The International Journal of Informatics Media and Communication Technology
ISSN (Print 2682-2105) (Online 2682-2881)

Particle Swarm Optimization to Solve Multi-Center Location Problem on Small Networks

Hassan Mohamed Rabie
Institute of National Planning, Cairo, Egypt
Hassan.rabie@inp.edu.eg

Abstract:

This paper addresses the multi-center location problem, in which a set of p centers to be located to minimize the maximum distance between a center p and its nearest demand point. The problem is NP-hard. Therefore, polynomial time-bounded algorithms are unlikely to yield an optimal solution. Particle swarm optimization (PSO) is a metaheuristic algorithm that has been shown to be effective in solving such hard and complex optimization problems.

This paper presented a new PSO for the multi-center location problem on small networks. Three objectives are considered as follows: (١) developing a new PSO algorithm to solve the problem, (٢) applying the developed algorithm to a set of well-known benchmark location problems, and comparing the results. (٣) verifying the stability of the algorithm. The result is a simple but effective algorithm for solving multi-center problems on networks. The new proposed PSO algorithm solved multi-center location problems, when all candidate solutions can be investigated. PSO was found to be stable and the average time of runs increases linearly as the number of centers increases. The advantage of this algorithm is that, it calculates the optimal solution for each candidate solutions

Keywords: Swarm Optimization – Center Location – Networks.

١. Introduction

Location science is a very attractive research area. Center location problems are especially prominent in emergency service locations, because saving lives is significantly more vital than any transportation costs involved in providing that service. The center location problem also is an important subject of strategic planning. It is also a very popular and common operations research problem. The location of a single center is easier than locating many facilities, which is known as the multi-center location problem. Location decisions arise in both government and private sectors. For example, governments, must decide where to locate emergency highway patrol cars, fire stations and ambulances. Offices, production and assembly plants, distribution hubs, and retail outlets must all be located optimally in the private sector. Poor location choices might result in higher expenses and lower competitiveness. (Daskin ٢٠١٣).

(Hakimi ١٩٦٤) was the first to propose the single center problem for locating a station in such a way that the station's maximum distance from a group of communities (demand points) connected by a highway system is reduced. By addressing a finite series of minimum set coverage problems, (Minieka ١٩٧٠, Christofides and Viola ١٩٧١) develped methods for solving multi-center (p-center) location ptoblems. (Christofides ١٩٧٥) shown that only a subset of the edges may be considered. However, this method failed to overcome generic p-center problems. (Kariv and Hakimi ١٩٧٩) showed that the multi-center location problem is NP-hard problem. This type of problems is hard to solve since it is non-linear, and non-convex (Brandeau and Chiu ١٩٨٩). The complexity and non-linearity of the multi-center location problem requires novel method to be developed. (Davidovica et al. ٢٠١١) developed a bee colony optimization method to solve vertex location problem on networks, also (Kaveh and Nasr ٢٠١١) solved the same problem using a modified harmony search technique. (Rabie et al. ٢٠١٣, Rabie et al. ٢٠١٣, Rabie ٢٠٢١) presented PSO algorithms for large-scale p-center and p-median location

problems. (Current et al. ٢٠٠١, Farahani and Hekmatfar ٢٠٠٩, Laporte et al. ٢٠١٥, Laporte et al. ٢٠١٩) provide an overview of p-center problem on networks.

Particle Swarm optimization (PSO) was introduced by Kennedy and Eberhart in ١٩٩٥, as an intelligence nature-inspired algorithm, based on natural swarm behavior. Since that, PSO has sparked a lot of attention and interests. PSO has been applied to many optimization areas (Yang ٢٠١٤).

This paper aims to develop an efficient PSO algorithm to solve multi-center location problem on networks. To test the efficiency our developed PSO algorithm, we performed an extensive experiment on different multi-center location problems on small networks in the literature. The paper is divided as follows; Section ٢ defines the location problem. Section ٣ introduces PSO algorithm. Section ٤ shows how PSO solves multi-center location problem. Section ٥ is a numerical example, and section ٦ is for the computational experiments. Finally, is the conclusion.

٢. Multi-Center Location Problem on Networks

Multi-location problems on network, is a collection of vertices (nodes) v_1, v_2,...,vn (denoted by the set V), with non-negative weight w(v), and a collection of edges(links) e_1,e_2,...,em (denoted by the set E) joining all or some of these nodes has a positive length b(e) (Christofides ١٩٧٥). Multi-center "Minimax p-center" location problems on networks locates more than one facility/center on a network to minimize the maximum distance between a demand point and its nearest center (Daskin ١٩٩٥).

The study of networks is usually known in mathematical literature as graph theory and in engineering and applied science as network analysis, though many researchers have attempted to approach the network location problems (Handler and Mirchandani ١٩٧٩).

(Hakimi ١٩٦٤) was the first to tackle this type of location problems on a network. The author proposed an enumerative approach for finding single center (p=١). Since the publication of

Hakimi's study on the "absolute center and median", the literature on network location problems has exploded (ReVelle and Eiselt ٢٠٠٥). Many studies have been conducted to improve Hakimi's technique, including (Hakimi and Maheshwari ١٩٧٢, Hakimi et al. ١٩٧٨, Kariv and Hakimi ١٩٧٩, Minieka ١٩٨١, Sforza ١٩٩٠).

٢,١. Problem Formulation

Given a set connected nodes (V) and edges (E), the goal is to locate p centers anywhere on the network to minimize the maximum of the distances from each node to its nearest center. Let G(V, E) be a finite undirected connected network with no loops. b(e) is a positive edge length; e ∈ E. wv is a positive vertex weight; v ∈ V. X = [x١, x٢,..., xp], represent p locations to be found. Let d(v, X) = min [d(v, x١), d(v, x٢),..., d(v, xp)] represent the shortest path distance on G from any point v in G to a facility location.

$$\min_{X\ on\ G} f(X) = \max_{١\le i\le v} w_i\ d(v_i, X) \qquad (١)$$

For a given edge $e(uz)$ which link between node u and node z, and if we let t denote the distance between a point x on edge $e(uz)$ to u along the edge then the distance from a point v_i on G to x is:

$$d(v_i, x) = \min(d(v_i, u)+t\ ,\ d(v_i, z)+b(uz)-t) \qquad (2)$$

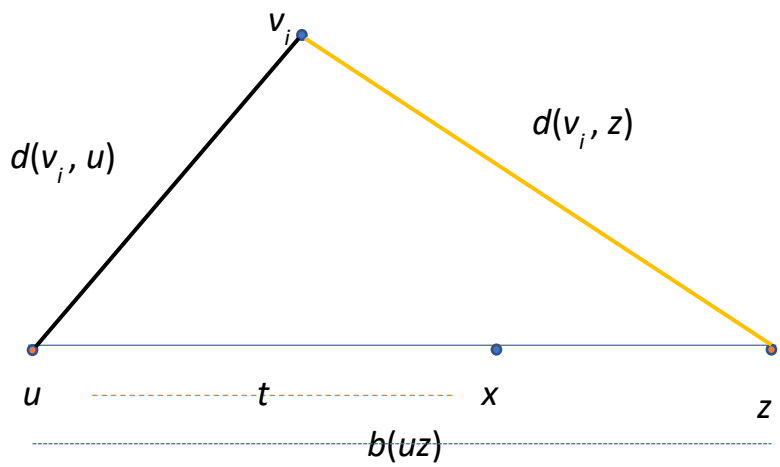

Figure ١: An illustrative example of a distance d(vi,x).

٣. Particle Swam Optimization (PSO)

PSO is a problem-solving algorithm, which iteratively attempts to enhance a set of candidate solutions. Each particle's move is governed by its local best-known position, and it moves toward the best-known places. PSO algorithm is a powerful optimization tool which appeared, likely more than other meta-heuristic algorithms (Sedighizadeh et al. ٢٠٢١).

PSO was introduced by (Kennedy and Eberhart ١٩٩٥), and there has been a steady increase in the number of research papers reporting its successful application in solving a variety of optimization problems, including finding optimal routes, scheduling problems and function optimization problems, since that time (Ahmed & Glasgow, ٢٠١٢).

The position and velocity of any particle are updated as follows (Kennedy and Eberhart ١٩٩٥):

$$Vel_i^{k+1} = wt_k Vel_i^k + c_1 r_1 \left(Pbest_i^k - X_i^k\right) + c_2 r_2 \left(Gbest^k - X_i^k\right) \quad (٣)$$

$$X_i^{k+1} = X_i^k + Vel_i^{k+1} \quad (٤)$$

where,

Vel_i^k Velocity of particle *i* at iteration *k*.

c_1, c_2 Positive constant weighting. They called the cognitive and social parameters. They typically set to equal ٢ (Sumathi and Paneerselvam ٢٠١٠).

r_1, r_2 Two random numbers uniformly selected in the range of [٠,٠, ١,٠].

X_i^k Position of particle *i* at iteration *k*.

$Pbest_i^k$ The best position of particle *i* at until iteration *k*.

$Gbest_i^k$ The position of the global best swarm particle until iteration *k*.

wt_k inertia weight updated as (Shelokar et al. ٢٠٠٧):

$$wt_k = (wt_{max} - wt_{min}) * \frac{(k_{max} - k)}{k_{max}} + wt_{\min,} \quad (٥)$$

$wt_{\min}$ and $wt_{\max}$ are the minimum and maximum values of w_{tk} respectively; k_{max} is the number of iterations.

According to Equations ٣ and ٤, Figure ٢ shows the basic flow of the original PSO algorithm can be described as shown below:

Basic flow of PSO

1. Initialize the swarm by randomly assigning each particle to an arbitrarily initial velocity and a position in each *d*-dimension in the solution space.
2. Evaluate the desired fitness function to be optimized for each particle's position.
3. For each individual particle, update its historically best position so far, *Pbest* if its current position is better than its historically best one.
4. Identify/Update the swarm's globally best particle that has the swarm's best fitness value, and set/reset its index as g and its position at *Gbest*.
5. Update the velocities of all the particles using equation 3.
6. Move each particle to its new position using equation 4.
7. Repeat steps 2–6 until convergence or a stopping criterion is met (e.g., the maximum number of allowed iterations is reached; a sufficiently good fitness value is achieved; or the algorithm has not improved its performance for a number of consecutive iterations).

Figure ٢: The basic flow of PSO (Ahmed and Glasgow ٢٠١٢)

٤. PSO for Multi-Center Location Problem on Networks

The basic idea of our PSO algorithm is that for each edge, in the set of candidate solution, we generate randomly swarm (points) on the edge in the range between ٠ and the length of the edge, then we evaluate the objective function for each particle in the swarm. After that we proceed to the PSO approach to improve the minimum value of the objective function.

As Figure ٣ shows, the proposed PSO algortihm starts by reading the network vertices and edges then the parameters have to determined and initialized.

In step ٣, the shoretest distance matrix is calculated using Dijkstra's algorithm (Christofides ١٩٧٥). Step ٤, the set of candidate solutions is generated, which contains all the combination of edges. For looping for each candidate solution starts. Step ٥, select the first candidate solution, then generate the swarm on its edges, in step ٦. Step ٧ evaluates the objective function of the swarm. Step ٨ update *Pbest* and *f*(*Pbest*) values. Step ٨, a new for looping starts to update swarm position and velocity. Step١٠ *Gbest* and *f*(*Gbest*). At the end of the iterations the minimum value of *f*(*gbest*) is reported.

Step 1. Read network $G(V, E)$, where V is the set of vertices/nodes, E is the set of edges, $b(e)$ is the length of edges. w_l is the set of weights that may be associated to the vertices; in case of no weights set w_l =1.

Step 2. Let p (*# centers*), ss (*swarm size*), *iterations* (*# iterations*), E (*the number of edges*), c_1 (*cognitive parameter*), c_2 (*social parameter*), wt_{max} (*the maximum value of inertia weight*), wt_{min} (*the minimum value of inertia weight*), and wt_k (*inertia weight*). Set $H=\binom{|E|}{p}$; which is the total number of candidate solutions.

Step 3. Calculate $D_{V \times V}$ using Dijkstra's algorithm (Christofides, 1975); the matrix of the shortest paths between each two nodes in G. V is the number of vertices/nodes.

Step 4. Generate the set of candidate solutions CS. The set contains all the combination of edges with size H in rows and p in columns.

for $h<=H$,

Step 5. Pick $CS_h=[e_{1st}, e_{2nd},\ldots, e_{pth}]$, the h^{th} combination from the set of candidate solutions.

Step 6. Generate randomly $X_{ss \times p}$ (*swarm on edges*); the particles position ranging from 0 to the length of each edge, and set $Vel_{ss \times p} = 0$ to initiate the velocity of the swarm.

Step 7. Evaluate the objective function:

$$\min_{X \text{ on } G} f(X) = \max_{1 \le l \le v} w_l \, d(v_l, X) \qquad (6)$$

where $d(v_l, X)$ represent the shortest path distance on G from the point v_l in G to a facility location.

Step 8. Let $Pbest = X$, with $f(Pbest) = f(X)$

Step 9. Update the swarm position and velocity

for $k <= iterations$,

for $i<= ss$,

- Let $Gbest= Pbest$ (arg(min($f(Pbest)$))),
- Generate r_1 and r_2 randomly following a uniform distribution values in range [0,1] with p dimension.
- Calculate inertia weight (wt_k),

$$wt_k = (wt_{max} - wt_{min}) * \frac{(iterations - k)}{iterations} + wt_{min} \qquad (7)$$

- Calculate the velocity (Vel_i^{k+1}),

$$Vel_i^{k+1} = wt_k Vel_i^k + c_1 r_1 \left(Pbest_i^k - X_i^k\right) + c_2 r_2 \left(Gbest^k - X_i^k\right) \qquad (8)$$

- Calculate the new position (X_i^{k+1})

$$X_i^{k+1} = X_i^k + Vel_i^{k+1} \qquad (9)$$

- If X_i^{k+1}<0; then X_i^{k+1}= 0; else if X_i^{k+1}> b(e); then X_i^{k+1}=b(e).
- Evaluate $f(X_i^{k+1})$ of the new position;
  - if $f(X_i^{k+1}) < f(X_i^k)$, then $X_i^k = X_i^{k+1}$.
  - if $f(X_i^{k+1}) < f(Pbest_i^k)$, then $Pbest_i^k = X_i^{k+1}$ with $f(Pbest_i^k) = f(X_i^{k+1})$.

  end for

  end for

Step 10. Store *Gbest* and f(*Gbest*) for candidate solution h,

end for

Step 11. Report the best solution *Gbest* with the objective function equal to *min*(f(*Gbest*)).

Figure ۳: The basic flow of the proposed PSO for multi-center location problems

٥. Numerical Example

This section introduces an example to illustrate the proposed PSO algorithm to solve multi-center location problem. For simplicity let us consider an example from (Christofides and Viola ١٩٧١) to find two multi-centers (p=٢), for the network presented in Figure ٤ which has ٦ nodes/vertices and ٩ edges.

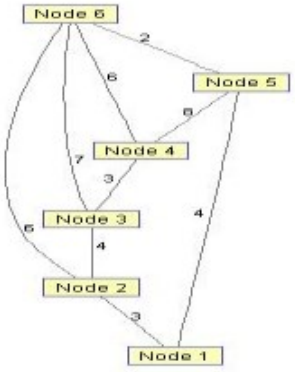

Figure ٤: Example of a network from Christofides and Viola (Christofides and Viola ١٩٧١)

The basic steps the proposed PSO for (p=٢):

Step ١. Read the following network G

edges	e_1	e_2	e_3	e_4	e_5	e_6	e_7	e_8	e_9
Node# ١	١	٥	٢	٢	٣	٦	٤	٦	٥
Node# ٢	٢	١	٣	٦	٤	٣	٥	٤	٦
Length *b*(*e*)	٣	٤	٤	٦	٣	٧	٨	٦	٢

Step ٢. Let p =٢, and set $H=\binom{٩}{٢}$ = ٣٦. The PSO parameters: $c_1=c_2=$٢ wt_{max} = ١,٢, and wt_{min} = ٠,١.

Step ٣. D (distance matrix):

	Node 1	Node 2	Node 3	Node 4	Node 5	Node 6
Node 1	0	3	7	10	4	6
Node 2	3	0	4	7	7	6
Node 3	7	4	0	3	9	7
Node 4	10	7	3	0	8	6
Node 5	4	7	9	8	0	2
Node 6	6	6	7	6	2	0

Step ٤. Create set of candidate solutions. for p=٢, we will check all the two combinations of edges with total ٣٦ candidate solutions.

		CS_1	CS_2	CS_3	CS_4	CS_5	CS_6	…	CS_{36}
1st Edge	1st Node	1	1	1	1	1	1	…	6
	2nd Node	2	2	2	2	2	2	…	4
2nd Edge	1st Node	5	2	2	3	6	4	…	5
	2nd Node	1	3	6	4	3	5	…	6

for h<=H,

Step ٥. Pick $CS_1 = [e_1, e_2]$ = [(Node١, Node٢), (Node٥, Node١)].

Step ٦. Generate randomly $X_{ss \text{ x } p}$ with size ss=١٠; ١٠ particles positions on range from [٠ to ٣] for e_1 and from [٠ to ٤] for e_2; set $Vel_{ss \text{ x } p}$ = ٠.

ss	1	2	3	4	5	6	7	8	9	10
$X_{ss,1}$	1.9672	0.1071	2.5474	2.8020	2.0362	2.2732	2.2294	1.1767	1.9664	0.5136
$X_{ss,2}$	2.8242	0.1273	1.1077	0.1847	0.3885	3.2938	2.7793	1.2684	3.8009	0.1378

Step ٧. Evaluate the objective function. For example, to evaluate the objective function of the first particle X_{1xp};

$f(X)$=max[min($X_{1,1}$+d(١,V), $b(e_1)$-$X_{1,1}$+d(٢,V), $X_{1,2}$+d(٥,V), $b(e_2)$-$X_{1,2}$+d(١,V))];

$X_{1,1}$=1.9672 $X_{1,2}$=2.8242	A_1= $X_{1,1}$ + d(1,V)	A_2= $b(e_1)$- $X_{1,1}$ + d(2,V)	A_3= $X_{1,2}$ + d(5,V)	A_4= $b(e_2)$- $X_{1,2}$ + d(1,V)	Min (A_1, A_2, A_3, A_4)
Node 1	1.9672	4.0328	6.8242	**1.1758**	1.1758
Node 2	4.9672	**1.0328**	9.8242	4.1758	1.0328
Node 3	8.9672	**5.0328**	11.8242	8.1758	5.0328
Node 4	11.9672	**8.0328**	10.8242	11.1758	8.0328
Node 5	5.9672	8.0328	**2.8242**	5.1758	2.8242
Node 6	7.9672	7.0328	**4.8242**	7.1758	4.8242
f($X_{1,p}$)=Max (Min (A_1, A_2, A_3, A_4))					8.0328

Therefore, the objective function (OF) value of the first particle is equal to ٨,٠٣٢٨, which means that, if we locate the first center on e_1 ١,٩٦٧٢ units far from Node ١ and the second center on e_2 ٢,٨٢٤٢ units far from Node ٥ the maximum distance will be

٨,٠٣٢٨. By the evaluation of all particles in the swarm the result will be:

ss	1	2	3	4	5	6	7*	8	9	10
X	1.9672	0.1071	2.5474	2.8020	2.0362	2.2732	2.2294	1.1767	1.9664	0.5136
	2.8242	0.1273	1.1077	0.1847	0.3885	3.2938	2.7793	1.2684	3.8009	0.1378
f(X)	8.0328	8.1273	7.4526	7.1980	7.9638	7.7268	7.7706	8.8233	8.0336	8.1378

Step ٨. Let *Pbest*=*X*, and *f*(*Pbest*)=*f*(*X*)

ss	1	2	3	4*	5	6	7*	8	9	10
Pbest	1.9672	0.1071	2.5474	2.8020	2.0362	2.2732	2.2294	1.1767	1.9664	0.5136
	2.8242	0.1273	1.1077	0.1847	0.3885	3.2938	2.7793	1.2684	3.8009	0.1378
f(Pbest)	8.0328	8.1273	7.4526	7.1980	7.9638	7.7268	7.7706	8.8233	8.0336	8.1378

Step ٩. Update both of swarm position and velocity

for k <= iterations,

for i <=*ss*,

- *Gbest*=*Pbest*(arg(min(*f*(*Pbest*)))) = [٢,٨٠٢٠, ٠,١٨٤٧], the current *Gbest* on [e_1 , e_2] with objective function value equal to ٧,١٩٨٠ i.e. the current best solution for the current candidate solution is the seventh particle.
- Generate r_1=[٠,٩٤٦٩,٠,٥٣٥٧] and r_2=[٠,٣١٩٩,٠,١٧٨٢].
- Calculate inertia weight (wt_k),

$$wt_1 = (1.2 - 0.1) * \frac{(20 - 1)}{20} + 0.1$$
$$= 1.145$$

- Calculate the velocity of first particle,

$$Vel_1^2 = wt_1 Vel_1^1 + c_1 r_1 \left(Pbest_1^1 - X_1^1\right) + c_2 r_2 \left(Gbest - X_1^1\right)$$

Vel_1^2=١,١٤٥*[٠ , ٠]
+٢*[٠,٩٤٦٩,٠,٥٣٥٧]*([١,٩٦٧٢,٢,٨٢٤٢] – [١,٩٦٧٢,٢,٨٢٤٢])
+٢*[٠,٣١٩٩,٠,١٧٨٢]*([٢,٨٠٢٠,٠,١٨٤٧] – [١,٩٦٧٢,٢,٨٢٤٢])
= [٠,٥٣٤٢, -٠,٩٤٠٧];

– Calculate the new position,
$X_1^2 = X_1^1 + Vel_1^2$
$X_1^2 = [1,9672,2,8242]+[0,5342, -0,9407] = [2,5014, 1,8835]$,
accept the new position, since it is between bounders of the two edges [0,3] for e_1 and [0, 4] for e_2.
– Evaluate the new position $f(X_1^2)$; $f(X_1^2) = 7.4986$
▪ if $f(X_1^2) < f(X_1^1)$ i.e.7,4986<8,0328, then $X_1^1 = X_1^2$ i.e. $X_1^1 = [2.5014, 1.8835]$.
▪ if $f(X_1^2) < f(Pbest_1^1)$ i.e.7,4986<8,0328, then $Pbest_1^1 = X_1^2$ i.e. $Pbest_1^1 = [2.5014, 1.8835]$ with $f(Pbest_1^1) = 7.4986$.
end for
end for

Step 10. Store *Gbest* and *f(Gbest)* for candidate solution *h*,
end for

Step 11. Report optimal solution, $\min_{h \epsilon CS}$ (*f*(*Gbest*))

We run the implemented code of the PSO with the following parameters: *p*=2, *ss*(*swarm-size*)=20, *iterations*=20 , $c_1=c_2=2$ and H =36, wt_{max} = 1,2, and wt_{min} = 0,1. The experiments were performed 20 runs independently. Table 1 shows the summarized results.

The first column of Table 1 provides the optimal solution from (Christofides and Viola 1971) is given. Second column shows the proposed PSO result. Column three and four show the average objective function values obtained from 20 runs and their standard deviations. The fifth column contains the average runs execution time in seconds. Table 1 shows that when all candidate solutions have been evaluated efficiently by using PSO and the optimal solution obtained for the absolute center problem (*p*=2), and also PSO has a steady performance.

Table ١: The PSO stability evaluation for Network in Example ٢ (٢٠ runs).

Optimal	PSO Results			
	Min	Average	Σ	Average CPU Time
٣,٥٠٠٠	٣,٥٠٠٠	٣,٥٠٠٤	٠,٠٠٠٥	٠,١٧٧

Table ٢ shows the optimal solution of all the candidate solutions for the network in above example for (p=٢). The optimal of the network is to locate the first center on edge e_2; ١,٠٥٤١ units far from Node ٥ and the second center on edge e_3; ٣,٥ units far from Node ١ and the maximum distance is equal to ٣,٥.

Table ٢: The result of all the ٣٦ candidate solutions.

CS	١st edge		٢nd edge		*Gpest*		*f*(*Gpest*)
	١st Node	٢nd Node٢	١st Node	٢nd Node			
e_1 , e_2	١	٥	٢	١	٣	٢,١٣٦	٧
e_1 , e_3	١	٢	٢	٣	٠	١,١٨٤٥	٦
e_1 , e_4	١	٢	٢	٦	١,٠٨١٢	٦	٦
e_1 , e_5	١	٣	٢	٤	١,٨٦٨٥	٣	٦
e_1 , e_6	١	٦	٢	٣	٠,٥٥٢٢	٥	٥
e_1 , e_7	١	٤	٢	٥	٢,١٦٩١	٥	٥
e_1 , e_8	١	٦	٢	٤	٣	٢	٤
e_1 , e_9	١	٥	٢	٦	١,٩١٩٥	٢	٦
e_2 , e_3	**٥**	**٢**	**١**	**٣**	**١,٠٥٤١**	**٣,٥**	**٣,٥***
e_2 , e_4	٥	٢	١	٦	٠,٧٩٥٨	٦	٧
e_2 , e_5	٥	٣	١	٤	١,٦٣٩٥	٠	٤
e_2 , e_6	٥	٦	١	٣	١,٢١٢١	٧	٤
e_2 , e_7	٥	٤	١	٥	٢,٥	١,٤٠٧٥	٤,٥
e_2 , e_8	٥	٦	١	٤	٣,٣٩١٦	٤,٥	٤,٥
e_2 , e_9	٥	٥	١	٦	٠,٩٢٢٣	٢	٧
e_3 , e_4	٢	٢	٣	٦	٢	٣,١٧٥٩	٥
e_3 , e_5	٢	٣	٣	٤	٠	١,٠٥٠٨	٧
e_3 , e_6	٢	٦	٣	٣	٢	٠,٤١٧٩	٥
e_3 , e_7	٢	٤	٣	٥	٣,٧٠٠٩	٨	٤
e_3 , e_8	٢	٦	٣	٤	٠,٦٥٥٥	٢	٤
e_3 , e_9	٢	٥	٣	٦	٣,١٢٦٢	٠	٤
e_4 , e_5	٢	٣	٦	٤	٢,٥	١,٩٠٤٧	٥,٥
e_4 , e_6	٢	٦	٦	٣	٢,٥	٤,٥٩٦٨	٥,٥
e_4 , e_7	٢	٤	٦	٥	٠	٥	٥
e_4 , e_8	٢	٦	٦	٤	٠	٢	٤
e_4 , e_9	٢	٥	٦	٦	١,٣٤٦	٢	٦
e_5 , e_6	٣	٦	٤	٣	٢,٥١٦٢	٠	٦
e_5 , e_7	٣	٤	٤	٥	٠	٨	٤
e_5 , e_8	٣	٦	٤	٤	١,٦٠٠١	٠	٦
e_5 , e_9	٣	٥	٤	٦	٠	٠	٤
e_6 , e_7	٦	٤	٣	٥	٧	٨	٤
e_6 , e_8	٦	٦	٣	٤	٥,٠٧٣٩	٠	٦
e_6 , e_9	٦	٥	٣	٦	٧	٠	٤
e_7 , e_8	٤	٦	٥	٤	١,١٥٤٨	٠	٦
e_7 , e_9	٤	٥	٥	٦	٠	٢	٦
e_8 , e_9	٦	٥	٤	٦	٣,٩١٤٧	٢	٦

٦. Computational experiments

The computational experiments were done by solving two different networks; (Daskin, ٢٠١٣) with *p* ranged from ١ to ٥, (Handler and Mirchandani ١٩٧٩) with *p* ranged from ١ to ٤. All computational experiments were carried out on a processor Intel Corei٧, with CPU ١,٨٠GHz and ٨ GB of RAM, under Windows ١٠, ٦٤-bit. The code was written and executed in MATLAB.

Frist, (Daskin 2013) with p from 1 to 5

To examine the stability of our PSO; extensive experiments have been conducted, through performing independently ٢٠ runs for different values of *p* ranging from ١ to ٥ to compare with the results of (Daskin ٢٠١٣) and different values for *ss* (swarm size) with the following values: ٥, ١٠, ١٥, ٢٠, ٢٥, and ٣٠. (Daskin ٢٠١٣) applied a relaxation algorithm to find the optimal solution for the *p* centers with ٦ nodes and ١٠ edges; figure ٣. In Table ٣ results are summarized. The first column contains the swarm size. The second is for the number of centers *p*. The third column shows the optimal results from (Daskin ٢٠١٣). The number of reached optimal solution by our PSO out from ٢٠ runs is in column four. Column five and six show the average values of objective function acquired from all ٢٠ runs, and also their standard deviations. In column seven, PSO's average running times to achieve the final answer in each of the ٢٠ runs are listed. The most important conclusion to be drawn from the Table ٣ is that PSO was able to obtain optimal solutions in ٥٦٨ of ٦٠٠ runs (٩٥ %). It's important to note that in all of these situations, a swarm size of ٥ can be used to obtain the number of centers. The majority of bad results come from solving the problem with *p*=٢. Table ٣ also shows that PSO exhibits stable performances.

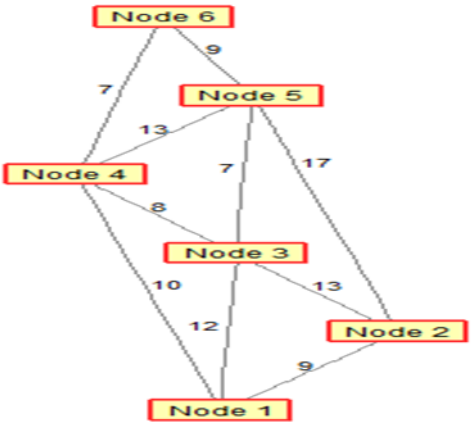

Figure ٥:: Example of a network from (Daskin ٢٠١٣)

Table ٣: The PSO stability evaluation for (Daskin ٢٠١٣) Network.

Swarm size (*ss*)	Centers (*p*)	Optimal (Daskin ٢٠١٣)	PSO Results			
			#Reached optimal	Average PSO	Σ	CPU Time (s)
٣٠	١	١٤,٠٠	٢٠	١٤,٠٠	٠,٠٠	٠,٣
	٢	٩,٠٠	١٧	٩,٠٨	٠,١٨	٠,٩
	٣	٤,٥٠	٢٠	٤,٥٠	٠,٠٠	٢,٩
	٤	٣,٥٠	٢٠	٣,٥٠	٠,٠٠	٥,٩
	٥	٣,٥٠	٢٠	٣,٥٠	٠,٠٠	٨,٢
٢٥	١	١٤,٠٠	٢٠	١٤,٠٠	٠,٠٠	٠,٣
	٢	٩,٠٠	١٧	٩,٠٨	٠,١٨	١,٧
	٣	٤,٥٠	٢٠	٤,٥٠	٠,٠٠	٥,٦
	٤	٣,٥٠	٢٠	٣,٥٠	٠,٠٠	١١,٠
	٥	٣,٥٠	٢٠	٣,٥٠	٠,٠٠	١٥,٣
٢٠	١	١٤,٠٠	٢٠	١٤,٠٠	٠,٠٠	٠,٢
	٢	٩,٠٠	١٩	٩,٠٣	٠,١١	١,٣
	٣	٤,٥٠	٢٠	٤,٥٠	٠,٠٠	٤,٣
	٤	٣,٥٠	٢٠	٣,٥٠	٠,٠٠	٨,٥
	٥	٣,٥٠	٢٠	٣,٥٠	٠,٠٠	١١,٧

Swarm size (*ss*)	Centers (*p*)	Optimal (Daskin ٢٠١٣)	PSO Results			
			#Reached optimal	Average PSO	Σ	CPU Time (s)
١٥	١	١٤,٠٠	٢٠	١٤,٠٠	٠,٠٠	٠,٢
	٢	٩,٠٠	**١٣**	٩,١٨	٠,٢٤	١,١
	٣	٤,٥٠	٢٠	٤,٥٠	٠,٠٠	٣,٥
	٤	٣,٥٠	٢٠	٣,٥٠	٠,٠٠	٧,١
	٥	٣,٥٠	٢٠	٣,٥٠	٠,٠٠	٩,٥
١٠	١	١٤,٠٠	٢٠	١٤,٠٠	٠,٠٠	٠,١
	٢	٩,٠٠	**١٢**	٩,٢٠	٠,٢٤	٠,٨
	٣	٤,٥٠	٢٠	٤,٥٠	٠,٠٠	٢,٥
	٤	٣,٥٠	٢٠	٣,٥٠	٠,٠٠	٥,٢
	٥	٣,٥٠	٢٠	٣,٥٠	٠,٠٠	٦,٣
٥	١	١٤,٠٠	٢٠	١٤,٠٠	٠,٠٠	٠,١
	٢	٩,٠٠	**١٣**	٩,١٥	٠,٢١	٠,٤
	٣	٤,٥٠	**١٧**	٤,٥١	٠,٠٢	١,٤
	٤	٣,٥٠	٢٠	٣,٥٠	٠,٠٠	٢,٨
	٥	٣,٥٠	٢٠	٣,٥٠	٠,٠٠	٣,٧

Figure ٦ shows that as the number of swarm size increased, the number of reached best known increased; however, the best swarm size is ٢٠ particles.

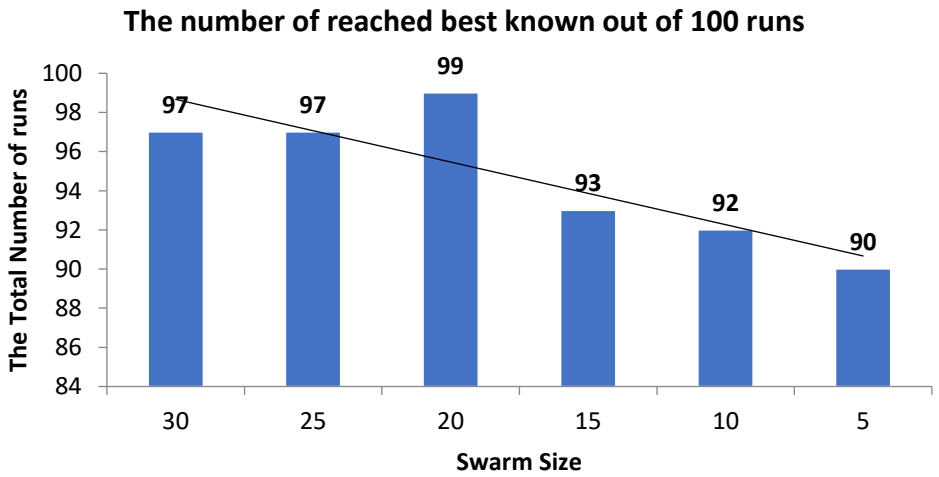

Figure ٦: The relationship between the size of the problem and the time it takes to solve it.

The average run time has a linear relationship with the problem size, as seen in Figure ٧. This linear relationship has been verified using ANOVA test and all the *p*-values are less than ٠,٠٥.

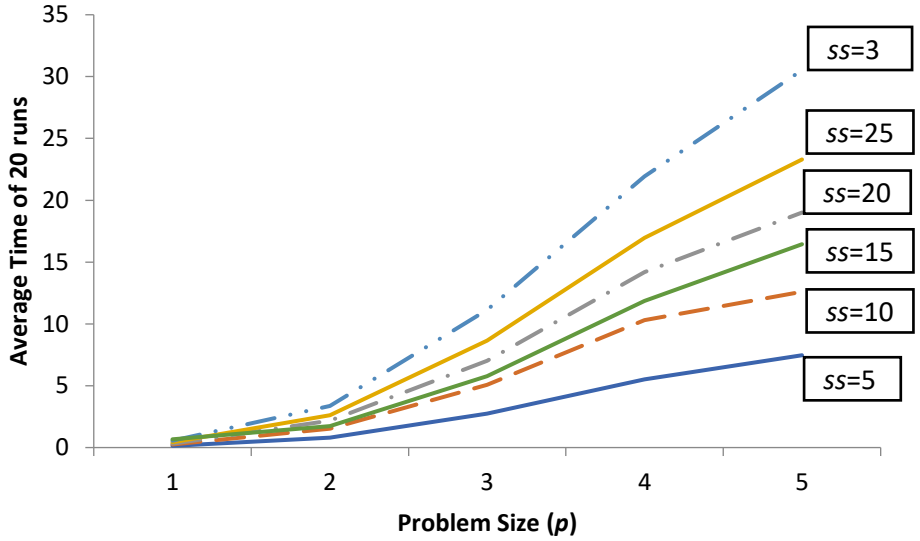

Figure ٧: The relationship between the problem size and the run time

The solution of (Daskin ٢٠١٣) network example for *p* values ranging from ١ to ٥ is presented as follows:

p=1	١st Node	٢nd Node	*Gpest*	*f*(*Gpest*)
١st center	٣	٤	١	١٤

p=2	١st Node	٢nd Node	*Gpest*	*f*(*Gpest*)
١st center	١	٤	٩	٩
٢nd center	٢	٥	٨,٠٢٥	

p=3	١st Node	٢nd Node	*Gpest*	*f*(*Gpest*)
١st center	١	٢	٤,٥	٤,٥
٢nd center	٥	٣	٢,٦٧	
٣rd center	٦	٤	٤,٤	

p=4	١st Node	٢nd Node	*Gpest*	*f*(*Gpest*)
١st center	١	٤	٠	٣,٥
٢nd center	٢	٥	٠	
٣rd center	٥	٣	٣,٥	
٤th center	٦	٤	٣,٥	

p=5	١st Node	٢nd Node	*Gpest*	*f*(*Gpest*)
١st center	٣	١	٠	٣,٥
٢nd center	١	٤	٠	
٣rd center	٢	٥	٣,٣٣	
٤th center	٦	٤	٣,٥٠	
٥th center	٥	٦	٠	

Second, (Handler and Mirchandani 1979) p=1 to 4

We also tested our PSO algorithm on larger network from (Handler and Mirchandani ١٩٧٩). The authors solved optimally multi-center location problem from *p*=١ to ٤, using a relaxation algorithm and applied the algorithm on a network with ٥٣ nodes with ٨١ edges for *p*=٤ (Figure ٨).

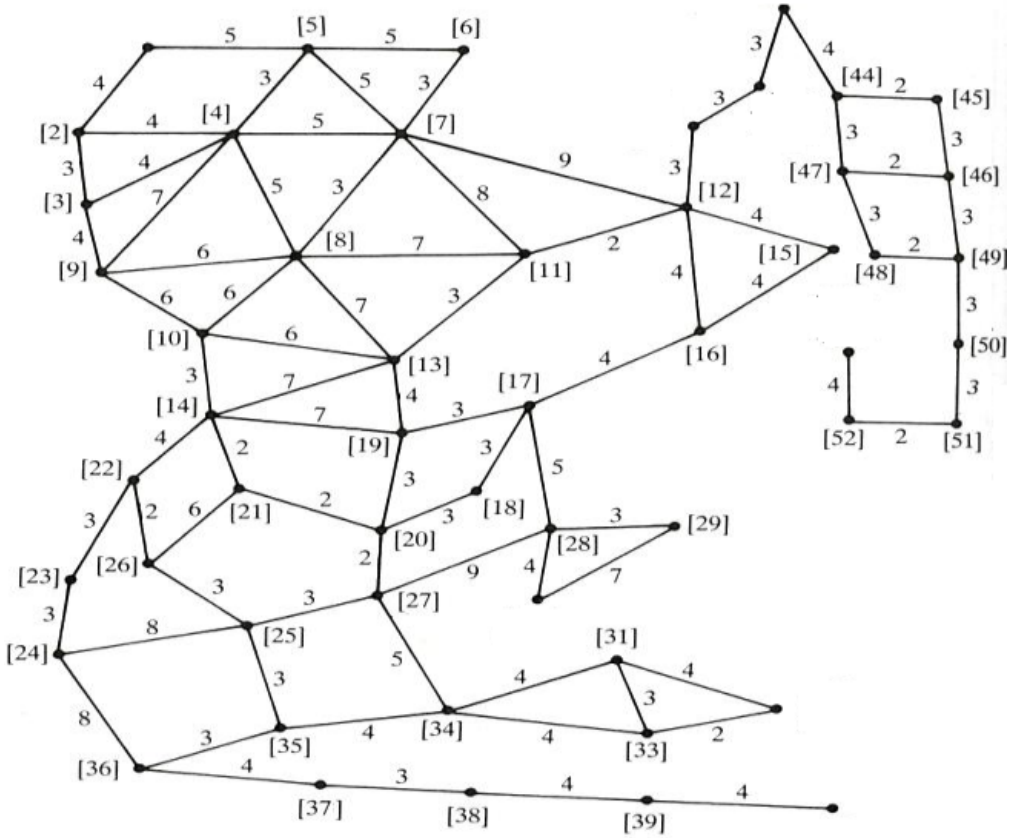

Figure ٨: Example of a network Handler (Handler and Mirchandani ١٩٧٩)

For all the bellow results, we run the implemented code of the PSO with the following parameters: *ss*=٢٠, *iterations*=٣٠ , c_1=c_2=٢, wt_{max} = ١,٢, and wt_{min} = ٠,١. The results show that PSO approach can find optimal solution for Handler network. These results match with the optimal solution obtained by Handler in (Handler and Mirchandani ١٩٧٩).

p=1	١st Node	٢nd Node	*Gpest*	*f*(*Gpest*)
١st center	١٣	١١	٢,٥	٣٥,٥

p=2	١st Node	٢nd Node	*Gpest*	*f*(*Gpest*)
١st center	٢٧	٢٠	١	٢٥
٢nd center	٥٣	٥٢	٣,١٥	

p=3	١st Node	٢nd Node	*Gpest*	*f*(*Gpest*)
١st center	٣٧	٣٨	٠	
٢nd center	١٣	٨	٢	١٨
٣rd center	٤٧	٤٦	٠	

p=4	١st Node	٢nd Node	*Gpest*	*f*(*Gpest*)
١st center	١٢	٧	٦,٥	
٢nd center	٢٨	٢٧	٨,١٥	١٢,٥
٣rd center	٤٩	٤٨	٠	
٤th center	٤٠	٣٦	٠	

٧. Conclusions

In this paper, a new particle swarm algorithm was develpoed to solve the multi-center location problem on networks. PSO effectively searches the search space for near-optimal and optimal solutions in a reasonable amount of time.

We have conducted an extensive empirical study. The computation results show that the propsed PSO approach despite of its simplicity and ease-of-use, it is an effective approach to find good feasible solution for the multi-center location problems.

The new proposed PSO algorithm which can solve *p*-center location problems, when all candidate solutions can be investigated. PSO was found to be stable and the average time of runs increases linearly as the number of centers increases. This algorithm has the advantage of obtaining the optimal solution for all the candidate solutions. The results demonstrate that the PSO method can provide good optimal solutions for multi-center locating problems on small networks.

ACKNOWLEDGMENT

I would like to acknowledge Prophet Muhammad (peace be upon him) who has taken all human being from the darkness to the lightness.

References

Ahmed, H. and J. Glasgow (٢٠١٢). Swarm Intelligence: Concepts, Models and Applications Technical Report ٢٠١٢-٥٨٥ School of Computing, Queen's University, Kingston, Ontario, Canada **K٧L٣N٦**

Brandeau, M. l. and S. s. Chiu (١٩٨٩). "An Overview of Representative Problems in Location Research." Management science **٣٥**(٦).

Christofides, N. (١٩٧٥). Graph Theory, an Algorithmic Approach, Academic Press, New York.

Christofides, N. and P. Viola (١٩٧١). "The Optimum Location of Multi-Centres on a Graph." Operational Research Quarterly **٢٢**(٢): ١٤٥-١٥٤.

Current, J., M. Daskin and D. Schilling (٢٠٠١). Facility location: applications and theory. In: Facility Location: Applications and Theory. Edited by Z. Drezner and H.W. Hamacher, Springer-Verlag.

Daskin, M. S. (١٩٩٥). Network and Discrete Location: Models, Algorithms and Applications, John Wiley and Sons, Inc., New York.

Daskin, M. S. (٢٠١٣). Network and Discrete Location: Models, Algorithms, John Wiley and Sons, Inc., New York.

Davidovica, T., D. Ramljak, M. Selmic and D. Teodorovic (٢٠١١). "Bee colony optimization for the p-center problem." Computers & Operations Research **٣٨**: ١٣٦٧–١٣٧٦.

Farahani, R. Z. and M. Hekmatfar (٢٠٠٩). Facility Location Concepts, Models, Algorithms and Case Studies, Springer-Verlag Berlin Heidelberg.

Hakimi, S. L. (١٩٦٤). "Optimum Locations of Switching Centers and the Absolute Centers and Medians of a Graph." Operations Research **١٢**(٣): ٤٥٠-٤٥٩.

Hakimi, S. L. and S. N. Maheshwari (١٩٧٢). "Optimum Locations of Centers in Networks." Operations Research **٢٠**(٥): ٩٦٧-٩٧٣.

Hakimi, S. L., E. F. Schmeichel and J. G. Pierce (١٩٧٨). "On *p*-centers in networks." Transportation science **١٢**: ١-١٥.

Handler, G. and P. Mirchandani (١٩٧٩). Location on networks: Theory and algorithms, MIT Press, Cambridge.

Kariv, O. and S. L. Hakimi (١٩٧٩). "An Algorithmic Approach to Network Location Problems. I: The p-Centers." SIAM Journal on Applied Mathematics ٣٧(٣): ٥١٣-٥٣٨.

Kariv, O. and S. L. Hakimi (١٩٧٩). "An algorithmic approach to network location problems. Part ١: *p*-centers." SIAM Journal of Applied Mathematics ٣٧(٣): ٥١٣-٥٣٨.

Kaveh, A. and H. Nasr (٢٠١١). "Solving the conditional and unconditional p-center problem with modified harmony search: A real case study." Scientia Iranica ١٨(٤): ٨٦٧–٨٧٧.

Kennedy, J. and R. Eberhart (١٩٩٥). "Particle Swarm Optimization." Proceedings of IEEE International Conference on Neutral Networks: ١٩٤٢-١٩٤٨.

Laporte, G., S. Nickel and F. S. d. Gama (٢٠١٥). Location Science, Springer

Laporte, G., S. Nickel and F. Saldanha da Gama (٢٠١٩). Location Science, Springer.

Minieka, E. (١٩٧٠). "The m-Center Problem." SIAM Review ١٢(١): ١٣٨-١٣٩.

Minieka, E. (١٩٨١). "A Polynomial Time Algorithm for Finding the Absolute Center of a Network." Networks ١١ ٣٥١-٣٥٥.

Rabie, H. M. (٢٠٢١). Particle Swarm Optimization and Grey Wolf Optimizer to Solve Continuous p-Median Location Problems. Machine Learning and Big Data Analytics Paradigms: Analysis, Applications and Challenges, Springer**:** ٤١٥-٤٣٥.

Rabie, H. M., I. El-Khodary and A. A. Tharwat (٢٠١٣). "Applying Particle Swarm Optimization for the Absolute p-center Problem." International Journal of Computer and Information Technology, http://ijcit.com/Vol٢Issue٥. php ٢: ١٠١٠-١٠١٥.

Rabie, H. M., I. A. El-Khodary and A. A. Tharwat (٢٠١٣). "A particle swarm optimization algorithm for the continuous absolute p-center location problem with Euclidean distance." International Journal of Advanced Computer Science and Applications (IJACSA) ٤(١٢): ١٠١-١٠٦.

ReVelle, C. S. and H. A. Eiselt (٢٠٠٥). "Location Analysis: A Synthesis and Survey." European journal of operational research ١٦٥: ١-١٩.

Sedighizadeh, D., E. Masehian, M. Sedighizadeh and H. Akbaripour (٢٠٢١). "GEPSO: A new generalized particle swarm optimization algorithm." Mathematics Computers in Simulation ١٧٩: ١٩٤-٢١٢.

Sforza, A. (١٩٩٠). "An algorithm for finding the absolute center of a network." European journal of operational research ٤٨: ٣٧٦-٣٩٠.

Shelokar, P. S., P. Siarry, V. K. Jayaraman and B. D. Kulkarni (٢٠٠٧). "Particle swarm and ant colony algorithms hybridized for improved continuous optimization." Applied Mathematics and Computation ١٨٨: ١٢٩–١٤٢.

Sumathi, S. and S. Paneerselvam (٢٠١٠). Computational Intelligence Paradigms: Theory & Applications using MATLAB, Taylor and Francis Group, LLC.

Yang, X.-S. (٢٠١٤). Nature-Inspired Optimization Algorithms, Elsevier