Number of Ants Versus Number of Iterations on Ant Colony Optimization Algorithm for Wireless Sensor Layout

\S. Fidanova and P. Marinov

Institute of Information and Communication Technologies—BAS, Acad. G. Bonchev Str. Block 25A, 1113 Sofia, Bulgari, (stefka, pencho\}@parallel.bas.bg,

The Wireless Sensor Networks (WSN) have large applications in our modern life. Some of the applications are monitoring, security, civil engineering and so on. The most important in network construction is fool coverage and lifetime of the network. The objectives of the problem are to position minimal number of sensors with minimal energy consumption. It is a hard combinatorial optimization problem for which is unpractical to apply exact algorithms or traditional numerical methods. The most appropriate is to apply metaheuristic method. We chose multi-objective Ant Colony Optimization (ACO) to solve this important telecommunication problem. We study the influence of the number of ants on the quality of the solution.

1. INTRODUCTION

Nowadays, telecommunication networks are highly decentralized, multi-node networks. From small size-limited local area networks the evolution has led to the huge worldwide Internet. Thus the Wireless Sensor Networks become a hot topic in research. A WSN allows to automatically monitor almost any phenomenon with a precision unseen to the date. WSN have been employed in military for reconnaissance, surveillance, and target acquisition [2], environmental activities such as forest fire prevention, volcano eruptions study [14], biomedical purposes such as health data monitoring [16] or civil engineering [11].

The WSN consists of sensors which can sense things of different types such as seismic, acoustic, chemical, optical, etc., and the communication is performed wireless. The small size and energy capacity of the sensors prevent them to send collected information directly to the base. They transmit their data to a high energy communication node (HECN), which communicates with the main computer for further processing. All sensors must be able to transmit their data to this node, either directly or via hops, using nearby sensors as communication relays.

Metaheuristics offer good versatility for solving hard combinatorial optimization problems. These methods imply a progressive migration from a continuous and weak informational content toward a strong informational content. In this paper we propose a solution method for the WSN layout problem using ACO. We focus on both minimizing the energy depletion of the nodes in the network and minimizing the number of nodes, while the full coverage of the network and connectivity are considered as constraints. Our research is focused on the influence of the number of used ants versus number of iterations.

Jourdan [7] solved an instance of WSN layout using a multi-objective genetic algorithm. In their formulation a fixed number of sensors had to be placed in order to maximize the coverage. In some applications most important is the network energy. In [6] is proposed ACO algorithm and in \cite{Wolf} is proposed evolutionary algorithm for this variant of the problem. In [3] is proposed ACO algorithm taking in to account only the number of the sensors and in [12] the problem is converted to mono-objective. In [9] are proposed several evolutionary algorithms to solve the problem. In [8] is proposed genetic algorithm which achieves similar solutions as the algorithms in [9], but it is tested on small test problems.

The paper is organized as follows. In 2 the WSN is introduced and the positioning problem is formulated. In Section 3 the multi-objective optimization is described. Section 4 presents the ACO algorithm. In Section 5 we show the experimental results. Conclusion is in the Section 6.

2. WIRELESS SENSOR NETWORK LAYOUT PROBLEM

A WSN consists of spatially distributed sensors which monitor physical or environmental conditions, such as temperature, sound, vibration, pressure, motion pollutants etc. Each node in a sensor network is equipped with a wireless communications device, a small micro-controller, and an energy source, usually a battery. Each sensor node sens an area around itself called its sensing area. The sensing radius determines the sensitivity range of the sensor node and thus the sensing area. The nodes communicate among themselves using wireless communication links. These links are determined by a communication radius. A special node in the WSN called High Energy Communication Node (HECN) is responsible for external access to the network. Therefore, every sensor node in the network must have communication with the HECN. Since the communication radius is often much smaller than the network size, direct links are not possible for peripheral nodes. A multi-hop communication path is then established for those nodes that do not have the HECN within their communication range.

To determine the energy spent by communications, the number of transmissions every node performs is calculated. The WSN operates by rounds: In a round, every node collects data and sends it to the HECN. Every node transmits the information to the neighbor that is closest to the HECN or the HECN itself if it is within the communication range. When several neighbors are tied for the shortest distance from the HECN, the traffic is distributed among them. That is, if a node has n neighbors tied for shortest distance from HECN, each one receives 1/n of its traffic load. Therefore, every node has a traffic load equal to 1 (corresponding to its own sent data) plus the sum of all traffic loads received from neighbors that are farther from the HECN.

The WSN layout problem a non-fixed amount of sensor nodes has to be placed in a terrain providing full sensitivity coverage. The positions of the nodes have to be chosen in a way that minimizes the energy of spent in communications by any single node, while keeps the connectivity of the network. These are opposed objectives since the more nodes there are the lesser share of retransmissions they bear.

The sensing area of the WSN is the union of the individual areas of all nodes. The designer wants the network to cover the complete sensing area. On the other hand, the number of sensor nodes must be kept as low as possible, since using many nodes represents a high cost of the network, possibly influences of the environment and also provokes a probability of detection (when stealth monitoring is designed). The objective of this problem is to minimize network energy and the number of sensors deployed while the area is fully covered and connected. In this paper, we formulate the problem of optimal placement of sensors as a bi-criteria optimization problem, emphasizing the two major objectives which derive from the main functions of the sensors: communication and sensing.

3. MULTI-OBJECTIVE OPTIMIZATION

In multi-objective optimization (MOP) are optimized two or more conflicting objectives subject to certain constraints. If a multi-objective problem is well formed, there should not be a single solution that simultaneously optimizes each objective. In each case an objective must have reached a point such that, when attempting to optimize the objective further, other objectives suffer as a result. Finding such a solution, and quantifying how much better this solution is compared to many other such solutions, is the goal when setting up and solving a multi-objective optimization problem.

Multi-objective optimization has his roots in the nineteenth century in the work of Edgeworth and Pareto in economics [10]. The optimal solution for MOP is not a single solution as for mono-objective optimization problems; it is a set of solutions defined as Pareto optimal solutions. A solution is Pareto optimal if it is not possible to improve a given objective without aggravate at least another objective. The main goal of the resolution of a multi-objective problem is to obtain the Pareto optimal set and consequently the Pareto front.

One solution dominates another if minimum one of its components is better than the same component of other solution and other components are not worse. The Pareto front is the set of non dominated solutions. The main of goal of metaheuristics is to obtain an approximation of the Pareto front.

4. ANT COLONY OPTIMIZATION APPROACH

We solve the problem with ant colony optimization (ACO). The ACO algorithm uses a colony of artificial ants. They behave as cooperative agents in a mathematic space and they search and reinforce pathways (solutions) in order to find the optimal ones. The problem is represented by graph and the ants walk on the graph to construct solutions. A solution is represented by a path in the graph or by tree in a graph. Ants construct feasible solutions, starting from random nodes, then the pheromone trails are updated. At each step ants compute a set of feasible moves and select the best one (based on heuristic function) to carry out the rest of the tour. The transition probability, to choose the node *j* when the current node is *i*, is based on the heuristic information and on the pheromone trail levelof the move, where $i,j=1,\dots,n$.

(1)

It is more profitable to select a move with higher value of the pheromone and the heuristic information. In the beginning, the initial pheromone level is set to a small positive constant value and then ants update this value after completing the construction stage [1]. ACO algorithms adopt different criteria to update the pheromone level.

In our implementation we use MAX-MIN Ant System (MMAS) [13], which is one of the best ant approaches for which is proven to converge to the global optima. The main feature of MMAS is using a fixed upper bound and a lower boundof the pheromone trails. Thus the accumulation of big amounts of pheromone by part of the possible movements and repetition of same solutions is partially prevented. The main features of MMAS are:

- Strong exploration to the space around the best found solution. This can be achieved by allowing only the best ant to add pheromone after each iteration. We modify this feature. Only the ants found non-dominated solutions allowed to add pheromone. Thus the algorithm becomes more appropriate for multi-objective optimization.

- \Wide exploration of the best solution. After the first iteration, the pheromone trails are reinitialized to.

The pheromone trail update rule is given by:

(2)

Where

Here $F(k)$ is the fitness function of the solution achieved by the k^{th} ant and $i,j = 1,\dots,n$, models evaporation in the nature. The fitness function is a function which is used to estimate solution. The aim is to add more pheromone on non-dominated solutions and thus to force the ants to search around them for new non-dominated solutions. The fitness function we constructed is as follows:

(3)

Where is the number of the sensors achieved by the k^{th} ant and \$f_2(k)\$ is the energy of the solution of the k^{th} ant, or these are the objective functions of the WSN layout problem. We divided the values of the two objective functions with their maximal achieved values from the first iteration. This fitness function we use as objective function in our previous work [5], where we solve the problem like mono-objective.

To avoid stagnation of the search, the range of possible pheromone values on each movement is limited to an interval . is an asymptotic maximum of and ,
while .

The WSN layout problem is represented by graph as follows: the terrain is modeled by grid; the pheromone is related with location sites, the initial pheromone can be a small value, for example . The point, where the HECN is located, is included in the solutions like first point (zero point). Every ant starts to create the rest of the solution from a random node which communicates with central one, thus the different start of every ant in every iteration is guaranteed. The ant chooses the next position by the ACO probabilistic rule (equation 1). It chooses the point having the highest probability. If there is more than one point with same probability, the ant chooses one of them randomly. The construction of heuristic information is one of the crucial points of the ACO algorithm. The heuristic information needs to be constructed thus, to manage the ants to look for better solutions. For some kinds of problems it is not obvious how to prepare it. One needs to combine different elements of the problem to most appropriate way.

Our heuristic information is as follows:

(4)

where

is the number of points which the new sensor will cover,

(5)

b is the solution matrix and the matrix element when there is sensor on this position otherwise . With we try to locally increase the covered points, more new covered points leads eventually to less number of sensors. With we guarantee that all sensors will be connected; with rule we guarantee that the position is not chosen yet and no more than one sensor will be mapped on the same position. When for all values of *i* and *j,* the search stops. Thus, the construction of the solution stops if no more free positions, or all points are covered or new communication is impossible.

5. \EXPERIMENTAL RESULTS

Every ant starts to create their solution from random point. In our case it is a random point which communicates with the HECN. Thus the ant algorithm uses small number of agents (ants). Less number of ants means less memory, which is important when we solve large problems. The aim of this work is to learn the influence of the number of the sensors versus number of iterations.

We have created a software which realizes our ant algorithm. Our software can solve any rectangular area, the communication and the coverage radius can be different and can have any positive value. The HECN can be fixed in any point in the area. The program is written in C language and the tests are run in computer with Intel Pentium processor with 2.8 GHz. In our tests we use an example where the area is square and consists of 500 points in every side. The coverage and communication radii are cover 30 points. The HECN is fixed in the center of the area.

In our previous work [5] we show that our ant algorithm outperforms existing algorithms for this problem. There after several runs of the algorithm we specify the most appropriate values of its parameters. We apply MAX-MIN ant algorithm with the following parameters: , .

In ACO if we fix the number of iterations and double the number of ants the execution time will be doubled. If we fix the number of ants and double the number of iterations the execution time will e doubled too. To have correct study of the influence of the number of ants versus number of iterations we fix the product of the number of ants and iteration to be 60. Thus all runs will take the same time. We vary the number of ants to have following values $\{1, 2, 3, 4, 5, 6, 10, 12, 15, 20, 60\}$ and the number of the iterations to be respectively {60, 20, 15, 12, 10, 6, 5, 4, 3, 2, 1\}. We run the ACO algorithm 30 times with every of the combinations *Number-ants*Number-iterations* = 60, and find the Pareto front.

Table 1. Pareto fronts

ant senso r	1	2	3	4	5	6	10	12	15	20	30	60
240			53									
241			54	50	50	50	50	50	50	50	50	50
238	53	53		53				52				
237												
236								54				
235				54	51	51	51		51	51	51	51
234				56				56	55	55	56	
233										57		
232						52	52		57			
230	58	58										
229	60	59	56		56		57		58	58	57	57
228		60				57		57	60		58	58
227		61	57		58		58	68	61	59	61	61
226			68	57		60						
225			79									

Analyzing Table 1we observe that the algorithm achieves the best Pareto front when the number of ants is 6 and the number of iterations is 10 and when the number of ants is 10 and the iterations is 6.

We also use the Hyper volume [17] as a quality indicator of achieved solutions. Mathematically, for each solution a hypercube is constructed with a reference point W and the solution as the diagonal corners of the hypercube. The reference point can be found simply by constructing a vector of worst objective function values. Thereafter, a union of all hyper cubes is found and its hyper volume (IHV) is calculated. Algorithms with larger IHV values are desirable. The hyper volume in the case (6, 10) is equal to 0.9508 and the hyper volume in the case (10, 6) is equal to 0.9344. The hyper volumes in the both case are very similar. Thus the case 10 ants is slightly better because they find larger Pareto front than the case 6 ants.

6. CONCLUSION

We propose a multi-objective ACO algorithm which solves the Wireless Sensor Network layout problem. We focus on learning the influence of number of ants versus number of iteration on algorithm performance. Observing the achieved results we conclude that using a lot of number of ants and small number of iterations and small number of ants and a lot of number of iterations does not lead to achieving good solutions. The best Pareto front we achieve with 6 and 10 ants.

Acknowledgment
This work has been partially supported by the European Commission under the grant ACOMIN.

REFERENCES

1.E. Bonabeau, M. Dorigo, G. Theraulaz, Swarm Intelligence: From Natural to Artificial Systems, Oxford University Press, 1999.
2.K. Deb, A. Pratap, S. Agrawal, T. Meyarivan, A Fast and Elitist Multi-objective Genetic Algorithm: Nsga-ii, IEEE Transactions on Evolutionary Computation, Vol.{6}(2), 2002, 182--197.
3.S. Fidanova, P. Marinov, E. Alba, Ant Algorithm for Optimal Sensor Deployment, Computational Inteligence, K. Madani, A.D. Correia, A. Rosa, J. Filipe (eds.), Studies of Computational Inteligence {399}, Springer, 2012, 21 - 29.
4.S. Fidanova, K. Atanasov, Generalized Net Model for the Process of Hibride Ant Colony Optimization,Comptes Randus de l'Academie Bulgare des Sciences, Vol.{62}(3), 2009, 315 - 322.
5.S. Fidanova, M. Shindarov, P. Marinov, Multi-Objective Ant Algorithm for Wireless Sensor Network Positioning, Comptes Randus de l'Academie Bulgare des Sciences, Vol. (63)(2), 2013.
6.H. Hernandez, C. Blum, Minimum Energy Broadcasting in Wireless Sensor Networks: An ant Colony Optimization Approach for a Realistic Antenna Model, J. of Applied Soft Computing, Vol.{11}(8), 2011, 5684--5694.
7.D.B. Jourdan, Wireless Sensor Network Planning with Application to UWB Localization in GPS-denied Environments, Massachusets Institute of Technology, PhD thesis, 2000.
8.A. Konstantinidis, K. Yang, Q. Zhang, D. Zainalipour-Yazti, A multi-objective Evolutionary Algorithm for the deployment and Power Assignment Problem in Wireless

sensor Networks, J. of Computer networks, Vol.{54}(6), 2010, 960--976.
9.G. Molina, E. Alba, El-G. Talbi, Optimal Sensor Network Layout Using Multi-Objective Metaheuristics, Universal Computer Science Vol.{14}(15), 2008, 2549--2565.
10.V. K. Mathur}, How Well do we Know Pareto Optimality?,J. of Economic Education Vol.{22}(2), 1991, 172 - 178.
11.J. Paek, N. Kothari, K. Chintalapudi, S. Rangwala, R. Govindan, The Performance of a Wireless Sensor Network for Structural Health Monitoring, In Proc. of 2nd European Workshop on Wireless Sensor Networks, Istanbul, Turkey, Jan 31 -- Feb 2, 2005.
12.M. Shindarov, S. Fidanova, P. Marinov, Wireless Sensor Positioning Algorithm, In. Proc. of IEEE Conf. on Intelligent Systems, Sofia, Bulgaria, 2012, 419 - 424.
13.T. Stutzle, H.H. Hoos, MAX-MIN Ant System, Future Generation Computer Systems Vol.{16}, 2000, 889--914.
14.G. Werner-Allen, K. Lorinez, M. Welsh, O. Marcillo, J. Jonson, M. Ruiz, J.Lees, Deploying a Wireless Sensor Network on an Active Volcano, IEEE Internet Computing Vol.{10}(2), 2006, 18--25.
15.S. Wolf, P. Mezz, Evolutionary Local Search for the Minimum Energy Broadcast Problem, in C. Cotta, J. van Hemezl (eds.), VOCOP 2008, Lecture Notes in Computer Sciences No. 4972, Springer, Germany, 2008, 61--72.
16.M. R. Yuce, S. W. Ng, N. L. Myo, J. Y. Khan, W. Liu, Wireless Body Sensor Network Using Medical Implant Band, Medical Systems Vol.{31}(6), 2007, 467--474.
17.E. Zitzler, L. Thiele, Multiobjective Evolutionary Algorithms: A Comparative Case Study and the Strength Pareto Approach, IEEE Transactions on Evolutionary Computation, Vol.{3}(4), 1999, 257--27