Applying Particle Swarm Optimization for the Absolute p-center Problem

Hassan M. Rabie*
PhD Researcher, Decision Support, Faculty of Computers and Information, Cairo University

Ihab El-Khodary
Decision Support, Faculty of Computers and Information, Cairo University

Assem A. Tharwat
Department of Engineering & Business, Canadian International College

Email: *Hassan.Rabie {at} gmail.com

Abstract— **Locating facilities on anywhere on a network is known as the absolute *p*-center location problem and it is proven to be NP-hard problem. Most of the recent approaches solve large-scale vertex *p*-center location problem in which facilities can be located only on the nodes of the network. However, rarely algorithms are developed to solve large-scale absolute *p*-center problem. Particle swarm optimization (PSO) is a metaheuristic approach, which recently proved to be a successful approach in solving complex continuous optimization problems. In this paper we present a PSO algorithm for the absolute *p*-center problem to minimize the maximum distance from each customer to his/her nearest facility. We have tested our proposed algorithm on a set of 12 problems from "Beasley OR Library" and compared the results of vertex location problem to those of the absolute location problem. The numerical experiments show that PSO algorithm can solve optimally large-scale location problems with networks up to 16,200 edges.**

Keywords- particle swarm optimization;location problem;absolute p-center.

I. Introduction

The focus of this paper is on the absolute *p*-center location problem [1], in which locating one or more facilities on any location of a network to service a set of demand points at known locations. It is also known as the minimax location problem where every demand receives its service from a closest facility, and the maximum distance between a demand and a closest facility is as small as possible [2]. The location problems deal with the optimal location of emergency facilities where the concern for saving human life is far more important than any transportation costs that may be incurred in providing that service. This kind of location problem is suggested by Hakimi[1, 3], and some of its applications are used to locate fire stations, hospital emergency services, data file location, police stations, and so on. This type of problems is non-convex, nonlinear optimization problems and, such problems are difficult to solve [4], and many authors have proved that the problem is NP-hard [5, 6].

Metaheuristics have become very powerful tools for solving hard combinatorial optimization problems. Particle swarm optimization (PSO) is a class of them which has been inspired from the flocking of birds and fish, and simulated evolution. PSO is an evolutionary computation method developed by James Kennedy and Russell Eberhart[7] to solve continuous non-linear functions. Since its conception in 1995, PSO has been noted for two main features: optimization via social evolution and its relative ease of use, although it is comparatively a new metaheuristic algorithm, in various applications; it has been proven to be a robust and efficient tool [8, 9].

To approach the *p*-center location problem, we propose a PSO algorithm and to test the efficiency of our implementation, we performed a set of numerical experiments on well-known benchmark problems. Reported results show that PSO can solve optimally large-scale *p*-center location problems.

The rest of this paper is organized as follows. Section 2 is devoted to the description of *p*-center location problem, while the description of PSO is given in Section 3. The proposed PSO algorithm for *p*-center problem and the implementation of PSO for solving location problem is explained in Section 4. Section 5 contains experimental results and Section 6 concludes the paper.

II. The *p*-Center Location Problem

Center location problem answers the question of where to locate a facility or a service and it is considered a major part of Operations Research and Management Science. There are often concerns about the location of a single facility or service to be established; such as ambulance, fire station, etc... In particular; if a graph represents a road network with its vertices representing communities, one may have the problem of locating optimally the location of an emergency facility. In a more general form, a large number of such facilities may be required to be located. In this case the furthest vertex of the graph must be reachable from at least one of the facilities within a minimum distance. The resulting facility locations are then called the *p*-center of a network [10]. The problems are natural extension of those of single facility location, however, there are two important conditions [11]: (1) at least two facilities are to be located, and (2) each new facility is linked to at least one center.

There are two main types of the *p*-center location problem on networks; they differ by the possible location of the service points [12]:

1. Problems in which the facilities can be located anywhere on the network (i.e., on the nodes and on the edges/links of the network) and are known as *absolute center problems*.
2. Problems in which facilities can be located only on the nodes of the network and are known *vertex center problems*.

The *p*-center problem is one of the fundamental problems in the location science. Due to its hardness and importance, it has always been a challenge for researchers who approached it from different perspectives. Recently many authors [2, 13-15] approached large-scale vertex location problems with number of nodes ranging from 100 to 900, but another interesting problem is the absolute *p*-center problem in which the facilities can be located anywhere on the network (i.e., on the nodes and on the edges/links of the network). In this paper we consider only solving the absolute *p*-center location problem using PSO algorithm.

A. Problem Formulation

The absolute *p*-center location problem could be defined in the following way: given a set of *V* nodes (customers) and a set of *E* edges which connect between nodes and the distances between any pair of nodes is given. This structure is usually referred to as network, see Fig. 1. The goal is to locate *p* facilities (centers) anywhere on a network in such a way to minimize the maximum of the distances from each node to its nearest center, where *p* represents the total number of centers that should be located on the network. Centers could be located at any point on the network.

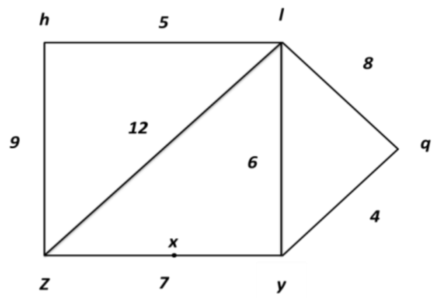

Figure 1. Network Example

In order to describe a formulation of the absolute *p*-center problem, we introduce the following notation: let $G(V, E)$ be a finite undirected connected graph with no loops, with a set V of vertices/nodes and set of edges/links E. A positive length $b(e)$, is associated with every edge $e \in E$ and a positive weight, $w(v)$, may be associated with each vertex $v \in V$.

For any two points $y,z \in G$, may be a vertex/node or on an edge/link, $d(y,z)$ denotes the length of the shortest path connecting these two points. Therefore, for $x \in (y,z)$ and given $l \in V$, $d(x, l) = Min[d(y, l)+b(x, y), d(z, l)+b(y, z) - b(x, z)]$. Given a set of p facility points in G (assuming $p \leq V$), say $X_p = (x_1, ...,x_p)$, then each vertex/node is assigned to, and its distance from the set of facilities is determined by the nearest facility to it. Therefore, for $l \in V$, its distance from X_p is $d(l, X_p)=min[d(l,x_i)| \; x_i \in X_p, \; i=1,...,p]$. In the *p*-center problem the value of such a choice of locations for the *p* facilities, is defined to be the weighted distance of the furthest vertex from the set of facilities, namely at $Z(X_p)=max[w(l) * d(l, X_p)| \; l \in V]$. The *p*-center problem is to minimize $[Z(X_p)|X_p \subset G]$. Given the location of the *p* facilities, X_p, the set V is partitioned into *p* exhaustive and disjoint subsets, each served by its own facility [16].

B. LiteratureReview

For a review on *p*-center problem network location models see Farahani et al. [11] and Current et al. [17]. Hakimi[1] defined and solved the absolute 1-center problem by examining the piecewise linear objective function on each edge and finding the edge-restricted minimum at one of the breakpoints. The smallest among the edge-restricted minima is the absolute 1-center of the network. Hakimi et al. [18] reduced the computational effort of his algorithm in [1]. Minieka[19] developed a method for solving *p*-center for *p*>1 location problem by solving a finite series of minimum set covering problems. Christofides and Viola [20] also employed the idea of using the set-covering problem in their algorithm. Christofides[10] showed that one needs to consider only a subset of the links for an optimal location. However, this approach was unable to solve general *p*-center problems. Minieka[19], and Kariv and Hakimi[21] showed that the optimal solution of the problem is restricted to a finite set of points on the network. Daskin[12] presents an optimal algorithm which solves the absolute *p*-center problem by performing a binary search over possible solution values. This algorithm solves maximal covering sub-problems, rather than the set-covering sub-problems solved by Minieka. However, The common characteristic of most approaches for solving *p*-center problem is that they all rely on Minieka[19] algorithm, which has also been considered to be inefficient since it solves successive instances of a minimal set covering problem, which is NP-complete itself on general graphs [22].

Recently most of the approaches describe an efficient methods for large-scale vertex center problems, such as: Iihan et al. [23] developed an algorithm that iteratively sets a maximum distance value within which it tries to assign all demand points. The algorithm is a two-stage variant of an algorithm by Minieka[19]. Caruso et al. [13] presented Dominant algorithm with four different versions, two of them optimally solve the vertex location problem. Chen et al. [15] presented a relaxation algorithm for the vertex location problem. Davidovic et al. [14] designed a bee colony optimization algorithm. Kaven and Nasr [2] solved *p*-center problem using a modified harmony search algorithm. This music inspired algorithm is a simple metaheuristic that was proposed recently for solving combinatorial and large-scale engineering and optimization problems. Good computational results are reported for each of an extensive list of test problems derived from "Beasley OR Library" with network size up to 900 nodes.

However, rarely algorithms were developed to solve large-scale absolute *p*-center problem. In this paper, we focus only on the absolute *p*-center problem, with the objective to locate p new facilities, called centers, on network *G* in order to minimize the maximum distance between a node and its nearest facility in which centers could be located on any vertex or edge of network.

III. Particle Swarm Optimization

Particle swarm optimization (PSO) is a metaheuristic intelligence approach, which recently proved to be a successful approach to solve complex optimization problems and it is considered a powerful optimization technique for solving continuous optimization problems and is known to be efficient and robust for solution of combinatorial optimization problems [24]. PSO is a population-based, co-operative search metaheuristic introduced in 1995 by Kennedy and Eberhart[24] for finding optimal solutions for complex search spaces through the interaction of individuals in a population of particles.

Following [24], in PSO, particles are the candidate solutions of PSO population cooperate simultaneously based on knowledge sharing with neighboring particles. Each particle through flying in the search space generates a solution using directed velocity vector and each particle modifies its velocity to find a better solution (position) by applying its own flying experience (i.e. memory having best position found in the earlier flights) and experience of neighboring particles (i.e. best-found solution of the population). Particles update their positions and velocities as shown below [25]:

- A population of particles is randomly initialized with positions X_i and velocities Vel_i and a function *f* is evaluated, using the particle's positional coordinates as input values. Positions and velocities are adjusted and the function evaluated with the new coordinates at each time step.
- When a particle discovers a pattern that is better than any it has found previously, it stores the coordinates in a vector $Pbest_i$.
- The difference between $Pbest_i$ (the best point found by *i* so far) and the individual's current position is added to the current velocity. Also, the difference between the neighborhood's best position $Gbest_i$ and the individual's current position is also added to its velocity, adjusting it for the next time step. These adjustments to the particle's movement through the space cause it to search around the two best positions.

Following [26], variables X_i and Vel_i are regarded as vectors that show various positions and velocities of particle and in order to find the optimum position of the best position of particle *i* and its neighbors' best position are recorded as: $Pbest_i$and $Gbest_i$, respectively. To improve the velocity and position of each particle, the modified velocity and position in the next iteration is calculated as follows:

$$Vel_i^{k+1} = w_k Vel_i^k + c_1 r_1 \left(Pbest_i^k - X_i^k\right) + c_2 r_2 \left(Gbest^k - X_i^k\right) \quad (1)$$

$$X_i^{k+1} = X_i^k + Vel_i^{k+1} \quad (2)$$

where,

- Vel_i^k, velocity of particle *i* at iteration *k*.
- w_k, inertia weight factor which is reduced dynamically to decrease the search area in a gradual fashion. The variable wk is updated as [24]*:*

  $$w_k = (w_{max} - w_{min}) * \frac{(k_{max} - k)}{k_{max}} + w_{\min},$$

  where, w_{max} and w_{min} denote the maximum and minimum of $w_k$respectively; k_{max} is a given number of maximum iterations.
- c_1 *and* c_2, acceleration coefficients of the self-recognition component and coefficient of the social component, respectively. The choice of value is c_1=c_2= 2; and generally referred to as learning factors [8].
- r_1 *and* r_2, random numbers between 0 and 1.
- X_i^k ,position of particle *i* at iteration *k*.
- $Pbest_i^k$,best position of particle *i* at until iteration *k*.
- $Gbest_i^k$,best position of the group at until iteration *k*.

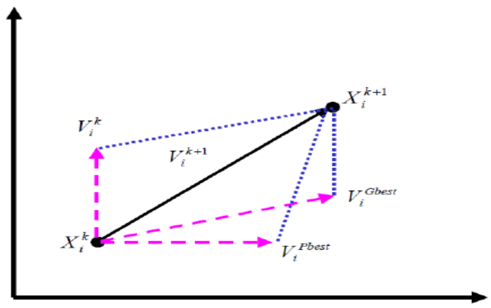

Figure 2: Updating the position of PSO

IV. PSO for P-Center Location Problem

The *p*-center location problems as mentioned before are non-convex, nonlinear optimization problems and such problems are difficult to solve [4], also many authors have proved that the problem is NP-hard [5, 6]. Due to its complexity, hardness and importance it has always been a challenge for researchers who approached it from different perspectives. In this paper we develop a PSO algorithm for the absolute *p*-center location problem which have been developed by Hakimi[1].

In [27], the authors reviewed the algorithms that have been applied to continuous location problems where demand points are in space and not a network, then they applied two algorithms models; PSO algorithm turned out to be the more efficient one. In [28], the authors proposed a PSO approach to find the optimal location of single power system stabilizer. Also, in [26], PSO was employed to determine the optimal location for installing a new electricity generating unit. Farahani et al. [29] reviewed recent efforts in location theory using metaheuristic techniques and they mentioned that PSO is a promising approach for solving location problems. However, rarely algorithms have been developed to tackle the large-scale absolute *p*-center location.

A. The Proposed PSO Algorithm

The basic idea of designing PSO is to build a multi-agent system (swarm of artificial birds) that will search for good solutions of various combinatorial optimization problems [14]. PSO is a population-based search algorithm and searches in parallel using a group of particles based on the analogy of swarm of bird [26]. Artificial particles investigate through the search space looking for the feasible solutions. In order to find better and better solutions, artificial particles collaborate and exchange information. Using collective knowledge and sharing information among themselves, artificial particles concentrate on more promising areas, and slowly abandon solutions from the less promising ones. Step by step, artificial particles collectively generate and/or improve their solutions. In this section, we describe our implementation of the PSO to be applied to the absolute *p*-center location problem.

The next step is to solve the *p*-center location problem by using PSO. PSO algorithm has the task to search different combination of *p* edges through generating particles on each edge and compare the worst case minimum value for each particle. The PSO search is running in iterations until some predefined stopping criteria is satisfied (Number of iterations). The procedure is then repeated with the remaining combinations in order to find the combination with the best minimum value. The corresponding combination is the optimal location. In order to better clarify, the PSO algorithm applied to the p-center location problem; the proposed approach can be described as follows:

Step 1. Let *p* represents the number of centers, and *n* represents the population size (swarm size, number of particles).

Step 2. Pick one of the candidate solutions ($CS \subset E$) – combination of edges with *p* size – such as $CS=[E_{1st}, E_{2nd}, \ldots, E_{pth}]$; where the total number of candidate solutions is $\binom{E}{p}$.

Step 3. Generate randomly the initial particles positions *X* in the range of upper and lower limits for *p* edges, and set velocities*Vel* to zero.

Step 4. The objective function and fitness value of each particle with its *Pbest* is calculated. The best among the *Pbest* is denoted as *Gbest*.

Step 5. The velocity and position of each particle is modified according to equation (1) and (2), respectively.

Step 6. The objective function of each particle is compared with its *Pbest*. If the current value is better than *Pbest* then *Pbest* value is set equal to the current value and *Pbest* position is set equal to the current position.

Step 7. If the current fitness value is better than the *Gbest*, then update *Gbest* to current best position and fitness value.

Step 8. Step 5 to 7 is repeated until the maximum number of iterations is met.

V. EXPERIMENTAL RESULTS

In order to examine the performance of our PSO algorithm, we select to apply our algorithm on “Beasley OR Library”, which has been proposed by Beasley in 1985 for the *p*-median problems, one of the problems from the location analysis well known by its hardness. The library consists of 40 problems and available on [30]. The range of the problems is from 100 to 900 nodes, and from 200 to 16,200 edges/links and the number of centers to be determined is from 5 to 200. This well-known library has been widely used, recently in [2, 13-15]; however these researches deal with the vertex location problems, therefore we used the results from them as an upper bound for our algorithm [15], i.e. the PSO optimal solution must be less than or at least equal to the optimal solutions from [2, 13-15].

To test our algorithm, we select a sample from OR Library (12 out of 40 problems about 30%), since the absolute location problem is more complex and hard to solve than the vertex one, for example, in order to solve the first problem in OR Library, which consists from 100 nodes, 200 edges and the required centers are 5, we have to investigate:

1. $\binom{100}{5}$candidate solutions on nodes-more than 75.287 million – for the vertex case.
2. $\binom{200}{5}$candidate solutions on edges -more than 2.535 billion – for the absolute case.

It is too difficult to investigate all the candidate solutions in OR Library. Therefore, first, we solved the problem with *p*=1, and get the edge of optimal solution, and then we solved the problem with *p*=2 using the optimal edge from *p*=1 as the first element in the candidate solutions for *p*=2 while the second elements will be all edges. For example, if the optimal edge in *p*=1 is on E_5, therefore, for *p*=2, the candidate solutions will be:

$$CS_{(p=2)} = \begin{bmatrix} E_5\ E_1 \\ E_5 E_2 \\ \vdots \\ E_5\ \ E_h \end{bmatrix}$$

Getting the solution for *p*=2, which must be less than or at least equal to *p*=1. To make sure this is the optimal solution for *p*=2, we have to replace the first element –such as E_5– by the second element *p*=2 solution of and reform *CS*; for example if the optimal solution for *p*=2, is on edge E_5 and E_{12}, therefore, we have to find the optimal solution for the following candidate solutions:

$$CS_{(p=2)} = \begin{bmatrix} E_{12}\, E_1 \\ E_{12}\, E_2 \\ \vdots \\ E_{12}\;\; E_h \end{bmatrix}$$

If the second solution is larger than the first for *p*=2, then the first solution is optimal for *p*=2; else; use the second solution and check its optimality as before and do till no improvement in the optimal solution for *p*=2, then stop and solve *p*=3 with the same manner for *p*=2 and so on till obtaining the optimal solution for all *p*.

Table 1 shows the results of the Particle Swarm Optimization (PSO) for solving the absolute *p*-center problem on a sample from "Beasley OR Library" with edges ranging from 200 to 16,200 and centers from 5 to 33. The table shows that by applying the PSO on the more difficult absolute *p*-center problems, we managed to achieve a minimum distance value which is in most cases lower than the obtained for the vertex and in worst cases achieved the same results.

TABLE 1: RESULTS OF SOLVING ABSOLUTE P-CENTER LOCATION PROBLEM USING PSO

File Name	Nodes (*V*)	Edges (*E*)	Centers (*P*)	Best known Of Vertex	Best PSO Absolute
Pmed1	100	200	5	127.00	125.32
Pmed2	100	200	10	98.00	96.23
Pmed3	100	200	10	93.00	93.00
Pmed4	100	200	20	74.00	73.40
Pmed5	100	200	33	48.00	45.96
Pmed6	200	800	5	84.00	82.52
Pmed7	200	800	10	64.00	63.52
Pmed11	300	1,800	5	59.00	57.65
Pmed16	400	3,200	5	47.00	46.63
Pmed26	600	7,200	5	38.00	37.00
Pmed35	800	12,800	5	30.00	30.00
Pmed38	900	16,200	5	29.00	28.50

VI. CONCLUSION

PSO algorithm for the absolute *p*-center location problem has been described. We used our algorithm to solve a sample from the well-known "Beasley OR Library" benchmark problems. The results show that the algorithm is capable of solving large-scale absolute location problems.

ACKNOWLEDGMENT

This work was supported by Al Ezz Ceramics and Porcelain Company (GEMMA).

REFERENCES

[1] S. L. Hakimi, "Optimum Locations of Switching Centers and the Absolute Centers and Medians of a Graph," *Operations Research,* vol. 12, pp. 450-459, 1964.

[2] A. Kaveh and H. Nasr, "Solving the conditional and unconditional p-center problem with modified harmony search: A real case study," *Scientia Iranica,* vol. 18, pp. 867–877, 2011.

[3] S. L. Hakimi, "Optimum Distribution of Switching Centers in a Communication Network and Some Related Graph Theoretic Problems," *Operations Research,* vol. 13, pp. 462-475, 1965.

[4] M. l. Brandeau and S. s. Chiu, "An Overview of Representative Problems in Location Research," *Management Science,* vol. 35, 1989.

[5] G. Cornuejols, M. L. Fisher, and G. L. Nemhauser, "Location of Bank Accounts to Optimize Float: An Analytic Study of Exact and Approximate Algorithms," *Management Science,* vol. 23, pp. 789-810, 1977.

[6] O. Kariv and S. L. Hakimi, "An Algorithmic Approach to Network Location Problems. I: The p-Centers," *SIAM Journal on Applied Mathematics,* vol. 37, pp. 513-538, 1979.

[7] J. Kennedy and R. Eberhart, "Particle Swarm Optimization," *Proceedings of IEEE International Conference on Neutral Networks,* pp. 1942-1948, 1995.

[8] S. Sumathi and S. Paneerselvam, *Computational Intelligence Paradigms: Theory & Applications using MATLAB*: Taylor and Francis Group, LLC, 2010.

[9] H. Yapicioglu, A. E. Smith, and G. Dozier, "Solving the semi-desirable facility location problem using bi-objective particle swarm," *European Journal of Operational Research,* vol. 177, pp. 733–749, 2007.

[10] N. Christofides, *Graph Theory, an Algorithmic Approach*: Academic Press, New York, 1975.

[11] R. Z. Farahani and M. Hekmatfar, *Facility Location Concepts, Models, Algorithms and Case Studies*: Springer-Verlag Berlin Heidelberg, 2009.

[12] M. S. Daskin, *Network and Discrete Location: Models, Algorithms and Applications*: John Wiley and Sons, Inc., New York., 1995.

[13] C. Caruso, A. Colorni, and L. Aloi, "Dominant, an algorithm for the p-center problem," *European Journal of Operational Research,* vol. 149 pp. 53–64, 2003.

[14] T. Davidovica, D. Ramljak, M. Selmic, and D. Teodorovic, "Bee colony optimization for the p-center problem," *Computers & Operations Research,* vol. 38, pp. 1367–1376, 2011.

[15] D. Chen and R. Chen, "New relaxation-based algorithms for the optimal solution of the continuous and discrete p-center problems," *Computers & Operations Research,* vol. 36, pp. 1646 -- 1655, 2009.

[16] J. Halpern and O. Maimon, "Algorithms for the m-center problems: A survey," *European Journal of Operational Research,* vol. 10, pp. 90-99, 1982.

[17] J. Current, M. Daskin, and D. Schilling, *Facility location: applications and theory. In: Facility Location: Applications and Theory. Edited by Z. Drezner and H.W. Hamacher*: Springer-Verlag, 2001.

[18] S. L. Hakimi, E. F. SCHMEICHEL, and J. G. PIERCE, "On P-centers in networks," *Transportation science,* vol. 12, pp. 1-15, 1978.

[19] E. Minieka, "The m-Center Problem," *SIAM Review,* vol. 12, pp. 138-139, 1970.

[20] N. Christofides and P. Viola, "The Optimum Location of Multi-Centres on a Graph," *Operational Research Quarterly,* vol. 22, pp. 145-154, 1971.

[21] O. Kariv and S. L. Hakimi, "An algorithmic approach to network location problems. Part 1: p-centers.," *SIAM Journal of Applied Mathematics,* vol. 37, pp. 513-538, 1979.

[22] B. C. Tansel, R. L. Francis, and T. J. Lowe, "Location on Networks: a Survey Part I: The p-Center and p-Median Problems," *Management Science,* vol. 29, pp. 482-497, 1983.

[23] T. Ilhan, F. A. Ozsoy, and M. C. Pinar, "An Effiient Exat Algorithm for the Vertex p-Center Problem and Computational Experiments for Different Set Covering Subproblems," 2002.

[24] P. S. Shelokar, P. Siarry, V. K. Jayaraman, and B. D. Kulkarni, "Particle swarm and ant colony algorithms hybridized for improved continuous optimization," *Applied Mathematics and Computation* vol. 188, pp. 129–142, 2007.

[25] M. Clerc and J. Kennedy, "The Particle Swarm—Explosion, Stability, and Convergence in a Multidimensional Complex Space," *IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION,* vol. 6, pp. 58-73, 2002.

[26] M. Eslami, H. Shareef, A. Mohamed, and M. Khajehzadeh, "Optimal Location of PSS Using Improved PSO with Chaotic Sequence," presented at the International Conference on Electrical, Control and Computer Engineering, 2011.

[27] L. Li, Q. YanYou, and L. Wei, "Overview of Optimization Algorithms in facility Allocation Problems," presented at the Sixth International Conference on Natural Computation (ICNC 2010), 2010.

[28] M. N. Suharto, M. Y. Hassan, M. P. Abdullah, M. S. Majid, and F. Hussin, "Optimal Location of a New Generating Unit Using Particle Swarm Optimization," presented at the Power Engineering and Optimization Conference (PEOCO), 2011 5th International 2011.

[29] R. Z. Farahani, M. SteadieSeifi, and N. Asgari, "Multiple Criteria Facility Location Problems: A Survey," *Applied Mathematical Modelling,* vol. 34, pp. 1689–1709, 2010.

[30] "http://people.brunel.ac.uk/~mastjjb/jeb/orlib/files/."