16

Finding Base-Station Locations in Two-Tiered Wireless Sensor Networks by Particle Swarm Optimization*

Tzung-Pei Hong[1, 2], Guo-Neng Shiu[3] and Yeong-Chyi Lee[4]
[1]Department of Computer Science and Information Engineering, National University of Kaohsiung
[2]Department of Computer Science and Engineering, National Sun Yat-sen University
[3]Department of Electrical Engineering, National University of Kaohsiung
[4]Department of Information Management, Cheng-Shiu University
Taiwan

1. Abstract

In wireless sensor networks, minimizing power consumption to prolong network lifetime is very crucial. In the past, Pan *et al.* proposed two algorithms to find the optimal locations of base stations in two-tiered wireless sensor networks. Their approaches assumed the initial energy and the energy-consumption parameters were the same for all application nodes. If any of the above parameters were not the same, their approaches could not work. Recently, the PSO technique has been widely used in finding nearly optimal solutions for optimization problems. In this paper, an algorithm based on particle swarm optimization (PSO) is thus proposed for general power-consumption constraints. The proposed approach can search for nearly optimal BS locations in heterogeneous sensor networks, where application nodes may own different data transmission rates, initial energies and parameter values. Experimental results also show the good performance of the proposed PSO approach and the effects of the parameters on the results. The proposed algorithm can thus help find good BS locations to reduce power consumption and maximize network lifetime in two-tiered wireless sensor networks.
Keywords: wireless sensor network, network lifetime, energy consumption, particle swarm optimization, base station.

2. Introduction

Recently, a two-tiered architecture of wireless sensor networks has been proposed and become popular [1]. It is motivated by the latest advances in distributed signal processing

---

* This is a modified and expanded version of the paper "A PSO heuristic algorithm for base-station locations," presented at The Joint Conference of the Third International Conference on Soft Computing and Intelligent Systems and the Seventh International Symposium on Advanced Intelligent Systems, 2006, Japan.

and source coding and can offer a more flexible balance among reliability, redundancy and scalability of wireless sensor networks. A two-tiered wireless sensor network, as shown in Figure 1, consists of sensor nodes (SNs), application nodes (ANs), and one or several base stations (BSs).

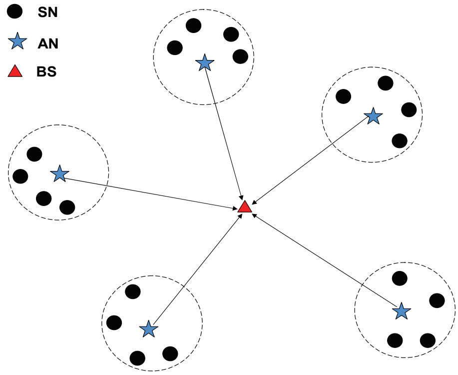


Figure 1. A two-tiered architecture of wireless sensor networks

Sensor nodes are usually small, low-cost and disposable, and do not communicate with other sensor nodes. They are usually deployed in clusters around interesting areas. Each cluster of sensor nodes is allocated with at least one application node. Application nodes possess longer-range transmission, higher-speed computation, and more energy than sensor nodes. The raw data obtained from sensor nodes are first transmitted to their corresponding application nodes. After receiving the raw data from all its sensor nodes, an application node conducts data fusion within each cluster. It then transmits the aggregated data directly to the base station or via multi-hop communication. The base station is usually assumed to have unlimited energy and powerful processing capability. It also serves as a gateway for wireless sensor networks to exchange data and information to other networks. Wireless sensor networks usually have some assumptions for SNs and ANs. For instance, each AN may be aware of its own location through receiving GPS signals [11] and its own energy.

In the past, many approaches were proposed to efficiently utilize energy in wireless networks. For example, appropriate transmission ways were designed to save energy for multi-hop communication in ad-hoc networks [16][10][5][19][7][6][20]. Good algorithms for allocation of base stations and sensors nodes were also proposed to reduce power consumption [12][15][16][8][9]. Thus, a fundamental problem in wireless sensor networks is to maximize the system lifetime under some given constraints. Pan *et al.* proposed two algorithms to find the optimal locations of base stations in two-tiered wireless sensor networks [13]. Their approaches assumed the initial energy and the energy-consumption parameters were the same for all ANs. If any of the above parameters were not the same, their approaches could not work.

In this paper, an algorithm based on particle swarm optimization (PSO) is proposed to find the base-station locations for general power-consumption constraints. The PSO technique was proposed by Eberhart and Kennedy in 1995 [2][3] and has been widely used in finding solutions for optimization problems. Some related researches about its improvement and applications has also been proposed [4][14][17][18]. It maintains several particles (each represents a solution) and all the particles continuously move in the search space according to their own local optima and the up-to-date global optimum. After a lot of generations, the optimal solution or an approximate optimal solution is expected to be found. The proposed approach here can search for nearly optimal BS locations in heterogeneous sensor networks. Experimental results also show the performance of the proposed PSO approach on finding the BS locations and the effects of the parameters on the results.

The remaining parts of this paper are organized as follows. Some related works about finding the locations of base stations in a two-tiered wireless networks is reviewed in Section 3. An algorithm based on PSO to discover base stations in a two-tiered wireless networks is proposed in Section 4. An example to illustrate the proposed algorithm is given in Section 5. Experimental results for demonstrating the performance of the algorithm and the effects of the parameters are described in Section 6. Conclusions are stated in Section 7.

3. Review of Related Works

As mentioned above, a fundamental problem in wireless sensor networks is to maximize the system lifetime under some given constraints. Pan *et al.* proposed two algorithms to find the optimal locations of base stations in two-tiered wireless sensor networks [13]. The first algorithm was used to find the optimal locations of base stations for homogenous ANs, and the second one was used for heterogeneous ANs. Homogenous ANs had the same data transmission rate and heterogeneous ANs might have different data transmission rates. In their paper, only the energy in ANs was considered. If a single SN ran out of energy, its corresponding AN might still have the capability to collect enough information. However, if an AN ran out of energy, the information in its coverage range would be completely lost, which was dangerous to the whole system.

Let d be the Euclidean distance from an AN to a BS, and r be the data transmission rate. Pan *et al.* adopted the following formula to calculate the energy consumption per unit time:

$$p(r, d) = r(\alpha_1 + \alpha_2 d^b) \ , \tag{1}$$

where a_1 is a distance-independent parameter, a_2 is a distance-dependent parameter, and b is the Euclidean dimension. The energy consumption thus relates to Euclidean distances and data transmission rates.

Pan *et al.* assumed each AN had the same a_1, a_2 and initial energy. For homogenous ANs, they showed that the center of the minimal circle covering all the ANs was the optimal BS location (with the maximum lifetime).

4. A General Base-Station Allocation Algorithm Based on PSO

The ANs produced by different manufacturers may own different data transmission rates, initial energies and parameter values. When different kinds of ANs exist in a wireless network, it is hard to find the optimal BS location. In this section, a heuristic algorithm based on PSO to search for optimal BS locations under general constraints is proposed. An initial set of particles is first randomly generated, with each particle representing a possible BS location. Each particle is also allocated an initial velocity for changing its state. Let $e_j(0)$ be the initial energy, r_j be the data transmission rate, a_{j1} be the distance-independent parameter, and a_{j2} be the distance-dependent parameter of the j-th AN. The lifetime l_{ij} of an application node AN_j for the i-th particle is calculated by the following formula:

$$l_{ij} = e_j(0) / r_j(\alpha_{j1} + \alpha_{j2} d_{ij}^b), \quad (2)$$

where d_{ij}^b is the b-order Euclidian distance from the j-th AN to the i-th particle. The fitness function used for evaluating each particle is thus shown below:

$$fitness(i) = \underset{j=1}{\overset{m}{Min}}\, l_{ij}, \quad (3)$$

where m is number of ANs. That is, each particle takes the minimal lifetime of all ANs as its fitness value. A larger fitness value denotes a longer lifetime of the whole system, meaning the corresponding BS location is better. The fitness value of each particle is then compared with that of its corresponding *pBest*. If the fitness value of the i-th particle is larger than that of $pBest_i$, $pBest_i$ is replaced with the i-th particle. The best $pBest_i$ among all the particles is chosen as the *gBest*. Besides, each particle has a velocity, which is used to change the current position. All particles thus continuously move in the search space. When the termination conditions are achieved, the final *gBest* will be output as the location of the base station. The proposed algorithm is stated below.

The proposed PSO algorithm for finding the best BS location:

- Input: A set of ANs, each AN_j with its location (x_j, y_j), data transmission rate r_j, initial energy $e_j(0)$, parameters a_{j1} and a_{j2}.
- Output: A BS location that will cause a nearly maximal lifetime in the whole system.
- Step 1: Initialize the fitness values of all *pBests* and the *gBest* to zero.
- Step 2: Randomly generate a group of n particles, each representing a possible BS location. Locations may be two-dimensional or three-dimensional, depending on the problems to be solved.
- Step 3: Randomly generate an initial velocity for each particle.
- Step 4: Calculate the lifetime l_{ij} of the j-th AN for the i-th particle by the following formula:

  $$l_{ij} = e_j(0) / r_j(\alpha_{j1} + \alpha_{j2} d_{ij}^b),$$

  where $e_j(0)$ is the initial energy, r_j is the data transmission rate, a_{j1} is a distance-independent parameter, a_{j2} is a distance-dependent parameter of the j-th AN, and d_{ij}^b is the b-order Euclidean distance from the i-th particle (BS) to the j-th AN.
- Step 5: Calculate the lifetime of the whole sensor system for the i-th particle as its fitness value ($fitness_i$) by the following formula:

  $$fitness(i) = \underset{j=1}{\overset{m}{Min}}\, l_{ij},$$

  where m is number of ANs and i = 1 to n.
- Step 6: Set $pBest_i$ as the current i-th particle if the value of *fitness(i)* is larger than the current fitness value of $pBest_i$.
- Step 7: Set *gBest* as the best *pBest* among all the particles. That is, let:

$$\text{fitness of } pBest_k = max_{i=1}^{n} \text{ fitness of } pBest_i,$$

and set $gBest=pBest_k$.

- Step 8: Update the velocity of the i-th particle as:

$$V_{id}^{new} = w \times V_{id}^{old} + c_1 \times Rand_1() \times (pBest_{id} - x_{id}) + c_2 \times Rand_2() \times (gBest_d - x_{id})$$

  where x_{id}^{new} is the new velocity of the i-th particle at the d-th dimension, x_{id}^{old} is the current velocity of the i-th particle at the d-th dimension, w is the inertial weight, c_1 is the acceleration constant for particles moving to $pBest$, c_2 is the acceleration constant for particles moving to $gBest$, $Rand_1()$ and $Rand_2()$ are two random numbers among 0 to 1, x_{id} is the current position of the i-th particle at the d-th dimension, $pBest_{id}$ is the value of $pBest_i$ at the d-th dimension, and $gBest_d$ is the value of $gBest$ at the d-th dimension.
- Step 9: Update the position of the i-th particle as:

$$x_{id}^{new} = x_{id}^{old} + V_{id}^{new},$$

  where x_{id}^{new} and x_{id}^{old} are respectively the new position and the current position of the i-th particle at the d-th dimension.
- Step 10: Repeat Steps 4 to 9 until the termination conditions are satisfied.

In Step 10, the termination conditions may be predefined execution time, a fixed number of generation or when the particles have converged to a certain threshold.

5. An Example

In this section, a simple example in a two-dimensional space is given to explain how the PSO approach can be used to find the best BS location that will generate the nearly maximal lifetime in the whole system. Assume there are totally four ANs in this example and their initial parameters are shown in Table 1, where "Location" represents the two-dimensional coordinate position of an AN, "Rate" represents the data transmission rate, and "Power" represents the initially allocated energy. All a_{j1}'s are set at 0 and all a_{j2}'s at 1 for simplicity.

AN	Location	Rate	Power
1	(1, 10)	5	10000
2	(11, 0)	5	10000
3	(8, 7)	4	6400
4	(4, 3)	4	6400

Table 1. The initial parameters of ANs in the example

For the example, the proposed PSO algorithm proceeds as follows.

- Step 1: The initial fitness values of all *pBests* and the *gBest* are set to zero.
- Step 2: A group of n particles are generated at random. Assume n is set at 3 in this example for simplicity. Also assume the three initial particles randomly generated are located at (4, 7), (9, 5) and (6, 4). Figure 2 shows the positions of the given ANs and the initial particles, where the triangles represent the particles and the circles represent the ANs.

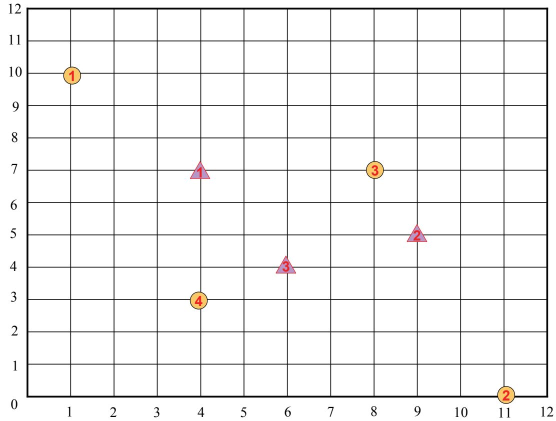


Figure 2. The positions of the given ANs and the initial particles

- Step 3: An initial velocity is randomly generated for each particle. In this example assume the initial velocity is set at zero for simplicity.
- Step 4: The lifetime of each AN for a particle is calculated. Take the first AN for the first particle as an example. Its lifetime is calculated as follows:

$$l_{11} = 10000/5[(4-1)^2 + (7-10)^2] = 111.11\,.$$

The lifetimes of all ANs for all particles are shown in Table 2.

Particle \ AN	1(1, 10)	2(11, 0)	3(8, 7)	4(4, 3)
1(4, 7)	111.11	20.41	100	100
2(9, 5)	22.47	68.97	320	55.17
3(6, 4)	32.79	48.78	123.08	320

Table 2. The lifetimes of all ANs for all particles

- Step 5: The lifetime of the whole sensor system for each particle is calculated as the fitness value. Take the first particle as an example. Its fitness is calculated as follows:

$$Fitness(1) = Min\{l_{11}, l_{12}, l_{13}, l_{14}\} = Min\{111.11, 20.41, 100, 100\} = 20.41.$$

In the same way, the fitness values of all the particles are calculated and shown in Table 3.

Particle	Location	Fitness
1	(4, 7)	20.41
2	(9, 5)	22.47
3	(6, 4)	32.79

Table 3. The fitness values of all the particles

- Step 6: The fitness value of each particle is compared with that of its corresponding *pBest*. If the fitness value of the *i*-th particle is larger than that of $pBest_i$, $pBest_i$ is replaced with the *i*-th particle. In the first generation, the fitness values of all the *pBest*s are zero, smaller than those of the particles. The particles are then stored as the new *pBest*s. The resulting *pBest*s are shown in Table 4, where the field "appearance generation" represents the generation number in which a particle is set as the current *pBest*.

Particle	Location	Fitness	Appearing Generation
1	(4, 7)	20.41	1
2	(9, 5)	22.47	1
3	(6, 4)	32.79	1

Table 4. The *pBest*s after the first generation

- Step 7: The best $pBest_i$ among all the particles is chosen as the *gBest.* In this example, $pBest_3$ has the largest fitness value and is set as the *gBest.*
- Step 8: The new velocity of each particle is updated. Assume the inertial weight w is set at 1, the acceleration constant c_1 for particles moving to *pBest* is set at 2, and the acceleration constant c_2 for particles moving to *gBest* is set at 2. Take the first particle as an example to illustrate the step. Its new velocity is calculated as follows:

$$\begin{aligned} V_{1x}^{new} &= w \times V_{1x}^{old} + c_1 \times Rand_1() \times (pBest_{1x} - x_{1x}) \\ &+ c_2 \times Rand_2() \times (gBest_d - x_{1x}) \\ &= 1 \times 0 + 2 \times 0.5(4-4) + 2 \times 0.25(6-4) \\ &= 1, \text{ and} \\ V_{1y}^{new} &= w \times V_{1y}^{old} + c_1 \times Rand_3() \times (pBest_{1y} - x_{1y}) \\ &+ c_2 \times Rand_4() \times (gBest_d - x_{1y}) \\ &= 1 \times 0 + 2 \times 1(7-7) + 2 \times 0.125(4-7) \\ &= -0.75, \end{aligned}$$

where the four random numbers generated are 0.5, 0.25, 1 and 0.125, respectively. In the same way, the new velocities of the other two particles can be calculated. The results are shown in Table 5.

Particle	Old Location	Velocity
1	(4, 7)	(1, -0.75)
2	(9, 5)	(-1.2, -0.2)
3	(6, 4)	(0, 0)

Table 5. The new velocities of all the three particles

- Step 9: The position of each particle is updated. Take the first particle as an example. Its new position is calculated as follows:

$$\begin{aligned} x_{1x}^{new} &= x_{1x}^{old} + V_{1x}^{new} \\ &= 4 + 1 \\ &= 5, \text{ and} \\ x_{1y}^{new} &= x_{1y}^{old} + V_{1y}^{new} \\ &= 7 + (-0.75) \\ &= 6.25. \end{aligned}$$

In the same way, the new positions of all the other two particles can be found. The results are shown in Table 6.

Particle	Old Location	Velocity	New Location
1	(4, 7)	(1, -0.75)	(5, 6.25)
2	(9, 5)	(-1.2, -0.2)	(7.8, 4.8)
3	(6, 4)	(0, 0)	(6, 4)

Table 6. The new positions of all the three particles

- Step 10: Steps 4 to 9 are then repeated until the termination conditions are satisfied. The lifetime evolution along with different generations is shown in Figure 3.

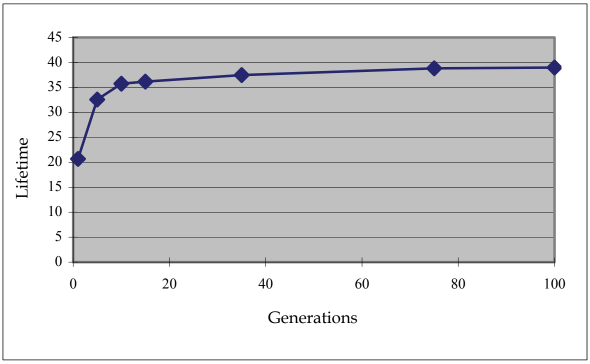


Figure 3. The evolution of the maximal lifetime for the example

6. Experimental Results

Experiments were made to show the performance of the proposed PSO algorithm on finding the optimal positions of base stations. They were performed in C language on an AMD PC with a 2.0GHz processor and 1G main memory and running the Microsoft Window XP operating system. The simulation was done in a two-dimensional real-number space of 1000*1000. That is, the ranges for both x and y axes were within 0 to 1000. The data transmission rate was limited within 1 to 10 and the range of initial energy was limited between 100000000 to 999999999. The data of all ANs, each with its own location, data transmission rate and initial energy, were randomly generated. Note that the data transmission rates and the initial energy amounts of real-life sensors may not fall in the above range. But the proposed approach is still suitable since the lifetime is proportional to the initial energy amount and inversely proportional to the transmission rate.

Experiments were first made to show the convergence of the proposed PSO algorithm when the acceleration constant (c_1) for a particle moving to its *pBest* was set at 2, the acceleration constant (c_2) for a particle moving to its *gBest* was set at 2, the inertial weight (w) was set at 0.6, the distance-independent parameter (a_{j1}) was set at zero, and the distance-dependent parameter (a_{j2}) was set at one. The experimental results of the resulting lifetime along with different generations for 50 ANs and 10 particles in each generation are shown in Figure 4.

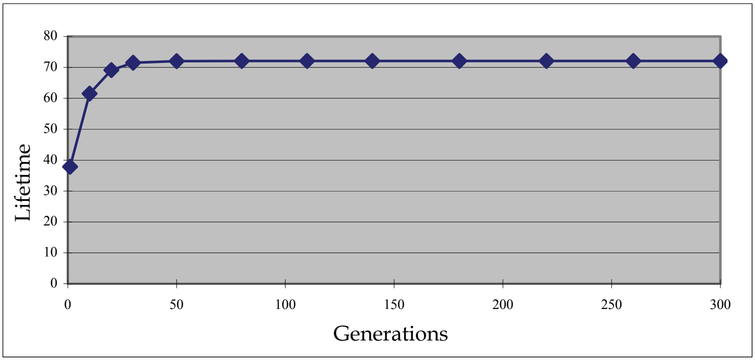


Figure 4. The lifetime for 50 ANs and10 particles

It is easily seen from Figure 4 that the proposed PSO algorithm could converge very fast (below 50 generations). Next, experiments were made to show the effects of different parameters on the lifetime. The influence of the acceleration constant (c_1) for a particle moving to its *pBest* on the proposed algorithm was first considered. The process was terminated at 300 generations. When w = 1 and c_2 = 2, the nearly optimal lifetimes for 50ANs and 10 particles along with different acceleration constants (c_1) are shown in Figure 5.

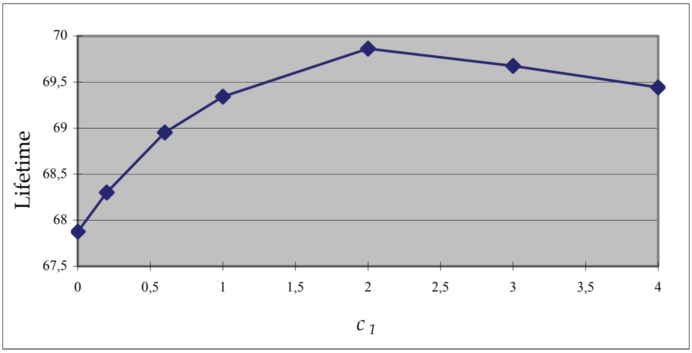


Figure 5. The lifetimes along with different acceleration constants (c_1)

It can be observed from Figure 5 that the lifetime first increased and then decreased along with the increase of the acceleration constant (c_1). When the value of the acceleration constant (c_1) was small, the velocity update of each particle was also small, causing the convergence speed slow. The proposed PSO algorithm might thus not get the optimal solution after the predefined number of generations. On the contrary, when the value of the acceleration constant (c_1) was large, the velocity change would be large as well, causing the particles to move fast. It was then hard to converge. In the experiments, the optimal c_1 value was about 2. Next, experiments were made to show the effects of the acceleration constant (c_2) for a particle moving to its *gBest* on the proposed algorithm. When w = 1 and c_1 = 2, the experimental results are shown in Figure 6.

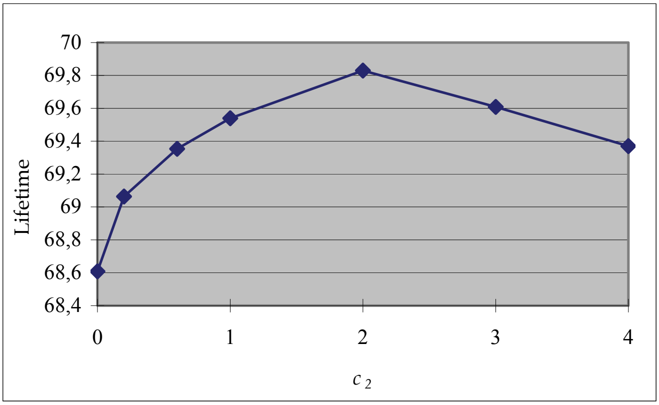


Figure 6. The lifetimes along with different acceleration constants (c_2)

It can be observed from Figure 6 that the lifetime first increased and then decreased along with the increase of the acceleration constant (c_2). The reason was the same as above. In the experiments, the optimal c_2 value was about 2. Next, experiments were made to show the effects of the inertial weight (w) on the proposed algorithm. When c_1 = 2 and c_2 = 2, the experimental results are shown in Figure 7.

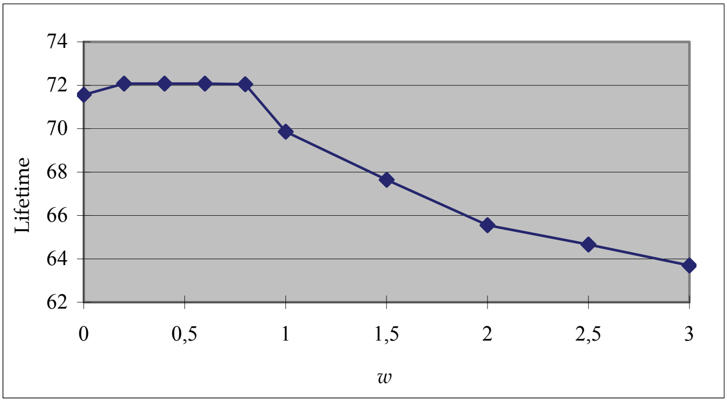


Figure 7. The lifetimes along with different inertial weights (w)

It can be observed from Figure 7 that the proposed algorithm could get good lifetime when the inertial weight (w) was smaller than 0.6. The lifetime decreased along with the increase of the inertial weight (w) when w was bigger than 0.6. This was because when the value of the inertial weight was large, the particles would move fast due to the multiple of the old velocity. It was then hard to converge. Next, experiments were made to show the relation between lifetimes and numbers of ANs. The experimental results are shown in Figure 8.

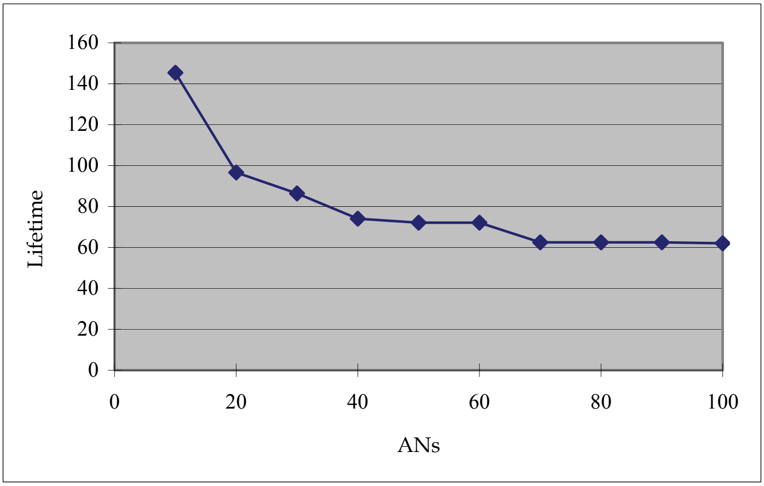


Figure 8. The lifetimes along with different numbers of ANs

It can be seen from Figure 8 that the lifetime decreased along with the increase of the number of ANs. It was reasonable since the probability for at least an AN in the system to fail would increase when the number of ANS grew up. The execution time along with different numbers of ANs is shown in Figure 9.

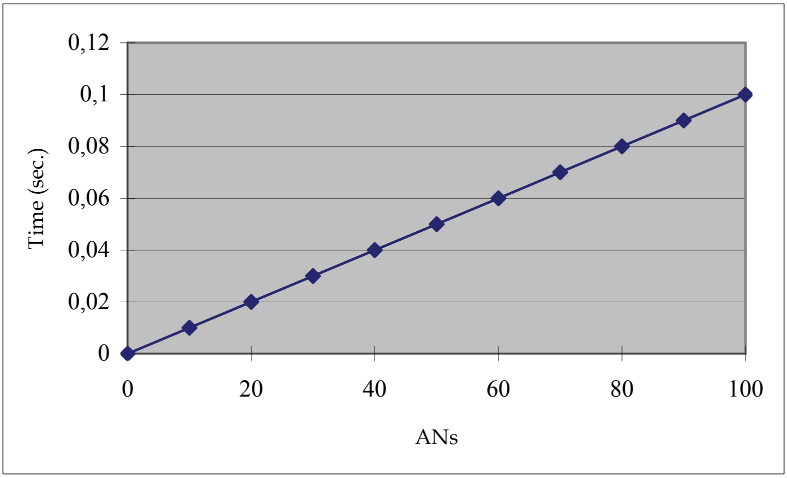


Figure 9. The execution time along with different numbers of ANs

It can be observed from Figure 9 that the execution time increased along with the increase of numbers of ANs. The relation was nearly linear. Experiments were then made to show the relation between lifetimes and numbers of particles for 50 ANs and 300 generations. The internal weight was set at 1. The experimental results are shown in Figure 10.

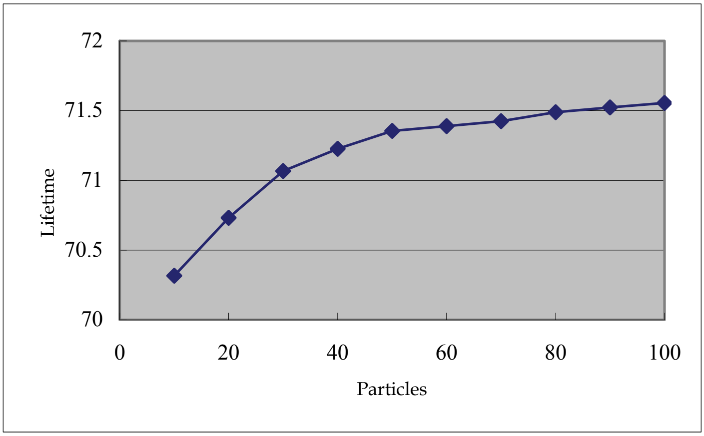


Figure 10. The lifetimes along with different numbers of particles

It can be seen from Figure 10 that the lifetime increased along with the increase of numbers of particles for the same number of generations. The execution time along with different numbers of particles for 300 generations is shown in Figure 11.

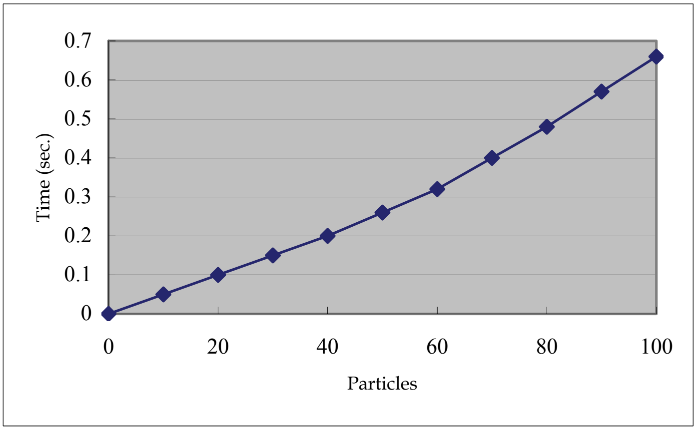


Figure 11. The execution time along with different numbers of particles for 50 ANs

Method	Lifetime
The proposed PSO algorithm	72.0763
The exhaustive grid search (grid size = 1)	72.0048
The exhaustive grid search (grid size = 0.1)	72.0666
The exhaustive grid search (grid size = 0.01)	72.0752

Table 7. A lifetime comparison of the PSO approach and the exhaustive grid searc

It can be observed from Figure 11 that the execution time increased along with the increase of numbers of particles. The relation was nearly linear. This was reasonable since the execution time would be approximately proportional to the number of particles.

Note that no optimal solutions can be found in a finite amount of time since the problem is NP-hard. For a comparison, an exhaustive search using grids was used to find nearly optimal solutions. The approach found the lifetime of the system when a BS was allocated at any cross-point of the grids. The cross-point with the maximum lifetime was then output as the solution. A lifetime comparison of the PSO approach and the exhaustive search with different grid sizes are shown in Table 7.

It can be observed from Table 7 that the lifetime obtained by our proposed PSO algorithm was not worse than those by the exhaustive grid search within a certain precision. The lifetime by the proposed PSO algorithm was 72.0763, and was 72.0048, 72.0666 and 72.0752 for the exhaustive search when the grid size was set at 1, 0.1 and 0.01, respectively. For the exhaustive grid search, the smaller the grid size, the better the results.

7. Conclusion

In wireless sensor networks, minimizing power consumption to prolong network lifetime is very crucial. In this paper, a two-tiered wireless sensor networks has been considered and an algorithm based on particle swarm optimization (PSO) has been proposed for general power-consumption constraints. The proposed approach can search for nearly optimal BS locations in heterogeneous sensor networks, where ANs may own different data transmission rates, initial energies and parameter values. Finding BS locations is by nature very similar to finding the food locations originated from PSO. It is thus very easy to model such a problem by the proposed algorithm based on PSO. Experiments have also been made to show the performance of the proposed PSO approach and the effects of the parameters on the results. From the experimental results, it can be easily concluded that the proposed PSO algorithm converges very fast when compared to the exhaustive search. It can also be easily extended to finding multiple base stations.

8. Acknowledgement

This research was supported by the National Science Council of the Republic of China under contract NSC 97-2221-E-390-023. We also like to thank Mr. Cheng-Hsi Wu for his help in making part of the experiments.

9. References

J. Chou, D. Petrovis and K. Ramchandran, A distributed and adaptive signal processing approach to reducing energy consumption in sensor networks, *The 22nd IEEE Conference on Computer Communications (INFOCOM)*, pp. 1054-1062, San Francisco, USA, March, 2003. [1]

R. C. Eberhart and J. Kennedy, A new optimizer using particles swarm theory, *The Sixth International Symposium on Micro Machine and Human Science*, pp. 39-43, Nagoya, Japan, October, 1995. [2]

R. C. Eberhart and J. Kennedy, Particle swarm optimization, *The IEEE International Conference on Neural Networks*, Vol. 4, pp. 1942-1948, Perth, Australia, November, 1995. [3]

Z. L. Gaing, Discrete particle swarm optimization algorithm for unit commitment, *The IEEE Power Engineering Society General Meeting*, Toronto, Canada, July, 2003. [4]

W. Heinzelman, J. Kulik, and H. Balakrishnan, Adaptive protocols for information dissemination in wireless sensor networks, *The Fifth ACM International Conference on Mobile Computing and Networking*, pp. 174-185, Seattle, USA, August, 1999. [5]

W. Heinzelman, A. Chandrakasan, and H. Balakrishnan, Energy-efficient communication protocols for wireless microsensor networks, *The Hawaiian International Conference on Systems Science,* Hawaii, USA, January, 2000. [6]

C. Intanagonwiwat, R. Govindan and D. Estrin, Directed diffusion: a scalable and robust communication paradigm for sensor networks, *The ACM International Conference on Mobile Computing and Networking*, Boston, USA, August, 2000. [7]

Vikas Kawadia and P. R. Kumar, Power control and clustering in ad hoc networks, *The 22nd IEEE Conference on Computer Communications (INFOCOM),* San Francisco, USA, March, 2003. [8]

N. Li, J. C. Hou and L. Sha, Design and analysis of an mst-based topology control algorithm, *The 22nd IEEE Conference on Computer Communications (INFOCOM)*, San Francisco, USA, March, 2003. [9]

S. Lee, W. Su and M. Gerla, Wireless ad hoc multicast routing with mobility prediction, *Mobile Networks and Applications*, Vol. 6, No. 4, pp. 351-360, 2001. [10]

D. Niculescu and B. Nath, Ad hoc positioning system (APS) using AoA, *The 22nd IEEE Conference on Computer Communications (INFOCOM)*, pp. 1734-1743, San Francisco, USA, March, 2003. [11]

J. Pan, Y. Hou, L. Cai, Y. Shi and X. Shen, Topology control for wireless sensor networks, *The Ninth ACM International Conference on Mobile Computing and Networking*, pp. 286-299, San Diego, USA, September, 2003. [12]

J. Pan, L. Cai, Y. T. Hou, Y. Shi and S. X. Shen, Optimal base-station locations in two-tiered wireless sensor networks, *IEEE Transactions on Mobile Computing*, Vol. 4, No. 5, pp. 458-473, 2005. [13]

Y. Q. Qin, D. B. Sun, N. Li and Y. G. Cen, Path planning for mobile robot using the particle swarm optimization with mutation operator, *The IEEE Third International Conference on Machine Learning and Cybernetics*, pp. 2473-2478, Shanghai, China, August, 2004. [14]

V. Rodoplu and T. H. Meng, Minimum energy mobile wireless networks, *IEEE Journal on Selected Areas in Communications,* Vol. 17, No. 8, 1999. [15]

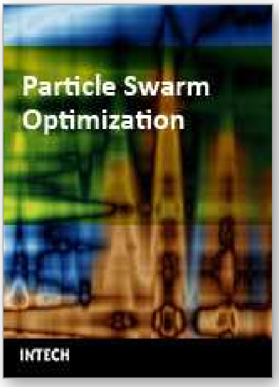
R. Ramanathan and R. Hain, Topology control of multihop wireless networks using transmit power adjustment, *The 19th IEEE Conference on Computer Communications (INFOCOM)*, Tel Aviv, Israel, March, 2000. [16]

Y. Shi and R. C. Eberhart, A modified particle swarm optimizer, *The IEEE International Conference on Evolutionary Computation*, pp. 69-73, Anchorage, USA, May, 1998, [17]

A. Stacey, M. Jancic and I. Grundy, Particle swarm optimization with mutation, *The IEEE Congress on Evolutionary Computation*, pp. 1425-1430, Canberra, Australia, December, 2003. [18]

D. Tian and N. Georganas, Energy efficient routing with guaranteed delivery in wireless sensor networks, *The IEEE Wireless Communication and Networking Conference,* Orleans, USA, March, 2003. [19]

F. Ye, H. Luo, J. Cheng, S. Lu and L. Zhang, A two-tier data dissemination model for large scale wireless sensor networks, *The ACM International Conference on Mobile Computing and Networking,* Atlanta, USA, September, 2002. [20]



Particle Swarm Optimization

Edited by Aleksandar Lazinica

ISBN 978-953-7619-48-0
Hard cover, 476 pages
Publisher InTech
Published online 01, January, 2009
Published in print edition January, 2009

Particle swarm optimization (PSO) is a population based stochastic optimization technique influenced by the social behavior of bird flocking or fish schooling.PSO shares many similarities with evolutionary computation techniques such as Genetic Algorithms (GA). The system is initialized with a population of random solutions and searches for optima by updating generations. However, unlike GA, PSO has no evolution operators such as crossover and mutation. In PSO, the potential solutions, called particles, fly through the problem space by following the current optimum particles. This book represents the contributions of the top researchers in this field and will serve as a valuable tool for professionals in this interdisciplinary field.

InTech Europe
University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China
Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821