
Multi-objective ACO algorithm for WSN layout: performance according to number of ants

Stefka Fidanova* and Pencho Marinov

Institute of Information and Communication Technologies,
Bulgarian Academy of Sciences,
Acad. G. Bonchev Str. bl 25A,
1113 Sofia, Bulgaria
E-mail: stefka@parallel.bas.bg
E-mail: pencho@parallel.bas.bg
*Corresponding author

Marcin Paprzycki

System Research Institute,
Polish Academy of Sciences,
Warsaw, Poland
E-mail: marcin.paprzycki@ibspan.waw.pl

Abstract: Wireless sensor networks monitor physical or environmental conditions. One of the key objectives during their deployment is full coverage of the monitoring region with a minimal number of sensors and minimised energy consumption of the network. This problem is hard, from the computational point of view. Thus, the most appropriate approach to solve it is application of some metaheuristics. In this paper we apply multi-objective ant colony optimisation to solve this important telecommunication problem. The number of the agents (ants) is one of the important algorithm parameters in the ant colony optimisation metaheuristics. The needed computational resources for algorithm performance depends on number of ants. When the number of ants increases the computational time and used memory increase proportionally. Thus it is important to find the optimal number of agents needed to achieve good solutions with minimal computational resources. Therefore, the aim of the presented work is to study the influence of the number of ants on the algorithm performance.

Keywords: wireless sensor network; WSN; metaheuristics; ant colony optimisation; ACO; multi-objective optimisation; MOP.

Reference to this paper should be made as follows: Fidanova, S., Marinov, P. and Paprzycki, M. (2014) 'Multi-objective ACO algorithm for WSN layout: performance according to number of ants', *Int. J. Metaheuristics*, Vol. 3, No. 2, pp.149–161.

Biographical notes: Stefka Fidanova received her PhD in Computer Science from the Sofia University, Bulgaria. She is an Associate Professor in the Institute of Information and Communication Technology at the Bulgarian Academy of Science. Her research interests include combinatorial

optimisation, metaheuristics, mathematical modelling and parallel computing. She is an author of a great deal of research studies published at international journals, conference proceedings as well as books and book chapters.

Pencho Marinov received his PhD in Mathematical Modelling from Bulgarian Academy of Science. He is an Associate Professor in the Institute of Information and Communication Technology at the Bulgarian Academy of Science. His research interests are related to mathematical modelling, numerical methods and combinatorial optimisation.

Marcin Paparzycki received his PhD in Mathematics from Southern Methodist University, Dallas, Texas, USA. He is an Associate Professor in the Systems Research Institute at Polish Academy of Science. His fields of research include agent-based systems, grid computing and parallel and distributed computing.

This paper is a revised and expanded version of a paper entitled 'Influence of the number of ants on multi-objective ant colony optimization algorithm for wireless sensor network layout' presented at Large Scale Scientific Computing, Sozopol, Bulgaria, 3–8 June 2013.

1 Introduction

A sensor is a device, which can collect and transmit data. The sensors can sense any various phenomena or materials, such as temperature, voltage, or chemical substances. A wireless sensor network (WSN) allows automatic monitoring. Initial deployments of WSNs were completed by the military, for reconnaissance and surveillance (Deb et al., 2002). Examples of other possible applications of WSNs are: forest fire prevention, volcano eruption study (Werner-Allen et al., 2006), health data monitoring (Yuce et al., 2007), civil engineering (Paek et al., 2005) and others.

The energy for collecting data and its transmission comes from the battery of a node. In battery-powered systems, higher data rates and more frequent radio use consume more power. One of the nodes of the WSN has special role. It is a high energy communication node (HECN), which collects data from across the network and transmits it to the 'main computer' to be processed. The sensors transmit their data to the HECN, either directly or via hops, using closest sensors as communication relays. When deploying a WSN, the positioning of the sensor nodes becomes one of major concerns. The coverage obtained with the network and the economic cost of the network depend directly on it. Note that, the WSN can have large numbers of nodes, and therefore the task of selecting the geographical positions of the nodes for an optimally designed network can be very complex. Thus, it is unpractical to solve the problem directly, with traditional numerical methods. Instead, one of the best choices is to apply some metaheuristic method.

The problem of designing a WSN is multi-objective, with two objective functions. These are

- 1 minimise the energy consumption of the nodes in the network

2 minimise the number of nodes.

The full coverage of the network and connectivity are considered as constraints. Observe that this is an NP-hard multi-objective problem. In our work we propose a multi-objective ant [ant colony optimisation (ACO)] algorithm, which solves the WSN layout problem. Our aim is to study the influence of the number of ants on the algorithm performance and quality of the achieved solutions, and to find the minimal number of ants which are enough to achieve good solutions.

In the past, Jourdan (2000) solved an instance of the WSN layout using a multi-objective genetic algorithm. In their formulation, a fixed number of sensors had to be placed in order to maximise the coverage. However, in some applications the most important is the network energy. In this context, in Hernandez and Blum (2011) an ACO algorithm was proposed, but it applicable to a special case when the sensors are antennas and the work concerns only energy minimisation. In Wolf and Mezz (2008) an evolutionary algorithm was applied to this variant of the problem. In Fidanova et al. (2012) an ACO algorithm that took into account only the number of sensors was investigated. In Molina et al. (2008) several evolutionary algorithms to solve the problem were proposed. Finally, in Konstantinidis et al. (2010) a genetic algorithm, which achieves similar solutions as the algorithms in Molina et al. (2008) was studied, but tested only on small test problems.

In this paper we study the influence of the number of ants to the algorithm performance and quality of the achieved solutions. The computational resources, which the algorithm needs, are not negligible. The computational resources depends on the size of the solved problem and on the number of ants. Our aim in this work is to find a minimal number of ants which allow the algorithm to find good solution. Note that this estimates the minimal number of computational resources needed to solve the problem. We test our algorithm on problems with various sizes.

The paper is organised as follows. In Section 2 the WSN is introduced and the layout problem is formulated. Section 3 presents the ACO algorithm. In Section 4 we present and discuss the experimental results. Finally, Section 5 contains concluding remarks.

2 Problem formulation

A WSN consists of spatially distributed autonomous sensors that cooperatively monitor physical or environmental conditions, such as temperature, sound, vibration, pressure, motion, pollutants, movement, etc. The development of WSNs was motivated by military applications, such as battlefield surveillance. Currently, they are used in many industrial and civilian application areas, including industrial process monitoring and control, machine health monitoring, environment and habitat monitoring, healthcare applications, home automation, traffic control, and so on.

Each node in a sensor network is equipped with a wireless communications device and an energy source, usually a battery. A sensor node might vary in size and cost. Each sensor node senses an area around itself. The sensing radius (R_{sens}) determines the sensing area of the node. The nodes communicate among themselves using wireless communication links, determined by a communication radius (R_{com}). The HECN is responsible for the external access to the network. Therefore, every sensor node in the network must be able to communicate with the HECN. Since the communication radius

is often much smaller than the network size, direct links are not possible for most peripheral nodes. A multi-hop communication path is then established for those nodes that are far from the HECN. Overall, the quantity of the transmitted data defines the energy usage. The nodes of the network are divided by levels. On the first level are the nodes which can communicate directly with the HECN. Their distance to the HECN is one unit. On the second level are the nodes which cannot communicate with the HECN, but they communicate with at least one node from level 1. Their distance to HECN is two units. On the level N are nodes which cannot communicate with nodes from level N-2, but they can communicate with at least one node from the level N-1. Every node transmits its own data and the data, received by other nodes that communicated with it, which are far from the HECN. Furthermore, if the node communicates with more than one node closer to the HECN, the information is divided between them (to reduce their power consumption).

We indicate the quantity of own information collected by the node with 1. After adding the information coming from other sensors, which are far from the HECN, we calculate the quantity of the information passing through the node and respectively its energy consumption. The node with the highest energy consumption defines the energy of the network. Note that an unspecified number of sensor nodes has to be placed in a terrain to provide full coverage. Therefore, the objectives are to construct a network, with minimal number of sensors (cheapest for construction) and with minimal energy consumption (cheapest for exploitation), while keeping the connectivity of the network and full coverage of the sensing area. The problem has two opposing objective functions, thus we have a set of optimal solutions (non-dominated). Normally the sensors close to the HECN are most loaded and used more energy than others. Thus a possibility to decrease energy consumption is to add additional sensors close to the most loaded sensors, but in this way the number of sensors increases. Separately, every removal of a sensor leads to an increase of the energy consumption, or to disconnection of the network. The users decide, which of the solutions to use, by taking into account additional criteria related to their use case. For example, if the sensors are expensive and the batteries and their replacement is cheap, the user will prefer solution with a smaller number of sensors. If the sensors are cheap, but the batteries and/or their replacement are expensive (volcano or landslide monitoring), the user will prefer solution utilising less energy.

3 Multi-objective ACO for WSN layout

Multi-objective optimisation (MOP) has its roots in the nineteenth century, in the work in the area of economics, completed by Mathur (1991). The optimal solution for the MOP is not a single solution, as for the mono-objective optimisation problems, but 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 deteriorating at least one other 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 solutions and other components are not worse. The Pareto front is the set of non-dominated solutions. When metaheuristics are applied, the goal becomes to obtain solutions close to the Pareto front.

We apply multi-objective ACO to solve the problem. The idea for the ant algorithm comes from the real ant behaviour. When walking, they put on the ground chemical substance called pheromone. The ants smell the pheromone and follow the path with a stronger pheromone concentration. Thus they find shorter path between the nest and the food. The ACO algorithm uses a colony of artificial ants that behave as cooperating agents. With the help of the pheromone and the heuristic information they try to construct better solutions and to find the optimal ones. The pheromone corresponds to the global memory of the ants and the heuristic information is a some preliminary knowledge of the problem. The problem is represented by a graph and the solution is represented by a path in the graph or by tree in the graph. Ants start from random nodes and construct feasible solutions. When all ants construct their solution the pheromone is updated. The new, added, pheromone depends to the quality of the solution. The elements of the graph, which belong to better solutions will receive more pheromone and will be more desirable in the next iteration. To chose the next node to be included in the solution, the ants compute a set of feasible moves and select the best one, according to the transition probability rule. The transition probability p_{ij} , to chose the node j when the current node is i , is based on the heuristic information η_{ij} and on the pheromone level τ_{ij} of the move, where $i, j = 1, \dots, n$ and n is the number of the nodes of the graph, and has the form:

$$p_{ij} = \frac{\tau_{ij}^{\alpha} \eta_{ij}^{\beta}}{\sum_{k \in \{allowed\}} \tau_{ik}^{\alpha} \eta_{ik}^{\beta}} \quad (1)$$

The ant selects the move with the highest probability. The initial pheromone is set to a small positive value τ_0 and then ants update this value after completing the construction stage (Bonabeau et al., 1999; Fidanova and Atanassov, 2009). In our implementation, we use the MAX-MIN ant system (MMAS) (Dorigo and Stutzle, 2004), which is one of the most successful ant approaches. The main feature of the MMAS is using a fixed upper bound τ_{max} and a lower bound τ_{min} on the pheromone. Thus the accumulation of big amounts of pheromone by part of the possible movements and repetition of the same solutions is partially prevented.

In our case, the graph of the problem is represented by a square grid. The nodes of the graph are enumerated. The ants will deposit their pheromone on the nodes of the grid. We will deposit the sensors on the nodes of the grid too. The solution is represented by tree. An ant starts to create a solution starting from random node, which communicates with the HECN. Using transition probability [equation (1)], the ant chooses the next node to be included in the solution. If there is more than one node with the same probability, the ant chooses one of them randomly. An ant continues to include nodes in its solution till all area is covered. Construction of the heuristic information is a crucial point in the ant algorithms. Our heuristic information is a product of three values [equation (2)].

$$\eta_{ij}(t) = s_{ij} l_{ij} (1 - b_{ij}), \quad (2)$$

where s_{ij} is the number of the new points (nodes of the graph) which the new sensor will cover, and which are not covered by other sensors, and

$$l_{ij} = \begin{cases} 1 & \text{if communication exists;} \\ 0 & \text{if there is not communication,} \end{cases} \quad (3)$$

Here, b_{ij} is the solution matrix and the matrix element $b_{ij} = 1$ when there is sensor on this position otherwise $b_{ij} = 0$. With s_{ij} we try to increase the number of points covered by one sensor and thus to decrease the number of sensors we need. With l_{ij} we guarantee that all sensors will be connected. With b_{ij} we guarantee that maximum one sensor will be mapped on the same point. The search stops when $p_{ij} = 0$ for all values of i and j . It means that there are no more free positions, or that all area is fully covered.

At the end of every iteration the quantity of the pheromone is updated. The main idea is the elements of better solutions to receive more pheromone than others and to be more desirable in a next iteration. The quantity of the new added pheromone is proportional to the quality of the solution. The problem is multi-objective, therefore we need the fitness function, to assign new added pheromone. The pheromone trail update rule is given by:

$$\tau_{ij} \leftarrow \rho\tau_{ij} + \Delta\tau_{ij}, \quad (4)$$

$$\Delta\tau_{ij} = \begin{cases} 1/F(k) & \text{if } (i, j) \in \{\text{non-dominated solution constructed by ant } k\}, \\ 0 & \text{otherwise.} \end{cases}$$

We decrease the pheromone with a parameter $\rho \in [0, 1]$. This parameter models evaporation in the nature and decreases the influence of old information on the search process. After that, we add the new pheromone, which is proportional to the value of the fitness function. If the pheromone of some node becomes less than the lower bound of the pheromone, we put it to be equal to the lower bound and thus we prevent the pheromone of some nodes to become very low (close to 0) and to be undesirable. It is a kind of diversification of the search. As an upper bound is used the approximate upper bound of the pheromone. The used lower bound is $\tau_{min} = 0.085\tau_{max}$. We add new pheromone to the nodes, which belong to non-dominated solutions and thus we force the ants to search around them for new non-dominated solutions. The fitness function is constructed as follows:

$$F(k) = \frac{f_1(k)}{\max_i f_1(i)} + \frac{f_2(k)}{\max_i f_2(i)} \quad (5)$$

where $f_1(k)$ is the number of sensors proposed by the k -th ant and $f_2(k)$ is the energy of the solution of the k -th ant. These are also the objective functions of the WSN layout problem. We normalise the values of two objective functions with their maximal achieved values from the first iteration. We do this because the value of the number of sensors is much larger than the value of the energy. However, when normalised, they have similar influence in the fitness function.

4 Experimental results

As stated above, every ant starts to create its solution from random node of the graph. In our case it is a node, which communicates with the HECN. Thus the ant algorithm uses small number of agents (ants). Smaller number of ants means less memory, which is important when large problems are to be solved. The aim of this work is to learn

the influence of the number of the ants on the quality of the solution. Our purpose is to find the minimal number of ants, which are enough to achieve ‘good’ solutions. It is known that the ACO algorithm (Dorigo and Stutzle, 2004), can performs with only one ant (because of the random start), but in this case the quality of the solutions is low.

We have implemented software, which realises our ant algorithm. Our software can solve the problem at any rectangular area, the communication and the coverage radius can be different and can have any positive value. We can have regions in the area where is not possible, or forbidden, to place sensors. Using these regions, we can model non-rectangular areas. The HECN can be fixed at any point in the area. The program was written in C language, and the tests were run on computer with an Intel Pentium 2.8 GHz processor. In our tests we use an example where the area is square. The coverage and communication radii cover 30 points. The HECN is fixed in the centre of the area. For the tests we have used areas with three sizes: 350×350 points, 500×500 points and 700×700 points. We use the example 500×500 for comparison with other algorithms, because other authors used one with the same size.

In our previous work (Fidanova et al., 2013), we showed that our ant algorithm outperforms the existing algorithms for this problem. There, after several runs of the algorithm we were able to specify the most appropriate values of its parameters. We apply MAX-MIN ant algorithm with the following parameters: $\alpha = \beta = 1$, $\rho = 0.5$, $\tau_0 = 0.5$. In the ACO algorithms, if we fix the number of iterations and double the number of ants the execution time will be doubled. Thus it is important to find the minimal number of ants which find good solution. It means minimisation of required computational resources, like time and memory, which we need to solve the problem. We study the influence of the number of ants on the quality of the solutions. We fixed the number of the iterations to be 60 (about three hours per ant) and the number of ants to have following values $\{1, 2, 3, 4, 5, 6, 7, 8, 9, 10\}$.

We run our ACO algorithm 30 times for each number of ants. We extract the Pareto front from the solutions of these 30 runs. In Tables 1 to 3 we show the achieved non-dominated solutions (approximate Pareto fronts) for case 350×350 , 500×500 and 700×700 respectively.

In the left column represents the number of sensors and in other columns we present the energy corresponding to this number of sensors and the number of ants. Non-dominated points are represented with bold font.

Analysing Table 1 (case 350×350) we observe that the approximate Pareto front achieved by 3 ants is dominated by other approximate Pareto fronts. The approximate Pareto fronts achieved by 1, 2, 4, 5, 6 and 9 ants are part of the approximate Pareto front achieved by 7, 8 and 10 ants. More ants leads to more computational time. Thus the best algorithm performance in the case 350×350 is achieved by 7 ants.

Analysing the Table 2 (case 500×500) we observe that the approximate Pareto front achieved by 6 ants dominates the approximate Pareto fronts achieved by 1, 2, 3, 4 and 5 ants. There is not dominance between approximate Pareto fronts achieved by 6, 7, 8, 9 and 10 ants and we cannot say which of them is better. More over the hyper-volumes achieved by different number of ants are rather similar.

Table 1 Approximate Pareto fronts, example 350×350

<i>Sensors</i>	<i>Ants</i>									
	<i>1</i>	<i>2</i>	<i>3</i>	<i>4</i>	<i>5</i>	<i>6</i>	<i>7</i>	<i>8</i>	<i>9</i>	<i>10</i>
111				30	30	30	30	30	30	30
112										
113	28	35	28				28	28	28	28
114	26	26	26	26	26	26	26	26	26	26
115										
116							25	25		25

Table 2 Approximate Pareto fronts, example 500×500

<i>Sensors</i>	<i>Ants</i>									
	<i>1</i>	<i>2</i>	<i>3</i>	<i>4</i>	<i>5</i>	<i>6</i>	<i>7</i>	<i>8</i>	<i>9</i>	<i>10</i>
223					89	81				
224	61				88	65	61	59	57	71
225				58	60	58	57	58		57
226	59	95	73	57	59	57	56			
227		57	57		57	56		57		
228									54	
229	58			55				56		56
230	57					52		54		
231		55			55		53			
232			55	51	54	50	52	51		48
233							51			
234					53			48	53	
235			54						50	
236										
237										
238			53							
239	56			50						
240	53	53								
241										
242										
243										
244					52					

Analysing the Table 3 (case 700×700) we observe that the approximate Pareto front achieved by 6 ants dominates the approximate Pareto fronts achieved by other number of ants, except by 7 ants. Approximate Pareto fronts achieved by 6 and 7 ants are non-dominated and we can not say which of them is better. In this case again the hyper-volumes achieved by 6 and 7 ants are rather similar.

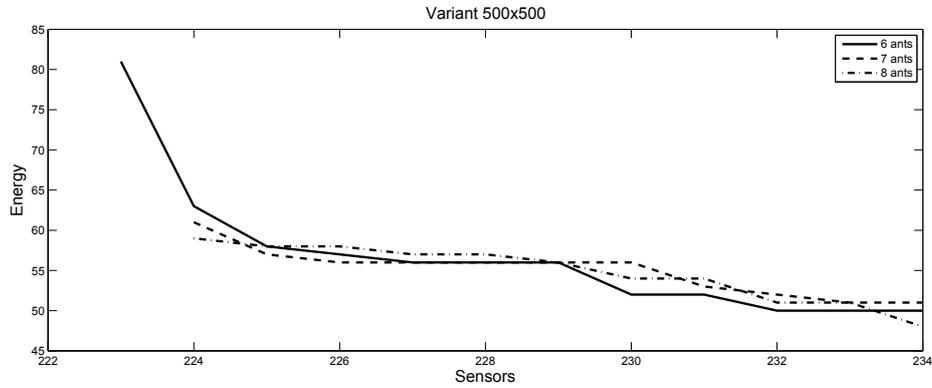
Table 3 Approximate Pareto fronts, example 700×700

Sensors	Ants									
	1	2	3	4	5	6	7	8	9	10
437						118	168	172		
438							112	117	260	
439					140	112	110	115	131	
440					115	93		114	111	162
441				122	111					110
442				114				112		
443		150	123							
444	124	112	112	106	107			102		105
445	117								108	
446			105	105	105		107		104	
447							105		102	
448	115	111								
449							102	99		
450	113									
451		109							97	
452							99			104
453										
454				105	96		96	96	96	
455	106	106								97

We prepare a Pareto front achieved by all runs of the algorithm with any number of ants (from 6 to 10) and we call it a common Pareto front. In the case 500×500 the common Pareto front is $\{(232, 48), (230, 52), (228, 54), (226, 56), (224, 57), (223, 81)\}$, for the case 350×350 it is $\{(111, 30), (113, 28), (114, 26), (116, 25)\}$ and for case 700×700 it is $\{(437, 118), (438, 112), (439, 110), (440, 93)\}$. The set of number of sensors is from 223 to 244 for the case 500×500 , from 111 to 116 for the case 350×350 and from 437 to 454 for the case 700×700 respectively. If for some number of sensors there is not a corresponding energy in the common Pareto front, we put the energy to be equal to the point of the front with lesser number of sensors. We can do this because, if we take some solution and if we include a sensor close to the HECN it will not increase the value of the energy and will increase by 1 only the number of the sensors. Thus, there is a corresponding energy to any number of nodes. This front we will call the *extended front*. In the case 500×500 the extended front is $\{(234, 48), (233, 48), (232, 48), (231, 52), (230, 52), (229, 54), (228, 54), (227, 56), (226, 56), (225, 57), (224, 57), (223, 81)\}$. In the case 350×350 the extended front is $\{(111, 30), (112, 30), (113, 28), (114, 26), (115, 26), (116, 25)\}$. In the case 700×700 the extended front is $\{(437, 118), (438, 112), (439, 110), (440, 93), (441, 93), (442, 93), (443, 93), (444, 93), (445, 93), (446, 93), (447, 93), (448, 93), (449, 93), (450, 93), (451, 93), (452, 93), (453, 93), (454, 93)\}$.

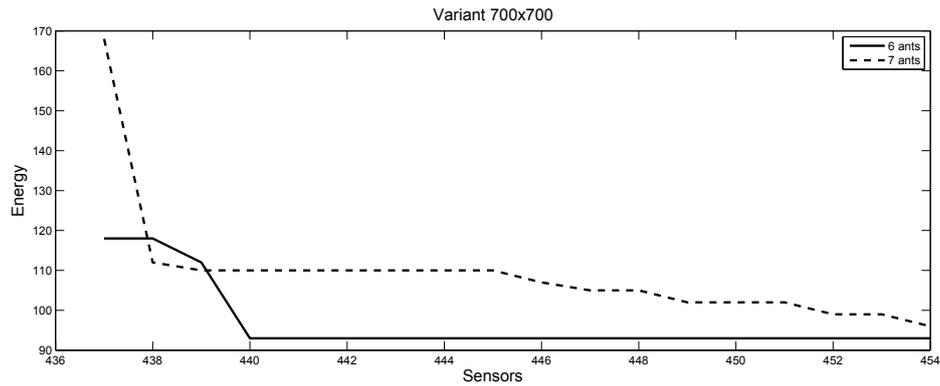
On Figure 1, with black line, we present the extended Pareto front achieved by 6 ants, with a dash line the extended Pareto front achieved by 7 ants and with dash-dot line is the extended Pareto front achieved by 8 ants.

Figure 1 ACO solution case 500×500



On Figure 2, with black line, we represent the extended Pareto front achieved by 6 ants and with a dash line the extended Pareto front achieved by 7 ants.

Figure 2 ACO solution case 700×700



We have included additional criteria to decide which approximate Pareto front is better in the case when there is no dominance between approximate Pareto fronts and the hyper-volumes are similar. We calculated the distance between a approximate Pareto fronts and the extended front. To calculate the distance, we extend every element of the approximate Pareto front in a similar way as the extended front. The distance between the approximate Pareto front and the extended front is the sum of distances between the points with a same number of sensors, or it is the difference between their energy. These distances are always positive because the extended front dominates the approximate Pareto fronts. Thus, by this criteria, the best approximate Pareto front will be the closest to the extended front.

In Table 4 we show the distances between the extended front and the approximate Pareto fronts achieved by 6, 7, 8, 9 and 10 ants. Analysing the Table 4 we conclude that the distance between the extended front and the approximate Pareto front achieved by 6 ants is the shortest. Thus, by our criteria, the approximate Pareto front (solutions) achieved by 6 ants in the case 500×500 is better.

Table 4 Distances from extended front case 500×500

Ants	6	7	8	9	10
Distance	20	23	21	22	29

In Table 5 we show the distances between the extended front and the approximate Pareto fronts achieved by 6 and 7 ants. Analysing the Table 5 we conclude that the distance between the extended front and the approximate Pareto front achieved by 6 ants is shorter. Thus, by our criteria, the approximate Pareto front (solutions) achieved by 6 ants in the case 700×700 is better.

Table 5 Distances from extended front case 700×700

Ants	6	7
Distance	6	210

In all discussed cases the approximate Pareto fronts achieved by 6 and 7 ants outperform others. Thus it is the best number of ants for our sensor layout problem.

Let us compare our results with results achieved by other authors. In Table 6 are reported best found results (with respect to the sensors and with respect to the energy) achieved by several metaheuristic methods for the case of 500×500 . We compare the results of our ACO algorithm with the results obtained by the evolutionary algorithms proposed in Molina et al. (2008). We observe that our algorithm achieves solution with much less sensors and less energy than other algorithms. The solution with worst number of sensors achieved by our algorithm consists of 247 sensors, which is less than the number of sensors achieved by the best solution of other methods. According to the energy criterion, our algorithm achieves solution with similar energy but with much less number of sensors. Thus we can conclude that our algorithm performs better.

Table 6 Experimental results

<i>Algorithm</i>	<i>Min sensors</i>	<i>Min energy</i>
MOEA	(260, 123)	(291, 36)
NSGA-II	(262, 83)	(277, 41)
IBEA _{HD}	(265, 83)	(275, 41)
ACO	(223, 81)	(232, 48)

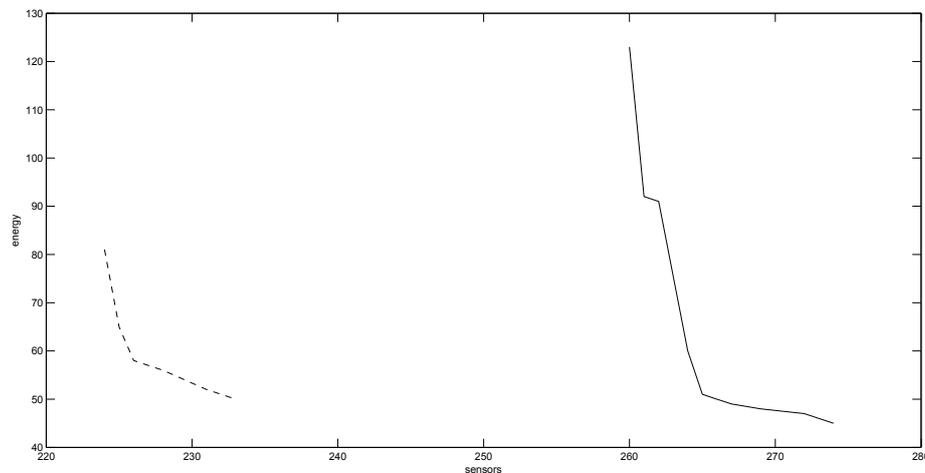
In Figure 3 we show the approximate Pareto fronts achieved by our algorithm (with dash line) and this achieved by the MOEA algorithm (with black line). We observe that Pareto front of our algorithm dominates the one achieved by MOEA.

5 Conclusions

In this paper we have studied the influence of the number of ants on the performance of the ACO algorithm, applied to the WSN. Smaller number of ants leads to the shorter running time and minimises memory use, which is important for complex/large cases.

We varied the number of ants, while fixing the number of iterations. Furthermore, we included the concept of an extended front, as an additional tool to compare approximate Pareto fronts that do not dominate each other. The best approximate Pareto front and the best performance were achieved when the number of ants was equal to 6 in the cases 700×700 and 500×500 , and 7 in the case 350×350 . Finally, our algorithm outperforms the results achieved by other methods.

Figure 3 Approximate Pareto fronts achieved by ACO (dash line) and MOEA (black line)



Acknowledgements

This work is a part of the Poland-Bulgaria bilateral Grant ‘Parallel and distributed computing practices’.

References

- Bonabeau, E., Dorigo, M. and Theraulaz, C. (1999) *Swarm Intelligence: From Natural to Artificial Systems*, Oxford University Press, UK.
- Deb, K., Pratap, A., Agrawal, S. and Meyarivan, T. (2002) ‘A fast and elitist multi-objective genetic algorithm: NSGA-II’, *IEEE Transactions on Evolutionary Computation*, Vol. 6, No. 2, pp.182–197.
- Dorigo M. and Stutzle T. (2004) *Ant Colony Optimization*, MIT Press, London, UK.
- Fidanova, S., Marinov, P. and Alba, E. (2012) ‘Ant algorithm for optimal sensor deployment’, in K. Madani, A.D. Correia, A. Rosa and J. Filipe (Eds.): *Studies of Computational Intelligence*, Vol. 399, pp.21–29, Springer.
- Fidanova, S. and Atanasov, K. (2009) ‘Generalized net model for the process of hybrid ant colony optimization’, *Comptes Rendus de l’Academie Bulgare des Sciences*, Vol. 62, No. 3, pp.315–322.
- Fidanova, S., Shindarov, M. and Marinov P. (2013) ‘Multi-objective ant algorithm for wireless sensor network positioning’, *Comptes Rendus de l’Academie Bulgare des Sciences*, Vol. 66, No. 3, pp.353–360.

- Hernandez, H. and Blum, C. (2011) 'Minimum energy broadcasting in wireless sensor networks: an ant colony optimization approach for a realistic antenna model', *J. of Applied Soft Computing*, Vol. 11, No. 8, pp.5684–5694.
- Jourdan, D.B. (2000) *Wireless Sensor Network Planning with Application to UWB Localization in GPS-denied Environments*, Massachusetts Institute of Technology, PhD thesis.
- Konstantinidis, A., Yang, K., Zhang, Q. and Zainalipour-Yazti, D. (2010) 'A multi-objective evolutionary algorithm for the deployment and power assignment problem in wireless sensor networks', *J. of Computer networks*, Vol. 54, No. 6, pp.960–976.
- Molina, G., Alba, E. and El Talbi, G. (2008) 'Optimal sensor network layout using multi-objective metaheuristics', *Universal Computer Science*, Vol. 14, No. 15, pp.2549–2565.
- Mathur, V.K. (1991) 'How well do we know pareto optimality?', *J. of Economic Education*, Vol. 22, No. 2, pp.172–178.
- Paek, J., Kothari, N., Chintalapudi, K., Rangwala, S. and Govindan, R. (2005) 'The performance of a wireless sensor network for structural health monitoring', *In Proc. of 2nd European Workshop on Wireless Sensor Networks*, Istanbul, Turkey.
- Stutzle, T. and Hoos, H.H. (1999) 'MAX-MIN ant system', *Future Generation Computer Systems* Vol. 16, No. 8, pp.889–914.
- Werner-Allen, G., Lorinez, K., Welsh, M., Marcillo, O., Jonson, J., Ruiz, M. and Lees, J. (2006) 'Deploying a wireless sensor network on an active volcano', *IEEE Internet Computing*, Vol. 10, No. 2, pp.18–25.
- Wolf, S. and Mezz, P. (2008) 'Evolutionary wireless sensor network on an active volcano', in C. Cotta and J. van Hemezl (Eds.): *VOCOP, Lecture Notes in Computer Sciences*, No. 4972, pp.61–72, Springer, Germany.
- Yuce, M.R., Ng, S.W., Myo, N.L., Khan, J.Y. and Liu, W. (2007) 'Wireless body sensor network using medical implant band', *Medical Systems*, Vol. 31, No. 6, pp.467–474.
- Zitzler, E. and Thiele, L. (1999) 'Multiobjective evolutionary algorithms: a comparative case study and the strength Pareto approach', *IEEE Transactions on Evolutionary Computation*, Vol. 3, No. 4, pp.257–271.