A modified particle swarm optimization algorithm for location problem

A modified particle swarm optimization algorithm for location problem

I A Osinuga[1], A A Bolarinwa[1] and L A Kazakovtsev[2,3]

[1]Federal University of Agriculture, PMB 2240, Alabata Road, Abeokuta, Nigeria,
[2]Reshetnev Siberian State University of Science and Technology,
31 Krasnoyarskiy Rabochiy av., Krasnoyarsk, 660037, Russia,
Siberian Federal University, 79 Svobodny av., Krasnoyarsk, 660041, Russia

E-mail: levklevk@gmail.com

Abstract. In the Weber location problem which was proposed for optimal location of industrial enterprises, the aim is to find the location of a point such that the sum of weighted distance between this point and a finite number of existing points is minimized. This popular model is widely used for optimal location of equipment and in many sophisticated models of cluster analysis such as detecting homogeneous production batches made from a single production batch of raw materials. The well-known iterative Weiszfeld does not converge efficiently to the optimal solution when the solution either coincides or nearly coincides with one of the demands point which is not the optimum. We propose a modified Particle Swarm Optimization (PSO) algorithm. The velocity update of the PSO is modified to enlarge the search space and enhance the global search ability. The preliminary results of these algorithms are analyzed and compared.

1. Introduction

Facility location problems seek to optimize the placement of facilities such that the demands of customers can be met at the lowest cost and/or shortest distance. Location optimization problems have numerous applications in the field of mathematics, economics, physics and engineering. Many kinds of the distance functions can be employed, distances in the Weber problems are often taken to be Euclidean distances [1,2].

Weber problem (also often called the Fermat Steiner Weber problem, median problem, minisum problem, or, in the special case of three existing facilities having equal weights, as the Fermat Torricelli problem):

$$min\ f(X)=\sum_{n=1}^{m} w_i d(X,P_I) \tag{1}$$

where $d(X,P_i)=\sqrt{(x-a_i)^2+(y-b_i)^2}$ is the distance between the new facility and the existing facility i, $X=(x,y)$ is the coordinate of the location of new facility, $P_i=(a_i,b_i)$ are the coordinates of the location of existing facility i and $w_i=w(P_I)$ are positive weights that specify the demand of the existing facility P_i. The objective is to minimize the total travel cost (as the case may be) between the new facility X and P_i modeling, for example, a warehouse, and a set of customers, respectively.

Researchers have worked on Weber location problem especially on the distance function (see: Osinuga et al. [4], Drezner and Hamacher [5], Drezner et al. [6], Osinuga and Bamigbola [7], Brimberg et al. [8], Farahani and Hekmatfar [9], Kazakovtsev et al. [10], Osinuga et al. [11], Stanimirovic et al. [12]).

2. Known Methods

Weiszfeld procedure is a one-point iterative method that involves the calculation of gradient in each iteration, and it is widely used in solving single facility Weber problem with Euclidean distances and also in multi-facility procedures as a step in the solution algorithm [13], although it was observed that the method can reach a non-optimal point, the situation described as "getting stuck", and starting points of method leading to this situation are called "bad" starting points [14].

Choose an initial solution, X_0 and set $k = 0$

1. Iteration step:

$$x^{k+1} = \frac{\sum_{i=1}^{n} \frac{w_i a_i}{\sqrt{(x^k - a_i)^2 + (y^k - b_i)^2 + \in}}}{\sum_{i=1}^{n} \frac{w_i}{\sqrt{(x^k - a_i)^2 + (y^k - b_i)^2 + \in}}}; \; y^{k+1} = \frac{\sum_{i=1}^{n} \frac{w_i b_i}{\sqrt{(x^k - a_i)^2 + (y^k - b_i)^2 + \in}}}{\sum_{i=1}^{n} \frac{w_i}{\sqrt{(x^k - a_i)^2 + (y^k - b_i)^2 + \in}}}.$$

2. $k = k + 1$.The stopping criterion rule is usually number of iterations or reaching a tolerance.

The general purpose optimization method known as PSO was an intelligent technology first presented in 1995 by Eberhart and Kennedy [15], Shi and Ebehart [16]. Stochastic search methods which do not require any properties of the objective function have been developed. They include, among others, genetic algorithms (GA) [17], differential evolution (DE) [18], particle swarm optimization (PSO) [15] and ant colony optimization (ACO) [19]. These methods evaluate the objective function in a random sample of points from the search space and subsequently manipulate the sample; they are often referred to as population set-based global optimization methods. This work deals with solving (1) using a modified PSO. It is widely used to find the global optimum solution in a complex search space. The velocity of each particle is updated using its updated velocity per iteration in the algorithm. Thus, as compared to other population set-based methods, e.g., genetic algorithm or differential evolution, PSO has memory. Previously visited best positions are remembered, while in GA and DE, these are forgotten once the current population changes [20]. Also, Fathali and Jamalian [21] used the concept of PSO in their efforts at providing an efficient method for goal square Weber location problem (a variant of Weber location problem).

PSO is initially a group of random particles (random solutions), and then the optimum solutions are found by repeated searching. In every iteration, a particle will follow two bests to renew itself: the best position found for a particle called $P_{best;i}$ and the best position found for the whole swarm called G_{best}.

Suppose x^t denote the position vector of particle in the multidimensional search space at time step t, then the position of each particle is updated in the search space by

$$x_i^{t+1} = x_i^t + v_i^{t+1} \text{ with } x_i^0 \approx \cup(x_{min}, x_{max}) \tag{2}$$

where x_i^t is the position vector of particle i at time step t, v_i^t - is the velocity vector of particle i that drives the optimization process, $(x_{\min}, x_{\max})$ is the uniform distribution where $x_{\min}$ and $x_{\max}$ are its minimum and maximum values respectively.

Global best PSO (or gbestPSO) is a method where the position of each particle is influenced by the best-fit in the entire swarm. It is the best value of particle in the swarm. $P_{best;i}$ is the personal best position corresponds to the position in search space where particle i had the smallest value as determined by the objective function *f*, considering a minimization problem. G_{best} is the position yielding the lowest value amongst all the personal best $P_{best;i}$

Local Best PSO (lbest PSO) is a method which only allows each particle to be in influenced by the best- t particle chosen from its neighbourhood.

$$P_{best,i}^{t+1} = \begin{cases} P_{best,i}^{t} & \text{if} \quad f(x_i^{t+1}) > f(P_{best,i}^{t}) \\ x_i^{t+1} & \text{if} \quad f(x_i^{t+1}) \le f(P_{best,i}^{t}) \end{cases}, \tag{3}$$

G_{best}(at time t) = min[$P_{best;i}^{t}$]. For Gbest PSO method, the velocity of particle *i* is calculated by

$$v_{ij}^{t+1} = v_{ij}^{t} + c_1 r_{1j}^{t}\left[P_{best,i}^{t} - x_{ij}^{t}\right] + c_2 r_{2j}^{t}\left[G_{best} - x_{ij}^{t}\right], \tag{4}$$

v_{ij}^{t} is the velocity vector of particle i in dimension j at time t, x_{ij}^{t} is the position vector of particle i in dimension j at time t, $P_{best;i}^{t}$ - is the personal best position of particle i in dimension j found from initialization through time t, G_{best} is the global best position of particle i in dimension j found from initialization through time t, c_1 and c_2 are coefficients of learning factors are positive acceleration constants, which are used to level the contribution of the cognitive and social component respectively, r_{1j}^{t} and r_{2j}^{t} are random numbers from uniform distribution at time t.

For lbest PSO method, we have

$$v_{ij}^{t+1} = v_{ij}^{t} + c_1 r_{1j}^{t}\left[P_{best,i}^{t} - x_{ij}^{t}\right] + c_2 r_{2j}^{t}\left[L_{best} - x_{ij}^{t}\right] \tag{5}$$

where $L_{best;i}$ is the best position that any particle has had in the neighbourhood of particle i found from initialization through time t.

The velocity components are essential for updating particles velocity. In (4) and (5), v_{ij}^{t+1} is called inertia component that provides a memory of the previous flight direction that means movement in the immediate past; $c_1 r_{1j}^{t}\left[P_{best,i}^{t} - x_{ij}^{t}\right]$ is called cognitive component which measures the performance of the particles relative to past performances; $c_2 r_{2j}^{t}\left[G_{best} - x_{ij}^{t}\right]$ for gbest PSO or $c_2 r_{2j}^{t}\left[L_{best} - x_{ij}^{t}\right]$ for lbest PSO is called social component, it represents the process of learning from the experiences of other particles on the part of certain particle, and it also shows the information sharing and social cooperation among particles.

Algorithm (PSO)

1.Initialize iteration counter. Initialize N random position of particles and store them in S. Initialize N random velocities and store them in V. Initialize N pbest and store them in P. Set $gbest^{t}$ equal the best pbest in P.

2.Iteration Step: For each ith particle:

2.1Update V: Calculate velocity V_i^{t+1} using (4). Update S: calculate position x_i^{t+1} using (2). Update P: calculate position $pbest_i^{t+1}$ using (3) . Update gbest: $gbest^{t+1}$= min[$P_{best;i}^{t+1}$]; i ϵ [1,…,n]. t:=t+1.

The stopping criterion rule is usually maximum number of iterations, the maximum CPU time, the number of successive best objective function values without any improvements or reaching a tolerance.

Shi and Eberhart introduced the inertia weight, w, a scaling factor associated with the velocity during the previous time step, resulting in a new velocity update equation, so that (4) becomes $v_{ij}^{t+1} = w v_{ij}^{t} + c_1 r_{1j}^{t}\left[P_{best,i}^{t} - x_{ij}^{t}\right] + c_2 r_{2j}^{t}\left[G_{best} - x_{ij}^{t}\right]$.

The original PSO velocity update equation can be obtained by setting $w = 1$. Shi and Eberhart indicate that choosing w ϵ [0.8, 1.2] results in faster convergence, but that larger w values (> 1.2) result in more failures to converge [16].

Clerc and Kennedy [21] also introduced another interesting modification to the canonical PSO in the form of a constriction coefficient, K, which controls all the three components in velocity

Uupdate rule (3.4). This has the effect of reducing the velocity as the search progresses and also insure theconvergence. In this modification, the velocity update is given as $v_{ij}^{t+1} = K\left(v_{ij}^{t} + c_1 r_{1j}^{t}\left[P_{best,i}^{t} - x_{ij}^{t}\right] + c_2 r_{2j}^{t}\left[G_{best} - x_{ij}^{t}\right]\right)$, $K = \frac{2}{\left[2-\varphi-\sqrt{\varphi^2-4\varphi}\right]}$ where $\varphi = c_1 + c_2 > 4$.

3. New Algorithms

The MPSO proposed in this work is aim at expanding the scope of the best positions which are saved by each particle and the swarm, in order to improve the performance of the PSO. To achieve this, full advantage of the excellent positions encountered by each particle and the swarm are observed, we take cognizance of the second personal best position ($sP_{best;i}{}^{t}$) and the second global best position(sG_{best}) alongside the normal personal best position($P_{best;i}{}^{t}$) and the global best position(G_{best}) of the PSO and we try to insure convergence by the introduction of K, the constriction coefficient.

The following equations illustrate these positions.

$$v_{ij}^{1}(t+1) = K\left(v_i^t + c_1 r_{11}^{t}\left[P_{best,i}^{t} - x_{ij}^{t}\right] + c_2 r_{21}^{t}\left[G_{best} - x_{ij}^{t}\right]\right),$$
$$v_{ij}^{2}(t+1) = K\left(v_i^t + c_1 r_{12}^{t}\left[sP_{best,i}^{t} - x_{ij}^{t}\right] + c_2 r_{22}^{t}\left[G_{best} - x_{ij}^{t}\right]\right),$$
$$v_{ij}^{3}(t+1) = K\left(v_i^t + c_1 r_{13}^{t}\left[P_{best,i}^{t} - x_{ij}^{t}\right] + c_2 r_{23}^{t}\left[sG_{best} - x_{ij}^{t}\right]\right),$$
$$v_{ij}^{4}(t+1) = K\left(v_i^t + c_1 r_{14}^{t}\left[sP_{best,i}^{t} - x_{ij}^{t}\right] + c_2 r_{24}^{t}\left[sG_{best} - x_{ij}^{t}\right]\right),$$

and the ith particle's iterative velocity is $v_{ij}(t+1) = \max_{1\le k\le 4} f(x_{ij}(t) + v_{ij}^{k}(t+1))$ where $f(x)$ is the fitness value at x position. and K, the constriction coefficient, is a function of φ:

$$K = \frac{2}{\left[2-\varphi-\sqrt{\varphi^2-4\varphi}\right]} \text{ where } \varphi = c_1 + c_2 > 4.$$

At the next iterative position, we would have had the group's G_{best} and sG_{best} updated alongside the renewal of the particles's $P_{best;i}$ and $sP_{best;i}$ at the end of the previous iterative position as follows:

$$sP_{best,i}^{t+1} = \begin{cases} sP_{best,i}^{t} & \text{if} \quad f(x_i^{t+1}) > f(P_{best,i}^{t}), \\ x_i^{t+1} & \text{if} \quad f(P_{best,i}^{t}) \ge f(x_i^{t+1}) > f(sP_{best,i}^{t}), \\ sP_{best,i}^{t} & \text{if} \quad f(x_i^{t+1}) \le f(sP_{best,i}^{t}), \end{cases} \tag{6}$$

$$P_{best,i}^{t+1} = \begin{cases} P_{best,i}^{t} & \text{if} \quad f(x_i^{t+1}) \le f(P_{best,i}^{t}), \\ x_i^{t+1} & \text{if} \quad f(x_i^{t+1}) > f(P_{best,i}^{t}), \end{cases} \tag{7}$$

$$sG_{best} = \begin{cases} G_{best} & \text{if} \quad f(P_{best,i}^{t}) > f(G_{best}), \\ P_{best,i}^{t} & \text{if} \quad f(G_{best}) \ge f(P_{best,i}^{t}) > f(sG_{best}), \\ sG_{best} & \text{if} \quad f(P_{best,i}^{t}) \ge f(sG_{best}), \end{cases} \tag{8}$$

$$G_{best} = \begin{cases} G_{best} & \text{if} \quad f(P_{best,i}^{t}) \le f(G_{best}), \\ P_{best,i}^{t} & \text{if} \quad f(P_{best,i}^{t}) > f(G_{best}), \end{cases} \tag{9}$$

$i \in [1,\cdots,n]$ and $n > 1$. Here, n is the number of particles in the particle swarm.

Algorithm (MPSO):

- Initialization: setting the maximum iterative steps and the number of the particles producing randomly the position and velocity of each particle in the particle swarm.
- Evaluate the fitness value of each particle.
- Renew the $P_{best;i}$ and $sP_{best;i}$ position of each particle according to (4.6) and (4.7).
- Renew the G_{best} and sG_{best} position in the particle swarm according to (4.8) and (4.9).
- Change the velocity of each particle according to (4.1-4.4).
- Move each particle to the new position according to (3.1) and return to 2 above.
- Repeat 2-6 until a stopping criterion is satisfied.

4. Computational examples

To justify the effectiveness of our proposed method, the MPSO, we test the algorithm on some problems. Example 1 (Francis and White [2]). Five existing facilities are located at (2.5,4.5), (3,2.5), (5,2), (5.5,4) & (8,5). The weights applied to these facilities are 2, 5, 7, 10 & 12 respectively. In Example 2, we have 10, 15, 20, 25, 30, 40 and 50 points problems were randomly generated with their weights.

In both examples 1 and 2, we used the WA, the classical PSO and the MPSO proposed in this work to solve all the points problems. In each problem, the objective function, the CPU time (in seconds), the iteration number and the corresponding point, that is, the new facility position were all estimated.

All computations were carried out on a Computer with Intel (R) processor with 4GB of RAM and CPU 2.16 GHz, and algorithms were coded in JavaScript software. In all methods, the iteration step of the algorithm is repeated until the tolerance level is reached. For WA, the iteration is stopped when none of the successive values of the new facility being sought in the iteration changes.

It is noticed empirically that larger number of swam size and larger iterations only increased the computational complexity per iteration and more time consuming. They also slow down the algorithm without leading to any better solutions. It is also noteworthy that the CPU time for all the problems as shown in table 1, indicated that the CPU times of WA are less than that of PSO and the proposed MPSO in all cases. However, this being an improved method, MPSO could find more promising solutions to all the problems and with considerable advantage in iteration numbers and was able to solve all the problems minimally.

Table 1. Numerical Results of example 2 in terms of objective function, CPU time and the number of iterations.

Problems	OBJECTIVE FUNCTION			CPU TIME			ITERATION NUMBER		
	WA	PSO	MPSO	WA	PSO	MPSO	WA	PSO	MPSO
10 points	82.1039	82.1087	82.0970	0.02	0.37	0.1	41	38	32
15 points	114.8334	115.7032	113.7089	0.01	0.34	0.10	21	24	19
20 points	225.6470	226.1577	225.6289	0.01	0.14	0.11	18	20	18
25 points	211.9472	212.0919	210.6591	0.02	0.68	0.19	24	25	18
30 points	261.4967	261.7235	261.3874	0.01	0.63	0.17	21	21	18
40 points	358.9703	359.0490	358.1918	0.01	0.88	0.2	17	20	16
50 points	478.0019	481.0474	476.2441	0.02	1.05	0.26	18	19	16

5. Conclusion

We showed that the modified PSO algorithm is a good option to the commonly used WA procedure, and this method has empirically been shown to be able to solve single facility Weber location problems effectively and efficiently. However, there are still areas that can be explored by researchers. The proposed MPSO can be worked upon to solve some benchmark problems and also, it can be hybridize with other meta-heuristic algorithms for solving general nonlinear optimization problems. A constrained

Weber location problem with the search direction in some specific regions can be solved using the MPSO.

References

[1] Jiang J L and Xu Y 2006 Minisum location problem with farthest Euclidean distances *Mathematical Methods of Operations Research* **64(2)** 285-308

[2] Hamacher H W and Nickel S 1998 Classification of location models *Location Science* **6** 229-42

[3] Francis R l and White, J A 1974 *Facility layout and location: An analytical approach* (Prentice Hall, Englewood Cliffs, NJ)

[4] Osinuga I A, Stanimirovic P S, Kazakovtsev L A and Akinleye S A 2015 A modeling framework on distance predicting functions for location models in continuous space *Facta Universitatis: Series Mathematics and Informatics* **30(4)** 419-43

[5] Drezner Z and Hamacher H 2002 *Facility Location: Applications and Theory*. Springer, Berlin.

[6] Drezner Z, Mehrez A and Wesolowsky G O 1991 The facility location problem with limited distances. *Transportation Science* **25** 183-7

[7] Osinuga I A and Bamigbola O M 2007 On the minimum solution to Weber problem. *Proceedings of 25th annual SAMSA Conference*, Windhoek, Namibia **2** 54-60

[8] Brimberg J, Juel H and Schobel A 2009 Locating a minisum in the plane *Discrete Applied Mathematics* **157** 901-12

[9] Farahani R Z and Hekmatfar M 2009 *Facility Location: Concepts, Models, Algorithms, and Case Studies*, Contributions to Management Science, Physica-Verlag Heidelberg, Germany

[10] Kazakovtsev L A, Stanimirovic P S, Osinuga I A, Gudyma M N and Antamoshkin A N 2014 Algorithm for location problems based on angular distance *Advances in Operations Research* **2014** 701267 doi:10.1155/2014/701267

[11] Osinuga I A., Kazakovtsev L A and Stanimirovic P S 2013 Planar Weber location problem with french metro metric. *Review Bulletin Calcutta Mathematical Society* **21(1)** 7-20

[12] Stanimirovic P S, Ciric M S, Kazakovtsev L A and Osinuga I A 2011 Single-facility Weber location problem based on the lift metric *Facta Universitatis: Series Mathematics and Informatics* **27(2)** 175-90

[13] Morris J G and Verdini W A 1979 Minisum l_p distance location problems solved via a perturbed problem and Weiszfeld's algorithm *Operations Research* **27** 1180-8

[14] Beck A and Sabreh S 2014 Weiszfeld’s method: Old and New Results, Springer science+Business Media, NY

[15] Kennedy J and Eberhart R 1995 Particle swarm optimization *Proceedings of IEEE International Conference on Neutral Networks*, Perth, Australia **4** 1942-8

[16] Shi Y and Eberhart R C 1998 Particle swarm optimizer *Proceedings of 1998 IEEE International Conference on Evolutionary Computation*, Anchorage, USA **4** 69-73

[17] Michalewicz Z 1996 *Genetic Algorithms + Data Structures = Evolution Programs* (Berlin: Springer-Verlag)

[18] Storn R and Price K 1997 Differential evolution - A simple and efficient heuristic for global optimization over continuous space *Journal of Global Optimization* **11** 341-59

[19] Dorigo M and Caro G D 1999 The ant colony optimization meta-heuristic *New Ideas in Optimization* 11-32

[20] Ali M M and Kaelo P 2008 Improved particle swarm algorithm for global optimization *Applied Mathematics and Computation* **196** 578-93

[21] Fathali J and Jamalian A 2017 Efficient methods for goal square Weber location problem. *Iranian Journal of Numerical Analysis and Optimization* **7** 65-82

[22] Clerc M and Kennedy J 2002 The particle swarm-explosion, stability and convergence in a mathematical complex space *IEEE Transactions on Evolutionary Computational* **6(1)** 58-73