BULGARIAN ACADEMY OF SCIENCES

CYBERNETICS AND INFORMATION TECHNOLOGIES • Volume **15**, No 6
Special Issue on Logistics, Informatics and Service Science

Sofia • 2015

Print ISSN: 1311-9702; Online ISSN: 1314-4081
DOI: 10.1515/cait-2015-0077

Genetic Algorithm Based Clustering for Large-Scale Sensor Networks

Hai Lin, Ruoshan Kong, Jiali Liu

International School of Software, Wuhan University, Wuhan, 430072 China
Email: lin.hai@whu.edu.cn

Abstract: *Despite the success of various clustering algorithms for Wireless Sensor Networks* (*WSNs*)*, there are few works that consider the interference between clusters. Obviously, interference-free clustering makes the communication more efficient and achieves energy saving. In this paper we propose a new clustering method for large-scale sensor networks. With this method the network is partitioned into clusters. Intra-cluster communication in a cluster has no interference by its neighbor clusters. Moreover, the proposed clustering is based on a Genetic Algorithm* (*GA*)*, which can achieve optimal performance in terms of the number of isolated nodes. This is demonstrated by the simulation analysis.*

Keywords: *Clustering, genetic algorithm, interference-free, WSN.*

1. Introduction

Research on Wireless Sensor Networks (WSNs) is one of the most rapidly growing scientific domains. This is because of the development of advanced sensor nodes with extremely low cost, and the potential applications of such sensor nodes are ever growing. As one of the efficient ways of network topology control, clustering-based protocols are considered as the best choice for large-scale WSNs [1]. It has some benefits, such as energy saving and scalability [2].

A cluster includes at least a Cluster Head (CH) and some cluster members. The CH is responsible for coordinating the nodes within its cluster and it periodically transmits aggregated data to the sink node. Clustering of WSNs has attracted much attention. The early work on clustering is LEACH [3], in which CHs are selected based on a predetermined probability. Other nodes choose a cluster to join by estimating which of the selected CHs is the closest one. However, LEACH addresses only one-hop transmission between CHs and the sink. It is not applicable for large-scale WSNs. HEED [4] is designed for multi-hop WSNs. It focuses on CH selection by considering both the residual energy and intra-communication cost. UCR [5] focuses on load balancing in order to address hot spot issues. This is done by designing smaller clusters as they approach the sink. As for the recent works, EDIT [6] selects CHs based not only on energy, but also on delay. In [7] the authors address load balancing by considering different hop distances for clusters. SEECH [8] proposes a relay selection scheme, where the relay function is separated from the CH node.

These previous clustering works usually do not consider the communication interference between the clusters and let MAC layer handle the communication issues. To increase the communication efficiency, MAC layer often resorts to contention free protocols [9, 10]. In [9] the authors propose a TDMA-based MAC protocol for cluster-based networks, in order to reduce the energy consumptions on nodes with low data traffic and to decrease the transmission latency on nodes with heavy data traffic. After the cluster formation, each CH collects its members' transmission information and allocates time slots according to the requests. E-BMA [10] also applies the TDMA-based MAC protocol within a cluster. The CH within a cluster allocates the time slots for its members according to the information collected via the contention period, also via the piggybacking information of the data packet. These works focus on TMDA scheduling within a cluster. It does not consider the interference between the adjacent clusters. Obviously, the interference from adjacent clusters makes the allocated time slots disable. To address this issue, combining of two medium access technologies is often applied. One solution is to combine TDMA and CDMA as done in LEACH [3]. That is, within a cluster the TDMA technology is used, while different clusters use different CDMA codes for inter-cluster interference avoidance. Another one is to combine FDMA and TDMA schemes [11]. That is, different frequencies are used for clusters, while TDMA is used within a cluster. Obviously, both solutions require a sensor node to support two medium access technologies, which is normally beyond the ability of the current sensor nodes. Besides, using two medium access technologies degrades the communication performance.

The previous works also resort to inter-cluster cooperation to avoid the interference between the adjacent clusters. The main idea behind these works is that the interference can be avoided if each cluster allocates the time slots by considering its adjacent clusters' time allocation. However, due to the complexity of inter-cluster cooperation management, it is often impractical for WSNs which are resource limited. In this paper we propose a clustering method with which a cluster can independently allocate the time slots to its members without the inter-cluster cooperation.

In the large-scale WSNs, a sensor node only covers a small fraction of the whole sensor field. Hence, if the CHs are carefully selected, it is possible that the communication interference between clusters can be avoided. For example, if two CHs are three-hop away and only a CH's direct neighbour nodes can join its cluster, it is guaranteed that these two clusters have no interference to each other. In this paper we try to partition the large-scale sensor network into clusters without interference during the data collection period. In consequence, each CH can allocate time slots to its member independently, which not only facilitates time slot management, but improves the communication efficiency as well.

In summary, we intend to propose a clustering method which can satisfy two requirements: one is no interference between adjacent clusters for collision avoidance; the other is direct communication between a CH and its members for time slots allocation. However, to grant these two requirements, it is often impossible to let all sensor nodes join clusters. Those nodes that cannot join any cluster are called isolated nodes. Since the isolated node degrades the network performance, e.g. data collection ratio and energy efficiency, the number of isolated nodes should be minimized. For this purpose, our clustering method resorts to a Genetic Algorithm (GA) to minimize the isolated nodes number and we call it Genetic Algorithm Clustering (GAC). GA provides an optimization method that, by defining an appropriate fitness function, identifies the optimal or sub-optimal solutions to satisfy all constraints. In fact, GA for clustering optimization has gained some attention [12, 13]. In [12] the authors propose a GA-based method that optimizes heterogeneous sensor node clustering by considering multiple heterogeneity and clustering factors, such as remaining energy, network location and distance to the base-station. In [13] the proposed GA-based algorithm not only minimizes the energy consumption and maximizes the network revenue, but also produces clusters with uneven size to balance the energy consumption among the cluster heads. However, so far, there is no proposition using GA to optimize interference-free clustering.

The contribution of this work is two-fold: First, an interference-free clustering method is proposed to facilitate TMDA management, since a CH only needs to allocate the time slots for its members without considering the allocation of its neighbour clusters, or without using other medium access protocols, e.g., FDMA. Second, a GA-based optimization method is developed that encodes the network clustering structure with integrity validation and employs a simple fitness function.

In the rest of this paper, Section 2 describes our first work on interference-free clustering. In Section 3 our proposition of GAC is described in details. Section 4 analyzes the performance through simulations, in which we can see that GA-based optimization significantly improves the performance. We conclude our work in Section 5.

2. Interference free clustering

As above described, the proposed clustering method should grant the two requirements for interference-free clustering. Granting the requirement of direct communication between a CH and its members is a trivial work. That is, only a CH’s direct neighbour nodes can join this CH. According to our analysis, the requirement of no interference between adjacent clusters can be interpreted as follows: *as long as CHs have no common neighbour nodes, the interference between clusters can be avoided*. This limits a member (node) to cover only one CH. Hence, during the data collection period, i.e., the communication from the members to their CH, a node's communication will never interfere CHs of other clusters, so the collision is avoided. The requirement can be further intercepted as: any CHs should be three or more hop away to each other; or CHs can be direct neighbour (one-hop away), but never be two-hop away.

Based on this observation, our first interference-free clustering method is described as follows:

- Each node randomly sets a timer for CH competition;
- When a node's timer fires, the node becomes a CH and its neighbours become its members. All two-hop neighbour nodes quit the competition.
- Repeat the above steps until all nodes finish the competition.

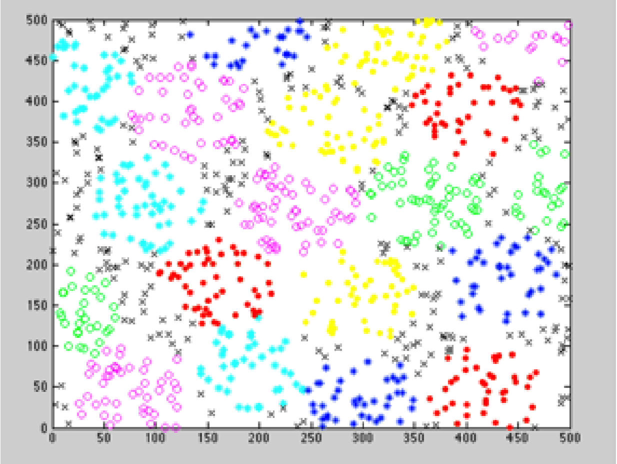


Fig. 1. Clusters without interference

With this method, a network can be partitioned into clusters without interference. Fig. 1 shows the clustering result of 1000 nodes by using this method. Different colours represent different clusters and the communication for data collection within a cluster is exempt from interference of other clusters. However, we also observe that lots of the nodes (black cross nodes – in the figure they are denoted with ×) fail to join any clusters and these nodes become isolated nodes. To minimize the isolated nodes number, in the next section we propose GAC to minimize the isolated nodes number.

3. Genetic algorithm based clustering

3.1. Gene and chromosome

In our GA-based clustering, a chromosome represents the selected CHs. That is, a gene value is the CH identity. Each chromosome has a fixed length size, which indicates the total CH number, i.e., the clusters number.

For example, a network has 1000 nodes and is partitioned into 10 clusters. A chromosome can be described as [554 472 989 865 9 539 435 412 195 520], where each number represents a CH identity.

In our genetic algorithm, the number of clusters, i.e. chromosome length, is fixed. This can minimize the algorithm iterations, but it requires an optimal (or near to optimal) value of the clusters number to be calculated at the initiation step. Intuitively, this value depends on CH's coverage area. We use the following formula for this calculation:

$$\mathrm{len}_{\mathrm{chrom}} = [\gamma(r)/(l * w/\pi r^2)], \tag{1}$$

where l and w are the length and width of the network field respectively; r is the transmit radius of a sensor node. $l * w/\pi r^2$ is considered as an ideal value for the cluster number ($l * w$ is a network area and $\pi r^2$is a cluster area), so we add a coefficient $\gamma(r)$ to revise it, i.e., the clusters number is a little smaller than the ideal value. In the performance analysis section, we will calculate $\gamma(r)$ based on experiments.

After obtaining this value, a *population initiation process* is carried out (a population is a collection of individual chromosomes). For each individual, it is initiated as:

Step 1. Let all nodes be CH candidates.

Step 2. Randomly choose a node from CH candidates and add it to CH group.

Step 3. Delete this node and all its two-hop nodes from CH candidates.

Step 4. Repeat Steps 2 and 3 until the number of CHs is equal to the obtained value.

Then, repeat the above steps to choose *N* individuals for the initial population. In our simulation, *N* is set to 20.

3.2. Population evaluation

The goal of GA is to minimize the isolated nodes number, so the fitness function defined in our proposition is simple:

$$f = 1/n_{\mathrm{iso}}, \tag{2}$$

where n_{iso} is the number of isolated nodes.

In our genetic algorithm, the elitist strategy was introduced into GA to preserve the best individuals, so m chromosomes with better fitness are kept for the next generation. Besides these m chromosomes, the method selects $N - m$ best individuals from the current generation. These $N - m$ individuals are selected in a way that is proportional to their fitness, e.g. roulette wheel selection based on the fitness function. The probability that a chromosome will be selected is

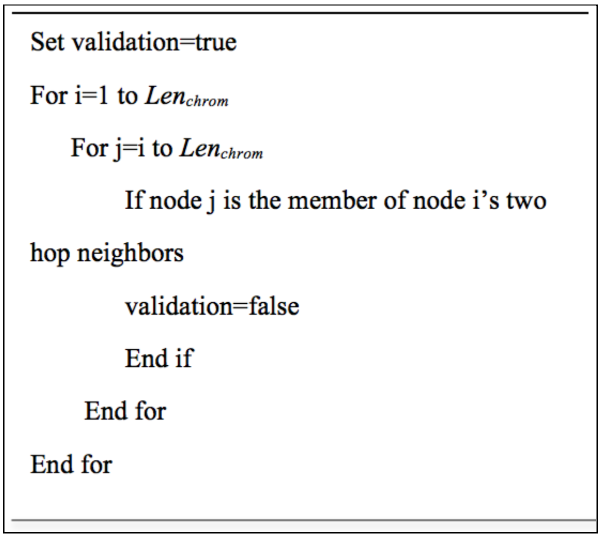
$$p_i = \frac{f_i}{\sum_{j=1}^{N} f_i}, \tag{3}$$

where f_i is the fitness value of each individual i in the population, *N* is the number of individuals in the population. It is worth to note that one and the same individual may be selected multiple times.

With the selected $N - m$ chromosomes, the population is evaluated via a crossover and mutation process.

3.2.1. Crossover

Among the above selected chromosomes, GAC randomly selected two chromosomes that are called “parents”. The parents will generate two new chromosomes. However, the new pair of chromosomes may not meet the requirement that *CHs should not have common neighbour nodes*. Thus, a validation process should be performed for each new chromosome. If validated, the new chromosome is kept. Otherwise, it is just discarded. The validation process (Fig. 2) checks if any CH is a member of other CHs’s two-hop neighbors. If so, the validation fails.



```
Set validation=true
For i=1 to $Len_{chrom}$
    For j=i to $Len_{chrom}$
        If node j is the member of node i’s two hop neighbors
        validation=false
        End if
    End for
End for
```

Fig. 2. Pseudo code of the validation process

The crossover process is described as follows:

Step 1. Randomly select two chromosomes.

Step 2. Randomly select an integer *i* between 2 and $\mathrm{len}_{\mathrm{chrom}}-1$.

Step 3. Cross the two chromosomes from *i*.

Step 4. Validate the two generated chromosomes. The invalid chromosome is just discarded.

Step 5. Repeat Step 1-4 until the number of validated chromosomes is $N-m$.

After the crossover process, the mutation process is executed.

3.2.2. Mutation

In the mutation operation, a gene is randomly selected and its value is changed to any other node except the nodes which already exist in this chromosome. The selection probability for a gene is fairly small. In our simulation, this probability is set to $1/\mathrm{len}_{\mathrm{chrom}}$, so the expected number of genes that need to be changed is 1. Similarly, the new generated chromosome may be invalid, so the validation process is also executed for each new chromosome. If validated, the chromosome is kept for the next generation. If not, it should be changed back to the previous form. The operation of mutation adds variation in the new generation.

Then, the $N-m$ chromosomes are combined with m best chromosomes to form the new generation.

The process is repeated until there is no change between two successive generations or a fixed number of generations is reached. Finally, an optimal CH distribution is found which can bring optimal clustering results.

4. Performance analysis

In this section we analyze GAC, using MATLAB simulation. We simulate such scenario: within the range of 500×500 m^2, with randomly deployed 1000 nodes. The other parameters are listed in Table 1.

Table 1. Simulation parameters

Parameters	Values
Node number	1000
Simulation area	500×500 m
Node transit radius	60 m
Size of population	20
Number of remained elites	2
Crossover probability	0.8
Mutation probability	0.03
Chromosome length	19

First, we try to obtain the coefficient $\gamma(r)$ of Equation (1) by experiments. Fig. 3 shows the generated cluster number vs. node's transmission radius. We simulate three scenarios. The blue curve (i.e., ideal curve) represents the number of clusters obtained by the function $lw/\pi r^2$, which is considered the optimal cluster number as described in Section 3.1. The other two scenarios are based on the initiation process described in Section 3.1. But different to the process, here we repeat Steps 2 and 3 until it is impossible to select more CHs, in order to have the maximum clusters number as possible. For every transmission radius, the process is executed 20 times. The max curve represents the maximum CH number obtained among these 20 operations, while the average curve represents the average CH number obtained.

According to Equation (1), $\gamma(r)$ is written as

$$\gamma(r) = \frac{\mathrm{len}_{\mathrm{chrom}}}{(lw/\pi r^2)}, \tag{4}$$

where $\mathrm{len}_{\mathrm{chrom}}$ is the obtained clusters number by experiments. If we choose the maximum CH number to calculate $\gamma_{\max}(r)$, then the calculation of $\mathrm{len}_{\mathrm{chrom}}$, based on $\gamma_{\max}(r)$ can bring the maximum clusters number. This could minimize the number of isolated nodes: the greater the cluster number is, the less isolated nodes are left. But the tradeoff is that the population initiation process should be executed many times (expected 20 times) to have this maximum clusters number. Hence, in our solution we take the average clusters number for $\gamma(r)$ calculation.

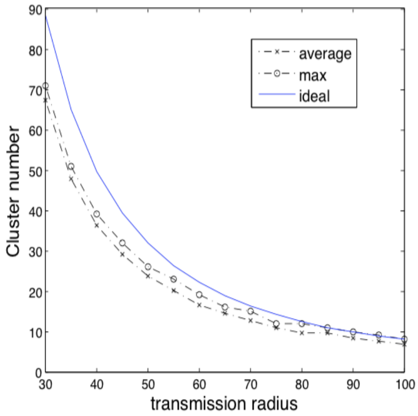


Fig. 3. Cluster number vs. transmission radius

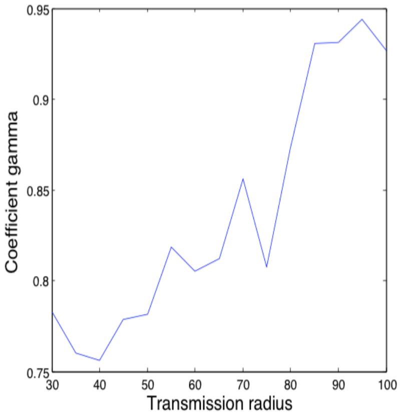


Fig. 4. Coefficient vs. transmission radius

Based on the average curve in Fig. 3, we have $\gamma(r)$ for each transmission radius. Fig. 4 shows the relationship between $\gamma(r)$ and the transmission radius. With the curve fitting method, we obtain the equation:

$$\gamma(r) = 0.0028r + 0.6565. \quad (5)$$

For the parameters listed in the Table II, where the node's transmission radius is set to 60, we have $\gamma(60) = 0.82$. According to Equation (1), the chromosome length is set to 18, i.e. the network should be partitioned into 18 clusters.

In the following lines we analyze the convergence of our GAC algorithm. Fig. 5 shows the isolated node proportion (the number of isolated nodes/the total nodes number) when the chromosomes evolve (the iteration time increases). Clearly, when the chromosomes evolve, better results are obtained. That is, more nodes can find a cluster to join, which results in less isolated nodes. Two curves are shown in the figure. The one with a dotted line presents the average proportion value of all individual chromosomes. The other with a solid line presents the proportion value of the best individual. We can observe that about 42 iterations later, one of the chromosomes evolves with the optimal fitness function, i.e., achieves the best clustering. About 52 iterations later, most of the chromosomes evolve to the optimal chromosomes. Also, we observe that with GAC only 21% nodes become isolated nodes, which is much less than our first proposition (called competition clustering), in which, about 37% of the total nodes are isolated nodes. We analyze in details GAC by comparing it with the competition clustering.

The comparison is done by varying the node's transmission radius and nodes number. Fig. 6 shows the isolated node proportion by varying the node's transmission radius. The curve of GAC is on an uptrend with the increase of the transmission radius, i.e., the number of isolated nodes increases with the transmission radius. This is because when the nodes' coverage area increases, the number of clusters decreases according to Equation (1). In consequence, more nodes fail to join a cluster. By comparing the two curves, GAC outperforms the competition clustering in terms of the isolated nodes number. With the competition clustering, about 36-38% of the total nodes fail to join any clusters and become isolated nodes, while only 22-24.5% of the total nodes are isolated nodes with GAC.

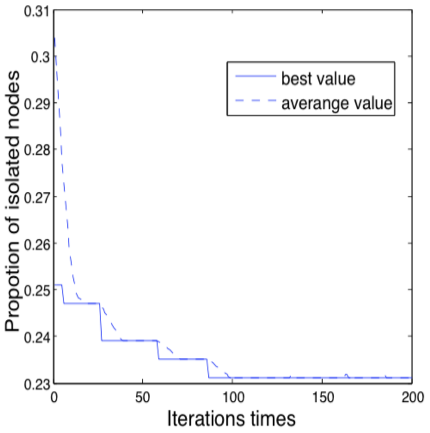


Fig. 5. Proportion of isolated nodes

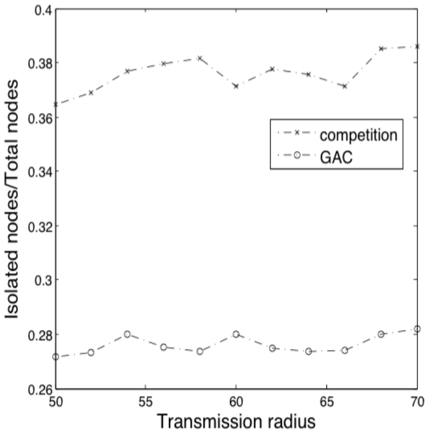


Fig. 6. Comparison by varying the transmission radius

Fig. 6 shows the isolated node proposition by varying the nodes number. We observe that the number of isolated nodes increases with the increase of the nodes number. This is because, when the nodes number increases, the density also increases. To achieve interference free mode, more nodes should not join the clusters for communication competition. That is why the number of isolated nodes increases. Thus, the proportion of isolated nodes increases. Similarly, we observe that GAC is much better than the competition clustering. With GAC, only about 21-25% of the total nodes fail to join any clusters, while about 35-39% of the total nodes become isolated nodes with the competition clustering.

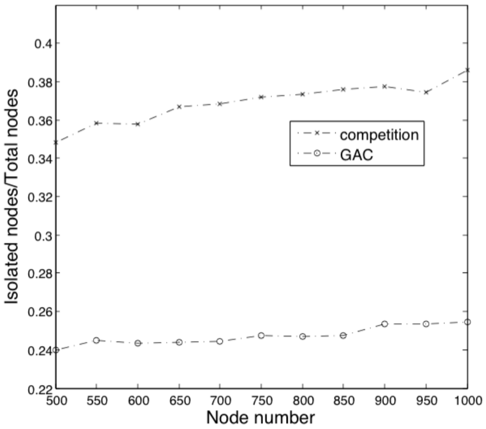


Fig. 7. Comparison by varying the nodes number

5. Conclusion

In this paper, we propose a GA-based clustering method for large-scale sensor networks. The clustering method partitions a network into interference-free clusters and minimizes the isolated nodes number. Based on this clustering, a CH manages the time slots easily without considering the inter-cluster cooperation. The simulation results show that our clustering algorithm has good performance and largely reduces the number of isolated nodes. In this paper, the interference-free clustering only takes into account the isolated nodes number. In our future works we will consider more factors, such as load balancing and communication efficiency, for performance improvement.

***Acknowledgements*:** The work presented in this paper was supported by the International Science and Technology Cooperation Program of China under grant No 2013DFA12460 and the Fundamental Research Funds for the Central Universities of China under grant No 2042015kf0053.

References

1. Younis, M. Krunz, S. Ramasubramanian. Node Clustering in Wireless Sensor Networks: Recent Developments and Deployment Challenges. – IEEE Network, Vol. **20**, May-June 2006, No 3, pp. 20-25.
2. Al-Karaki, J. N., A. E. Kamal. Routing Techniques in Wireless Sensor Networks: A Survey. – IEEE Wireless Commun., Vol. **11**, December 2004, No 6, pp. 6-28.
3. Heinzelman, W. B., A. P. Chandrakasan, H. Balakrishnan. An Application-Specific Protocol Architecture for Wireless Microsensor Networks. – IEEE Trans. Wireless Commun., Vol. **1**, October 2002, No 4, pp. 660-670.
4. Younis, S. Fahmy. HEED: A Hybrid Energy-Efficient Distributed Clustering Approach for Ad Hoc Sensor Networks. – IEEE Trans. Mob. Comput., Vol. **3**, 2004, No 4, pp. 366-379.
5. Chen, G., C. Li, M. Ye, J. Wu. An Unequal Cluster-Based Routing Protocol in Wireless Sensor Networks. – Wireless Networks, Vol. **15**, February 2009, No 2, pp. 193-207.
6. Thakkar, A., K. Kotecha. Cluster Head Election for Energy and Delay Constraint Applications of Wireless Sensor Network. – IEEE Sensors Journal, Vol. **14**, August 2014, No 8, pp. 2658-2664.
7. Liao, Y., H. Qi, W. Li. Load-Balanced Clustering Algorithm with Distributed Self-Organization for Wireless Sensor Networks. – IEEE Sensors Journal, Vol. **13**, May 2013, No 5, pp. 1498-1506.
8. Tarhani, M., Y. S. Kavian, S. Siavoshi. SEECH: Scalable Energy Efficient Clustering Hierarchy Protocol in Wireless Sensor Networks. – IEEE Sensors Journal, Vol. **14**, November 2014, No 11, pp. 3944-3954.
9. Hsu, T.-H., P.-Y. Yen. Adaptive Time Division Multiple Access-Based Medium Access Control Protocol for Energy Conserving and Data Transmission in Wireless Sensor Networks. – Communications, IET, Vol. **5**, December 2011, No 18, pp. 2662-2672.
10. Shafiullah, G. M., S. A. Azad, A. B. M. S. Ali. Energy-Efficient Wireless MAC Protocols for Railway Monitoring Applications. – Intelligent Transportation Systems, IEEE Transactions, Vol. **14**, June 2013, No 2, pp. 649-659.
11. Gherairi, S., S. Ouni, F. Kamoun. Optimized TDMA Multi-Frequency Scheduling Access Protocols for Sensor Networks. – In: Proc. of 2011 International Conference on Communications, Computing and Control Applications (CCCA), March 2011, pp. 1-6.
12. Elhoseny, M., X. Yuan, Z. Yu, C. Mao, H. El-Minir, A. Riad. Balancing Energy Consumption in Heterogeneous Wireless Sensor Networks Using Genetic Algorithm. – Communications Letters, IEEE, 2014, No 99.
13. Wu, Y., W. B. Liu. Routing Protocol Based on Genetic Algorithm for Energy Harvesting-Wireless Sensor Networks. – Wireless Sensor Systems, IET, Vol. **3**, June 2013, No 2, pp. 112-118.