Special Issue of International Journal of Computer Applications (0975 – 8887)
on Wireless Communication and Mobile Networks, No.1. Jan.2012, ww.ijcaonline.org

Imperialist Approach to Cluster Head Selection in WSN

S. M. Hosseinirad
Department of Computer Science,Banaras Hindu University,Varanasi-221005, India

S.K. Basu
Department of Computer Science,Banaras Hindu University,Varanasi-221005, India

ABSTRACT

Finding cluster head (CH) is an important issue in WSN. A new optimization algorithm Imperialist Competitive Algorithm (ICA) has been introduced recently, inspired by socio-political process of imperialistic competition. We use ICA for CH selection according to the communication energy (CE) cost. We demonstrate that ICA is an effective method for selection of CH in WSN. ICA either finds one or at most a few CHs within 500 decades. The tie is broken by use of a heuristic. CE stabilizes after 225 decades in the case of 300-size, after 150 decades for 200-size, and after 140 decades for 100-size WSNs. For 100-size, 1 CH is selected after 260 decades, for 200-size and 300-size 7 and 21 CHs, respectively are selected after 500 decades. For reducing the number of final CHs, the algorithm should be run for more than 500 decades for larger-size WSNs. For smaller size (100) networks, time increases very slowly with decades. For higher size networks, it increases nonlinearly and takes almost exponential shape with a network of 300 sensors. This is a preliminary study and we plan further investigation in this direction.

Keywords

WSN, Cluster Head, Imperialist

1. INTRODUCTION

Wireless sensor networks (WSNs) have gained importance in many scientific, military and other areas over the past decade [1, 2]. WSN consists of a large number of sensor nodes normally deployed randomly in the application environment [3]. Sensors are battery-operated (impossible to be replaced or recharged in the field), and usually equipped with low-power signal processing, storage, wireless radio communication capabilities [4]. They are now-a-days used in many industrial, military and civilian applications, such as industrial process monitoring and control, machine health monitoring, environment and habitat monitoring, healthcare, home automation, traffic control, etc [5]. Energy conservation, limited bandwidth, scalability, operation in hostile environments and data processing are some of the aspects that have to be carefully taken into account [6]. Energy efficiency and power control are important primary challenges in WSNs. To achieve good efficiency in communication energy, several solutions have been proposed [7]. Constructing WSNs based on clustered approach has been extensively investigated recently [8]. One of the best methods to design energy efficient, highly scalable and robust WSN is clustering. By utilizing clustering approach, communication energy is reduced; thereby energy consumption in the sensor nodes is decreased. Through fusion and aggregation of the sensors' data at the cluster heads (CH), the total amount of data sent to the sink is significantly reduced, saving energy and bandwidth resources [9].

In this paper we use Imperialist Competitive Algorithm (ICA) [13] to select CH in a cluster having nodes which are homogenous and randomly deployed. The paper is organized as: Section II deals with related work, section III with imperialist competition algorithm, section IV with cluster head selection through imperialist approach, section V with results, and section VI concludes the paper.

2. RELATED WORK

For cluster head selection, some algorithms such as Low Energy Adaptive Clustering Hierarchy (LEACH) [10], Adaptive Cluster Head Selection (ACHS) [12], and Extended Adaptive Cluster Head Selection (EACHS) [12] have been proposed to increase the WSN's lifetime and energy efficiency.The first hierarchical cluster-based LEACH [10] is a routing protocol and a classical algorithm in the WSN. It distributes the energy load to prolong WSN's lifetime [10]. It partitions the nodes into clusters, in each cluster a node with extra privileges is selected as CH. For initialization, all nodes assign itself a random number between 0 and 1. A new cluster head is selected according to the random number of the sensor nodes less than a predefined threshold. ACHS [12] and EACHS [12] have been proposed to have better load balancing in the clusters. If sensor nodes are deployed in clusters equally, the probability of the network load and energy consumption will highly increase [13]. In order to save time and energy, by reselecting clusters, EACHS improves the ACHS when those differ from the average number of sensor nodes.

3. IMPERIALIST COMPETITION ALGORITHM

To solve a real world optimization problem, different methods such as genetic algorithms [14], ant colony optimization [15], bird flocking algorithm optimization [14], etc. have been proposed. These methods are based on computer simulation of natural processes. For example, genetic algorithms (GA) belong to the larger class of evolutionary algorithms (EA). Based on search heuristic, GA generates optimized solution from a population of candidate solutions to a given problem [15]. Ant colony optimization is inspired by the foraging behaviour of real ants [16]. By using the notion of imperialism and imperialistic competition theory, Imperialist Competition Algorithm (ICA) has been introduced as a new evolutionary global heuristic search [13]. ICA simulates the socio-political process of imperialism and imperialistic competition. This algorithm uses a population, where each individual is called a country. A country is represented by a vector of n components as [p1, p2, p3,…, pn], where pi (1 ≤ i ≤ n) represents ith attribute of the country. Each variable associated with a country can be interpreted as a socio-political characteristic such as culture, language, economical policy, and even religion. The cost of a country is found by evaluation of the cost function f :

Cost t = f (country) = f (p1, p2, p3,…, pn).

The empires are initially selected based on the primary cost. All empires should have at least one colony. The maximum number of empires cannot be more than 50% of the population. To distribute the remaining countries (if any) among the empires, the cost of the countries are normalized by Cn= cn-max{ci}, where cn is the cost of the nth empire and Cn is its normalized cost. An empire with higher cost will have lower normalized value. After cost normalization, the total power of each empire is calculated as

$$Power_n = \left| \frac{C_n}{\sum_{i=1}^{N_{imp}} c_i} \right|,$$

where the Cn is the cost of an empire, and ci is the cost of colony i. The normalized power of each empire is determined by the total cost of its colonies. Therefore, the initial colonies are divided among empires based on their power. The initial number of colonies of the nth empire is $N.C_n = \lfloor p_n . N_{col} \rfloor$, where N.Cn is initial number of colonies of the nth empire and Ncol is the number of colonies. N.Cn of the colonies are randomly chosen and given to the nth empire. These colonies along with the nth imperialist form the nth empire. Initial empires and colonies are displayed in Fig. 1. The strongest empire occupies more number of colonies and the weakest may just have one colony.

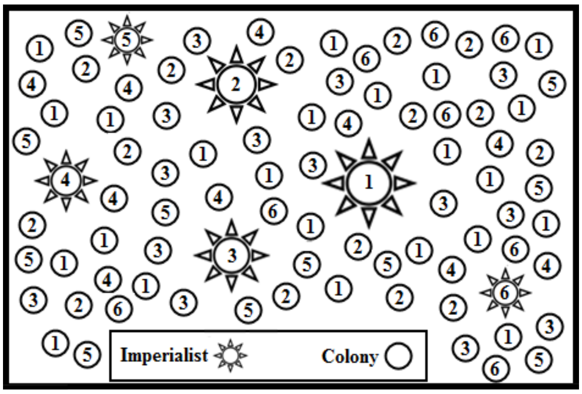

Figure 1. Imperialist and colonists

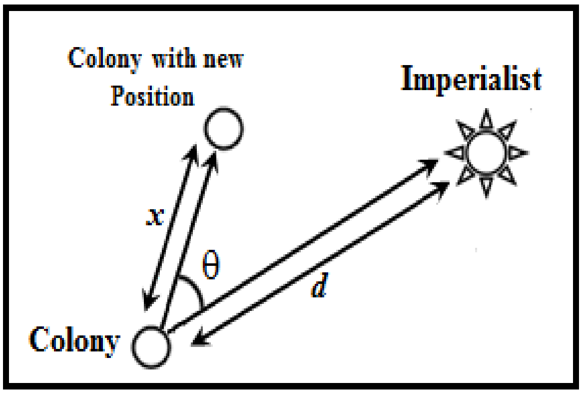

Figure 2. Colony moving toward Imperialist

Figure 2 shows movement of a colony, where *x* is a random variable (representing the distance between new and old positions of a colony) with uniform (or any proper) distribution, to model and increase the ability of searching more area around the empire. So, this direction of movement is not necessarily to be a vector from colony to the empire. β is a number greater than 1, and d is the distance between a colony and an imperialist position. If β >> 1, the colonies reach the status of empires rapidly. θ is a parameter (representing the angle of direct and indirect movement) with uniform (or any proper) distribution. ϴ~U(-γ,γ), γ is an adjustment parameter of deviation from the original direction. However, the values of β and γ are arbitrary. For good convergence, a value of about 2 for β and about π/4 (Radian) for γ has been used [13]. The total power of each empire is determined by the sum of its power of and average powers of its colonies.

$T.C_n$= Cost (imperialist) + ξ×mean{Cost (colonies of empire)} (1),

where $T.C_n$ is the total cost of the n^{th} empire and ξ is a positive number less than one. The continuation of the mentioned steps will hopefully cause the countries to converge to the global minimum of the cost function. Different criteria can be used to stop the algorithm. One idea is to use maximum number of iterations, called maximum decades, or the end of imperialistic competition, when there is only one empire. On the other hand, the algorithm can be stopped when its best solution in different decades cannot be improved for some consecutive decades. The main steps of ICA are: Select some random points and initialize the empires. Move colonies toward their relevant imperialist (Assimilation). Randomly change the position of some colonies (Revolution). If there is a colony in an empire which has higher cost than the imperialist, exchange the positions of that colony with the imperialist. Unite similar empires. Compute the total cost of all empires. Pick the weakest colony (colonies) from the weakest empires and hand over it to one of the empires (Imperialistic competition). Eliminate the powerless empires. Exit if stop conditions are satisfied, otherwise do further assimilation and continue.

4. CLUSTER HEAD SELECTION THROUGH IMPERIALIST APPROACH

We use ICA to select CH by observing two conditions; minimize communication energy consumption and position close to center of gravity (CoG) of deployment. Mostly, if the coordinate of a sensor is close to CoG of deployment, communication energy will also be minimum. Our WSN is a homogeneous and static network (the coordinates of the sensors do not vary with time). Sensor nodes are deployed randomly in the field. A sensor can operate as a CH or as an OS. It is assumed that CHs act as imperialists and OSs act as colonies.

4.1 Initialization

For topology control (TC), all nodes communicate with each other using maximum power to discover neighbours and calculate the energy consumption as primary communication energy (CE). The initial CHs are selected according to the primary CE. The CE of two sensors has direct relation with the distance between them:

$$CE \propto d2$$

and the distance d between two sensor nodes is calculated by:

$$d=\sqrt{(x_{S2}-x_{S1})^2+(y_{S2}-y_{S1})^2}$$

where x and y are the coordinates. An OS's CE is calculated based on the distance between the OS and its CH. If an OS is closer to its CH, it will consume less energy to communicate with its CH. According to the primary cost of sensors, at most half of the sensors can be selected as CH. After the selection of primary CHs, OSs will be assigned randomly to the CHs. The number of OSs assigned to each CH is equal except that the last one may have less or more OSs. To start the main part of program, the new costs are calculated for all sensors according to the new status.

4.2 Assimilation

All OSs try to assimilate to their CHs and change their operation mode from OS to CH; for each sensor a status property is maintained. In each decade, a random number is generated and it is multiplied by

β and added to the status property of OS. If the status of an OS becomes greater or equal to a CH status, the OS and CH positions are exchanged. After this exchange, the new status value is assigned to new OS randomly.

$$\text{Assimilate}_{OS} = \text{Assimilate}_{OS} + \beta \times \text{ranadom value}$$

If there is any exchange, in end of assimilation function, the new cost value is calculated for all sensors

4.3 Revolution

Revolution in ICA is very similar to mutation in GA. The revolution is repeated for all OSs of CHs. Each OS finds a chance randomly to change its mode suddenly from OS to CH and CH to OS. If the random number is less than revolution parameter, then revolution happens for that OS. In addition to exchange of their modes, some other properties such as position, status, etc are exchanged. In the event of a revolution, the new cost is calculated for new CHs and OSs. Revolution can increase cost value.

4.4 Exchange with the best OS

After assimilation and revolution, cost of some OSs or CHs may change. The cost of a new OS may be lower than that of its CH. For OS with lowest cost that is smaller than its CH, exchange its operating mode with its CH. After this exchange, new costs are calculated for all sensors.

4.5 Calculate Total Cost

The total cost of each CH is summation of its cost plus average cost of its OSs multiplied by ξ. ξ is total cost parameter that increases the effect of average cost of OSs in total cost of every CH.

4.6 Imperialistic Competition

CH with highest cost is found and inside it, OS with highest cost is also found. The CH loses the weakest OS and this weakest OS is assigned to another CH by some random method such as roulette wheel selection. New costs are calculated for all nodes. Sometimes after *n* decades, ICA cannot select one empire. In this case, heuristic conditions are used to select the best empire.

5. RESULTS

We do experiments with population sizes 100, 200, and 300 of sensors deployed randomly. Half of these sensors initially are selected as CH. We assume that the β=2 and PR (probability of revolution) = 0.1, and ζ=0.1 and the number of decades = 500.

To select the primary CHs and OSs, we assume that CE is the primary cost and sort the sensors according to this cost and select half of the sensors to act as CH as shown in figure 3.

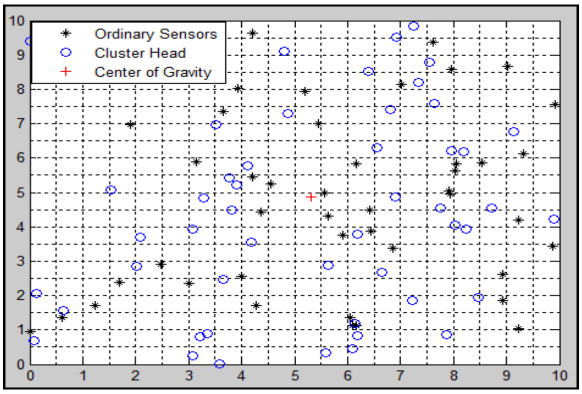

Figure 3. CHs and OSs initialization

Remaining sensors are assigned to these CHs randomly. After this initialization, new costs are calculated. Also, the status of an OS is assigned by a uniform random number between 0 and 10. The main body of ICA is started by assimilation. If the value of β is increased, every OS finds a better chance to change its mode and act as a CH. There is more fluctuation in cost for higher values of β. In revolution, OS finds a chance to change its mode suddenly. Like in the assimilation phase, with higher values of PR, cost values undergo more fluctuations as any OS may find a chance to act as CH affecting the cost value.

In each evolving cluster, there may be an OS that if it acts as a CH the energy cost will decrease. We assume that an OS can be selected as a CH if it has lower CE. If there is an OS with lower cost, it is selected as the new CH in this group of sensors. This rule is implemented for all CHs and OSs. Total cost is calculated by Equation (1). If the value of ζ is increased, the average cost of OS affects more in the total cost of a CH. Sometimes, it results in finding a good result and sometimes not.

After CH competition, if any CH exists without any OS, it changes mode to operate as an OS and randomly assigned to a CH. When the number of CH becomes 1, the program is terminated; otherwise the CH among the many contenders (less than 5) close to the CoG is selected as the winner in this competition. Figure 4 shows that finally one CH is selected out of 100 sensors and it is close to the CoG.

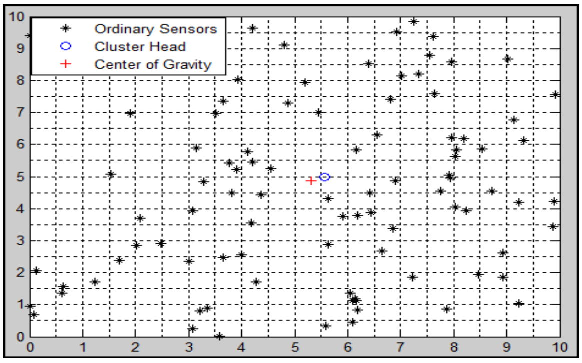

Figure 4. Final selection of CH

Figure 5 shows CE against decades for different size WSN. CE stabilizes after 225 decades in the case 300- size, after 150 decades for 200-size, and after 140 decades for 100-size WSNs.

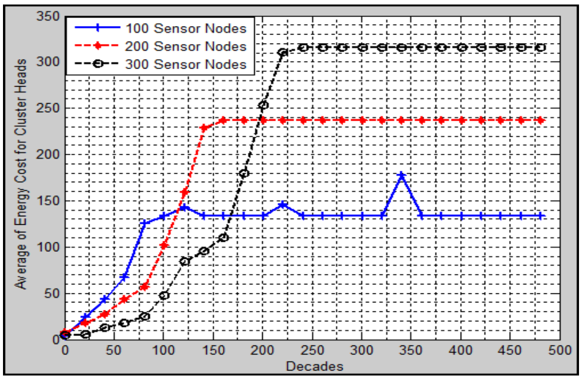

Figure 5. Energy cost versus decades for different size networks

Figure 6 shows the number of CHs for different number of sensors at different decades. It is seen that for larger deployment, the program should run for more decades to reduce the number of final CH. For 100-size, 1 CH is selected after 260 decades, for 200-size and 300-size, 7 and 21 CHs, respectively are selected after 500 decades.

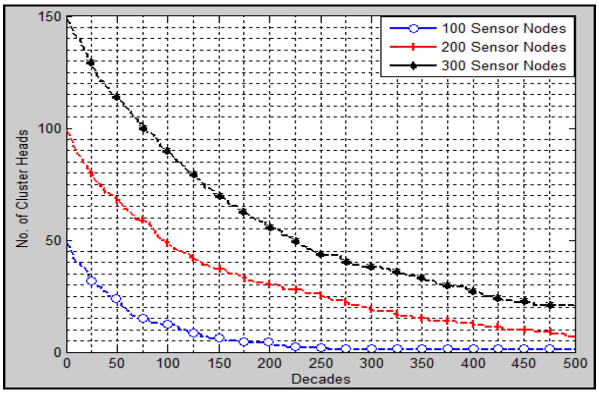

Figure 6. Progressive selection CHs at different decades

In figure7, for smaller size (100) networks, time increases very slowly with decades. For larger size networks, it increases nonlinearly and takes almost exponential shape with a network of 300 sensors.

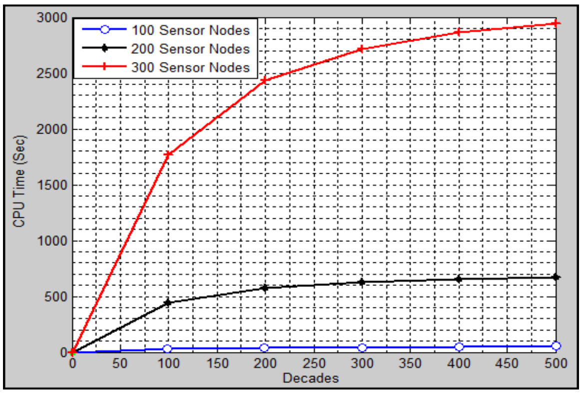

Figure 7. Time against decades

6. CONCLUSION

In this paper, we have demonstrated that ICA is an effective method to select CH in a WSN. ICA either finds one or at most a few CHs within 500 decades. The tie is broken by use of a heuristic. CE stabilizes after 225 decades in the case 300-size, after 150 decades for 200-size, and after 140 decades for 100-size WSNs. For 100-size, 1 CH is selected after 260 decades, for 200-size and 300-size, 7 and 21 CHs, respectively are selected after 500 decades. For reducing the number of final CH in larger size WSNs, the algorithm should be run for more than 500 decades. The algorithm is coded in MATLAB version 7 on Intel(R) core *i*5 CPU 650 3.2 GHz running Windows 7 professional. For smaller size (100) networks, time increases very slowly with decades. For larger size networks, it increases nonlinearly and takes almost exponential shape with a network of 300 sensors. This is a preliminary study and we plan further investigation in this direction including the effects of β and ξ on the outcome.

7. REFERENCES

[1] I.F. Akyildiz, W. Su, Y. Sankarasubramaniam, E. Cayirci, “A survey on sensor networks”, IEEE Communications Magazine 40 (8), pp 104 –112, 2002.

[2] Estrin D., Govindan R., Heidemann J., Kumar S., Next century challenges: Scalable coordination in sensor networks. In Proceedings of the ACM Mobicom, pp 263–270, 1999.

[3] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, E. Cayirci, “Wireless sensor networks: a survey”, Computer Networks 38, Elsevier, pp 393-422, 2002.

[4] Bhaskar Krishnamachari, Networking Wireless Sensors, Cambridge University Press, 2005.

[5] Ian F. Akyildiz, Ismail H. Kasimoglu, “Wireless sensor and actor networks: research challenges”, Ad Hoc Networks 2, Elsevier, pp 351-367, 2004.

[6] Paolo Santi, “Topology Control in Wireless Ad Hoc and Sensor Networks”, ACM Computing Surveys, Vol. 37, No. 2, pp 164–194, 2005.

[7] W. R. Heinzelman, A. Chandrakasan, and H. Balakrishnan, “Energy efficient communication protocol for wireless micro-sensor networks”, Proc. of the 33rd ICSS, 2000.

[8] Jun Zheng, Abbas Jamalipour, Wireless Sensor Networks: A Networking Perspective, John Wiley & Sons, 2009.

[9] A. Khosrozadeh, H. Motameni, “Survey in Stable Coverage Guarded for Wireless Sensor Network”,

American Journal of Scientific Research, Issue 22, pp 6-17, 2011.

[10] W. Heinzelman, A. Chandrakasa, and H. Balakrishnan, "Energy Efficient Communication Protocols for Wireless Microsensor Networks", Proceedings of Hawaian International Conference on Systems Science, Jan. 2000.

[11] C. Nam, H. Jeong, and D. Shin, “The Adaptive Cluster Head Selection in Wireless Sensor Networks", IEEE International Workshop on Semantic Computing and Application, pp 147-149, July 2008.

[12] C. Nam, Y. Ku, J. Yoon, and D. Shin, "Cluster Head Selection for Equal Cluster Size in Wireless Sensor Networks", in Proc. IEEE New Trends in Information and Service Science, NISS, pp 618-623, July 2009.

[13] Atashpaz-Gargari E., Lucas C., “Imperialist Competitive Algorithm: An algorithm for optimization inspired by imperialistic competition”, IEEE Congress on Evolutionary Computation 7, pp 4661–4666, 2007.

[14] R. L. Haupt and S. E. Haupt, Practical Genetic Algorithms, Second Edition, John Wiley & Sons, 2004.

[15] M. Melanie, An Introduction to Genetic Algorithms, MIT Press, 1999.

[16] M. Dorigo and C. Blum, “Ant colony optimization theory: A survey”, Theoretical Computer Science, 344, pp 243 – 278, 2005.