International Journal of Innovative Computing, Information and Control
Volume **8**, Number **12**, December **2012**
ICIC International ©2012 ISSN 1349-4198
pp. 8387–8398

PARTICLE SWARM OPTIMIZATION FOR LOCATION MOBILITY MANAGEMENT

Sung-Soo Kim[1,*], Gon Kim[2], Ji-Hwan Byeon[1] and Javid Taheri[3]

[1]Department of Industrial Engineering
[2]Division of Architecture
Kangwon National University
192-1, Hyoja 2 dong, Chunchon-si 200-701, Republic of Korea
[*]Corresponding author: kimss@kangwon.ac.kr; { gonkim; benjy86 }@kangwon.ac.kr

[3]School of Information Technology
The University of Sydney
Sydney, NSW 2006, Australia
javid.taheri@sydney.edu.au

Received September 2011; revised April 2012

Abstract. *In the generic mobile location problem for locating mobile terminals in a network, assignment of cells to either "reporting" or "non-reporting" cells is an NP-complete problem with known exponential complexity, also known as the reporting cell planning (RCP). The number of reporting cells as well as their locations must be carefully determined to balance the registration (location update) and search (paging) operations to minimize the cost of RCP. In this paper, we propose binary particle swarm optimization (BPSO) for optimal design of RCP. Our extensive set of experimental simulations demonstrates the effectiveness of BPSO; BPSO also proved to be a competitive approach in terms of quality of solution for the optimal design of several benchmark problems. Results also provide invaluable insights into the nature of this classical formidable problem and its effective solutions.*
Keywords: Mobile computing, Location management, Reporting cell planning (RCP), Binary particle swarm optimization (BPSO)

1. **Introduction.** Mobile communication systems pose hard challenges to planners involved in mobile terminal location tracking or location management. One of the problems faced by planners is that of providing a satisfactory level of QoS (Quality of Service) in a cost-effective manner. An important aspect of QoS is fast service access to customers. Fast service access involves keeping track of the location of users, while they are moving around the network without being involved in calls. In a mobile network, location databases are used to keep track of Mobile Terminals (MT) so that incoming calls can be directed to requested mobile terminals at all times. The commonly used off-line location management strategies [19] include always-update, never-update, time-based, movement-based, distance-based, paging cells, and location area (LA) schemes. In the LA scheme, a mobile network is partitioned into location areas (LAs). Each location area consists of a group of cells in the network. Mobile terminals (MTs) are then forced to update their location whenever they move from one location area (LA) to another. The system knows the current location areas (LAs) of all mobile terminals (MTs). Because of the partition of the network, when a call arrives (for a user), the search (paging) is restricted to only the cells within the location area (LA). Another approach used in location management is to designate each cell in the network as a *reporting cell* or a *non-reporting cell*. In this scheme, when a mobile terminal enters any of the *reporting cells* a location update

is performed. When a call arrives (for a user), the search is restricted to the last updated *reporting cell* and its neighboring *non-reporting cells*.

The important aspect of this location management in mobile computing networks in location area scheme is: what is the best/optimal topology of the location areas in the network, such that the network performance is best with minimum cost [18]. The two conflicting cost components considered in location management problem are: location update (registration) cost, and searching (paging cost). The total cost of the network is the sum of location update (registration) cost, and searching (paging cost). Minimizing only the location update cost usually increases the paging cost and paging delay. Minimizing only the paging cost usually results in greater cost of location updating. Hence, it is of great importance to balance these two conflicting cost components in location management. A good description of various issues involved in location management problem can be found in [13, 14].

Reporting Cell Planning (RCP): Another method used in location management is to designate each cell in the network as a *reporting cell* or a *non-reporting cell*. Determination of an optimal number of *reporting cells* (or reporting cell configuration) for a given network is reporting cell planning (RCP) problem. This problem (RCP) is a difficult combinatorial optimization problem and is shown to be NP-complete [2]. We can see that a cell in a network is either a *reporting cell* or a *non-reporting cell*. Hence, for a given network with N cells, the number of possible solutions is 2^N. Subrata and Zomaya [15, 16] used ant colony optimization, genetic algorithm, and taboo search methods, to find the best/optimal reporting cell configuration, for 4×4, 6×6 and 8×8 networks. Ant colony optimization method along with a selective paging (search) strategy is used to obtain the best/optimal set of *reporting cells* which is presented in [11], for 4×4, 6×6 and 8×8 networks. The use modified Hopfield network (MHN) to find the best reporting cell configuration for 4×4, 6×6, 7×9 and 9×11 networks, is given in Taheri and Zomaya [17]. Differential evolution is applied to obtain the best solution for reporting cell planning problem which is dealt in Almedia-Luz et al. [1], for 6×6, 8×8, 7×9 and 9×11 networks.

Contributions of this study: Particle Swarm Optimization (PSO) was originally developed to obtain optimal solution for continuous optimization problems [4, 9, 10]. This PSO was applied for power system and spam filtering [12]. Later, a binary particle swarm optimization (BPSO) was introduced to handle discrete optimization problems [8]. This BPSO can handle discrete binary variables and the relationship between PSO and BPSO is presented in [8]. This BPSO was also applied for the reliability optimization and clustering method [21]. As our contribution, in this paper we use the BPSO algorithm to find the best/optimal solution to the RCP problem. We present a comparison of the results obtained to RCP problem using BPSO with results of earlier studies. In earlier studies, differential evolution (DE), modified Hopfield neural network (HNN), simulated annealing (SA), genetic algorithm (GA), ant colony optimization (ACO) and taboo search (TS) methods were used for solution.

2. **Problem Formulation of RCP.**

The wireless mobile network consists of cells, usually represented in hexagonal shape. Thus, there are six possible neighbors for each cell. In the RCP location management method, some cells are designated as *reporting cells* to inform mobile terminals that they need to update their locations whenever they pass through them. Therefore, in the search process of locating users, they are only paged in their last updated reporting cell as well as its non-reporting neighbors without entering another reporting cell. Location management cost [15, 16] is mainly determined by weighted sum of the cost of two processes, i.e., location update cost and paging cost. The location management cost is [15, 16]

$$Total\ Cost = C \times N_{LU} + N_P \tag{1}$$

In the above equation, N_{LU} and N_P are the total number of location updates and the total number of paging transactions over a period of time, and C is a constant to represent the cost ratio of location update to paging. Because the cost of location update is usually much higher than the cost of paging, this constant is usually set 10, i.e., $C = 10$ [6, 22].

There are two weights associated with each cell i in the network: *movement weight* (w_{mi}) and *call arrival weight* (w_{ci}). The movement weight represents the frequency (or total number) of movement into a cell, whereas, the call arrival weight represents the frequency (or the total number) of call arrivals within a cell. If the cell i is a reporting cell, then the number of its location updates is directly related to its movement weight. The total number of paging (search) transaction is the product of the number *non-reporting cells* can be reached from a reporting cell i without entering any other reporting cell, with call arrival weights of each *non-reporting cell*. The total number of location updates (N_{LU}) and paging (N_P) for a given network can be calculated based on these two weights (w_{mi} and w_{ci}) as follows:

$$N_{LU} = \sum_{i \in S} w_{mi} \tag{2}$$

$$N_P = \sum_{j=0}^{N} w_{cj} \times v(j) \tag{3}$$

where N is the total number of cells in a network, S is the set of *reporting cells* in a network, N_{LU} is the total number of performed location updates, and N_P is the total number of paging transactions. We can see in Equation (3) that N_P is the product of w_{cj} and $v(j)$ for cell j. The vicinity value ($v(j)$) of cell j is defined as the maximum number of cells that can be searched, if an incoming call is received in this cell j. We will explain the vicinity value with an example in the next section. From Equations (2) and (3), we obtain location management cost, for a particular reporting cell configuration as

$$Total\ Cost = C \times \sum_{i \in S} w_{mi} + \sum_{j=0}^{N} w_{cj} \times v(j) \tag{4}$$

The cost per arrival is obtained by dividing the total cost by the total number of call arrivals and is

$$Cost\ per\ Call\ Arrival = \frac{Total\ Cost}{\sum_{j=0}^{N} w_{cj}} \tag{5}$$

Motivating Example: The *reporting cells* optimization problem (RCP) is that of finding an optimal set of *reporting cells* for a given network that minimizes the cost per call arrival. This optimization problem is considered and solved in earlier studies [6, 14, 15]. The vicinity value (v_i) of a reporting cell i is the maximum number of cells searched (paging), when a call arrives for a user whose last location is known to be cell i [14].

Example 2.1. *Consider a reporting cell configuration shown in Figure 1 for a* 4×4 *network. The cells are numbered from 0 to 15. In this figure, the shaded cells are reporting cells and the others are non-reporting cells. The values of movement weights* (w_{mi}) *and call arrival weights* (w_{cj}) *for all the cells in the network are given in Table 1.*

Table 1. Data set for 4×4 [15, 16]

Cell	w_{ci}	w_{mi}	Cell	w_{ci}	w_{mi}
0	517	518	8	251	445
1	573	774	9	224	2149
2	155	153	10	841	1658
3	307	1696	11	600	952
4	642	1617	12	25	307
5	951	472	13	540	385
6	526	650	14	695	1346
7	509	269	15	225	572

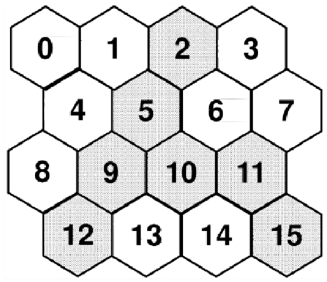


Figure 1. Configuration of 16 cells

In this reporting cell configuration cells numbered 2, 5, 9, 10, 11, 12 and 15 are *reporting cells* and cells numbered 0, 1, 3, 4, 6, 7, 8, 13 and 14 are *non-reporting cells.*

The vicinity value of the reporting cells (for the configuration given in Figure 1) is $v(2) = 8$, $v(5) = 8$, $v(9) = 7$, $v(10) = 6$, $v(11) = 6$, $v(12) = 7$ and $v(15) = 3$ [14]. The vicinity value of all non-reporting cells (for the configuration given in Figure 1) is $v(0) = 8$, $v(1) = 8$, $v(3) = 8$, $v(4) = 8$, $v(6) = 8$, $v(7) = 8$, $v(8) = 8$, $v(13) = 7$ and $v(14) = 7$ [14]. Using these vicinity values, the location management cost and cost per arrival for the reporting cell configuration given Figure 1 are obtained as 117,787 and 15.54 respectively. More details are available about this problem in earlier studies [6, 14, 15].

Let us consider a second configuration in which *all the cells* in the network are *reporting cells.* The location management cost and cost per arrival are 147,211 and 19.42 respectively.

Let us consider a third configuration in which *all the cells* in the network are *non-reporting cells.* The location management cost and cost per arrival are 121,296 and 16.00 respectively.

From the above three reporting cell configurations, we see that there is an optimal set of *reporting cells* for a given network that minimizes the cost per call arrival. Our interest is to find that optimal set of *reporting cells* for a given network.

3. **Binary Particle Swarm Optimization for RCP.** The original particle swarm optimization (PSO) methodology was used to find best/optimal solution for continuous optimization problems [4, 9, 10].

Particle Swarm Optimization method is a population based iterative algorithm. This starts with a population of solutions (particles), and at each time (iteration) step, the solutions are moved towards better solutions. The movement of particles in each iteration is based on its best solution that it has achieved so far (*pbest*) and overall best solution (*gbest*). The methodology of PSO for RCP problem has the following steps:

Step 1 Initialize a population of solutions (particles); each solution is a reporting cell configuration to the given RCP problem.

Step 2 For each particle (X_j in iteration i), evaluate the cost using Equation (4). This is represented as $X_j^i = j$ in our study.

Step 3 For each particle store $pbest_j^i$ and $gbest^i$ in this iteration. If $pbest_j^i$ is better than the $pbest_j$ obtain in earlier iterations then replace the $pbest_j^i$ as $pbest_j$. Similarly, if $gbest_j^i$ is better than the $gbest_j$ obtain in earlier iterations then replace the $gbest_j^i$ as $gbest_j$.

Step 4 Compute the velocity and change position of the particle according to Equations (6) and (7) given below.

Step 5 Increase the iteration i to $i+1$ and go to Step 2, until a termination criterion is met.

In each iteration, the velocity is computed as

$$v_j^{i+1} = w \times v_j^i + c_1 \times r_1 \times \left[pbest_j^i - X_j^i\right] + c_2 \times r_2 \times \left[gvest^i - X_j^i\right] \tag{6}$$

In each iteration, the particle position X_j is moved to X_j^{i+1} using

$$X_j^{i+1} = X_j^i + v_j^{i+1} \tag{7}$$

This algorithm has a maximum number of generations, number of particles and w, $c_1 \times r_1$ and $c_2 \times r_2$ are weighting factors.

Binary particle swarm optimization. In this section, we present the binary swarm optimization (BPSO) algorithm for solving the reporting cell planning (RCP) problem. We know that the RCP problem is a discrete optimization problem, and so we use discrete binary particle swarm optimization methodology given in [8]. A particle is represented in the solution space as a binary vector of size N. Here, N is the size of the network. In this binary vector, zero represents the cell to be a non-reporting cell, and one represents the cell to be a reporting cell. Hence, in this representation all the N cells are either reporting or non-reporting cells. In this manner, we can represent all the 2^N possible solutions to this problem. For example, the reporting cell configuration (solution) shown in Figure 1 is represented as shown in Table 2. The flowchart of BPSO for RCP problem is shown in Figure 2. For the RCP problem, the BPSO starts with a population of initial RCP solutions. Initial solutions can be generated randomly.

In our approach of BPSO, we first select the population of initial solutions, using the randomized algorithm given in [11]. All solutions are evaluated using objective function Equation (4). The best solution achieved so far by a given particle is *pbest*. The best value obtained so far by any particle in the neighborhood of that particle is *gbest*. The best solution is the one that has the minimum total cost. The basic concept of BPSO lies in moving each particle using its *pbest* and *gbest*, with a weighted acceleration using Equations (6) and (7) [8], in each iteration.

In BPSO methodology, the important aspect is the computation of velocity and the movement of a particle based on Equations (6) and (7). We will explain these with an example. We start our BPSO with 10 particles, $X_1^1, X_2^1, X_3^1, X_4^1, \ldots, X_{10}^1$ in generation 1.

Table 2. Solution representations for 16 cells configuration of Figure 1

Cell 0	Cell 1	Cell 2	Cell 3	Cell 4	Cell 5	Cell 6	Cell 7
0	0	1	0	0	1	0	0
Cell 8	Cell 9	Cell 10	Cell 11	Cell 12	Cell 13	Cell 14	Cell 15
0	1	1	1	1	0	0	1

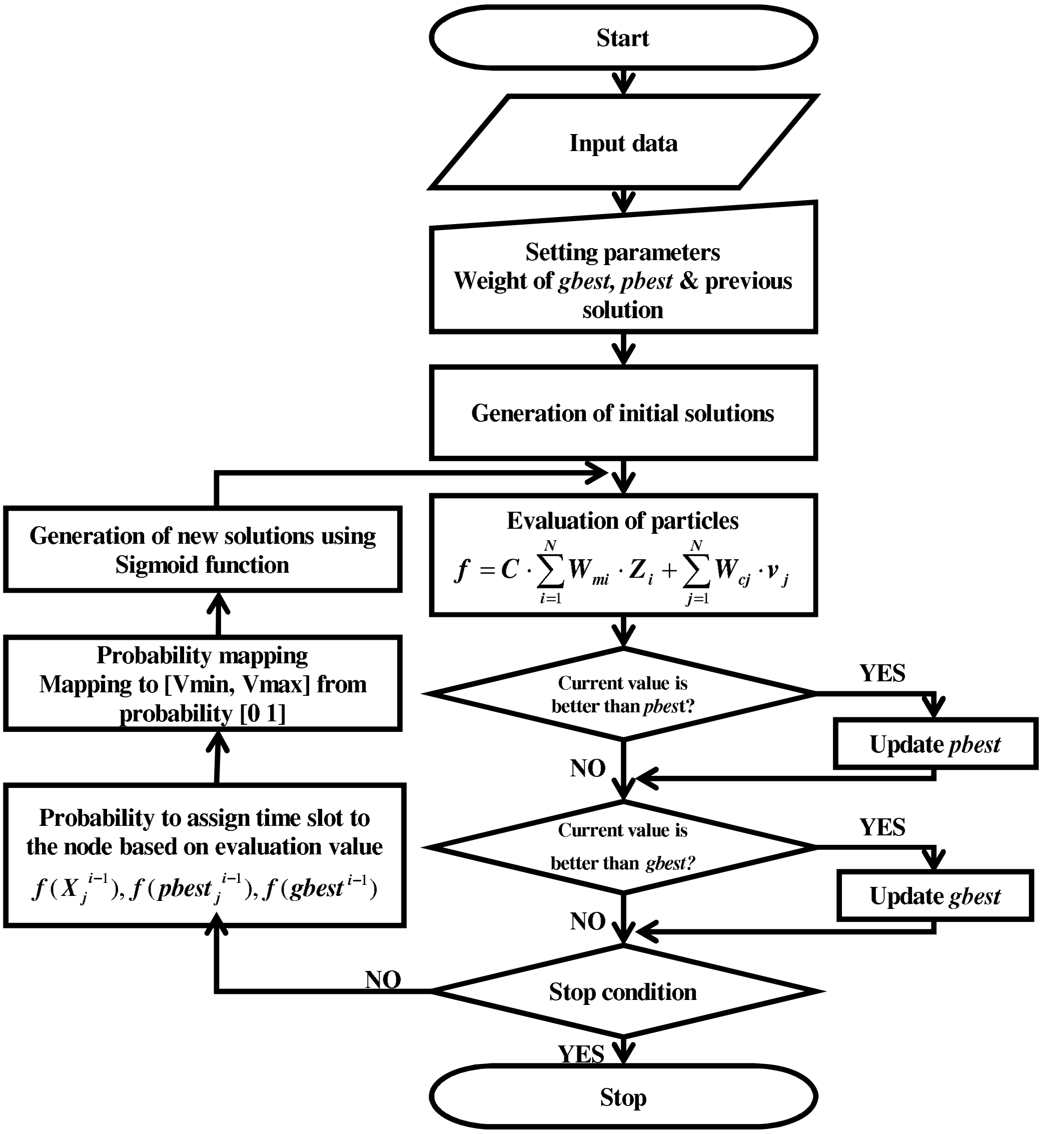


Figure 2. Binary particle swarm optimization for RCP

Each particle is a reporting cell configuration for the RCP problem. Also, each point has an objective function value given by the total cost of the configuration. For the particle j in iteration i the objective function value is denoted as $f(X_j^i)$. Let $f(X_1^1) = 20.7667$, $f(pbest_1^1) = 18.6522$ and $f(gbest^1) = 17.9732$ in the first iteration. This particle X_1^1 in iteration 1 will move to new point X_1^2 in generation 2, using Equations (6) and (7) as shown in Figure 3. If $w = 1.5$, $c_1 \times r_1 = 1$ and $c_2 \times r_2 = 0.7$, we obtain the following probabilities:

$$\frac{w \times \frac{1}{f(X_1^1)}}{w \times \frac{1}{f(X_1^1)} + c_1 \times r_1 \times \frac{1}{f(pbest_1^1)} + c_2 \times r_2 \times \frac{1}{f(gbest^1)}} = 0.4383$$

$$\frac{c_1 \times r_1 \times \frac{1}{f(pbest_1^1)}}{w \times \frac{1}{f(X_1^1)} + c_1 \times r_1 \times \frac{1}{f(pbest_1^1)} + c_2 \times r_2 \times \frac{1}{f(gbest^1)}} = 0.3253$$

$$\frac{c_2 \times r_2 \times \frac{1}{f(gbest^1)}}{w \times \frac{1}{f(X_1^1)} + c_1 \times r_1 \times \frac{1}{f(pbest_1^1)} + c_2 \times r_2 \times \frac{1}{f(gbest^1)}} = 0.2363$$

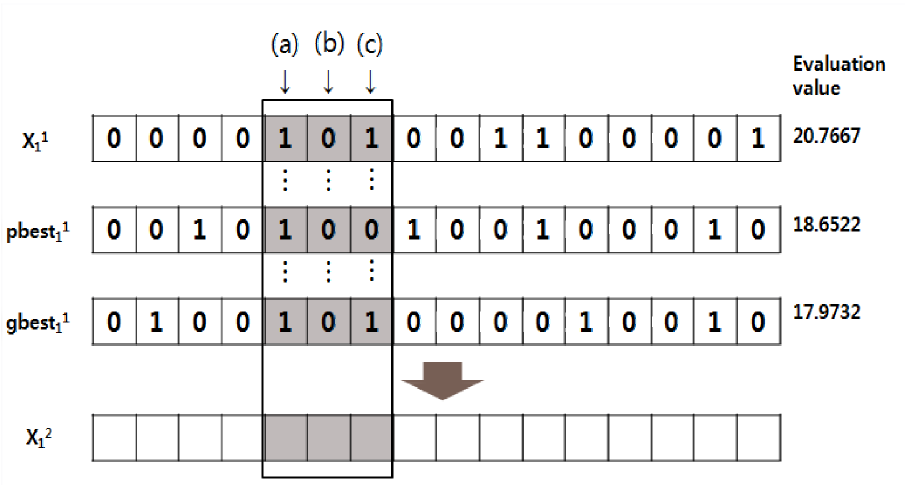


FIGURE 3. Particle representation of X_1^1, $pbest_1^1$, $gbest^1$ and generation of X_1^2

Consider the positions (a), (b) and (c) in Figure 3. In X_1^2 these positions to be a reporting cell with probabilities 1, 0 and 0.6747. These are obtained using X_1^1, $pbest_1^1$ and $gbest^1$. These probabilities 1, 0 and 0.6747 are mapping to $v_{\min} = -4$ and $v_{\max} = 4$, $-4 \leq v_i(t+1) \leq 4$. In an earlier study [20], it is mentioned that $v_{\max}$ is typically set to a value of 4.0, so that there is always at least a probability of 0.018 for any bit to change its state (8).

$$\begin{aligned} v_{(a)}(t+1) &= 1 \times 8 - 4 = 4 \\ v_{(b)}(t+1) &= 0 \times 8 - 4 = -4 \\ v_{(c)}(t+1) &= 0.6747 \times 8 - 4 = 1.3973 \end{aligned}$$

We can find the probability of assigning a reporting cell to positions (a), (b), (c) of new solution X_1^2 by using the Sigmoid function given in Equation (8), where $rand()$ generates a random number from the uniform distribution of $[0.0, 1.0]$.

$$S(v_i(t+1)) = \frac{1}{(1 + e^{(-v_i(t+1))})} \tag{8}$$

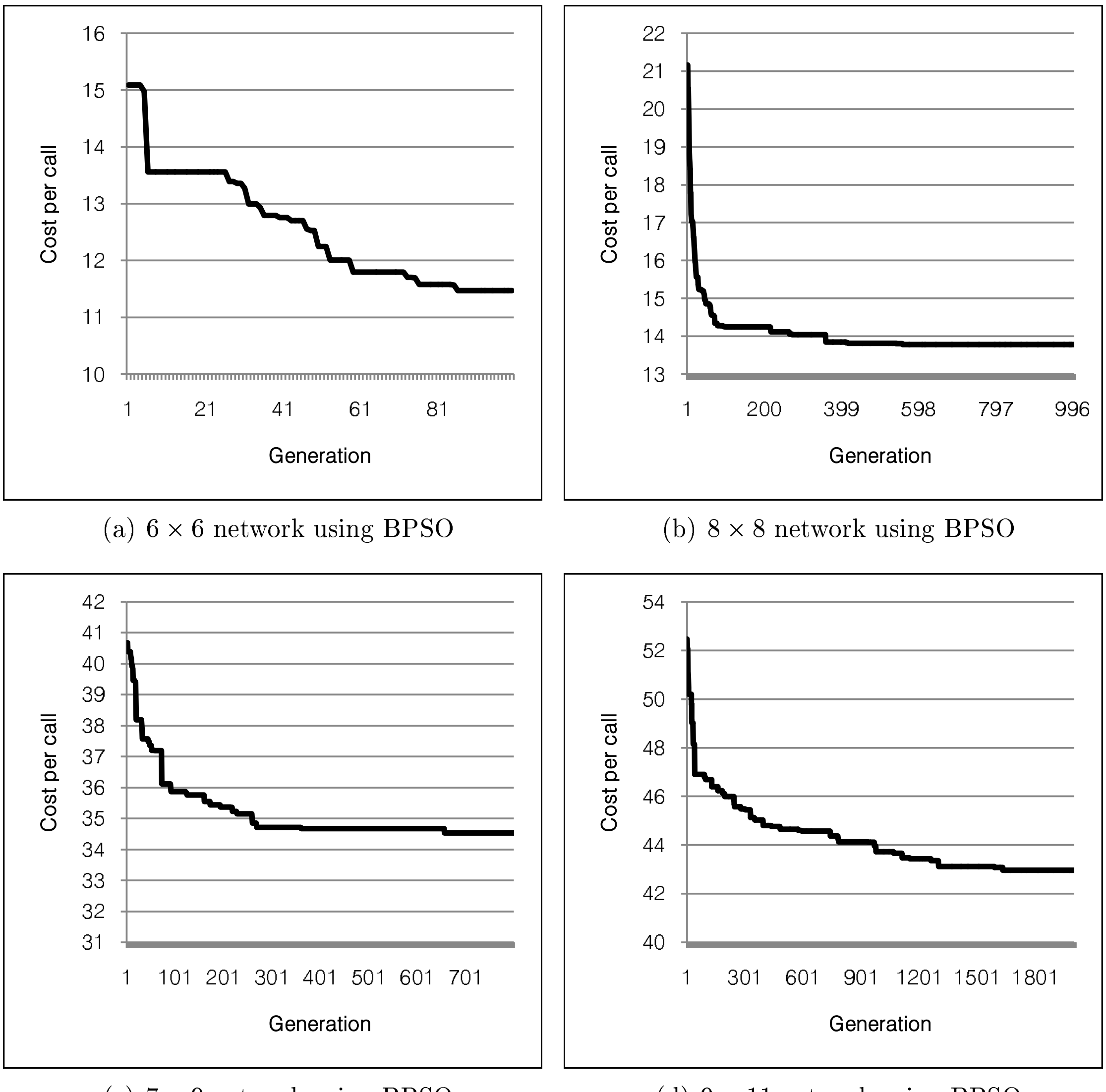
$$X_1^2 = \begin{cases} 0 & \text{if } rand() \geq S(v_i(t+1)) \\ 1 & \text{if } rand() < S(v_i(t+1)) \end{cases}$$

$$\begin{aligned} P_{((a))} &= S(v_{((a))}(t+1)) = \frac{1}{1 + e^{-4}} = 0.9820 \\ P_{((b))} &= S(v_{((b))}(t+1)) = \frac{1}{1 + e^{4}} = 0.0180 \\ P_{((c))} &= S(v_{((c))}(t+1)) = \frac{1}{1 + e^{-1.3973}} = 0.8018 \end{aligned}$$

Based on these calculations, position a (cell 5) of X_1^2 can be assigned as a reporting cell with probability 0.982. In the same way, position b (cell 6) of X_1^2 can be assigned as a reporting cell with probability 0.018. Position c (cell 7) of X_1^2 can be assigned as a reporting cell with probability 0.8018. A more detailed analysis of using the Sigmoid function is is given in [8].

4. **Simulation Results and Discussions.** In this section, we have used 6×6, 8×8, 7×9 and 9×11 networks as benchmark problems to show the performance of BPSO. The first two networks (6×6, 8×8) are given in [1, 15], and the other two (7×9 and 9×11) are used in [17]. The simulations were run on Intel(R) Core(TM) 2 Duo CPU (2.66GHz, 2G RAM). The best w, $c_1 \times r_1$ and $c_2 \times r_2$ are empirically set to 1.5, 1 and 0.7, respectively. The convergence of our BPSO method for the 4 different networks are shown in Figure 4. The best/optimal configurations obtained for 6×6, 8×8, 7×9 and 9×11 networks, using BPSO are shown in Figure 5. In this figure the reporting cells are shown in gray.

Discussion and analysis. We observe the following from the convergence of BPSO method to RCP problem. For all the four (6×6, 8×8, 7×9 and 9×11) networks, the convergence is fast in initial iterations and it is slow when the iterations increases. The reason is in the beginning BPSO starts with a non-optimal randomly generated initial solution with a high cost. After some iterations, the cost of each particles in the algorithm is close to each other and hence the convergence is slow in improving the results.



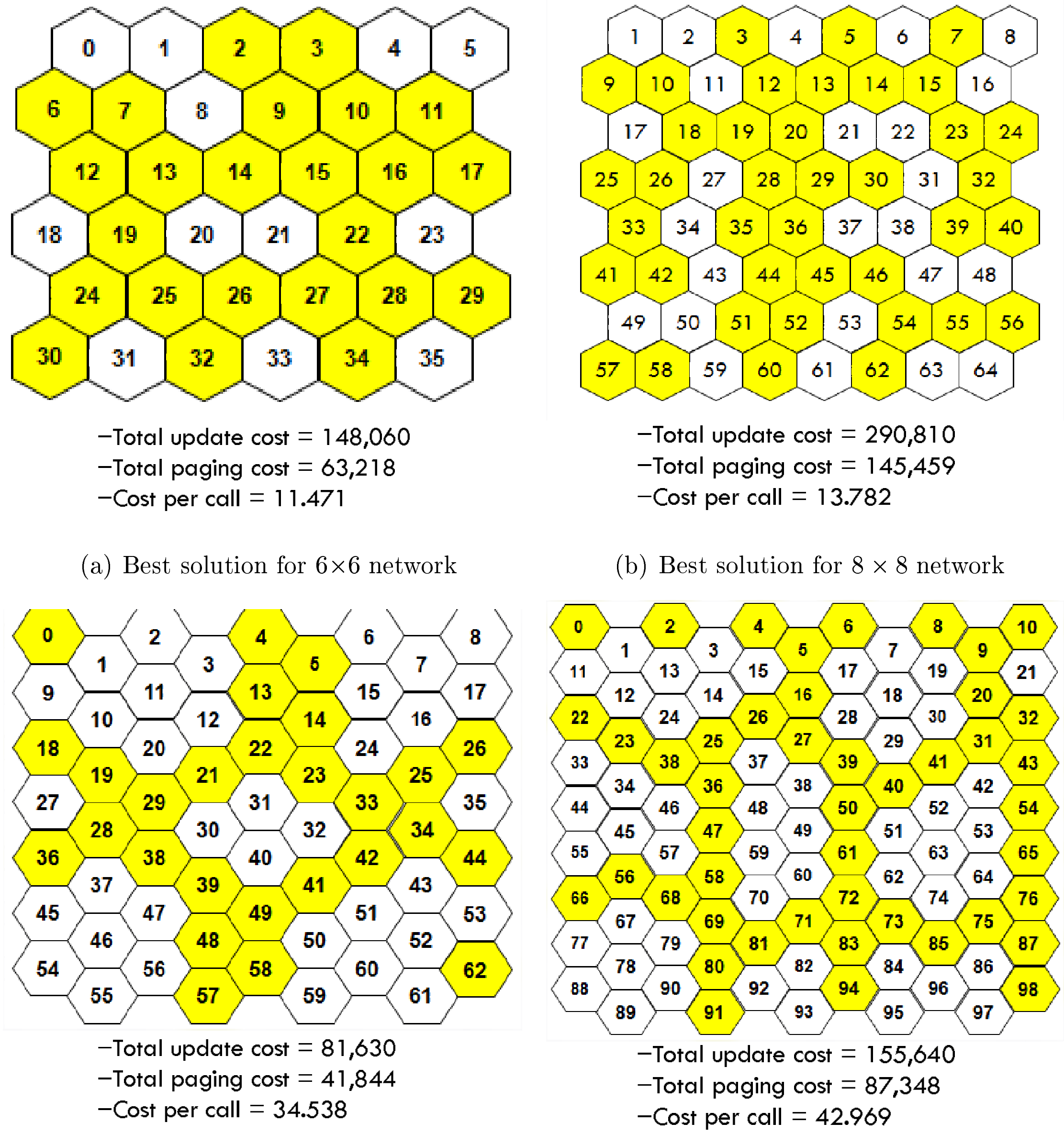
(a) 6×6 network using BPSO

(b) 8×8 network using BPSO

(c) 7×9 network using BPSO

(d) 9×11 network using BPSO

FIGURE 4. Trend of convergence for RCP benchmarking problems using



(a) Best solution for 6×6 network

(b) Best solution for 8 × 8 network

(c) Best solution for 7 × 9 network

(d) Best solution for 9 × 11 network

FIGURE 5. Configuration of best solutions for 4 different benchmarking RCP problems

The additional results we obtain for the four (6 × 6, 8 × 8, 7 × 9 and 9 × 11) networks are shown in Table 3. We notice from Table 3, that the ratios of reporting cells for 6 × 6, 8 × 8 networks are higher than those for 7 × 9 and 9 × 11 networks. The ratio of sum of call arrival to sum of movement weight in 6 × 6, 8 × 8 networks is higher than those of 7 × 9 and 9 × 11 networks.

In order to verify the robustness of BPSO algorithm for RCP problem, we conducted 100 simulation runs for each of 6 × 6, 8 × 8, 7 × 9 and 9 × 11 networks. From the results of these 100 runs, the minimum, maximum, average, standard deviation of the objective function value (cost per call arrivals) are shown in Table 4.

We compare the results of our BPSO with the earlier algorithms for the four (6×6, 8×8, 7×9 and 9×11) networks. In an earlier study [15], used the genetic algorithm (GA), taboo search (TS), and ant colony optimization (ACO) to solve the RCP problems. It is shown

Table 3. Ratio of reporting cells with arrival and movement weights for the best RCP solutions

Network	(a) Sum of cell arrival weight	(b) Sum of movement weight	(a)/(b)	Number of reporting cell	Ratio of reporting cells
6×6	18418	33192	0.555	24 out of 36	$24/36 = 0.667$
8×8	31656	64423	0.491	39 out of 64	$39/64 = 0.609$
7×9	3575	19765	0.181	27 out of 63	$27/63 = 0.429$
9×11	5655	40736	0.139	44 out of 99	$44/99 = 0.444$

Table 4. Ratio of reporting cells with arrival and movement weights for the best RCP solutions

Network	Minimum (best solution)	Maximum	Average	Standard deviation	Minimum found percentage
6×6	11.471	11.471	11.471	0	100% (100/100)
8×8	13.782	13.893	13.793	0.034	90% (90/100)
7×9	34.538	35.873	34.576	0.142	66% (66/100)
9×11	42.969	44.633	43.423	0.346	3% (3/100)

Table 5. Comparison of cost per call using BPSO and early studies

Network	BPSO	DE [1]	MHN [17]	ACO [15]
6×6	11.471	11.471	11.471	11.472(ACO), 11.471(TS), 12.464(GA)
8×8	13.782	13.782	N/A	13.801(ACO), 13.782(GA, TS)
7×9	34.538	33.819	34.538	N/A
9×11	42.969	43.140	43.044	N/A

in their study that TS performs best performance followed by ACO. The performance of GA is not comparable with TS and ACO. Other earlier studies [1, 17] also consider the RCP problem. Differential Evolution (DE) methodology has been used in [1] to obtain the optimal configuration of non-reporting cells for all the four networks. A modified Hopfield Network (MHN), is used for the RCP problem in [17].

A comparison of our results with the earlier results are presented in Table 5. From Table 5, we can see our results are also able to get the best results obtained in 6×6, 8×8 and 9×11 networks.

5. **Conclusions.** Reporting cell planning (RCP) for location management that arise in mobile computing networks is addressed in this paper. Finding the optimal set of reporting cells (or reporting cell configuration) for a given network is a known NP-complete combinatorial optimization problem. Hence, in this paper, we propose the binary particle swarm optimization (BPSO) to obtain the best/optimal set of reporting cells for 6×6, 7×9, 8×8 and 9×11 networks. From our simulation results, we see that the proposed BPSO is able to obtain the best/optimal reporting cell configuration of 6×6, 8×8 and

9×11 networks. Our results are compared with the results obtained using genetic algorithm (GA), ant colony optimization (ACO), taboo search (TS), differential evolution (DE) and modified Hopfield network (MHN). From the results obtained using BPSO, we conclude that this method can also be used to find best/optimal solution for RCP.

Acknowledgment. This research work was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education, Science and Technology (Nos. 2012-0000609). This research was supported by the Kangwon National University. The authors would like to thank Prof. V. Mani of Indian Institute of Science, Bangalore, India for some discussions and clarifications on this paper.

REFERENCES

[1] S. M. Almeida-Luz, M. A. Vega-Rodríguez, J. A. Gómez-Púlido and J. M. Sánchez-Pérez, Differential evolution for solving the mobile location management, *Applied Soft Computing*, vol.11, pp.410-427, 2011.

[2] A. Bar-Noy and I. Kessler, Tracking mobile users in wireless communication networks, *IEEE Transactions on Information Theory*, vol.39, no.6, pp.1877-86, 1993.

[3] M. F. Cátedra and J. Pérez, *Cell Planning for Wireless Communications*, Artech House, MA, USA, 1999.

[4] R. Eberhart and J. Kennedy, A new optimizer using particle swarm theory, *Proc. of the 6th International Symposium on Micro Machine and Human Science*, Nagoya, Japan, pp.39-43, 1995.

[5] R. C. Eberhart and Y. Shi, Particle swarm optimization: Development, applications and resources, *Proc. of 2001 Congr. Evolutionary Computation*, vol.1, pp.81-86, 2001.

[6] P. R. L. Gondim, Genetic algorithms and the location area partitioning problem in cellular networks, applications and resources, *Proc. of IEEE the 46th Vehicular Technology Conf.*, 1996.

[7] A. Hac and X. Zhou, Location strategies for personal communication networks: A novel tracking strategy, *IEEE Journal on Selected Areas in Communication*, vol.15, no.8, pp.1425-1436, 1997.

[8] J. Kennedy and R. C. Eberhart, A discrete binary version of the particle swarm algorithm, *IEEE International Conference on Systems, Man, and Cybernetics, 1997 Computational Cybernetics and Simulation*, 1997.

[9] J. Kennedy and R. Eberhart, Particle swarm optimization, *Proc. of IEEE International Conference on Neural Networks*, Perth, Australia, vol.4, pp.1942-1948, 1995.

[10] J. Kennedy, The particle swarm: Social adaptation of knowledge, *Proc. of IEEE International Conference on Neural Networks*, Indianapolis, Indiana, pp.303-308, 1997.

[11] S. S. Kim, I.-H. Kim, V. Mani and H. J. Kim, Ant colony optimization for reporting cell planning in mobile computing using selective paging strategy, *International Journal of Innovative Computing, Information and Control*, vol.5, no.6, pp.1587-1598, 2009.

[12] C.-C. Lai, C.-H. Wu and M.-C. Tsai, Feature selection using particle swarm optimization with application in spam filtering, *International Journal of Innovative Computing, Information and Control*, vol.5, no.2, pp.423-432, 2009.

[13] T. S. Rappaport, *Cellular Radio and Personal Communications: Selected Readings*, Piscataway, NJ, 1995.

[14] G. L. Stüber, *Principles of Mobile Communication*, Kluwer Academic, MA, 1996.

[15] R. Subrata and A. Y. Zomaya, A comparison of three artificial life techniques for reporting cell planning in mobile computing, *IEEE Transactions on Parallel and Distributed Systems*, vol.14, no.2, pp.142-153, 2003.

[16] R. Subrata and A. Y. Zomaya, Evolving cellular automata for location management in mobile computing networks, *IEEE Transactions on Parallel and Distributed Systems*, vol.14, no.6, pp.13-26, 2003.

[17] J. Taheri and A. Y. Zomaya, A modified Hopfield network for mobility management, *Wireless Communications and Mobile Computing*, vol.8, no.3, pp.355-367, 2008.

[18] J. Taheri and A. Y. Zomaya, A simulated annealing approach for mobile location management, *Computer Communications*, vol.30, no.4, pp.714-730, 2007.

[19] J. Taheri, A. Y. Zomaya and M. Iftikhar, Fuzzy online location management in mobile computing environments, *Journal of Parallel and Distributed Computing*, vol.71, no.8, pp.1142-1153, 2011.

[20] Y. del Valle, G. K. Venayagamoorthy, S. Mohagheghi, J. C. Hernandez and R. G. Harley, Particle swarm optimization: Basic concepts, variants and applications in power system, *IEEE Transactions on Evolutionary Computation*, vol.12, no.2, 2008.

[21] S. Wang and J. Watada, Reliability optimization of a series-parallel system with fuzzy random lifetimes, *International Journal of Innovative Computing, Information and Control*, vol.5, no.6, pp.1547-1558, 2009.

[22] H. Xie, S. Tabbane and D. J. Goodman, Dynamic location area management and performance analysis, *The 43rd IEEE Vehicular Technology Conference*, pp.536-539, 1993.