Optimal Wireless Sensor Network Coverage with Ant Colony Optimization

Stefka Fidanova, Pencho Marinov

IICT-BAS, Acad. G. Bonchev str bl. 25A,
1113 Sofia, Bulgaria
stefka@parallel.bas.bg, pencho@parallel.bas.bg

Abstract

Wireless Sensor Network is a fast growing and exciting research area that has attracted considerable research attention in the recent past. The creation of large-scale sensor networks interconnecting several hundred to a few thousand sensors nodes opens up several technical challenges and immense application possibilities. We discuss in this paper wireless sensor network coverage problem. This problem consists in placing sensors so as to get the best possible coverage while saving as many sensors as possible. To solve it we prepare an Ant Colony Optimization (ACO) algorithm. We compare our results with existing metaheuristic algorithms.

Key words

Wireless sensor network, metaheuristics, ant colony optimization

1 Introduction

At the present time, emerging new technologies, which allow fast data communications and new services and applications, are in widespread use around the globe. In this situation, there is a renewed interest in related technology and communication network problems. Mobile communications is a major area in the industry of the twenty-first century. These types of services are required more and more. Numerous companies have entered this area and began to compete to offer the best services of the low cost. Therefore, a great number of issues have arisen as problems to be solved in order to optimize the features of the service. Sensor networks find applications spanning several domains including military, medical, industrial, and home networks. Wireless Sensor Networks (WSN) have moved from the research domain into the real world with the commercial availability of sensors with networking capability.

Since the size of the existing networks is continuously enlarging, the underlying instances of related optimization problems frequently pose a challenge to existing algorithms. Wireless networks started as being composed by a small number of devices connected to a central node. Recent technological developments have enabled smaller devices with computing capabilities to communicate in the absence of any infrastructure by forming ad-hoc networks and goes toward a new paradigm: Wireless Sensor Networks [1]. A WSN allows an administrator to automatically and remotely monitor almost any phenomenon with a precision unseen to the date.

When deploring WSN, the positioning of the sensor nodes becomes one of the major concerns. The coverage obtained the network and the economic cost of the network depends directly of it. Since many of WSN can have large number of nodes, the task of selecting the geographical positions of the nodes for an optimal designed network can be very complex. Therefore, metaheuristics seem an interesting option to solve this problem.

In this paper we propose a solution method for the WSN coverage problem using ACO algorithm. We focus on minimizing the number of nodes, while maximizing the coverage of the network. Jordan [10] solved an instance of WSN layout using a multiobjective genetic algorithm. In there formulation a fixed number of sensors had to be placed in order to maximize the coverage. In [2] are proposed several algorithms to solve the problem. For solving the WSN layout problem, the coverage has to satisfy some restrictions and the biggest possible coverage will be preferred: the number of sensors or nodes should be kept low for economical reasons.

The rest of the paper is organized as follows. In Section 2 the WSN is described and the coverage problem is formulated. Section 3 presents the ACO algorithm employed for solving this problem. Then in Section 4 the experiments performed and the results obtained are analyzed. Section 5 shows the conclusions and directions for future work.

2 WSN coverage problem

The positioning of nodes in sensor network has received a notable attention in research. Zhang [11] study the positioning of sensors in a terrain from the point of view of data transmission. In [4] is studied different regular positioning for sensors: square, triangular and hexagonal grids. They observe a tradeoff between node density and fault-tolerance. In a similar approach in [8] is proposed systematic placing methods to ensure connected coverage to 2-dimensionnal sets of points, which approach the minimum number of sensor nodes required. In [6] are proposed two greedy algorithms that select the locations for a sensor network with minimal number of nodes. They consider a probabilistic coverage model for the sensors where the probability of coverage for any point by given sensor decreases exponentially with its distance from the sensor.

In this section we describe the coverage problem for WSN, then present the formulation employment for its resolution. A WSN allows monitoring some physical set of parameters in a region known as the sensor field. When a WSN is placed in the sensor field, every sensor monitors a region of the field, ideal the complete network is able to monitor the entire field by adding all the pieces of information together. It is the duty of the designer to establish the sensor field, which the WSN has to monitor. The area that a single sensor can sense can be modeled with a circle whose radius R_{sens} or sensing radius indicates the sensing range of the sensor. Similarly R_{comm}, the communication radius of a sensor, defines the circle where any sensor can establish a direct communication link with it. The value of this range depends on the environment, the radio hardware, the power employed etc.

When a WSN is deployed in the sensor field, the sensors form a wireless ad-hoc network in order to communicate their sensing results to a special station called the High Energy Communication Node (HECN). The data can then be analyzed by the HEKN processor or network administrator. The sensing area of the WSN is the union of the individual sensing areas of all the usual nodes. The designer wants the network to cover as much of the area as possible. On the other hand, the number of sensor nodes be kept as low as possible, since using many nodes represent a high cost of the network, possibly influence the environment, and also provide a high probability of detection (if stealth monitoring is desired).

The problem of designing the layout for a WSN can be defined as an extension of an existing problem: the radio network design problem (RND) [3]. The objective of this problem is to maximize the sensing area of the network while minimizing the number of sensors deployed. The Available Location Sites (ALS) for placing the sensors are given as an ordered list that constitutes the specific problem instance. Figure 1 shows a graphical example of WSN layout (random solution).

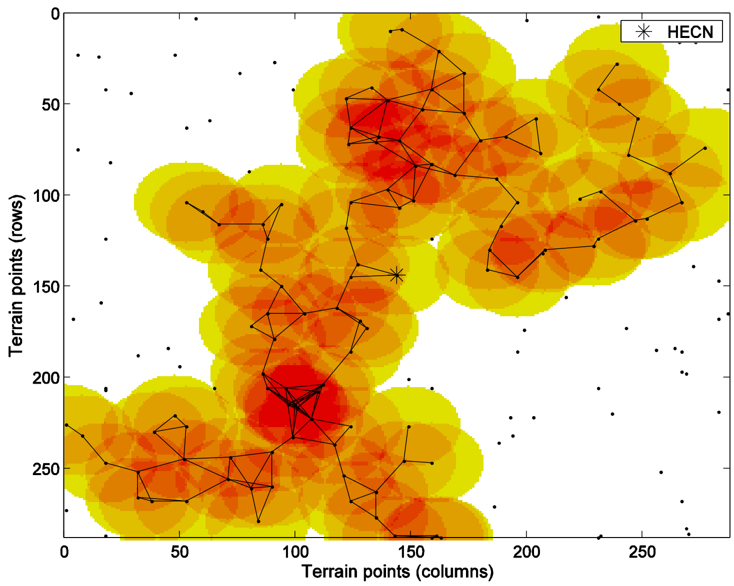

Figure 1 Random solution.

From the previous definition of the problem, a fitness function, that combines both objectives is employed (equation 1).The objective is to maximize the fitness value of the solution.

$$f(x) = \frac{Coverage(x)^2}{Nb.of\ sensors}, \qquad Coverage(x) = 100\frac{Covered\ points}{Total\ points} \tag{1}$$

3 Ant Colony Optimization

Many of the existing solutions to this problem come from the field of Evolutionary Computation [2,3]. After analyzing them, we noticed that these interesting developments are quite similar to ACO algorithms. The relation between ACO algorithms and evolutionary algorithms provides a structural way of handling constrained problems. They have in common the use of a probabilistic mechanism for recombination of individuals. This leads to algorithms where the population statistics are kept in a probability vector. These probabilities are used to generate new solutions, in each iteration. The new solutions are then used to adapt the probability vector.

Real ants foraging for food lay down quantities of pheromone (chemical cues) marking the path that they follow. An isolated ant moves essentially guided by an heuristic function and an ant encountering a previously laid pheromone will detect and decide to follow it with high probability thus taking more informed actions based on the experience of previous ants (and thereby reinforce it with a further quantity of pheromone). The repetition of the above mechanism represents the auto-catalytic behavior of real ant colony where the more the ants follow a trail, the more attractive that trail becomes.

The ACO algorithm uses a colony of artificial ants that behave as cooperative agents in a mathematic space were they are allowed to 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. The solution is represented by a path in the graph. After initialization of the pheromone trails, 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 (according to some probabilistic rules based on a heuristic guided function) to carry out the rest of the tour. The structure of ACO algorithms is shown in Figure 2.

Ant Colony Optimization

Initialize number of ants;
Initialize the ACO parameters;
while not end-condition **do**
  for k=0 **to** number of ants
    ant k starts from a random node;
    while solution is not constructed **do**
      ant k selects higher probability node;
    end while
  end for
  Local search procedure;
  Update-pheromone-trails;
end while

Figure 1: Pseudo code for ACO.

The transition probability p_{ij}, to choose the node *j* when the current node is *I*, is based on the heuristic information η_{ij} and on the pheromone trail level τ_{ij} of the move, where $i, j = 1, \ldots, n.$

$$p_{ij} = \frac{\tau_{ij}^{\alpha}\eta_{ij}^{\beta}}{\sum_{k \in allowed} \tau_{ij}^{\alpha}\eta_{ij}^{\beta}} \tag{2}$$

The higher value of the pheromone and the heuristic information, the more profitable is to select this move. In the beginning, the initial pheromone level is set to a small positive constant value τ_0 and then ants update this value after completing the construction stage [5]. ACO algorithms adopt different criteria to update the pheromone level. In our implementation we use MAX-MIN ANT System (MMAS) [9], which is one of the most popular ant approaches. The main feature of MMAS is using a fixed upper bound $\tau_{\max}$ and a lower bound $\tau_{\min}$ of the pheromone trails. Thus the accumulation of big amounts of pheromone by part of the possible movements and repetition of same solution is partially prevented. The main features of MMAS are:

- Strong exploration of the space search of the best found solution. This can be achieved by only allowing one single ant to add pheromone after each iteration (the best one).
- Wide exploration of the best solution. After the first iteration, the pheromone trails are reinitialized to $\tau_{\max}$. In the next iteration, only the movements that belong to the best solution receive a pheromone, while the rest pheromone values are only evaporated.

The aim to use only one solution is to make the solution components, which frequently occur in the best found solutions, get a larger reinforcement. The pheromone trail update rule is given by:

$$\tau_{ij} \leftarrow \rho\tau_{ij} + \Delta\tau_{ij} \tag{3}$$

$$\Delta\tau_{ij} = \begin{cases} 1/C(V_{best}) & if\ (i,j) \in best\ solution \\ 0 & otherwise \end{cases},$$

Where V_{best} is the iteration best solution and $i, j = 1, \ldots, n,\ \rho \in [0,1]$ models evaporation in the nature. To avoid stagnation of the search, the range of possible pheromone values on each movement is limited to an interval $[\tau_{\min}, \tau_{\max}]$. $\tau_{\max}$ is an asymptotic maximum of τ_{ij} and $\tau_{\max} = 1/(1-\rho)C(V^*)$, while $\tau_{\min} = 0.087\tau_{\max}$. Where *V** is the optimal solution, but it is unknown, therefore we use V_{best} instead of *V**.

The WSN layout problem is represented by graph as follows: the terrain is modeled by grid $G = \{g_{ij}\}_{N \times M}$; the pheromone is related with location sites $Ph = \{ph_{ij}\}_{N \times M}$, the initial pheromone can be a small value, for example $1/n_{ants}$. The central point, where the HECN is located, is included in the solution 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 higher probability.

The used heuristic information is:

$$\eta_{ij} = s_{ij} l_{ij} (1 - b_{ij}) \tag{4}$$

Where s_{ij} is the number of uncovered points which the new sensor will cover, and

$$l_{ij} = \begin{cases} 1 & \textit{if communication exists} \\ 0 & \textit{if there are is not communication} \end{cases}$$

B is the solution matrix and the matrix element $b_{ij} = 1$ when there is a sensor on this position, otherwise $b_{ij} = 0$. With s_{ij} we try to locally increase the covered points, with l_{ij} we guarantee that all sensors will be connected; with rule $(1 - b_{ij})$ we guarantee that the position is not chosen yet. When $p_{ij} = 0$ for all values of *i* and *j* the search stops. Other stopping criterion is the increasing of objective function, when new sensor is included. Thus, the construction of the solution stops if no more free positions or all points are covered or new communication is impossible or new sensor increases the objective function.

4 Computational Results

In this section we describe the experiment and the results obtained using ACO algorithm described in Section 3. The results are then analyzed and compared with results achieved by other algorithms. The solved instance in this work is very large (1000 available points). The sensor field is modeled by a 287×287 point grid. All sensors behave equally and both their sensing and communication radii are set to 22 terrain points. The ALS is formed by 1000 locations randomly distributed over the sensor field following a uniform distribution. On Figure 1 is illustrated a random solution for this instance using 167 sensors and covering 56.76% of the sensor field. The low quality achieved by the random search and the NP hardness of the problem suggest the utilization of metaheuristics.

The problem is solved using MAX-MIN ant algorithm with following parameters: $\alpha = \beta = 1$, $\rho = 0.5$, The number of used ants is 3. We analyze the algorithm effectiveness for solving the problem by inspecting the fitness obtained. The influence of the number of solution evaluations will also be studied by running several experiments using increasing number evaluations. The number of evaluations will range from 10 to 100. For every experiment the results are obtained by performing 30 independent runs, then averaging the fitness values obtained in order to ensure statistical confidence. We compare achieved by ACO algorithm results with results from [2] achieved by using Simulated Annealing (SA) and Cross generational elitist selection Heterogeneous recombination and Cataclysmic mutation (CHC), a kind of evolutionary algorithm, which works with a set of solutions (population). Table 1 summarizes the results obtained by SA and CHC. To obtain good results the both algorithms need a huge amount of fitness evaluations.

evaluations	50 000	100 000	200 000	300 000	400 000	500 000	1 000 000
SA	74.793	76.781	78.827	79.836	80.745	81.602	84.217
CHC	75.855	83.106	87.726	89.357	90.147	90.974	92.107

Table 1: Fitness results by SA and CHC

Table 2 shows the results obtained by ACO algorithm.

Evaluations	10	20	100
ACO	78.125	85.470	87.085

Table 2: Fitness results by ACO

On the tables above we observe that the average fitness obtained with either SA or CHC or ACO improves when the number of evaluations is increased. The SA average fitness goes from 74.793 for 50 000 evaluations to 84.217 for 1 000 000 evaluations. The CHC average fitness goes from 75.855 for 50 000 evaluations to 92.107 for 1 000 000 evaluations. The ACO average fitness goes from 78.125 for 10 evaluations to 87.085 for 100 evaluations.

Analysis of the data shows that the ACO algorithm needs much less evaluations to achieve good results in comparison with SA and CHC. Moreover after 20 evaluations ACO algorithm achieves better solutions than SA after 1 000 000 evaluations. After 100 evaluations ACO algorithm achieves similar results to CHC achieved after 200 000 (which is 2000 times more). It is proven in [7] that MAX-MIN ant algorithm converges to the global optimum, when the number of evaluations converges to the infinity. Thus we can conclude that our ACO algorithm proceeds better than SA and CHC for this problem.

5 Conclusion

We have defined a coverage problem for wireless sensor networks. A very large problem consisting 1000 available locations has been solved using ACO algorithm. We compare achieved results with results from [2] achieved by other metaheuristics: SA and CHC. In our experiments ACO is able to reach similar or higher fitness values with an effort (number of performed solution evaluations) than 2000 times less than the effort required by SA and CHC. The average fitness obtained by any of the algorithms improves it the allowed number of evaluations is increased (10 to 100 for ACO, 50 000 to 1 000 000 for SA and CHC). In a future work we plan to redefine the problem to be able to solve more complex WSN problems with regions in a sensing area where to put sensors is forbidden and network problem with obstacles. Other interesting direction is to study the robustness of the solutions to minimize the disturbance of the network, when single sensor fail and thus to avoid segmentation of the network.

Acknowledgements

This work has been partially supported by the Bulgarian National Scientific Fund under the grants ”Modeling Processes with fixed development rules” DID 02/29 and ”Effective Monte Carlo Methods for large-scale scientific problems” DTK 02/44.

References

[1] AKYILDIZ L., SU W., SANKASUBRAMANIAM Y., CAYICIi E., *A Survey on Sensor Networks,* IEEE Communications Magazine, 2002.

[2] ALBA E., MOLINA G., *Optimal Wireless Sensor Network Layout with Metaheuristics: Solving Large Scale Instance,* Large Scale Scientific Computing, I. Lirkov, S. Margenov, J Wasniewski eds., Lecture Notes in Computer Science No 4818, Springer Germany, pp. 527 – 535, 2008.

[3] ALBA E., MOLINA G., CHICANO F., *Optimal Placement of Antenae Using Metaheuristics,* Numerical Methods and Applications, Lecture Notes in Computer Science, Springer, Germany, 2007.

[4] BIAGIONI E., SASALAKI G., *Wireless Sensor Placement for Reliable and Efficient Data Collection,* Hawaii Int. Conf. Sys. Sci., 2003.

[5] BONABEAU E., DORIGO M., THERAULAZ G., *Swarm Inteligence: From Natural to Artificial Systems,* Oxford University Press, New York, 1999.

[6] DHILLON S., CHAKRABARTY K., *Sensor Placement for Effective Coverage and Surveillance in Distributed Sensor Networks,* IEEE Wirel. Comm. Netw. Conf., pp. 1609 – 1624, 2003.

[7] DORIGO M., STUTZLE T., *Ant Colony Optimization,* MIT Press, 2004.

[8] KAR K., BANERJEES N., *Node Placement for Connected Coverage in Sensor Networks,* In Proc of WiOpt, 2003.

[9] STUTZLE T., HOOS H., *Max-Min Ant System,* Future Generation Computer Systems 16, pp. 889 – 914, 2000,

[10] YORDAN D., *Wireless Sensor Network Planning with Application to UWB Localization in GPS-denied Environments,* PhD thesis, Massachusetts Institute of Technology, USA, 2000.

[11] ZHANG H., WIEKER S.B., *On the Optimal Distribution of Sensors in a Random Field,* ACM Trans, Sen. Netw. 1(2), pp. 272 – 306, 2005.