Metadata of the chapter that will be visualized in SpringerLink

Book Title	Large-Scale Scientific Computing
Series Title	558
Chapter Title	Influence of the Number of Ants on Multi-objective Ant Colony Optimization Algorithm for Wireless Sensor Network Layout
Copyright Year	2014
Copyright HolderName	Springer-Verlag Berlin Heidelberg

Corresponding Author	Family Name	**Fidanova**
	Particle	
	Given Name	**Stefka**
	Prefix	
	Suffix	
	Division	Institute of Information and Communication Technologies
	Organization	Bulgarian Academy of Sciences
	Address	Acad. G. Bonchev Street Bl25A, 1113, Sofia, Bulgaria
	Email	stefka@parallel.bas.bg
Author	Family Name	**Marinov**
	Particle	
	Given Name	**Pencho**
	Prefix	
	Suffix	
	Division	Institute of Information and Communication Technologies
	Organization	Bulgarian Academy of Sciences
	Address	Acad. G. Bonchev Street Bl25A, 1113, Sofia, Bulgaria
	Email	pencho@parallel.bas.bg
Author	Family Name	**Paprzycki**
	Particle	
	Given Name	**Marcin**
	Prefix	
	Suffix	
	Division	Systems Research Institute
	Organization	Polish Academy of Sciences
	Address	Warsaw, Poland
	Division	
	Organization	Management Academy
	Address	Warsaw, Poland
	Email	marcin.paprzycki@ibspan.waw.pl

Abstract

Wireless sensor networks monitor physical or environmental conditions. One of key objectives during their deployment is full coverage of the monitoring region with a minimal number of sensors and minimized energy consumption of the network. The 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 Optimization to solve this important telecommunication problem. The aim is to study the influence of the number of the ants on the algorithm performance.

Influence of the Number of Ants on Multi-objective Ant Colony Optimization Algorithm for Wireless Sensor Network Layout

Stefka Fidanova[1](✉), Pencho Marinov[1], and Marcin Paprzycki[2,3]

[1] Institute of Information and Communication Technologies,
Bulgarian Academy of Sciences, Acad. G. Bonchev Street Bl25A, 1113 Sofia, Bulgaria
{stefka,pencho}@parallel.bas.bg
[2] Systems Research Institute, Polish Academy of Sciences, Warsaw, Poland
[3] Management Academy, Warsaw, Poland
marcin.paprzycki@ibspan.waw.pl

Abstract. Wireless sensor networks monitor physical or environmental conditions. One of key objectives during their deployment is full coverage of the monitoring region with a minimal number of sensors and minimized energy consumption of the network. The 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 Optimization to solve this important telecommunication problem. The aim is to study the influence of the number of the ants on the algorithm performance.

1 Introduction

A sensor is a device which can collect and transmit data. First the wireless sensor networks were used by the military for reconnaissance and surveillance [2]. Examples of possible applications are forest fire prevention, volcano eruption study [14], health data monitoring [16], civil engineering [12], and others. Sensor networks depend on deployment of sensors. The sensors can sense any various phenomena or material such as temperature, voltage, or chemical substances. A Wireless Sensor Network (WSN) allows automatic monitoring.

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 depends 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

I. Lirkov et al. (Eds.): LSSC 2013, LNCS 8353, pp. 1–8, 2014.
DOI: 10.1007/978-3-662-43880-0_25,

complex. Thus, it is unpractical to solve the problem with traditional numerical methods. In this case, one of the best choices is to apply some metaheuristic method.

The problem is multi-objective with two objective functions. They are (1) minimizing the energy consumption of the nodes in the network, and (2) minimizing the number of the nodes. The full coverage of the network and connectivity are considered as constraints. It is an NP-hard multi-objective problem. We propose a multi-objective ant (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.

Jourdan [8] 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 maximize the coverage. In some applications the most important is the network energy. In this context, in [7] an ACO algorithm was proposed, while in [15] an evolutionary algorithm was applied to this variant of the problem. In [4] an ACO algorithm was investigated that took into account only the number of the sensors. In [10] several evolutionary algorithms to solve the problem were proposed. Finally, in [9] a genetic algorithm, which achieves similar solutions as the algorithms in [10] was studied, but tested on small test problems.

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

2 Problem Formulation

A wireless sensor network consists of spatially distributed autonomous sensors that cooperatively monitor physical or environmental conditions, such as temperature, sound, vibration, pressure, motion, or pollutants. The development of wireless sensor networks was motivated by military applications such as battlefield surveillance, and are now used in many industrial and civilian application areas, including industrial process monitoring and control, machine health monitoring, environment and habitat monitoring, health-care applications, home automation, and traffic control, etc.

Each node in a sensor network is equipped with wireless communications device and an energy source, usually a battery. A sensor node might vary in size and cost. Each sensor node sens an area around itself. The sensing radius determines the sensing area of the node. The nodes communicate among themselves using wireless communication links, determined by a communication radius. The HECN is responsible for the 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 the 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 used energy. The node with the highest energy 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 (cheapest for exploitation), while keeping the connectivity of the network.

3 Multi-objective ACO for WSN Layout

Multi-Objective Optimization (MOP) has his roots in the nineteenth century in the work in economics, of Edgeworth and Pareto [11]. The optimal solution for MOP is not a single solution as for mono-objective optimization 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 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 component 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 ant colony optimization to solve the problem. The idea for ant algorithm comes from the real ant behavior. 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 they try to construct better solutions and to find the optimal ones. 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 we update the pheromone. 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 $i, j = 1, \ldots, n$.

$$p_{ij} = \frac{\tau_{ij}^{\alpha}\,\eta_{ij}^{\beta}}{\sum\limits_{k \in \{allowed\}} \tau_{ik}^{\alpha}\,\eta_{ik}^{\beta}} \tag{1}$$

The ant selects the move with 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 [1,5]. In our implementation we use the MAX-MIN Ant System (MMAS) [3], which is one of the most successful ant approach. The main feature of the MMAS is using a fixed upper bound τ_{max} and a lower bound τ_{min} of the pheromone. Thus the accumulation of big amounts of pheromone by part of the possible movements and repetition of same solutions is partially prevented.

In our case the graph of the problem is represented by a square grid. The ants will deposit their pheromone on the nodes of the grid. We will deposit the sensors on the nodes of the grid. The solution is represented by tree starting by the high energy communication node. An ant starts to create the rest of the solution from a random node, which communicates with the HECN. Using transition probability (Eq. 1), the ant chooses the next node to visit. If there is more than one node with the same probability, the ant chooses one of them randomly. Construction of the heuristic information is a crucial point in ant algorithms. Our heuristic information is a product of three values (Eq. 2).

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

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

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

b 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. The search stops when $p_{ij} = 0$ for all values of i and j.

The pheromone trail update rule is given by:

$$\tau_{ij} \leftarrow \rho \tau_{ij} + \Delta\tau_{ij}, \tag{4}$$

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

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 in 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. The F is the fitness function. The role of the fitness function is to estimate the achieved solutions. 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 is constructed as follows:

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

Where $f_1(k)$ is the number of sensors achieved by the kth ant and $f_2(k)$ is the energy of the solution of the kth ant. These are also the objective functions of the WSN layout problem. We normalize the values of two objective functions with their maximal achieved values from the first iteration.

4 Experimental Results

Every ant start to create its solution from random point. In our case it is such point, 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 we solve large problems. The aim of this work is to learn the influence of the number of the ants on quality of the solution.

We have created a software which realizes 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. The HECN can be fixed in any point in the area. The program was written in C language and the tests were run on computer with Intel Pentium 2.8 GHz processor. 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 cover 30 points. The HECN is fixed in the center of the area. We use this example for comparison, because other authors use the same. We apply our algorithm on smaller test problem too. The area consists of 350 × 350 points. The HECN is fixed in the center of the area, the coverage and communication radii are as in a previous case.

In our previous work [6], we showed 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: $\alpha = \beta = 1$, $\rho = 0.5$. In the ACO, if we fix the number of iterations and double the number of ants the execution time will be doubled. 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 (H 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 and 2 we show the achieved non dominated solutions (Pareto fronts) for case 500 × 500 and 350 × 350 respectively. In the left column are the number of sensors and in other columns is the energy corresponding to this number of sensors and the number of ants. Analyzing the Table 1 (case 500 × 500) we observe that the Pareto front achieved by 6 ants dominates the Pareto fronts achieved by 1, 2, 3, 4 and 5 ants. The is not dominance between Pareto fronts achieved by 6, 7, 8, 9 and 10 ants and we cannot say which of them is better. Analyzing the Table 2 (case 350 × 350)we observe that the Pareto front achieved by 3 ants is dominated by other Pareto fronts. The Pareto fronts achieved by 1, 2, 4, 5, 6 and 9 ants are part of the Pareto front achieved by 7, 8 and 10 ants. More ants leads to more computational time. Thus the best Pareto front in the case 350 × 350 is achieved by 7 ants.

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)} and for the case 350 × 350 it is {(111, 30), (113, 28), (114, 26), (116, 25)}. Let us have a set of number of sensors from 223 to 244 for the case 500 × 500 and 111 to 116 for the case 350 × 350 respectively. If for

Table 1. Pareto fronts, example 500 × 500

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

some number of sensors there is not 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 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)}.

We have included additional criteria to decide which Pareto front is better in the case when there are not dominance between Pareto fronts. We calculated the distance between a Pareto front and the Extended front. To calculate the distance, we extend every element of Pareto fronts in a similar way as the Extended front. The distance between a 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 Pareto fronts. Thus, by this criteria, the best Pareto front will be the closest to the Extended front.

Table 2. Pareto fronts, example 350 × 350

Sensors	Ants									
	1	2	3	4	5	6	7	8	9	10
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 3. Distances from extended front case 500 × 500

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

In Table 3 we show the distances between the Extended front and the Pareto fronts achieved by 6, 7, 8, 9, and 10 ants. Analyzing the Table 3 we conclude that the distance between the Extended front and the Pareto front achieved by 6 ants is the shortest. Thus, by our criteria, the Pareto front (solutions) achieved by 6 ants in the case 500 × 500 is better.

5 Conclusion

In this paper we studied the influence of the number of ants on the performance of the ACO algorithm, applied to the wireless sensor network. Smaller number of ants leads to the shorter running time and minimizes 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 Pareto fronts that do not dominate each other. The best Pareto front and the best performance were achieved when the number of ants was equal in case 500 × 500 in the case 350 × 350.

Acknowledgments. This work has been partially supported by the Bulgarian National Scientific Fund under the grants DID 02/29 and DTK 02/44. It is a part of the Poland-Bulgaria bilateral grant “Parallel and distributed computing practices”.

References

1. Bonabeau, E., Dorigo, M., Theraulaz, G.: Swarm Intelligence: From Natural to Artificial Systems. Oxford University Press, New York (1999)
2. Deb, K., Pratap, A., Agrawal, S., Meyarivan, T.: A fast and elitist multi-objective genetic algorithm: NSGA-II. IEEE Trans. Evol. Comput. **6**(2), 182–197 (2002)
3. Dorigo, M., Stutzle, T.: Ant Colony Optimization. MIT Press, Cambridge (2004)

4. Fidanova, S., Marinov, P., Alba, E.: Ant algorithm for optimal sensor deployment. In: Madani, K., Correia, A.D., Rosa, A., Filipe, J. (eds.) Studies of Computational Inteligence: Computational Inteligence, vol. 399, pp. 21–29. Springer, Berlin (2012)
5. Fidanova, S., Atanasov, K.: Generalized net model for the process of hybrid ant colony optimization. C. R. Acad. Bulg. Sci. **62**(3), 315–322 (2009)
6. Fidanova, S., Shindarov, M., Marinov, P.: Multi-objective ant algorithm for wireless sensor network positioning. C. R. Acad. Bulg. Sci. (in press) AQ1
7. Hernandez, H., Blum, C.: Minimum energy broadcasting in wireless sensor networks: an ant colony optimization approach for a realistic antenna model. J. Appl. Soft Comput. **11**(8), 5684–5694 (2011)
8. Jourdan, D.B.: Wireless sensor network planning with application to UWB localization in GPS-denied environments. Ph.D. Thesis Massachusets Institute of Technology (2000)
9. Konstantinidis, A., Yang, K., Zhang, Q., Zainalipour-Yazti, D.: A multi-objective evolutionary algorithm for the deployment and power assignment problem in wireless sensor networks. J. Comput. Netw. **54**(6), 960–976 (2010)
10. Molina, G., Alba, E., Talbi, E.-G.: Optimal sensor network layout using multi-objective metaheuristics. Univ. Comput. Sci. **14**(15), 2549–2565 (2008)
11. Mathur, V.K.: How well do we know pareto optimality? J. Econ. Educ. **22**(2), 172–178 (1991)
12. Paek, J., Kothari, N., Chintalapudi, K., Rangwala, S., Govindan, R.: The performance of a wireless sensor network for structural health monitoring. In: Proceedings of 2nd European Workshop on Wireless Sensor Networks, Istanbul, Turkey (2005)
13. Stutzle, T., Hoos, H.H.: MAX-MIN ant system. Future Gener. Comput. Syst. **16**, 889–914 (2000) AQ2
14. Werner-Allen, G., Lorinez, K., Welsh, M., Marcillo, O., Jonson, J., Ruiz, M., Lees, J.: Deploying a wireless sensor network on an active volcano. IEEE Internet Comput. **10**(2), 18–25 (2006)
15. Wolf, S., Merz, P.: Evolutionary local search for the minimum energy broadcast problem. In: van Hemert, J., Cotta, C. (eds.) EvoCOP 2008. LNCS, vol. 4972, pp. 61–72. Springer, Heidelberg (2008)
16. Yuce, M.R., Ng, S.W., Myo, N.L., Khan, J.Y., Liu, W.: Wireless body sensor network using medical implant band. Med. Syst. **31**(6), 467–474 (2007)
17. Zitzler, E., Thiele, L.: Multiobjective evolutionary algorithms: a comparative case study and the strength pareto approach. IEEE Trans. Evol. Comput. **3**(4), 257–271 (1999)

Author Queries

Chapter 25

Query Refs.	**Details Required**	**Author's response**
AQ1	Please provide full details for the Ref. [6] , if applicable.	
AQ2	References [13, 17] are given in list but not cited in text. Please cite in text or delete from list.	

MARKED PROOF

Please correct and return this set

Please use the proof correction marks shown below for all alterations and corrections. If you wish to return your proof by fax you should ensure that all amendments are written clearly in dark ink and are made well within the page margins.

Instruction to printer	*Textual mark*	*Marginal mark*
Leave unchanged	• • • under matter to remain	Ⓥ
Insert in text the matter indicated in the margin	⋏	New matter followed by ⋏ or ⋏ⓧ2
Delete	/ through single character, rule or underline or ⊢——⊣ through all characters to be deleted	ꝺ or ꝺⓧ2
Substitute character or substitute part of one or more word(s)	/ through letter or ⊢——⊣ through characters	new character / or new characters /
Change to italics	— under matter to be changed	⌞
Change to capitals	≡ under matter to be changed	≡
Change to small capitals	= under matter to be changed	=
Change to bold type	~ under matter to be changed	~
Change to bold italic	≂ under matter to be changed	≂
Change to lower case	Encircle matter to be changed	≢
Change italic to upright type	(As above)	+
Change bold to non-bold type	(As above)	≁
Insert 'superior' character	/ through character or ⋏ where required	Y or ⅄ under character e.g. ²Y or ²⅄
Insert 'inferior' character	(As above)	⋏ over character e.g. ⋏₂
Insert full stop	(As above)	⊙
Insert comma	(As above)	,
Insert single quotation marks	(As above)	‘Y or ‘⅄ and/or ’Y or ’⅄
Insert double quotation marks	(As above)	“Y or “⅄ and/or ”Y or ”⅄
Insert hyphen	(As above)	⊢-⊣
Start new paragraph	_┌	_┌
No new paragraph	↩	↩
Transpose	⊔⊓	⊔⊓
Close up	linking ⁐ characters	⁐
Insert or substitute space between characters or words	/ through character or ⋏ where required	Y
Reduce space between characters or words	| between characters or words affected	T