

A Clustering Scheme for Wireless Sensor Networks Based on Genetic Algorithm and Dominating Set

Jeng-Shyang Pan^{1,2,3}, Lingping Kong², Tien-Wen Sung¹, Pei-Wei Tsai¹, Václav Snášel⁴

¹ Fujian Provincial Key Laboratory of Big Data Mining and Applications, College of Information Science and Engineering, Fujian University of Technology, China

² Innovative Information Industry Research Center, Harbin Institute of Technology Shenzhen Graduate School, China

³ Department of Information Management, Chaoyang University of Technology, Taiwan

⁴ Department of Computer Science, VSB Tech. University of Ostrava, Czech Republic

jengshyangpan@fjut.edu.cn, konglingping2007@163.com, twsung@foxmail.com, peri.tsai@gmail.com, vaclav.snasel@vsb.cz

Abstract

The basic K-center problem is a fundamental facility location problem. Given n vertices with some distances, one wants to build k facilities in different vertices, so as to minimize the maximum distance from a vertex to its corresponding facility. This problem is known as the NP-hard problem, and grouping sensor nodes into a cluster is an important mechanism in large multi-hop wireless sensor networks for obtaining scalability, reducing energy consumption, and achieving better network performance. This study proposed a new method for solving the K-center problem based on the Genetic algorithm and dominating (GADO) set, and it is called the GADO method for wireless sensor network. An evaluation of the proposed GADO shows a decrease in the number of the centers compared to the well-known Farthest-first traversal method and dominating set only-based methods. Not only is the total distance from the centers to the sink node less than the other two algorithms, but the proposed GADO also diminishes the data delay and increases the lifetime of the centers.

Keywords: Wireless sensor network, K-center problem, Genetic algorithm, Dominating set

1 Introduction

A wireless sensor network is a system composed of a few to hundreds or even thousands of nodes. The development of small, low-cost intelligent sensors with communication capabilities has prompted the emergence of wireless sensor networks (WSNs) in applications such as environmental monitoring and tracking [1], as well as measuring, and gathering based on some local decision process, in order to transmit the useful data to user [2]. The organization of large multi-hop wireless networks into clusters is essential for

achieving basic network performance. Clustering in WSNs is primarily characterized by data aggregation for each cluster head, which significantly reduces the traffic cost. Sensors in these multi-hop networks detect events and then communicate the collected information to a central location where the sink processes and controls the whole network. WSNs are typically deployed in hostile and remote environments such as dense jungles, battlefields, etc. [3].

It is expected that a large group of cheap, simple sensor devices can be randomly scattered over a field of interest, because of the characteristic of the node that could not be re-charged or replaced in an adverse environment. Thus, in addition to the sensing task, every node also has the burden of relaying interesting events from other sensors in the network to the sink, which causes these nodes that are closer to the sink node to easily die versus those nodes that are far away from the sink. No matter how we decrease the transmission energy cost, those nodes burdened by a lot of relaying tasks consume ten times or even hundreds of times of energy to finish the transmission. Even if the routing optimization algorithm the researcher has proposed is extremely perfect, the node closer to the sink in the critical position will always die earlier than the node located on the border edge with the same equipment. With the development of WSNs, the literature has designed the network hierarchical structure largely used to provide scalable solutions in many networking systems [4-5]. The basic hierarchical model requires two main methods [6]: periodic selection of cluster heads (CHs); and assignment of each node to one or multiple clusters. Each cluster is the cluster representative that is responsible for cluster management.

This paper proposed a new method to solve the energy balancing problem, also called the k service centers vs. client nodes model problem, and applied it

into a wireless sensor network. In this network the sink node serves as the first level for processing the data, and the service centers as a second layer are responsible for collecting the interesting event from the sensor clients and commuting with the sink node directly. The last level of this three-tier structure is the sensor node client level, in which sensing the environment information and transmitting the important event to the service site are the main works for the node clients. Here, k is the amount of service sites, which is the most important parameter for our network, and k should not be too large, which would increase the construction fee and other expenditures. However, this k should also not be too small, which would cause the incomplete coverage problem. Hence, the aim of this paper is to try and minimize the number of service sites as few as possible. In this way, the cost of construction for a three-tier structure network could decrease, while the network at the same time could achieve full coverage.

The rest of the paper is organized as follows. Section 2 introduces the K-center problem and two methods to solve the K-center problem and other related works. Section 3 describes our proposed algorithm for the K-center problem based on a genetic algorithm. Section 4 shows the performance comparing our algorithm with other methods, along with some analysis about them. Section 5 concludes the paper.

2 Related Works

The basic K-center problem is a fundamental facility location problem, where we are asked to locate K facilities in a graph and to assign vertices to facilities, so as to minimize the maximum distance from a vertex to the facility to which it is assigned. This problem is known as being NP-hard [7-8], and several optimal approximation algorithms that achieve an approximation factor of 2 have been developed for it. A ϵ -approximation algorithm for a minimization problem runs in polynomial time and always outputs a solution of value that is no more than ϵ times the optimal. Recent research has presented some variant versions about the K-center problem algorithm [9]. For example, in 2015, Du et al. [10] studied an incremental version of the K-center problem with centers constrained to lying on the boundary of a convex polygon. They presented a polynomial time incremental algorithm with a competitive ratio of around 2.6 and proved that no deterministic incremental algorithm can achieve a ratio better than 2 for the problem.

Liang et al. [11] in 2015 considered the connected K-center (CkC) problem, which can be seen as the classic K-center problem with the constraint of internal connectedness, where two nodes in a cluster are required to be connected by an internal path in the same cluster (CkC was first introduced by Ge et al. [12]

in 2008). They proposed a 6-approximation algorithm for the connected K-center problem and also showed that the algorithm given by Ge et al. [12] is invalid since computing the reaching distance is NP-hard. Feldmann [13] offered a method that considers the K-center problem and some of its generalizations. He disclosed how to get below the approximation factor of 2 by combining the parameters k and h , where k is the number of center vertices and h is the highway dimension parameter. He also noted that the K-center problem is a generalization of the Dominating Set problem. Chechik and Peleg [14] showed interest in the capacitated K-center by studying the fault-tolerant version of this problem, where one or more service centers might fail simultaneously, and how to fix the problem. In 2012, Shih et al. [16] propose a fault node recovery (FNR) algorithm to enhance the lifetime of a wireless sensor network when some of the sensor nodes shut down. The algorithm is based on the grade diffusion algorithm combined with the genetic algorithm [15-16]. The algorithm can result in fewer replacements of sensor nodes and more reused routing paths. However, the FNR still suffers the bottleneck problem, all the inside nodes deplete their power earlier than the outside nodes, and those inside nodes are the most of candidates for recovery.

2.1 Farthest-first traversal

A basic fact about the K-center problem is that it is NP-hard. Thus, there is no efficient algorithm that always returns the right answer. Farthest-first traversal method from Gonzalez [17-18] is a simple greedy approximation algorithm that achieves an approximation factor of 2. The same idea also used in 2002 by Harel and Koren [19].

The idea of Farthest-first traversal is simple. The first point is chosen arbitrarily, and each successive point is as far as possible from all previously chosen points. As the pseudocode shown in Figure 1, it lists the process of Farthest-first traversal. In the pseudocode, S is a set of points, and T is the final results for collecting the centers. Set T is empty at first. Point z is any point in the set S . The distance between a point x to all the centers from set T is denoted as $\rho(x,T)$.

```

Set  $S$ , set  $T = \{ \}$ .
    Pick any  $z \in S$  and set  $T = \{z\}$ 
while  $|T| < k$ :
     $z = \arg \max_{x \in S} \rho(x, T)$ 
     $T = T \cup \{z\}$ 
Output  $T$ 
    
```

Figure 1. Process of farthest-first traversal

2.2 Minimum dominating set

The minimum dominating set problem is one type of minimum set cover problems [13, 20-21], which is a series of main sub-problems when solving the K-center problem. The process of getting the dominating set [20] starts from an empty set T, in which set T is the final dominating nodes. Set T grows during the algorithm based on the “lazy” principle, which adds a new point into T as late as possible. At the same time, the two arrays of CovCnt [] and Score [] will be set, and both of the lengths of the arrays are the number of nodes. In CovCnt[v] array, v is the tag of nodes, the value of CovCnt[v] is the number of times the vertex is covered by the remaining vertices, and its initialized value is deg(v) +1, where deg(v) denotes the degree of node v. The initialization value of Score is equal to CovCnt[v]. Score [] is used to estimate the possibility of becoming a center.

We then choose one node with a minimal score and check this point’s neighbors CovCnt[v] value to see if its neighbors’ cover count value is equal to 1. Setting the node into set T, this node must be added to the dominating set T, because this node is the only possible remaining vertex in network that can cover its neighbor with the CovCnt[v] value equal to 1; otherwise, adjust the CovCnt[v] and Score value. Consequently, all the vertices of the network will be covered. The result of the algorithm is a dominating set. In this paper the final results will be output after running the dominating set algorithm and parametric pruning algorithm.

The author Robič presented this heuristic process for solving the minimum dominating set problem. However, the situation in that paper is much different from ours. That sensing radius and receiving package capacity are fixed values in our simulation tests. Thus, our implementation about the DO part only includes the above process, without the Bottleneck Graph part.

3 GADO Implementation

Genetic algorithms are a family of computational models inspired by evolution [22-25]. An implementation of a genetic algorithm begins with generating several initialization gene populations. To evaluate each gene population, one uses a fitness function and then repeats the following steps until a good result has been found. First, elect father and mother genes from the existing population. Second, recombine parents to generate a new child. Third, compute the fitness value for the child. Fourth, replace the old population by the child with a better fitness value. Finally, stop the output until it reaches the iteration times.

The process of the GADO method is presented as below, which also solves the K-center problem. Thus, the aim of the method is to minimize the number of k.

Step 1. Generate the potential service nodes using the dominating set method introduced in related works, in which nodes will be used to generate initial chromosomes. Note those nodes as SD.

Step 2. Initialize the M chromosome population, with each chromosome having n dimensions. In other words, an n-bit binary string is indicated as having one chromosome where n is the number of whole sensor nodes in the network. The usual 0 and 1 binary representation is the value of n sensor nodes; a value of 0 for each bit implies the node is a normal node; and the value 1 presents that the node is a center. One of the chromosome is a variant of SD. SD is a set of nodes, with those nodes occupying a bit position labeled by value ‘1’. The other bits are given a value ‘0’. The remainder of the population is generated from the one where reverses a certain bits of value ‘0’ into value ‘1’ randomly and non-repetitive. Figure 2 shows one structure of a chromosome. The first row is the serial number of sensor nodes, and the second row presents the character of nodes, whether it is a center or a common node.

1	2	3	4	5	...	n-1	n
1	0	1	0	0	...	1	0

Figure 2. An individual chromosome

From the figure, it is easy to tell that the number one, three, and (n-1) nodes are the center candidates. The others are the common nodes for now.

Step 3. Checkup function. This function is to confirm the validity of a chromosome. Each chromosome must be an effective resolution of the center-client model. Every sensor node could be covered by at least one center. If it returns 0, then this chromosome needs to be remedied. Change the uncovered node to be the center or the following strategy. List all uncovered nodes, keep adding one of the nodes into the center, and then remove the covered node by the new center until all the nodes are covered.

Step 4. Evaluate the M chromosome population using the fitness function. Store the best one as f_{best} .

A well designed fitness function could accelerate the evolving of the good solution. Here we present our fitness function to evaluate the individuals. The fitness function is composed of cv, ca, and oa parameters, and cv is the sum of the number of nodes that satisfy the two criteria. First, the bit value should be ‘0’, where the position represents this node. Second, this node could be covered by at least one center. The ca actually is a constant value, which calculates the total amount district of the network area. The equation is:

$$ca = [width \times length / (senseR)^2] \tag{1}$$

The width and length are the size of the network area, and senseR measures the communication radius of the sensor node. The last oa counts the practical sector that

those centers occupy. Coefficients α_1, α_2 are constants. The fitness function is:

$$F = \alpha_1 \times cv + \alpha_2 \times oa / ca \tag{2}$$

Step 5. Parent selection techniques. There are a lot of widely used methods for parent selections, which cover the task of assigning re-productive choices to each individual in the population. Divide all the chromosomes into two sets; the father chromosome is from one set, and the mother chromosome is from the other. Here, replace those chromosomes that have a worse than one-half chromosome fitness value by the children.

Step 6. For the crossover process, in our algorithm we only use one-point operators. The crossover operators work by fixed choice. The one-point is set as the middle point of the length of the chromosome. For example, Let C_1 (mother) and C_2 (father) be the parent chromosomes (see Figure 3), $C_{1,1}, \dots, C_{1,n}$ and $C_{2,1}, \dots, C_{2,n}$, respectively. A middle point is chosen, and then the child C_3 strings of C_1 and C_2 are:

$$C_3 := C_{1,1}, C_{1,2}, \dots, C_{1,[n/2]}, C_{2,[n/2]+1}, C_{2,[n/2]+2}, \dots, C_{2,n} \tag{3}$$

In order to make sure that the child of a newborn is not invalid for our work, the checkup process is for checking the child's effectiveness. In the end, replace the valid one child with one of the chromosomes in Set B .

1	2	...	k	k+i	...	n-1	n
1	0	...	0	1	...	0	1

1	2	...	k	k+i	...	n-1	n
0	1	...	1	0	...	1	0

1	2	...	k	k+i	...	n-1	n
1	0	...	0	0	...	1	0

Figure 3. Crossover process

Step 7. Mutation process. Mutation is usually applied to each child after a crossover, but here for the sake of ease and convenience, the mutation process is applied to each chromosome. The mutation process works by inverting a bit value in the chromosome with a small probability. A mutation is generally seen as a background operator that provides a small amount of random search. It helps to guard against loss of valuable genetic information by reintroducing information lost due to premature convergence and thereby expanding the search space. [26-27] suggested a mutation rate of $1/n$ as a lower bound on the optimal mutation rate, where n is the length of the chromosome. The mutation rate could be a variable. This lower bound rate exactly mutates one randomly chosen bit in one chromosome. Here, we use a constant 0.02, as the Figure 4 shows.

1	2	...	k	...	k+i	...	n-1	n
1	0	...	0	...	1	...	1	0

1	2	...	k	...	k+i	...	n-1	n
1	0	...	1	...	0	...	1	0

Figure 4. Mutation process

An overview of the GADO method steps:

- (1) Generate M random population solutions based on the results of the dominating set method. Check on their infeasibility.
- (2) Evaluate all the population fitness functions. Sort them by decreasing order.
- (3) Divide the population into two sets. Select two chromosomes $P1$ and $P2$ from the two population sets. $P1$ is not equivalent to $P2$.
- (4) Combine $P1$ and $P2$ using a middle-point operator to generate a child $P3$. Check $P3$'s feasibility. If it is not satisfied, then re-chose $P2$; otherwise replace one of the populations with a population having a lower fitness value by $P3$.
- (5) Mutate 0.02 percent randomly selected genes for any chromosome; check on the chromosome's feasibility; if not satisfied, use the remedy strategy.
- (6) Repeat steps 2-5 until the solution is satisfied.

4 Experiment Results

In this section we describe several implemented and tested algorithms used in our sensor node network. The sensor nodes are randomly deployed in a two-dimensional environment with the size of 200 times 200 square units. The number of sensor nodes differs from 100, 200, 300, and 400. Our algorithm GADO is used to find the K-centers for the network, and then other node could transfer its data to its closest center, while the centers communicate with the sink directly. The goal of this paper is to find the least number of service centers, while those centers could cover all other nodes. The population size of chromosomes is set at 16. In our simulation, the sensing radius and the communication radius range from 40 units to 20 units with a decreasing number of nodes. For 100 nodes, the sensing radius is 40; for 200 nodes, the sensing radius is 30; for 300 nodes, the sensing radius is 25; and for 400 nodes, the sensing radius is 20. Table 1 shows the center number for the three methods and the fitness value. In Table 1, the first row explains the number of nodes 100 to 400 and the sensing radius of nodes 40 to 20. In each method row, the up row shows the number of centers, whereby the smaller amount the better. The down row lists the fitness value, whereby the larger value the better.

Table 1 and Table 2 present the results of four methods: GADO, FF, DO, and GA in four different sizes of network. GA means the method of genetic algorithm. The experiments run in the same situation with the same iterations between GADO and GA,

where 100-node, 200-node, 300-node and 400-node run 500, 600, 700 and 800 iterations respectively. Here, 100/40 describes a 100-node network with a 40-unit sensing and communicating radius in a 200 square unit grid network.

Table 1. Fitness value results

	Node number/sensing radius			
	100/ 40	200/30	300/25	400/20
GADO	43.28	87.83	129.78	171.28
FF	40.86	85.77	127.85	166.83
DO	41.34	85.77	128.29	168.29
GA	41.84	79.88	113.87	151.34

Table 2. Node centers amount results

	Node number/sensing radius			
	100/ 40	200/30	300/25	400/20
GADO	14	25	41	58
FF	19	29	45	67
DO	18	29	44	64
GA	17	41	73	98

Table 2 shows that for 100 nodes of a network with 40 units of nodes' sensing radius, it only uses 14 nodes to cover all nodes in the network for the GADO algorithm and 18 nodes for the dominating set algorithm, while the Farthest-first traversal needs more nodes. For the network with 400 nodes, the sensing radius of the node is 20 units, and the result of GADO is better than the other methods. In Table 1, the result of fitness value also shows that GADO method obtains a good performance.

The dominating set algorithm and Farthest-first traversal both are one-time algorithms, but the GADO algorithm could optimize the results as long as the iteration times are increasing. They actually belong to different classes of solution solving methods. Thus, the

following table lists other results for comparing the merits of these three methods. Distance computing times (DCT), just as the name implies, count the number of times of computing distance between node to node used in the whole process of the algorithm. We could calculate the approximate scope DCT value of FF and DO by using the equation, even though there are some pruning strategies used in the program. First, label the DCT values for FF and DO as FF times and DO times. Next, take the two values as the base line to test the GADO method. Here, as Table 3 shows, it only lists 100-node results, and the results of GADO are still better than the other two methods. What is important is that for the 200-node to 400-node network, the DCT value for GADO is smaller than the others. This fully illustrates the advantage of GADO.

Table 3. GADO results compared to FF and DO

	Center Number	Fitness Value
FF times	17	41.82
DO times	16	42.30

We found that there is a problem in the Dominating set method. That is, when some nodes live without any neighbor, they will become isolated nodes and be ignored by the method. This will cause a wrong result. Therefore, in our experiment we remedy the problem by considering those isolated nodes and adding them into the coverage range. Except for the center quantity and fitness value results, the final three-tier network structure graphs are also plotted here (see Figure 5 to Figure 8). In the graph, the center blue point signifies the sink node, which is the controlling site of the network. Those points in red denote the service centers, and the others in black are the common sensor nodes for sensing and collecting interesting data.

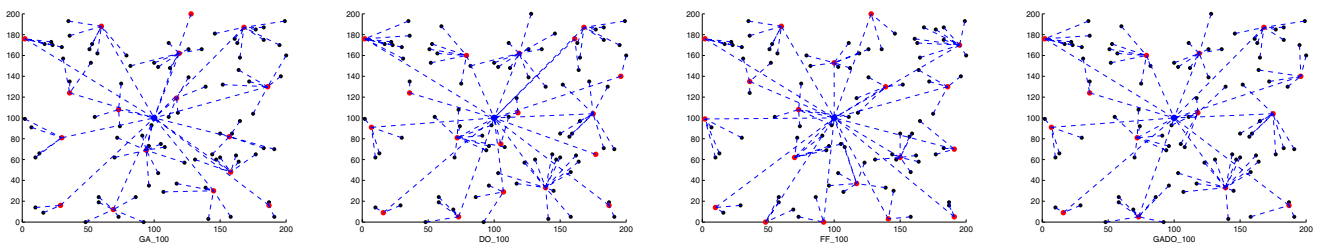


Figure 5. Network structure for 100 nodes

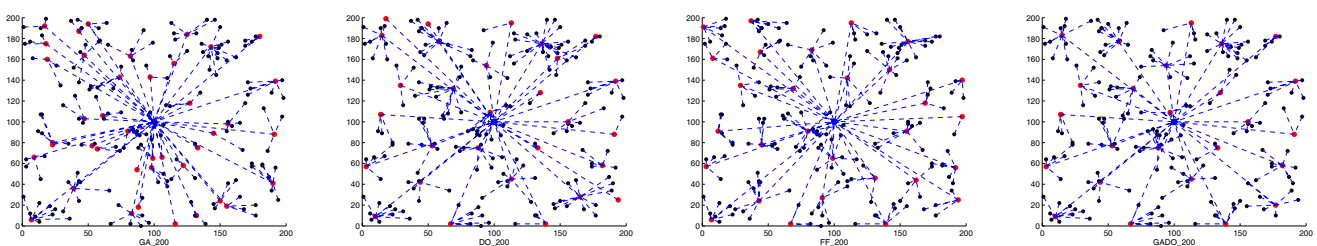


Figure 6. Network structure for 200 nodes

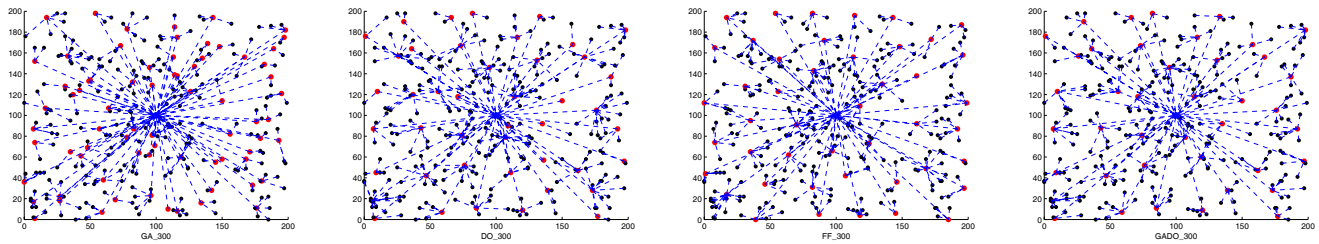


Figure 7. Network structure for 300 nodes

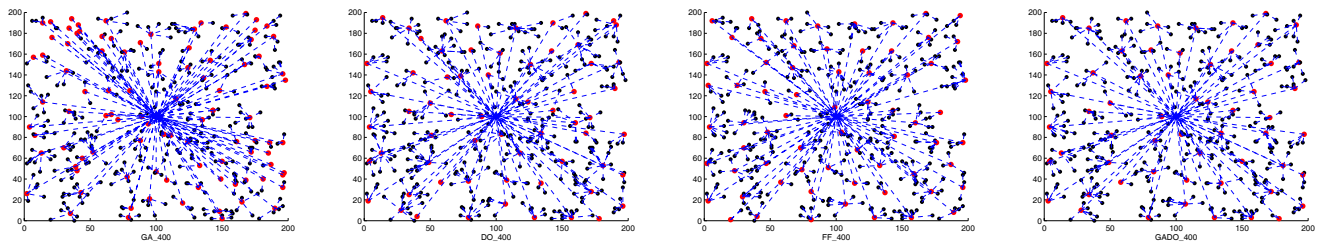


Figure 8. Network structure for 400 nodes

5 Conclusion

This paper builds a three-tier structure network with one sink node, several service centers, and common sensor nodes. The sink node is responsible for controlling network running and processing the data. Service centers as the middle level are in charge of communicating with both sides, collecting the data from common data, and forwarding the data to the sink node. This level thus plays a very important role. However, the number of service centers should not be too large, which would cause too high of a construction fee; yet the number should not be too small; otherwise, an incomplete coverage problem will arise. Hence, this research looks to find the appropriate number of service centers and proposes using the GADO method. The experimental results show that GADO could find a decreased number of centers compared to the existing two other well-known methods, which means that by using our method we need only deploy fewer centers for covering all the network nodes. After the experiments, it can be found that the fewer number of facilities, the more unconnected sensors. Those unconnected sensors cannot contribute to the sensor network. On the contrary, the more connected (working) nodes, the smaller workload for each node. This will realize prolonging the lifetime. Furthermore, the cost of a facility is much more than a general node. Deploying fewer facilities will realize the cost down for the construction of a wireless sensor network.

Furthermore, we make some improvements for the Dominating set method that cannot handle such networks, in which there are some isolated nodes (without neighbors).

Acknowledgments

This work is partially supported by the Fujian Provincial Natural Science Foundation, China, under project acceptance number 11171127 and partially supported by the Key Project of Fujian Education Department Funds (JA15323).

References

- [1] T.-W. Sung, C.-S. Yang, A Voronoi-based Sensor Handover Protocol for Target Tracking in Distributed Visual Sensor Networks, *International Journal of Distributed Sensor Networks*, Vol. 10, No. 3, pp. 1-14, March, 2014.
- [2] P. A. Forero, A. Cano, G. B. Giannakis, Distributed Clustering using Wireless Sensor Networks, *IEEE Journal of Selected Topics in Signal Processing*, Vol. 5, No. 4, pp. 707-724, August, 2011.
- [3] Y. Zhang, Y. Liu, Z. Zhang, H.-C. Chao, J. Zhang, Q. Liu, A Weighted Evidence Combination Approach for Target Identification in Wireless Sensor Networks, *IEEE Access*, Vol. 5, pp. 21585-21596, October, 2017.
- [4] T.-W. Sung, C.-S. Yang, An Adaptive Joining Mechanism for Improving the Connection Ratio of ZigBee Wireless Sensor Networks, *International Journal of Communication Systems*, Vol. 23, No. 2, pp. 231-251, February, 2010.
- [5] T.-T. Nguyen, T.-K. Dao, M.-F. Horng, C.-S. Shieh, An Energy-based Cluster Head Selection Algorithm to Support Long-lifetime in Wireless Sensor Networks, *Journal of Network Intelligence*, Vol. 1, No. 1, pp. 23-37, February, 2016.
- [6] T. Anker, D. Bickson, D. Dolev, B. Hod, Efficient Clustering for Improving Network Performance in Wireless Sensor Networks, in: R. Verdone (Ed.), *Wireless Sensor Networks, Lecture Notes in Computer Science*, Vol 4913, Springer, Berlin,

- 2008, pp. 221-236.
- [7] S. Khuller, R. Pless, Y. J. Sussmann, Fault Tolerant K-center Problems, *Theoretical Computer Science*, Vol. 242, No. 1-2, pp. 237-245, July, 2000.
- [8] J. Hartmanis, Computers and Intractability: A Guide to the Theory of NP-Completeness (Michael R. Garey and David S. Johnson), *Siam Review*, Vol. 24, No. 1, pp. 90-91, January, 1982.
- [9] V. V. Vazirani, *Approximation Algorithms*, Springer Science & Business Media, 2013.
- [10] H. Du, Y. Xu, B. Zhu, An Incremental Version of the K-center Problem on Boundary of a Convex Polygon, *Journal of Combinatorial Optimization*, Vol. 30, No. 4, pp. 1219-1227, November, 2015.
- [11] D. Liang, L. Mei, J. Willson, W. Wang, A Simple Greedy Approximation Algorithm for the Minimum Connected K-Center Problem, *Journal of Combinatorial Optimization*, Vol. 31, No. 4, pp. 1417-1429, May, 2016.
- [12] R. Ge, M. Ester, B. J. Gao, Z. Hu, B. Bhattacharya, B. Ben-Moshe, Joint Cluster Analysis of Attribute Data and Relationship Data: The Connected K-center Problem, Algorithms and Applications, *ACM Transactions on Knowledge Discovery from Data (TKDD)*, Vol. 2, No. 2, Article No. 7, July, 2008.
- [13] A. E. Feldmann, Fixed Parameter Approximations for K-Center Problems in Low Highway Dimension Graphs, in: M. Halldórsson, K. Iwama, N. Kobayashi, B. Speckmann (Eds.), *Automata, Languages, and Programming, ICALP 2015, Lecture Notes in Computer Science*, Vol 9135, Springer, Berlin, 2015, pp. 588-600.
- [14] S. Chechik, D. Peleg, The Fault-tolerant Capacitated K-center Problem, *Theoretical Computer Science*, Vol. 566, pp. 12-25, February, 2015.
- [15] J.-H. Ho, H.-C. Shih, J.-S. Pan, Hierarchical Gradient Diffusion Algorithm for Wireless Sensor Networks, *Journal of Internet Technology*, Vol. 16, No. 7, pp. 1201-1210, December, 2015.
- [16] H.-C. Shih, J.-H. Ho, B.-Y. Liao, J.-S. Pan, Fault Node Recovery Algorithm for a Wireless Sensor Network, *IEEE Sensors Journal*, Vol. 13, No. 7, pp. 2683-2689, July, 2013.
- [17] T. F. Gonzalez, Clustering to Minimize the Maximum Intercluster Distance, *Theoretical Computer Science*, Vol. 38, pp. 293-306, June, 1985.
- [18] CSE 291: Geometric Algorithms, <http://cseweb.ucsd.edu/~dasgupta/291-geom/>.
- [19] D. Harel, Y. Koren, Graph Drawing by High-dimensional Embedding, in: M. T. Goodrich, S. G. Kobourov (Eds.), *Graph Drawing. GD 2002, Lecture Notes in Computer Science*, Vol. 2528, Springer, Berlin, 2002, pp. 207-219.
- [20] B. Robič, J. Mihelič, Solving the k-center Problem Efficiently with a Dominating Set Algorithm, *Journal of Computing and Information Technology*, Vol. 13, No. 3, pp. 225-234, September, 2005.
- [21] S. Elloumi, M. Labbé, Y. Pochet, A New Formulation and Resolution Method for the p-center Problem, *INFORMS Journal on Computing*, Vol. 16, No. 1, pp. 84-94, February, 2004.
- [22] D. Whitley, A Genetic Algorithm Tutorial, *Statistics and Computing*, Vol. 4, No. 2, pp. 65-85, June, 1994.
- [23] T.-K. Dao, T.-S. Pan, T.-T. Nguyen, S.-C. Chu, A Compact Artificial Bee Colony Optimization for Topology Control Scheme in Wireless Sensor Networks, *Journal of Information Hiding and Multimedia Signal Processing*, Vol. 6, No. 2, pp. 297-310, March, 2015.
- [24] F.-C. Chang, H.-C. Huang, A Survey on Intelligent Sensor Network and Its Applications, *Journal of Network Intelligence*, Vol. 1, No. 1, pp. 1-15, February, 2016.
- [25] V. Snasel, L. Kong, P.-W. Tsai, J.-S. Pan, Sink Node Placement Strategies based on Cat Swarm Optimization Algorithm, *Journal of Network Intelligence*, Vol. 1, No. 2, pp. 52-60, May, 2016.
- [26] T. Bäck, Optimal Mutation Rates in Genetic Search, *Proceedings of the 5th International Conference on Genetic Algorithms*, Urbana-Champaign, IL, 1993, pp. 2-8.
- [27] J. E. Beasley, and P. C. Chu, A Genetic Algorithm for the Set Covering Problem, *European Journal of Operational Research*, Vol. 94, No. 2, pp. 392-404, October, 1996.

Biographies



Jeng-Shyang Pan received the Ph.D. degree in Electrical Engineering from the University of Edinburgh, U.K. in 1996. Currently, he is a Professor in Harbin Institute of Technology Shenzhen and Fujian University of Technology. His current research interests include soft computing, information security, big data mining and signal processing.



Lingping Kong is a Ph.D. student in Harbin Institute of Technology Shenzhen. Her current research interests include wireless sensor network, artificial intelligence etc..



Tien-Wen Sung received his Ph.D. degree in Computer and Communication Engineering from Department of Electrical Engineering at Cheng Kung University, Taiwan in 2014. He is currently an Associate Professor in College of Information Science and Engineering, Fujian University of Technology, China. His research interests include wireless sensor networks and e-learning.



Pei-Wei Tsai received his Ph.D. in Electronic Engineering at National Kaohsiung University of Applied Sciences. Currently, he is an associate professor in Fujian University of Technology in China. His research interests include swarm intelligence, information security, finance forecasting, digital image processing, fuzzy logic, and wireless sensor network.



Václav Snášel is a researcher and university professor at the VSB-Technical University of Ostrava. His research and development experience includes over 25 years in the Industry and Academia. He works in a multi-disciplinary environment involving artificial intelligence, conceptual lattice, information retrieval, semantic web, neural network, web intelligence etc..