,m,p) problem		
Small and large datasets	1	1
(n,|L|,p) problem		
Small and large datasets	5	-
(n,n,p) problem (Stage 3)		
Small dataset	min{max{p,10},20}	5
Large dataset	5	1

The results of our experiments are presented in several tables using the following notations:

- n: number of demand points
- p: number of new facilities to be located
- Z: objective function value with Z* and Z** being the optimal solution for the unconditional and conditional problems respectively.
- EM : Exact Method.
- Time: computational time in seconds.
- Deviation(%): this is the percent gap from the best known solution (or optimal if it exists) and is computed as:

  $\text{Deviation} = 100\left(\frac{Z_c - Z_b}{Z_b}\right)$, where Z_c and Z_b correspond to the Z value obtained with method 'c' and the best Z (or optimal Z) value respectively.

The next two subsections present experiments on the unconditional and the conditional p-centre problems respectively.

5.2 Computational results on the unconditional vertex p-centre problems

For the small dataset, each instance is solved with p varying from 5 to 100 with a step of 5 totalling 24 instances whereas we vary the value of p from 25 to 100 for the large dataset with an increment of 25 totalling 12 instances. For small dataset, we also give the average results with their respective standard deviations based on 10 runs.

Small dataset

The performance of our methods (TSA and GMA) on the small dataset is compared against the optimal solution (Z*) obtained by the exact method which is described in Figure 6, see Table 2.

Table 2 Statistical Results for the small unconditional p-centre problems (based on 10 runs)

p	Exact Method	Deviation from Z* (best) (%)		Z value TSA		Z value GMA		Avg Time (Seconds)		
	Z*	TSA	GMA	Avg	Std	Avg	Std	EM (Z*)	TSA	GMA
Oman Data (n = 1,979)										
5	1,876.83	**0.000**	**0.000**	1,876.83	0.000	1,876.83	0.000	67.50	36.49	78.46
10	1,160.70	**0.000**	**0.000**	1,160.70	0.000	1,160.70	0.000	52.53	30.29	59.57
15	867.52	**0.000**	**0.000**	867.52	0.000	867.52	0.000	36.65	38.92	63.29
20	750.53	**0.000**	**0.000**	762.97	8.582	764.74	7.491	38.90	70.12	66.36
25	638.79	**0.000**	**0.000**	642.83	3.852	641.69	1.020	30.31	81.10	62.22
50	380.90	1.798	**0.000**	395.31	6.507	382.21	2.331	30.47	127.45	127.59
75	284.80	1.926	**0.000**	303.29	6.589	289.74	1.735	27.64	147.60	103.67
100	220.32	6.176	1.773	237.69	1.983	225.48	3.019	45.09	181.67	152.42
Canada Data (n = 4,663)										
5	16,836.61	**0.000**	**0.000**	16,842.78	5.315	16,845.87	3.255	1,031.14	124.71	310.73
10	10,498.81	**0.000**	**0.000**	10,498.81	0.000	10,504.32	17.424	630.57	129.47	288.37
15	8,295.93	**0.000**	**0.000**	8,299.87	12.459	8,358.78	71.781	465.87	226.95	367.17
20	7,023.87	**0.000**	**0.000**	7,030.85	22.082	7,088.87	66.990	417.31	365.00	477.64
25	5,965.76	0.745	**0.000**	6,090.21	47.605	6,073.15	71.381	409.11	363.72	426.21
50	3,955.06	0.439	**0.000**	4,086.80	83.779	3,978.48	17.069	508.30	342.38	333.58
75	3,069.32	2.765	2.765	3,208.67	84.152	3,168.54	14.488	575.54	325.61	279.10
100	2,543.89	1.635	1.635	2,685.69	78.784	2,589.17	7.108	471.85	378.03	276.70
Tanzania Data (n = 6,117)										
5	2,917.86	**0.000**	**0.000**	2,918.43	1.805	2,918.43	1.805	3,725	543.74	1,324.64
10	1,902.12	**0.000**	**0.000**	1,915.88	12.293	1,929.79	33.982	11,366	366.25	883.26
15	1,527.98	1.400	0.475	1,558.23	8.919	1,564.66	21.517	35,875	727.24	1,142.96
20	1,278.30	0.318	1.002	1,293.42	11.538	1,309.82	12.965	25,375	1,120.98	1,398.05
25	1,152.05	1.114	1.114	1,178.33	9.692	1,184.32	10.021	362,943	934.22	1,152.35
		Z(TSA)	Z(GMA)							
50	N/A	**806.23**	**806.23**	820.18	11.051	824.71	820.74	N/A	637.26	903.44
75	N/A	**663.53**	663.74	679.38	9.903	676.17	671.85	N/A	585.96	719.04
100	N/A	579.75	**566.18**	596.81	12.974	588.02	589.02	N/A	618.42	686.61
	Average	0.872	**0.417**		19.330		17.399	21,148	317.24	446.40

The specification of the computer used to obtain the optimal solution for Tanzania Data (n=6,117) is slightly different as we need a greater capacity of memory (RAM). Here, we used a PC Intel Core 2Duo 2.6GHz, 8 GB of RAM to solve these problems optimally. According to Dongarra's (2013) transformation, this computer is approximately 80% faster

than the one that we used to execute other instances. In Table 2, for Tanzania data, the computing time required to obtain the optimal solution has been adjusted accordingly. The optimal solutions of this instance for p = 100 could not be obtained due to memory issue.

Table 2 shows that both TSA and GMA are able to find the optimal solutions when $p \leq 50$. In general, GMA performs slightly better than TSA as it found the optimal solutions in 15 out of 21 instances while TSA produces 11. Regarding the deviation from the optimal solution, GMA also yields a relatively smaller average deviation (0.417%) compared to the one of TSA (0.872%). This deviation increases with p and n. The effect of the increase of p appears to be more significant than the one of n. For both methods, the average computing time is found to be relatively much smaller than that of the exact method.

Large dataset

The computational results of our methods on large p-centre problems are given in Table 3. For these problems we do not have the optimal solutions or other results that we can compare with. We just analyse the deviation (%) and the computing time between TSA and GMA.

Table 3 Computational Results for the large unconditional p-centre problems

p	Best known (Z_b)	Deviation (%)		Time (Seconds)	
		TSA	GMA	TSA	GMA
Sweden Data (n = 24,978)					
25	1,329.37	6.6185	0.0000	10.80	1,300.90
50	925.71	3.8950	0.0000	621.42	1,499.67
75	759.02	0.1445	0.0000	919.13	1,080.59
100	685.77	0.7063	0.0000	652.09	897.27
Burma Data (n = 33,708)					
25	1,183.80	0.0000	0.0000	725.08	839.85
50	823.27	0.0000	0.5110	1,164.58	1,072.08
75	683.94	2.8109	0.0000	769.46	1,105.56
100	593.48	1.2540	0.0000	552.04	1,823.30
China Data (n = 71,009)					
25	4,428.72	1.4251	0.0000	7,837.74	7,543.40
50	3,107.56	2.1134	0.0000	7,603.29	7,536.78
75	2,554.32	0.0000	0.3072	7,538.15	7,524.98
100	2,168.97	1.7211	0.0000	7,499.61	6,818.05
	Average	1.9286	**0.0639**	2,991.12	3,253.53

When solving large p-centre problem, the local search (vertex substitution heuristic) of the VNS used to solve the original (disaggregated) problem is slightly modified to reduce the

computing time. Here, the substitution points are in the area B (see Figure 11) and their distance to customer i_m is less than $r_m \cdot (1\text{-}p/100)$. For the large dataset, we also limit the computing time of the VNS in Stage 3 of TSA to 1.5 hours and in Step 3 of GMA to 0.5 hours. In general, the methods run relatively fast (more or less 3,000 seconds on average). Similar to the results of the small dataset, GMA is found to be superior to TSA as it produces a smaller average deviation (0.0639).

5.3 Computational results on the conditional (p, q) centre problems

Our modified approaches for solving the conditional p-centre problem are also assessed on the TSP dataset that was tested on the unconditional p-centre problem. The existing q facilities in the (p, q) centre problem are taken from the solutions (the optimal solution for small instances) produced by solving the p-centre problem in the previous subsection. For instance, for the (p=10, q=5) centre problem, the existing 5 facility locations are the solution of the (p=5) centre problem. We compare the objective function of the (p=10, q=5) centre problem to the unconditional (p=15) centre problem. When the exact method is used, the value of the objective function of the (p=10, q=5) centre problem is obviously worse than or equal to the one of the (p=15) centre problem.

The above setting will demonstrate how much loss was produced by restricting some of the facilities when solving the new p-centre problem. In other words, with such a setting the solution of the latter acts as a lower bound for the conditional problem. From a managerial view point, this could also be used to evaluate whether or not to close some of these already opened facilities and replacing them by the new optimal (or best) ones if necessary. Another experiment, which can also be performed, would be to take q locations randomly from the optimal locations of the p-centre problem and solve the (p-q,q) conditional problem. This will enable us to see the effect of the subset of the optimal facilities within the p-centre problem and how much the additive property in the p-centre is violated.

Small dataset

The computational results of TSAq and GMAq on the small TSP dataset are presented in Tables 4 and 5 where the deviation (%) between the optimal solution found by the exact method (EM) for the (p, q) problems is presented. The tables also show the performance of TSAq and GMAq based on the deviation (%) and the computing time (in seconds). Tables 4

and 5 present the computational results on the small (p, q) problems for the small and the large values of p respectively.

Table 4 Computational Results on small (p, q)-centre problems for small p (10 runs)

p	q	Z**	Deviation from Z** (best) (%)		Z value TSAq		Z value GMAq		Avg Time (seconds)		
			TSAq	GMAq	Avg	Std	Avg	Std	EM	TSAq	GMAq
Oman Data (n=1,979)											
5	5	1,455.35	0.000	0.000	1,455.354	0.000	1,455.354	0.000	15	26	69
10	5	1,019.08	0.000	0.000	1,019.077	0.000	1,019.077	0.000	29	64	102
5	10	1,109.55	0.000	0.000	1,109.554	0.000	1,109.554	0.000	9	22	41
15	5	779.70	0.000	0.000	779.698	0.000	779.698	0.000	26	110	120
10	10	883.33	0.000	0.000	883.333	0.000	883.333	0.000	8	65	86
5	15	827.65	0.000	0.000	827.647	0.000	827.647	0.000	8	66	82
20	5	676.13	0.000	0.000	683.841	4.062	680.186	4.819	23	103	134
15	10	759.58	0.000	0.000	759.580	0.000	759.580	0.000	7	102	115
10	15	817.35	0.000	0.000	817.347	0.000	817.347	0.000	7	55	65
5	20	736.71	0.000	0.000	736.711	0.000	736.711	0.000	6	53	64
Canada Data (n=4,663)											
5	5	13,622.13	0.000	0.000	13,622.133	0.000	13,622.133	0.000	66	76	179
10	5	9,661.06	0.000	0.000	9,661.062	0.000	9,675.167	44.604	85	425	779
5	10	10,250.66	0.000	0.000	10,250.664	0.000	10,250.664	0.000	49	87	140
15	5	7,254.92	0.000	0.000	7,254.922	0.000	7,254.922	0.000	83	720	1,023
10	10	8,968.54	0.000	0.000	8,968.541	0.000	8,968.541	0.000	48	397	499
5	15	8,130.41	0.000	0.000	8,130.413	0.000	8,130.413	0.000	34	209	263
20	5	6,447.44	1.344	1.344	6,549.629	49.110	6,560.537	49.944	93	538	697
15	10	7,244.33	0.000	0.000	7,244.327	0.000	7,244.327	0.000	70	558	800
10	15	7,262.08	0.000	0.000	7,262.078	0.000	7,293.942	100.761	36	323	403
5	20	6,892.35	0.000	0.000	6,892.347	0.000	6,892.347	0.000	29	274	338
Tanzania Data (n=6,117)											
5	5	2,540.56	0.000	0.000	2,540.560	0.000	2,540.560	0.000	80	417	995
10	5	1,705.95	0.000	0.000	1,705.954	0.000	1,711.037	10.728	181	1,037	1,853
5	10	1,874.17	0.000	0.000	1,874.166	0.000	1,874.166	0.000	49	158	271
15	5	1,454.94	0.000	0.000	1,461.835	12.609	1,474.225	25.146	797	1,228	1,663
10	10	1,625.24	0.000	0.000	1,625.235	0.000	1,625.235	0.000	52	1,250	1,608
5	15	1,512.63	0.000	0.000	1,512.632	0.000	1,512.632	0.000	35	219	275
20	5	1,206.12	1.026	1.563	1,228.947	10.871	1,233.200	9.367	922	996	1,245
15	10	1,397.82	0.567	0.000	1,406.245	1.591	1,427.460	21.422	56	1,015	1,287
10	15	1,460.69	0.000	0.000	1,460.974	0.601	1,461.686	0.689	37	439	542
5	20	1,274.38	0.000	0.000	1,274.380	0.000	1,274.380	0.000	23	189	223
Average			0.10	0.10		2.63		8.92	99	374	532

Table 5 Computational Results on small (p, q)-centre problems for large p (10 runs)

p	q	Z**	Deviation from Z** (best) (%)		Z Value TSAq		Z Value GMAq		Avg Time (seconds)		
			TSAq	GMAq	Avg	Std	Avg	Std	EM	TSAq	GMAq
Oman Data (n=1,979)											
40	10	412.56	6.653	0.000	440.013	0.000	418.733	3.251	9.19	67.88	109.99
35	15	426.34	0.000	0.000	436.571	4.467	431.179	6.242	6.17	94.86	112.91
30	20	449.47	0.000	0.000	449.679	0.276	449.572	0.225	6.58	125.11	124.25
25	25	482.74	0.000	0.000	486.274	3.670	486.299	3.644	6.80	101.09	120.00
65	10	293.27	4.483	0.000	311.051	6.006	296.722	4.482	10.47	86.57	116.67
60	15	303.12	1.086	0.820	306.413	0.000	305.702	0.254	6.73	80.08	109.39
55	20	313.93	0.731	0.731	317.261	1.090	316.339	0.234	6.82	123.77	94.52
50	25	305.61	0.264	0.264	308.228	4.068	306.503	0.286	8.43	113.88	117.44
90	10	223.44	4.693	0.351	238.160	1.487	227.137	4.686	15.27	109.19	117.39
85	15	233.93	1.493	0.000	242.234	4.070	236.037	2.267	13.62	109.34	123.15
80	20	236.16	1.915	1.045	244.025	2.670	418.733	3.251	7.37	114.11	127.89
75	25	233.93	2.011	2.011	238.832	0.324	418.733	3.251	9.67	127.14	148.66
Canada Data (n=4,663)											
40	10	4,166.60	1.076	2.086	4,245.347	16.830	4,364.470	147.235	86.38	381.89	403.28
35	15	4,537.65	0.000	0.000	4,612.338	26.242	4,691.431	181.905	56.61	184.17	267.66
30	20	4,481.35	2.475	2.649	4,620.685	35.394	4,665.977	50.814	48.97	709.84	904.47
25	25	4,787.28	0.000	0.000	4,838.133	80.308	4,876.919	95.221	26.14	405.14	479.97
65	10	3,175.65	3.066	1.850	3,478.710	145.154	3,390.607	125.136	71.80	206.95	271.91
60	15	3,175.65	3.066	0.905	3,281.104	22.046	3,286.649	42.958	91.66	385.89	483.67
55	20	3,355.26	1.466	1.466	3,430.062	11.616	3,432.061	17.933	47.82	492.61	585.90
50	25	3,452.86	0.847	1.123	3,500.850	14.357	3,531.332	27.200	31.04	540.14	622.60
90	10	2,557.45	1.096	1.147	2,654.247	76.372	2,636.070	61.829	174.01	309.73	365.07
85	15	2,583.06	6.259	0.757	2,744.742	0.000	2,718.696	55.554	63.57	189.07	284.73
80	20	2,584.68	0.081	0.694	2,630.598	52.800	4,364.470	147.235	62.51	465.36	607.59
75	25	2,664.01	0.336	0.041	2,699.424	25.465	4,364.470	147.235	57.87	446.00	572.93
Tanzania Data (n=6,117)											
40	10	833.33	2.801	2.000	871.031	11.051	874.852	16.509	2,439.99	620.53	731.33
35	15	863.13	1.241	2.249	887.311	8.686	891.758	11.926	95.10	672.39	901.48
30	20	912.41	2.358	2.846	944.417	7.347	962.771	22.630	36.31	638.70	813.21
25	25	988.41	0.906	1.173	1,006.810	7.655	1,010.999	7.234	25.47	480.35	606.26
65	10	641.18	7.175	5.843	708.005	10.008	695.142	11.099	13,577.62	486.29	606.89
60	15	655.96	6.138	5.038	708.544	4.873	704.362	10.321	761.81	533.03	713.81
55	20	674.33	3.512	5.151	719.904	11.877	737.759	19.913	166.13	559.98	709.87
50	25	687.18	3.611	3.128	724.688	12.124	732.942	17.266	51.53	599.97	747.71
90	10	542.88	7.846	7.846	603.568	8.626	596.950	6.632	55,460.22	506.06	671.65
85	15	551.01	7.411	8.428	608.369	11.037	608.965	9.354	4,391.66	538.73	714.99
80	20	560.26	7.135	7.465	611.578	11.730	874.852	16.509	1,144.41	546.73	709.92
75	25	577.83	6.760	7.032	625.573	4.702	874.852	16.509	88.72	532.25	675.18
			2.78	0.80		17.90		36.06	2,199.01	352.36	440.95

The results show that solving p-centre problems using the exact method (EM) requires more than twice the computing time than solving (p, q) problems. One of the reasons is that when solving the SCP the (p, q) problems constraints (11) make the problem easier to solve by restricting the number of combinations (feasible set).

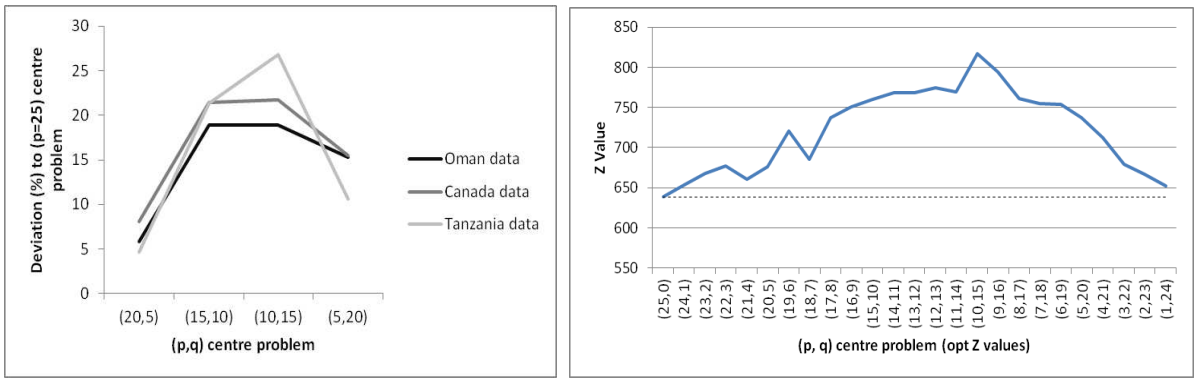
Table 4 shows that both TSAq and GMAq are able to find the optimal solutions for most instances. In general, GMAq performs better than TSAq as it produces a smaller average deviation from the optimal solutions. Table 5 shows that our methods run much faster than the exact method especially with large n and small q. Similar to the previous results, GMAq also performs better than TSAq. It is observed that it is quite hard to find the optimal solutions when p is relatively large.

Some observations

The comparison between the optimal results of the p-centre and (p, q) centre problems using the exact method are also shown in Table 4. The objective function value (Z) of a more restricted and less restricted problems appears to be smaller than the one in the middle. For instance, the Z value for (p=20, q=5) and (p=5, q=20) centre problems are smaller than that of (p=10, q=15) problem for all instances. Figure 13(a) shows the bell-shape pattern of the deviation (%) from the (p, q) problems to the (p = 25) centre problem. To get more detailed results reflecting the effect of the q value on the objective function value of the (p, q) centre problem, we solved the (p, q) problems on the Oman data optimally varying q = 0 to 24 in increments of 1 keeping p+q=25. The q existing facilities are set to the optimal solution of (p=q) centre problem. Figure 13(b) presents the pattern of the Z value on the (p, q) problems which confirms the statement that a more restricted or less restricted problems yield a smaller Z value.

Large dataset

Tables 6 presents the computational results of TSAq and GMAq on large (p, q)-centre problems. There is no known optimal solution for these problems. Due to their large sizes, as in the unconditional problems experiments, we also limit the computing time of the VNS in Stage 3 (the post-optimisation) of TSAq to 1.5 hours and in Step 3 of GMAq to 0.5 hours.



(a) (b)

Fig. 13 The pattern of the objective function of (p, q) centre problem

Table 6 Computational Results for the large (p, q)-centre problems

p	q	Best Kwown (Z_b)	Deviation (%)		Time (seconds)	
			TSAq	GMAq	TSAq	GMAq
Sweden Data (n = 24,978)						
25	25	1,101.14	0.00	0.00	680.50	954.14
50	25	819.89	0.00	1.883	874.99	908.75
25	50	874.48	0.00	0.308	495.61	847.36
75	25	706.32	0.00	2.448	768.94	1,233.95
50	50	763.40	0.00	1.279	554.22	667.55
25	75	726.10	0.00	0.000	445.62	660.98
Burma Data (n = 33,708)						
25	25	970.82	1.29	0.000	960.33	1,299.26
50	25	704.94	0.00	4.510	1,036.94	987.60
25	50	755.17	0.00	2.406	607.95	1,200.04
75	25	619.36	0.00	2.996	1,390.19	1,119.48
50	50	641.18	0.00	1.941	670.27	854.59
25	75	647.22	0.00	0.166	461.88	766.13
China Data (n = 71,009)						
25	25	3,637.15	1.74	0.000	7,313.58	7,512.75
50	25	2,752.35	0.00	0.278	7,288.20	7,515.19
25	50	2,937.21	0.00	0.891	7,279.92	7,297.55
75	25	2,310.13	5.54	0.000	7,328.08	7,511.97
50	50	2,539.07	2.72	0.000	7,283.31	7,514.36
25	75	2,504.44	0.00	0.011	7,264.37	7,363.80
Average			**0.63**	1.06	2,928.05	3,123.08

Contrarily to the previous results, TSAq generally performs better than GMAq when solving large (p, q) centre problems. The average deviation of TSAq is 0.63% which is about 40% smaller than the one of GMAq (1.06%).

6 Conclusion and suggestions for future research

Two meta-heuristics based on data aggregation, an efficient implementation of an exact method, and the use of a VNS is proposed to solve large unconditional and conditional vertex p-centre problems. The first approach called the three-stage approach (TSA) consists of three stages. The first stage is a learning process incorporating demand point aggregation and an exact method. The second stage uses a VNS to solve the disaggregated problem with the facilities identified from the previous stage as potential facility sites. A post-optimisation is performed, as the third stage, using the same VNS but on the original problem instead. The second approach is a guided multi-start approach (GMA). This is designed to provide flexibility in revisiting the aggregated problem several times so to produce a new and diverse solution configuration which is then fed into the VNS.

According to the computational results on the TSP dataset, our methods perform quite well and run relatively fast. For the small dataset ($n \leq 6{,}117$), the methods find the optimal solution on some instances for both the unconditional and conditional problems. These optimal values are obtained by our modified version based on set covering and new attributes to enhance its efficiency. These optimal solutions could be used for benchmarking purpose as well. In most cases, GMA performs better than TSA as GMA yields a smaller average deviation except for the conditional large dataset.

This research could be worthwhile expanding and adapting to other related problems such as clustering of large datasets with higher dimension as part of data mining.

Acknowledgment

The authors would like to thank both referees for their useful suggestions that improved both the content as well as the presentation of the paper. We are also grateful to the Indonesian Government and Department of Industrial Engineering – ITENAS Bandung, Indonesia for the sponsorship of the first author.

References

1. Al-Khedhairi, A., Salhi, S.: Enhancements to two exact algorithms for solving the vertex p-center problem. J. Math. Model. Algorithm. 4, 129-147 (2005)
2. Avella, P., Boccia, M., Salerno, S., Vasilyev, I.: An aggregation heuristic for large scale p-median problem. Comput. Oper. Res. 39(7), 1625-1632 (2012)
3. Barua, S., Sander, J.: Mining statistically sound co-location patterns at multiple distances. Proceedings of the 26th International Conference on Scientific and Statistical Database Management, Article No. 7 (2014)
4. Berman, O., Drezner, Z.: A new formulation for the conditional p-median and p-center problems. Oper. Res. Lett. 36, 481-483 (2008)
5. Berman, O., Simchi-Levi, D.: The conditional location problem on networks. Transp. Sci. 24, 77-78 (1990)
6. Bozkaya, B., Tansel, B.: A spanning tree approach to the absolute p-center problem. Locat. Sci. 6, 83-107 (1998)
7. Brimberg, J., Mladenovic, N.: A variable neighbourhood algorithm for solving the continuous location-allocation problem. Stud. Locat. Anal.10, 1-12 (1996)
8. Calik, H., Tansel, B.C.: Double bound method for solving the p-center location problem. Comput. Oper. Res. 40, 2991-2999 (2013)
9. Caruso, C., Colorni, A., Aloi, L.: Dominant, an algorithm for the p-center problem. Eur. J. Oper. Res. 149, 53-64 (2003)
10. Casillas, P.: Aggregation problems in location-allocation modeling. In: Gosh, A., Rushton, G. (eds.) Spatial analysis and location-allocation models, pp. 327-344. Van Nostrand Reinhold, New York (1987)
11. Chen, D., Chen, R.: New relaxation-based algorithms for the optimal solution of the continuous and discrete p-center problems. Comput. Oper. Res. 36, 1646-1655 (2009)
12. Chen, D., Chen, R.: A relaxation-based algorithm for solving the conditional p-center problem. Oper. Res. Lett. 38, 215-217 (2010)
13. Chen, R.: Conditional minisum and minimax location-allocation problems in euclidean space. Transp. Sci. 22, 158-160 (1990)
14. Cheng, T.C.E., Kang, L., Ng, C.T.: An improved algorithm for the p-center problem on interval graphs with unit lengths. Comput. Oper. Res. 34, 2215-2222 (2007)
15. Current, J.R., Schilling, D.A.: Elimination of source A and B errors in p-median location problems. Geogr. Anal. 19, 95-110 (1987)
16. Daskin, M.S.: Network and discrete location: models, algorithms, and applications. Wiley, New York (1995)
17. Daskin, M.S.: A new approach to solving the vertex p-center problem to optimality: algorithm and computational results. Commun. Oper. Res. Soc. Jpn. 45, 428–436 (2000)
18. Davidovic, T., Ramljak, D., Selmic, M., Teodorovic, D.: Bee colony optimization for the p-center problem. Comput. Oper. Res. 38, 1367-1376 (2011)

19. Davoodi, M., Mohadesa, A., Rezaeib, J.: Solving the constrained p-center problem using heuristic algorithms. Appl. Soft Comput. 11, 3321-3328 (2011)

20. Dongarra, J.J.: Performance of various computers using standard linear equation software. http://www.netlib.org/benchmark/performance.pdf (Accessed online 15 April 2013)

21. Drezner, Z.: The p-Centre problem-heuristic and optimal algorithms. J. Oper. Res. Soc. 35, 741-748 (1984)

22. Drezner, Z.: Conditional p-center problems. Transp. Sci. 23, 51-3 (1989)

23. Drezner, Z.: On the conditional p-median problem. Comput. Oper. Res. 22, 525-530 (1995)

24. Elloumi, S., Labbe, M., Pochet, Y.: A new formulation and resolution method for the p-center problem. INFORMS J. Comput. 16, 84–94 (2004)

25. Fortney, J., Rost, K., Warren, J.: Comparing alternative methods of measuring geographic access to health services. Health Serv. Outcomes Res. Methodol. 1, 173-184 (2000)

26. Francis, R.L., Lowe, T.J., Tamir, A.: Demand Point Aggregation Analysis for a Class of Constrained Location Models: A Penalty Function Approach. IIE Trans. 36, 601-609 (2004a)

27. Francis, R.L., Lowe, T.J., Tamir, A., Emir-Farinas, H.: Aggregation Decomposition and Aggregation Guidelines for a Class of Minimax and Covering Location Models. Geogr. Anal. 36, 332-349 (2004b)

28. Francis, R.L., Rayco, M.B.: Asymptotically optimal aggregation for some unweighted p-center problems with rectilinear distances. Stud. Locat. Anal. 10, 25-36 (1996)

29. Francis, R.L., Lowe, T.J., Rayco, M.B., Tamir, A.: Aggregation error for location models: survey and analysis. Ann. Oper. Res. 167, 171-208 (2009)

30. Gamal, M.D.H., Salhi, S.: Constructive heuristics for the uncapacitated continuous location-allocation problem. J. Oper. Res. Soc. 52, 821-829 (2001)

31. Hakimi, S.L.: Optimum locations of switching centers and the absolute centers and medians of a graph. Oper. Res. 12, 450-459 (1964)

32. Hansen, P., Brimberg, J., Urosevic, D., Mladenovic, N.: Solving large p-median clustering problems by primal-dual variable neighborhood search. Data Min. Knowl. Discov. 19, 351-375 (2009)

33. Hansen, P., Mladenovic, N.: Variable neighbourhood search for the p-median. Locat. Sci. 5, 207-225 (1997)

34. Hansen, P., Mladenovic, N., Perez, J.A.M.: Variable neighbourhood search: methods and applications. Ann. Oper. Res. 175, 367-407 (2010)

35. Hillsman, E.L., Rhoda, R.: Errors in measuring distances from populations to service centers. Ann. Reg. Sci. 12, 74-88 (1978)

36. Ilhan, T., Pinar, M.: An efficient exact algorithm for the vertex p-center problem. 2001. URL (http://www.ie.bilkent.edu.tr/~mustafap/pubs/).

37. Irawan, C.A., Salhi, S.: Solving large p-median problems by a multistage hybrid approach using demand points aggregation and variable neighbourhood search. J. Glob. Optim. (2013). doi :10.1007/s10898-013-0080-z.

38. Irawan, C.A., Salhi, S., Scaparra, M.P.: An adaptive multiphase approach for large unconditional and conditional p-median problems. Eur. J. Oper. Res. 237, 590–605 (2014)

39. Jaeger, M., Kariv, O.: Algorithms for finding p-centers on a weighted tree (for relatively small p). Netw. 15, 381–389 (1985)

40. Kariv, O., Hakimi, S.L.: An algorithmic approach for the vertex p-center problem. Part I: The p-centers. SIAM J. Appl. Math. 37, 513-538 (1979)

41. Kaveh, A., Nasr, H.: Solving the conditional and unconditional p-center problem with modified harmony search: A real case study. Sci. Iran. A. 18, 867-877 (2011)

42. Liu, S., Liu, Y., Ni, L. M., Fan, J., Li, M.: Towards Mobility-based Clustering. Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining, 919-928 (2010)

43. Lu, C.: Robust weighted vertex p-center model considering uncertain data: An application to emergency management. Eur. J. Oper. Res. 230,113-121 (2013)

44. Lu, C., Sheu, J.: Robust vertex p-center model for locating urgent relief distribution centers. Comput. Oper. Res. 40, 2128-2137 (2013)

45. Minieka, E.: The m-center problem. SIAM Rev. 12, 138-139 (1970)

46. Minieka, E.: Conditional Centers and Medians on a Graph. Netw. 10, 265-272 (1980)

47. Mladenovic, N., Labbe, M., Hansen, P.: Solving the p-center problem with tabu search and variable neighbourhood search. Netw. 42, 48-64 (2003)

48. Qu, Q., Liu, S., Yang, B. Jensen, C.S.: Integrating non-spatial preferences into spatial location queries. Proceedings of the 26th International Conference on Scientific and Statistical Database Management, Article No. 8 (2014)

49. Rayco, M.B., Francis, R.L., Lowe, T.J.: Error-bound driven demand point aggregation for the rectilinear distance p-center model. Locat. Sci. 4, 213-235 (1997)

50. Rayco, M.B., Francis, R.L., Tamir, A.: A p-center grid-positioning aggregation procedure. Comput. Oper. Res. 26, 1113-1124 (1999)

51. Salhi, S., Al-Khedhairi, A.: Integrating heuristic information into exact methods: The case of the vertex p-centre problem. J. Oper. Res. Soc. 61, 1619-1631 (2010)

52. Salhi, S., Sari, M.: A multi-level composite heuristic for the multi-depot vehicle fleet mix problem. Eur. J. Oper. Res. 103, 95-112. (1997)

53. Shaw, D.X.: A unified limited column generation approach for facility location problems on trees. Ann. Oper. Res. 87, 363-382 (1999)

54. Tansel, B.C., Francis, R.L., Lowe, T.J.: State of the art – location on network: a survey. Part I: the p-center and p-median problems. Manag. Sci. 29, 482-497 (1983a)

55. Tansel, B.C., Francis, R.L., Lowe, T.J.: State of the art – location on network: a survey. Part II: exploiting tree network structure. Manag. Sci. 29, 498-511 (1983b)

56. Tansel, B.C., Francis, R.L., Lowe, T.J., Chen, M.: Duality and distance constraints for the nonlinear p-center and covering problem on a tree network. Oper. Res. 30, 725-744 (1982)