A particle swarm optimization algorithm for the continuous absolute *p*-center location problem with Euclidean distance

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

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

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

***Abstract*—The *p*-center location problem is concerned with determining the location of *p* centers in a plane/space to serve *n* demand points having fixed locations. The continuous absolute *p*-center location problem attempts to locate facilities anywhere in a space/plane with Euclidean distance. The continuous Euclidean *p*-center location problem seeks to locate *p* facilities so that the maximum Euclidean distance to a set of *n* demand points is minimized. A particle swarm optimization (PSO) algorithm previously advised for the solution of the absolute *p*-center problem on a network has been extended to solve the absolute *p*-center problem on space/plan with Euclidean distance. In this paper we develop a PSO algorithm for the continuous absolute *p*-center location problem to minimize the maximum Euclidean distance from each customer to his/her nearest facility, called “PSO-ED”. This problem is proven to be NP-hard. We tested the proposed algorithm “PSO-ED” on a set of 2D and 3D problems and compared the results with a branch and bound algorithm. The numerical experiments show that PSO-ED algorithm can solve optimally location problems with Euclidean distance including up to 1,904,711 points.**

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

I. INTRODUCTION

The *p*-center location problem (also called minimax facility location problem) is a major class of location problems. Continuous location problem with Euclidean distance is a main variant of the *p*-center location which is concerned with determining the location of *p* centers in a plane/space to serve *n* demand points having fixed locations. The Euclidean *p*-center location problem seeks to locate a facility so that the maximum Euclidean distance to a set of *n* demand points is minimized [1]. The problem is equivalent to finding the center of the smallest circle enclosing all points [1].

Our main objective is to locate new *p* centers/facilities in the space/plane in such a way that the maximum distance between demand points and their nearest facility becomes minimum. It is assumed that all the facilities are identical and provide the same service to the customers, and there is no limit for the number of customers who can get service from the centers [2]. This kind of location problem is suggested by Hakimi [3, 4], and some of its applications are used to locate fire stations, hospital emergency services, data file location, police stations, and so on. Megiddo and Supowit [5] have shown that the continuous Euclidean *p*-center location problem in the plane is NP-hard, and, such problems are difficult to solve.

James Kennedy and Russell Eberhart [6] in 1995, developed a new metaheuristic algorithm, so-called Particle Swarm Optimization (PSO) algorithm, which is inspired from the flocking of birds, and simulated evolution. Although PSO is comparatively a new metaheuristic algorithm, in various applications; it has been proven to be a robust and efficient tool [7, 8]. PSO has been used mostly to solve continuous optimization problems. The purpose of this paper is to describe a simple put efficient PSO algorithm to solve large-scale Euclidean distance absolute location problem and to test the efficiency of our algorithm. Reported results show that PSO can solve optimally large-scale Euclidean distance *p*-center location problems.

The rest of this paper is organized as follows. Section 2 is devoted to the description of continuous Euclidean *p*-center location problems, while the description of PSO algorithm is given in Section 3. The proposed PSO-ED algorithm for *p*-center problem and the implementation of PSO-ED for solving location problem are explained in Section 4. Section 5 contains experimental results. Section 6 concludes the paper and section 7 contains future work.

II. THE CONTINUOUS EUCLIDEAN *P*-CENTER LOCATION PROBLEM

The minimax location problem seeks to locate a facility so that the maximum distance to a set of demand points is minimized. Therefore, the *p*-center problem involves locating *p* identical facilities to minimize the maximum distance between demand nodes and their closest facilities, i.e. to minimize the worst case possible time spent on the way in providing service. Using Euclidean distances in the plane/space, this problem is equivalent to finding the center of the smallest circle enclosing all points, hence the term “center” regarding this problem [1]. According to [9], usually, the utilization of minimax criterion arises when location of emergency facilities is considered. The facilities will be located in such a way that the response time to the farthest customer will be minimal. Most of the applications arise in emergency service locations such as determining optimal locations of ambulances, fire stations and police stations where the human life is at stake.

In many cases, the distances between demand and service points are Euclidean [10]. The Euclidean distance location problem seeks to locate p new facilities at some points (x_j, y_j), j = 1, .., p in R^2, within an existing n demand points (a_i, b_i), i = $1,..., n$. According to [11], the location of p-center facilities in two-dimensional R^2 Euclidean space can be formulated as:

$$\min_{x_j,y_j}\max_{i}\min_{j}\left[(a_i-x_j)^2+(b_i-y_j)^2\right]^{1/2}, j=1,\dots,p. \qquad (1)$$

where (a_i, b_i), $i=1,\dots,n$, are the coordinates of the demand points; (x_j, y_j), $j=1,\dots,p$ are the coordinates of the service facilities (which are to be determined); and $\min_j$ selects for each demand point its closest facility and the $\min_{xj,yj}$ $\max_i$ operations are to be performed.

The continuous absolute p-center location problem with Euclidean distance is an NP-hard problems, even the simplest p-center problems or the approximation to the problem was found to be NP-Hard [12]. The optimal solution can be found in time $O(p^n)$, which is impractical, even for small p and small n [12].

Mainly, there exist two types of the continuous Euclidean p-center location problem, distinguished through the possible location of the service points. The first type, includes facilities which can be located anywhere in the space including the demand points; known as the absolute center location problems. Whereas the second type is the vertex p-center location problems in which facilities can be located only on the demand points. Usually the solution of vertex location problem can be used as an upper bound for the solution for the absolute location problem [13]. In this paper we consider only solving the continuous Euclidean absolute p-center location problem in which each center/facility can be located anywhere in the plane/space including the demand points.

For example, as presented in [9], suppose we need to find 3 centers for the 10 demand points represented as [♦] in blue in Figure 1. Therefore, the continuous Euclidean p-center location problem searches for the optimal location of 3 (p) points (centers) within the problem space in such a way that the maximum distance from these 3 centers to n demand points is minimum than any other 3 points in the space. As Figure 1 shows the location of these 3 centers [○] in red are p_1, p_2 and p_3 with maximum distance equal to 24.0208. Accordingly, the solution vector consists of the coordinates of the 3 center points.

According to [12], many approximation algorithms have been suggested for solving p-center problems. However, recently, we developed a new PSO algorithm for solving the absolute p-center problem on networks, that algorithm has the task to randomly generate swarms on the arcs of network, then, the algorithm has the task to search optimal solution from different combination of swarms [13]. This paper extended that algorithm to solve the absolute p-center location problem on space/plan with Euclidean distance.

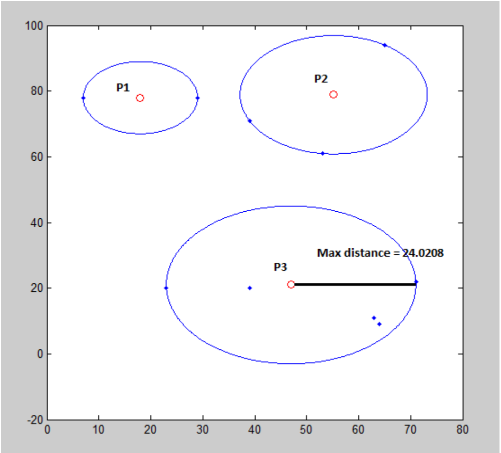


Fig. 1. A set of 10 demand points (in Blue) and service points (in Red) and the maximum distance; from Ref [9].

A. Literature Review

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 the researchers who approached it from different perspectives. When searching for a single center point (p=1); the solution of the problem will be the center of the smallest circle enclosing n given points in the plane [10]. The single center location problem with Euclidean distance was first suggested by James Sylvester in 1814 [1]. Chrystal suggested an algorithm that starts with a large circle that encloses all the points and reduces the radius of the circle iteratively until the smallest circle is obtained [1].

According to [11], the continuous Euclidean p-center location problem was first mentioned in 1958 by Miehle [14] and formulated by Cooper [15] in 1963. Chen suggested a differentiable approximation method to solve the problem [11]. Handler and Mirchandani used relaxation approaches to solve this problem [16]. Daskin [17] presented an optimal algorithm which solves the absolute p-center problem by performing a binary search over possible solution values [17]; the algorithm solved maximal covering sub-problems rather than the set-covering sub-problems solved by Minieka [18].

Recently, Ilhan et al. [19] developed an exact method for solving the vertex location p-center problem in which centers must be chosen only from demand points. The algorithm solved problems with sizes up to 657 points in space. Chen and Chen [10] presented a new relaxation based algorithms for the solution of vertex and absolute continuous p-center problems. The algorithm solved problems with sizes up to 1,817 points in space. Kaveh and Nasr [2] suggested a metaheuristic algorithm called harmony search algorithm. The latter algorithm solved the vertex location problems with sizes up to 4,461 points in space. Calik and Tansel [20] proposed a new integer programming formulation for the p-center problem, in which the optimal p-center solution is obtained by solving a series of simple structured integer programs. The algorithm successfully solved problems with sizes up to 3,038 points in space.

Fayed and Atiya [12] suggested a mixed breadth-depth first strategy to speed up the traversing of the branch and bound tree in order to solve the continuous Euclidean absolute *p*-center location problem. The algorithm was capable of optimally solving problems with size up to 1,904,711 points in space.

According to [10], most of the methods developed for solving the continuous Euclidean problem are geometrical in nature, which involves complex, time consuming search methods for finding the smallest enclosing circle. This includes the repeated solution of relaxed, smaller sub-problems. However, few researches solved the large-scale continuous Euclidean absolute p-center location problems such as [12]. As mentioned before the continuous Euclidean absolute *p*-center problem has been proved to be NP-hard [5] and to approach the *p*-center location problem, we propose a simple algorithm based on PSO (PSO-ED) for solving this problem.

III. Particle Swarm Optimization (PSO)

PSO is a population-based, co-operative search metaheuristic approach introduced in 1995 by Kennedy and Eberhart [6]. PSO inspired from the sociological behavior associated with bird flocking. It is a natural observation that birds can fly in large groups with no collision for extended long distances, making use of their effort to maintain an optimum distance between themselves and their neighbors [21]. PSO was originally used to solve non-linear continuous optimization problems, but more recently it has been used in many practical, real-life application problems [21]. PSO proved to be a successful approach to solve complex continuous problems and is proved to be efficient and robust for solution of combinatorial optimization problems [22].

PSO finds solution for problems that can be represented as a set of points in an *n*-dimensional solution space. PSO is a population-based search algorithm that finds optimal solutions using a set of flying particles with velocities that are dynamically adjusted according to their historical performance, as well as their neighbors in the search space. The population consist form particles which are described as the swarm positions in the *k*-dimensional solution space. Each particle is set into motion through the solution space with a velocity vector representing the particle's speed in each dimension. Each particle has a memory to store its historically best solution (i.e., its best position ever attained in the search space so far, which is also called its experience) [21].

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 [23]:

- A population of particles is randomly initialized with point 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 [24], 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:

$$Vel_i^{k+1} = w_k Vel_i^k + c_1 r_1 \left(Pbest_i^k - X_i^k\right) + c_2 r_2 \left(Gbest^k - X_i^k\right) \quad (2)$$

$$X_i^{k+1} = X_i^k + Vel_i^{k+1} \quad (3)$$

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 w_k is updated as [22]:

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

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, 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 [7].
- r_1, 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*.

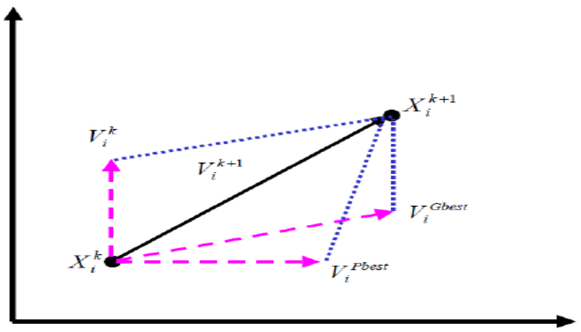


Fig. 2. Updating the position of PSO

IV. PSO-ED for *p*-Center Location Problem

This paper develops a PSO-ED algorithm for the continuous Euclidean absolute *p*-center location problem which has been proved to be NP-hard. Due to its complexity, hardness and importance it has always been a challenge for researchers who approached it from different perspectives. In [13], we suggested a PSO algorithm for the absolute location problem on networks, in this paper a modified algorithm "PSO-ED" is presented. The main modification is that, we generate the swarm on space/plane limits instead of generating it on arcs of networks as in [13].

PSO-ED algorithm has the task to randomly generate a swarm of birds with size (*Swarm_Size* * *Number_Centers*) for each dimension in *R* within the space/plane which contain the demand points. For each particle in the swarm; we compare the minimax value from each particle – which contains the coordination of centers *p* – to all demand points. Therefore each center will serve a set of demand points i.e. the space/plane will be divided into *p* sections. The procedure is then repeated with the remaining particles in order to find the combination with the best minimum values. The corresponding minimax combination is the optimal location. The PSO-ED search runs in iterations until some predefined stopping criteria is satisfied (*Number of iterations*). The PSO-ED proposed algorithm to solve the continuous Euclidean *p*-center location problem in R^2 can be described as follows:

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

Step 2. Generate randomly the initial particles positions (x_j, y_j) in R^2 in the range of upper and lower limits for each dimension for each center with size (*s* * *p*). Set the swarm velocities *Vel* to zero.

Step 3. The objective function and fitness value of each particle according to Equation (1) and the *Pbest* is calculated. The best among the *Pbest* is denoted as *Gbest*.

Step 4. The velocity and position of each particle is modified/updated according to Equations (2) and (3), respectively.

Step 5. 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 6. If the current fitness value is better than the *Gbest*, then update *Gbest* to current best position and fitness value.

Step 7. Steps 4 to 6 are repeated until the maximum number of iterations is met.

V. Experimental Results

For our computational experiments, we applied our algorithm on the 2D and 3D data sets which have been used in [12]. The 2D is from the common TSPLIB library which represents cities/locations in different countries (available at [25, 26]). According to [12], solutions of these problems are very useful in facility location problems where the objective is to minimize the maximum time to reach any location. In TSPLIB instances, the coordinates of points are provided. The number of points range from 7,146 to 1,904,711 points. The 3D geometric model data sets can be found at http://www.ocnus.com/models/, and the number of points ranges from 352 to 437,645.

Fayed and Atiya [12] applied the exact algorithm of branch and bound to large scale location problems on the above mentioned 2D and 3D datasets, with *p* ranging from 3 to 8 centers. They reported comprehensive results of applying the algorithm while trying to achieve accuracies (ϵ) of 10^{-2}, 10^{-3}, and 10^{-4}; the results of which are provided in Table 1. The presented results in Table 1 show, for each problem, the maximum distance between a demand point and its closest center. In our paper, we used the results of $\epsilon = 10^{-2}$ obtained by [12] as an upper bound for our PSO-ED results; presented also in Table 1.

Therefore, Table 1 provides a comparison between using the exact algorithm of branch and bound and the developed PSO-ED algorithm. Comparing the results of the PSO-ED algorithm with those of the branch and bound algorithm at $\epsilon = 10^{-2}$, it is evident that the PSO-ED algorithm provides more accurate results for both the 2D and 3D datasets.

On the other hand, when comparing with the branch and bound with $\epsilon = 10^{-4}$, the bolded values indicate that the PSO-ED slightly outperforms the branch and bound algorithm in many cases for the 2D location problems, and is slightly more accurate for some of the 3D location problems. In either case, even for the instances where the branch and bound outperforms the PSO-ED algorithm, the differences are so small. Accordingly, experimentally we have managed to demonstrate the efficiency of PSO-ED algorithm despite of its simplicity and ease-to-use.

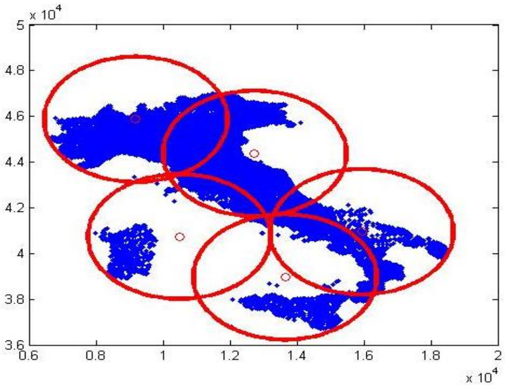
TABLE I. Maximum Distance Comparison between the PSO-ED and Branch and Bound algorithms for different *p* centers

Problem, Dimension	Demand Points (*n*)	Centers (*p*)	Branch and Bound (max distance)		PSO-ED (max distance)
			10^{-2}	10^{-4}	
Egypt, **2D**	7,146	5	2,363.67	2,352.03	**2,351.90**
		6	2,073.79	2,057.22	**2,056.90**
		7	1,848.18	1,839.08	**1,839.00**
		8	1,722.14	1,713.66	**1,713.60**
USA, **2D**	13,509	5	100,486.74	99,991.80	**99,987.00**
		6	91,950.86	91,310.62	91,411.00
		7	79,917.17	79,566.79	**79,565.00**
		8	75,836.19	75,533.03	75,585.00
Germany, **2D**	15,112	5	2,071.56	2,061.15	**2,061.10**
		6	1,798.07	1,792.68	**1,792.60**
		7	1,647.17	1,645.65	**1,645.60**
		8	1,551.99	1,546.27	1,547.00
Italy, **2D**	16,862	5	2,746.53	2,732.82	2,734.20
		6	2,248.18	2,239.11	**2,239.00**
		7	1,984.57	1,970.92	**1,970.80**
		8	1,880.49	1,860.87	1,878.30

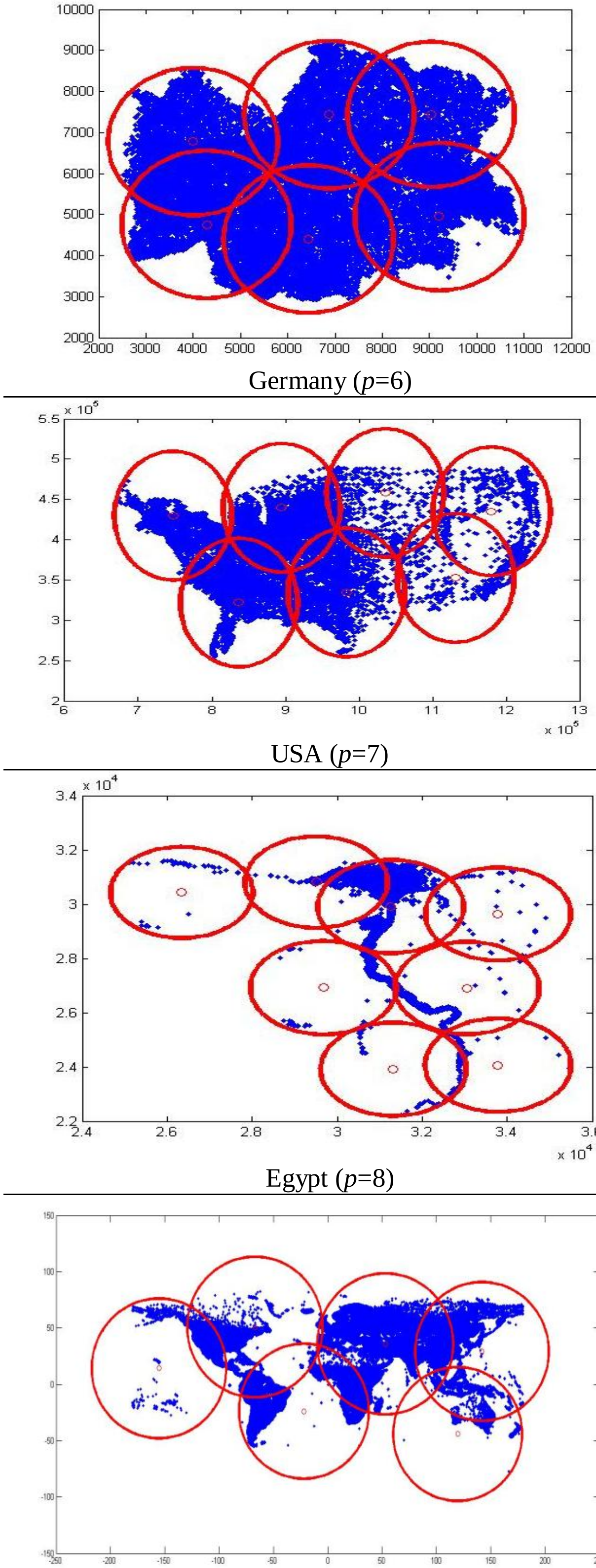
Problem, Dimension	**Demand Points (n)**	**Centers (p)**	**Branch and Bound (max distance)**		**PSO-ED (max distance)**
			10^{-2}	10^{-4}	
World, **2D**	1,904,711	5	72.22	71.41	**71.40**
		6	62.55	62.12	62.43
Cat, **3D**	352	3	0.068	0.067	**0.0669**
		4	0.058	0.057	0.0575
		5	0.052	0.051	0.0515
Seashell, **3D**	18,033	3	0.697	0.694	0.6942
		4	0.607	0.605	0.6053
		5	0.549	0.547	0.5485
Bunny, **3D**	35,947	3	0.066	0.065	0.0653
		4	0.056	0.056	**0.0557**
		5	0.051	0.051	**0.0509**
Dragon, **3D**	437,645	3	0.066	0.065	0.0654
		4	0.061	0.060	0.0601

As an example, the location of p emergency centers for some countries is illustrated in Figure 3. For the different countries, the cities are demonstrated through the blue points, while the service/emergency centers are the red points. The locations of the p centers are optimally selected through the PSO-ED algorithm such that their respective areas encompass all the regions within the optimal maximum distance calculated.

As evident from Figure 3, the service regions of some centers could overlap. For example, the first country presented in Figure 3 is Italy, where the best locations of 5 emergency centers are provided with a maximum coverage distance of 2,734.20.



Italy (p=5)



Germany (p=6)

USA (p=7)

Egypt (p=8)

World (p=6)

Fig. 3. Location of p-center for some countries

VI. CONCLUSION

The continuous absolute *p*-center location problem with Euclidean distance is a complex, and NP-hard problem. Particle swarm optimization (PSO) is a simple and effective algorithm to optimally solve complex continuous problems.

In this paper, a new PSO algorithm for the absolute *p*-center location problem has been developed (PSO-ED). The developed algorithm is simple, easy to apply, and as experimentally shown is an efficient algorithm. PSO-ED has the task to randomly generate a swarm for each dimension in *R* within the space and for each particle; compares the minimax value from each particle. The procedure is then repeated in order to find the combination with the best minimum values.

Results on several well-known test problems are compared with an exact method from the literature. The PSO-ED algorithm used to solve a common 2D and 3D datasets up to 1,904,711 points. We compared our results with a branch and bound algorithm, and the results showed that the PSO-ED algorithm is capable of solving continuous absolute *p*-center location problems optimally.

VII. FUTURE WORK

In this paper, we put forth a new algorithm for the absolute *p*-center location problem that is simple and efficient. The PSO-ED is devised to solve the problem. Although its effectiveness, a hybrid version is recommended in which PSO may be combined with anther metaheuristic technique in order to achieve more accurate results for all instances. We may also expand the usage of the algorithm to solve α-neighbor *p*-center problem in which each demand point is assigned to α service facilities, so that each demand point could withstand the failure of α-1 service facilities.

ACKNOWLEDGMENTS

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

REFERENCES

[1] H. A. Eiselt and V. Marianov, Foundations of Location Analysis vol. 155: Springer, 2011.

[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 Locations of Switching Centers and the Absolute Centers and Medians of a Graph," Operations Research, vol. 12, pp. 450-459, 1964.

[4] 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.

[5] N. megiddo and K. j. Supowits, "On the Complexity Of Some Common Geometric Location Problems," Sofiety for Industrial and Applied Mathematics, vol. 13, pp. 182–196, 1984.

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

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

[8] 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.

[9] R. Chen and G. Y. Handler, "Relaxation method for the solution of the minimax location-allocation problem in euclidean space," Naval Research Logistics vol. 34, pp. 775–788, 1987.

[10] 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.

[11] R. Chen, "Solution of minisum and minimax location–allocation problems with Euclidean distances," Naval Research Logistics Quarterly, vol. 30, pp. 449-459, 1983.

[12] H. A. Fayed and A. F. Atiya, "A mixed breadth-depth first strategy for the branch and bound tree of Euclidean k-center problems," Computational Optimization and Applications, vol. 64, pp. 675-703, 2013.

[13] H. M. Rabie, I. El-Khodary, and A. A. Tharwat, "Applying Particle Swarm Optimization for the Absolute p-center Problem," International Journal of Computer and Information Technology, http://ijcit.com/Vol2Issue5.php, vol. 2, pp. 1010-1015, 2013.

[14] W. Miehle, "Link-Length Minimization in Networks," Operations Research, vol. 6, 1958.

[15] L. Cooper, "Location-Allocation Problems," Operations Research, vol. 11, pp. 331-343, 1963.

[16] G. Handler and P. Mirchandani, Location on networks: Theory and algorithms: MIT Press, Cambridge, 1979.

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

[18] E. Minieka, "The Centers and Medians of a Graph," Operations Research, vol. 25, pp. 641-650, 1977.

[19] 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.

[20] H. Calik and B. C. Tansel, "Double bound method for solving the p-center location problem," vol. 40, pp. 2991–2999, 2013.

[21] H. Ahmed and J. Glasgow, "Swarm Intelligence: Concepts, Models and Applications," School of Computing, Queen's University 2012.

[22] 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.

[23] 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.

[24] 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.

[25] National Traveling Salesman Problems. Available: http://www.math.uwaterloo.ca/tsp/world/countries.html#EG

[26] World TSp. Available: http://www.math.uwaterloo.ca/tsp/world/index.html