Ant Colony Optimization with environment changes: an application to GPS surveying

Antonio Mucherino*, Stefka Fidanova†, Maria Ganzha‡

*IRISA, University of Rennes 1, Rennes, France.
antonio.mucherino@irisa.fr

†BAS, University of Sofia, Sofia, Bulgaria.
stefka@parallel.bas.bg

‡SRI, Polish Academy of Science, Warsaw, Poland.
maria.ganzha@ibspan.waw.pl

Abstract—We propose a variant on the well-known Ant Colony Optimization (ACO) general framework where we introduce the environment to play an important role during the optimization process. Together with diversification and intensification, the environment is introduced with the aim of avoiding the search to get stuck at local optima. In this work, the environment is simulated by means of the Logistic map, that is used in ACO for perturbing the update of the pheromone trails. Our preliminary experiments show that our environmental ACO (*e*ACO), with variable environment, outperforms the standard ACO on a set of instances of the GPS Surveying Problem (GSP).

I. INTRODUCTION

GLOBAL optimization consists in finding the global optimum of a given objective function which is generally subject to a given number of constraints. Many real-life problems can be formulated as a global optimization problem where the objective function often contains several local optima. A major issue in solving such problems is to "escape" from local optima for converging towards the global optimum of the objective function [12].

Meta-heuristics are general-purpose methods for global optimization. They are usually based on the simulation of animal or natural behaviors. They consist in a list of actions (generally repetitive) to be performed for finding an approximation of the global optimum of a given (and generally hard) optimization problem [23]. In recent years, several meta-heuristic approaches have been proposed in the scientific literature. A classical example is the Simulating Annealing (SA) proposed in the 80s, which is still used in some applications, such as the ones where it is necessary to deal with biological molecules [16]. Other well-known examples of meta-heuristics are the Genetic Algorithms (GAs), Differential Evolution (DE), the Tabu Search (TS), the Variable Neighborhood Search (VNS), etc. In this paper, we consider the Ant Colony Optimization (ACO) approach (see Section II-A). For a quick survey on meta-heuristics and the main references for the meta-heuristics mentioned above, the reader can refer, for example, to [17].

Every meta-heuristic is developed in order to find the best trade-off between two main concepts: *diversification* and *intensification* [23]. While the former tries to widely extend the search in the domain of the optimization problem, the latter improves candidate solutions by focusing on local neighbors of the current best known solutions. In ACO, diversification is guaranteed by the simulation of the typical ant behavior, while intensification is performed by applying a local search to a set of candidate solutions.

In this work, we introduce the *environment* in our ACO implementation. Pheromone updates are not supposed to be performed only on the basis of found solutions, but also on the basis of the current environment "surrounding" the ants. As an example, particular real-life environment conditions, such as strong wind, may alter the perception of the deposited pheromone. We will simulate environment changes by employing the Logistic map [25]. This simulation will help our artificial ants to escape from local optima, which may otherwise focus on firstly discovered paths and completely skip better ones. We warn the reader that the Logistic map has already been employed in optimization for performing chaotic searches [3], [26]. However, its use is different from the one considered in this work for the simulation of environment changes.

This paper is organized as follows. In Section II, we will extend the classic ACO approach for managing environment changes, which will affect the fitness values used in the pheromone updating rule during the execution of ACO. We will refer to our extended ACO as "environmental ACO" (*e*ACO). In Section III, we will describe the problem we consider in our preliminary computational experiments: the GPS Surveying Problem (GSP). Computational experiments will be presented in Section IV: they show that our *e*ACO, with environment changes, outperforms the standard ACO on the set of considered instances. Finally, conclusions will be given in Section V.

II. ACO WITH ENVIRONMENT CHANGES

We propose an extension of ACO for managing variable environments during the execution of the meta-heuristic. In Section II-A, we give a brief overview of this meta-heuristic search, and the reader who is interested in additional details is referred to the references given in the same section. In our ACO implementation, the pheromone updating rule is modified in order to take into consideration the current environment, and not only the objective function values of obtained solutions. Section II-B shows how to simulate environment changes in ACO. The environment is simulated by means of the Logistic map.

A. Ant Colony Optimization

Ants foraging for food deposit a substance named pheromone on the paths they follow. An isolated ant would just move randomly. Ants encountering previously laid pheromone marks are however stimulated to follow the same paths. This way, the pheromone trails are reinforced around the optimal ones, so that the probability for the other ants to follow optimal paths increases with time. The repetition of this mechanism represents the auto-catalytic behavior of ant colonies in nature [1], [6].

Ant Colony Optimization (ACO) is inspired by this ant behavior. A colony of artificial ants working into a mathematical space is simulated. These ants search for candidate solutions of a given optimization problem, while possible paths are marked by artificial pheromone for guiding other ants in the regions of the search space where good-quality solutions were already found. In ACO, therefore, the artificial ants generally create a sort of environment by themselves, by depositing the pheromone on marked paths. As explained in details in Section II-B, we will perturb this environment by means of the Logistic map.

The ants' search space is represented by the so-called *construction graph*, which is a weighted graph $G_C = (S_C, E_C, \eta)$ where vertices in S_C are solution components and edges in E_C indicate the possibility to combine the two connected components for obtaining a partial solution. The weight $\eta :$ $(u, v) \in E_C \longrightarrow \Re$ associated to each edge (u, v) is named heuristic information, whose equation is generally tailored to the problem at hand. A path on G_C allows to combine several partial solutions and to construct one complete solution.

Alg. 1 is a sketch of the ACO-based meta-heuristic. The transition probability p_{uv}, necessary in the algorithm when the ants need to decide on which edge (u, v) to walk, is based on the heuristic information η_{uv} and on the current pheromone level τ_{uv}:

$$p_{uv} = \tau_{uv}^{\alpha} \eta_{uv}^{\beta} \left[\sum_{(u,w) \in E_S : w \notin X} \left(\tau_{uw}^{\alpha} \eta_{uw}^{\beta} \right)^{-1} \right], \qquad (1)$$

where α and β are transition probability parameters. When the algorithm starts, small positive values are given to every τ_{uv}, which represent the current pheromone values on the edges $(u, v) \in E_C$. Then, every time a new solution is identified, the

---

Algorithm 1 Ant Colony Optimization

1: **ACO** (*in*: N, G_C, α, β; *out*: X_{best})
2: **let** $X_{best} = \emptyset$;
3: **while** (stopping criteria not satisfied) **do**
4:    **for** (k = 1, N) **do**
5:       **place** k^{th} ant on a random vertex $u \in S_C$;
6:       **let** $X = \{u\}$;
7:       **while** (X is incomplete) **do**
8:          **select** the vertex v in the star of u having higher probability p_{uv} (*see equ.* (1));
9:          **let** $X = X \cup \{v\}$;
10:         **let** $u = v$;
11:       **end while**
12:       **update** pheromone (*see equ.* (2));
13:       **apply** local search starting from X (*optional*);
14:       **if** (X is better than X_{best}) **then**
15:          **let** $X_{best} = X$;
16:       **end if**
17:    **end for**
18: **end while**

---

edges considered by the ants are marked with a new level of pheromone. In ACO, one possible updating rule (for a minimization problem) is the following:

$$\tau_{uv} = \tau_{uv} + \frac{1}{f(X)}, \qquad (2)$$

where X is the current solution, and f is the objective function of the considered problem.

In ACO, the general ant behavior allows to perform a wide search on the search domain (diversification), while the local search (see line 13 in Alg. 1) from constructed solutions X allows to focus on promising neighbors (intensification). Our implementation of ACO makes use of MaxMin Ant System (MMAS). The reader is referred to the paper [22] for a wider explanation of the ACO implementation considered in this work.

B. Simulating the environment

The Logistic map is a quadratic dynamical equation proposed in 1938 as a demographic model [25]. It is a rather simple quadratic polynomial

$$x_{n+1} = r x_n (1 - x_n), \qquad n > 0, \qquad (3)$$

where x_n represents the population size at time n and r is a constant, named growth coefficient. Given $x_0 \in [0, 1]$ and a value for $r \in [0, 4]$, this dynamical equation can either converge or be chaotic. In the first case, given any $x_0 \in [0, 1]$, x_n tends to the so-called "attraction domain". In the second case, x_n never converges, but it can rather take, in an apparent random way, any possible value in the range $[0, 1]$.

Fig 1 shows the behavior of the Logistic map for different values of r in the range $[2, 4]$ (its behavior is linear in the range $[0, 2]$). On the x-axis, we consider a discrete subset of 3000 equidistant values for r between 2 and 4; on the y-axis,
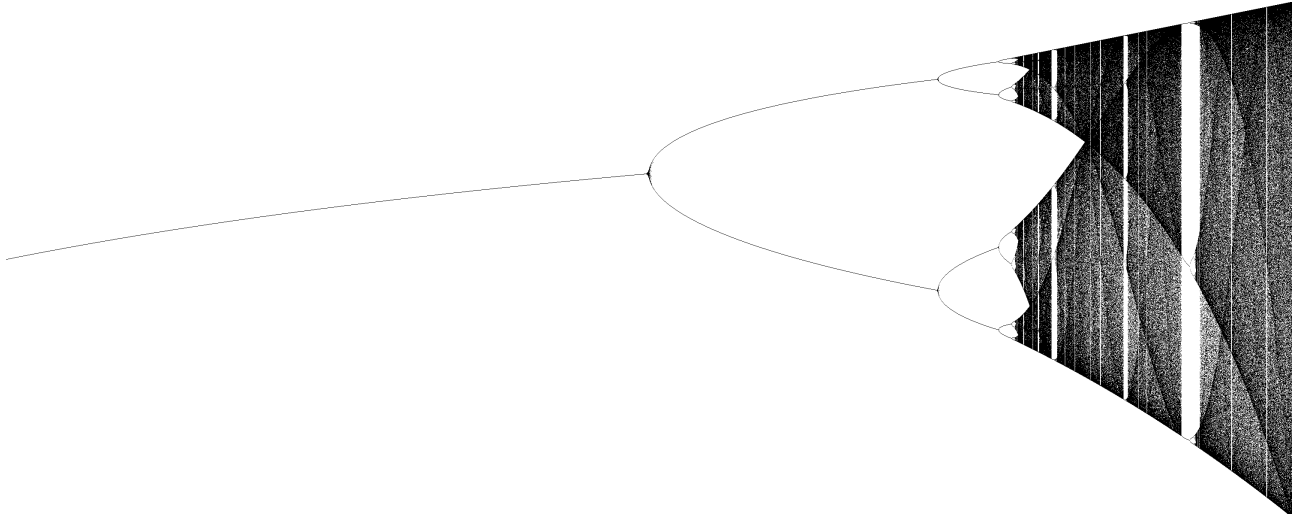
Fig. 1. The behavior of the Logistic map for different values of r (on the x-axis, for $r = 2$ to 4 in the figure). The attraction domain can be either regular or chaotic.

for every considered value for r, we report the corresponding attraction domain. In order to identify the attraction domains, we take 1500 equidistant points in the interval $[0,1]$ and we apply equation (3) 1000 times for each of them. For small values of r, the Logistic map always converges to one single point, i.e. the attraction domain consists of one point only. The first bifurcation appears when $r = 3$, where the attraction domain consists of 2 points; then there is another bifurcation when $r = 1 + \sqrt{6}$, where the attraction domain consists of 4 points. For larger values for r, the Logistic map experiences other bifurcations, and it can be chaotic for some subintervals of r. However, in these chaotic regions, it is still possible to identify regular attraction domains. For example, in Fig 2, the same graphic reported in Fig. 1 is zoomed in the region $r = [3.901, 3.908]$, where this phenomenon is clearly shown. Regular regions, that can be glimpsed in Fig. 1, still contain bifurcations. Moreover, we can notice that the whole graphic in Fig. 1 reappears in our zoomed region. Other regular attraction domains can be identified by looking at tighter subintervals of r, as well as other copies of the entire graphic. The graphic in Fig. 1 is in fact a fractal, because of its self-similarity [7], [19].

We simulate regular and chaotic changes of environment in ACO by introducing the Logistic map in equation (2), which is used in ACO for updating the pheromone trails. In the hypothesis the objective function of the considered problem is positive and greater than 1, the term $1/f(X)$ in equation (2) has always values ranging between 0 and 1. It can therefore take the place of x_0 in the Logistic map, so that a perturbed value x_1 can be computed, for a given value of r in $[0,4]$. The equation for updating the pheromone therefore becomes:

$$\tau_{uv} = \tau_{uv} + r \cdot \frac{1}{f(X)} \cdot \left(1 - \frac{1}{f(X)}\right). \tag{4}$$

With this simple change in the rule for updating the pheromone, we artificially perturb the environment of the ants, which would otherwise only depend on the solution fitness values. Different values for r can produce different environment changes, depending on the behavior of the Logistic map. For values of r for which the Logistic map converges, the pheromone levels added to τ_{uv} tend to be constant, reducing in this way the effects of good-quality solutions, that might mislead the ants towards a local optimum. For values of r for which the Logistic map behaves instead chaotically, the environment is dominant on the choices of the ants, as the pheromone update mostly depend on the simulated environment, rather than on the actual fitness value.

We refer to ACO with environment changes as *environmental* ACO (*e*ACO). In this work, we present some preliminary experiments (see Section IV) where *e*ACO is employed for solving the GSP (see next section).

III. GPS SURVEYING

The Global Positioning System (GPS) was originally developed in the US for military purposes, even if it was soon after used as well for civil applications [11]. It consists of a certain number of satellites that constantly orbit around earth and that are provided with sensors able to communicate with machines located on earth. The power that is necessary for establishing a satellite-earth communication allows for estimating the distance between the two communicating machines. Since the machine located on earth lies over a sphere that does not contain the satellite, a very precise information about the distance between the earth surface and the satellite would allow for determining the precise location of the machine on earth [15]. Moreover, the precision in locating sensor machines on earth can still be high when the distance information is not very precise, but the communication with more than one satellite can be established [14].
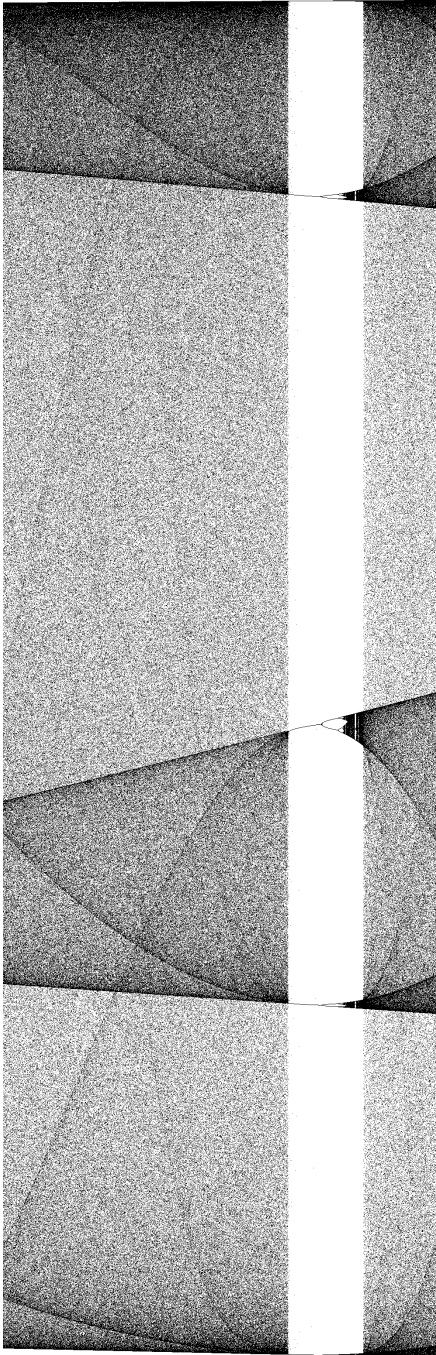
Fig. 2. The behavior of the Logistic map for values of r in the interval [3.901, 3.908]. This region of the Logistic map is generally chaotic, but regular attraction domains (with the typical bifurcations) can still be identified.

GPS technology can in fact provide very accurate locations for all sensors forming a given sensor network. The related costs can however be too high when it is necessary to deal with large networks. For this reason, over the last years, researchers have been trying to design and install local ground networks having the task of recording satellite signals with the aim of decreasing the overall network functioning cost [5], [20]. A network is composed by a certain number of *receivers* working at different *stations* in different moments. Therefore, given a certain number of *sessions*, representing the temporarily assignment of a given number of receivers to a set of distinct stations, the problem is to find a suitable *order* for such sessions for reducing the overall cost. This cost is in fact strictly related to the order of the sessions, because receivers need to be moved from one station to another when stepping from one session to another. Therefore, the distance between two involved stations is important for the computation of the costs. However, there are also additional costs that we might need to consider: if the number of working days necessary to perform the operation is more than one, then the need of planning an over-night stop at a company office can make the cost of the operation increase. The session order is also named *session schedule*. Generally, in order to alleviate the impact of measurement errors in the data, at least two receivers per session are considered [24].

The GPS Surveying Problem (GSP) can be formalized as follows. Let

$$S = \{s_1, s_2, \ldots, s_n\}$$

be a set of stations, and let

$$R = \{r_1, r_2, \ldots, r_m\}$$

be a set of receivers, with $m < n$. Sessions can be defined by a function $\sigma : R \longrightarrow S$ that associates one receiver to one station. Considering that no more than one receiver should be assigned to the same station, σ can be represented by an m-vector $(\varsigma_1, \varsigma_2, \ldots, \varsigma_m)$ containing, for each of the m receivers, the labels of the chosen stations. Since $m < n$ (and generally fixed to 2 or 3 in the applications), the number of permutations of m objects from n distinguishable ones is $n!/(n-m)!$, which can be huge when the network is large. Notice, however, that not all permutations may actually be possible, depending on the problem at hand.

Let C be an $n \times n$ matrix providing the costs $c(\varsigma_u, \varsigma_v)$ for moving one receiver from the station ς_u to the station ς_v. This matrix can be symmetric when moving between ς_u and ς_v is independent from the directionality; the non-symmetric case is however more realistic.

An instance of the GSP can be represented by a weighted undirected multigraph $G = (V, E, c)$ where vertices represent sessions σ_v and arcs (σ_u, σ_v) indicate the possibility to switch from session σ_u to session σ_v. The upper bound on the cardinality of V is $n!/(n-m)!$, which corresponds to the maximum number of possible sessions. The weight associated

		ACO	eACO			
instances	$|V|$		$r=1$	$r=2$	$r=3$	$r=4$
Malta	38	899.50	**897.00**	**897.00**	**897.00**	900.33
Seyshels	71	922.06	905.73	905.60	**887.33**	906.73
kro124p	100	40910.60	**40725.40**	40799.30	40753.00	40803.76
ftv170	171	3341.93	3314.20	**3313.76**	3319.83	3338.53
rgb323	323	1665.90	1654.40	**1648.66**	1649.43	1649.53
rgb358	358	1692.66	**1679.63**	1689.00	1682.80	1685.95
rgb403	403	3428.56	3413.63	3392.23	3393.76	**3386.10**
rgb443	443	3765.80	3749.93	3742.86	**3742.43**	3754.50

TABLE I
COMPARISON BETWEEN ACO AND eACO ON A SET OF GSP INSTANCES.

to the arcs provides the cost $c(\sigma_u, \sigma_v)$ for moving every receiver from the station $\varsigma_{u,i}$ to the station $\varsigma_{v,i}$, for each i:

$$c(\sigma_u, \sigma_v) = \sum_{i=1}^{m} c(\varsigma_{u,i}, \varsigma_{v,i}).$$

The graph G is not simple in general, because it might be feasible to switch from session σ_u to session σ_v, as well as from σ_v to σ_u, but with a different total cost. The problem consists in finding an optimal path on G, i.e. a path for which all selected arcs give the minimal total cost, while covering the entire vertex set V [5].

The GSP can be seen as the classic Traveling Salesman Problem (TSP), which asks for determining the optimal route for a salesman to visit a given number of cities while minimizing the traveled distance [13]. If the two cities are replaced with sessions σ_v and the distances are replaced with the weights associated to the arcs (σ_u, σ_v) of our graph G, the equivalence between the two problems becomes evident. More precisely, since the weights on the arcs (σ_u, σ_v) and (σ_v, σ_u) are generally different, GSP better fits with the Asymmetric TSP (ATSP), where distances between the cities depend upon the order in which the cities are reached. Finally, we also remark that the necessity of an over-night office stop can be formalized by adding a fictive session in G where receivers come back to base offices. This special session might need to be "traveled" more than once: we can therefore say that, in general, the GSP can be seen as a Multiple Asymmetric TPS (MATSP) [2]. The TSP and these variants are NP-hard [18]. Therefore, the GSP is NP-hard as well.

First attempts for solving the GSP were based on the idea of transforming GSP instances into instances of the class of TSP-like problems, and to employ existing methods and algorithms. In [4], a branch-and-bound approach was employed, which is actually able to find the optimal solutions for small GSP instances. As the size of the networks increases, the complexity grows and the time necessary for a branch-and-bound to converge becomes prohibitive. On the other side, in real-life applications, there is generally the need to obtain GSP solutions as fast as possible, even if only approximated ones. Therefore, heuristic approaches particularly developed for this application have been proposed over the last years for identifying optimal or near-optimal session orders [8], [9], [20]. In this work, we consider a set of instances of the GSP for

testing our ACO approach with environment changes described in Section II.

IV. COMPUTATIONAL EXPERIMENTS

We apply our eACO (see Section II) for solving instances of the GSP (see Section III). In this application, the construction graph G_C corresponds to the weighted undirected multigraph G where vertices are sessions σ_u and edges indicate the possibility to switch from one session to another. As test cases, we consider data from two real networks: Malta [20], composed by 38 sessions, and Seychelles [21], composed by 71 sessions. We also consider larger instances designed for testing the ATSP, which are freely available on the Internet.[1]

Our eACO implementation is based on Alg. 1, where the transition probability p_{uv} is computed by equation (1), and the pheromone update is performed by applying equation (4). The heuristic information η_{uv} is given by the formula:

$$\eta_{uv} = \frac{1}{c(\sigma_u, \sigma_v)},$$

where $c(\sigma_u, \sigma_v)$ is the total cost for switching from session σ_u to session σ_v (see Section III). Finally, the values for the transition probability parameters α and β are fixed to 1 and 2, respectively. These values were identified in previous works as the optimal ones for ACO when solving instances of the GSP [10]. In the following experiments, we will focus on the quality of the found solutions, rather than on the algorithms' performances. In fact, the increase in complexity for using equation (4), rather than equation (2), can be neglected when considering the overall algorithms' complexity. We do not compare our results to the best results currently known for the considered instances. In the present work, our aim is not to improve those results, but only to give a preliminary validation to the usefulness of the presented idea.

Table I shows some computational experiments for different values of r. Average values over 30 runs are reported in the table. eACO is able to identify better quality solutions in all experiments and for almost all used values for r. This shows that, in fact, a variable environment for the ants, instead of a constant one, gives benefits to the search. For values of r equal to 1, 2 and 3, the Logistic map converges to one unique value; it is instead chaotic for $r = 4$. It seems therefore that

---

[1] http://www.informatik.uni-heidelberg.de/groups/comopt/software/TSLIB95/ATSP.html

the best results can be achieved when the environment tends to homogenize the pheromone trails. Notice, however, that only one experiment with $r = 4$, when the environment is chaotic, was not able to provide a better solution (w.r.t the one provided by the standard ACO).

V. CONCLUSIONS

We introduced a variant of ACO where the ant environment is not constant but it is rather subject to change over time. This way, in our ACO implementation, there are less chances for the meta-heuristic search to be trapped at a local optimum. In this work, we have tested this idea on instances of the GSP. In order to simulate the environment in our *e*ACO, we employed the well-known Logistic map, which can have either a regular or a chaotic behavior, depending on the values assigned to its parameter r.

We believe this is the first research contribution where an "environment" is introduced in a meta-heuristic search. In this work, we showed that this novel idea seems to be promising when working with ACO. Evidently, the idea still needs to be studied in details from a theoretical point of view, as well as from a practical one. The effect of a suitable environment should be studied in conjunction with other meta-heuristic frameworks. Moreover, a much wider experimental analysis should be performed, on a larger set of instances of the GSP, as well as on instances of other known hard problems. This will be our main direction for future works.

ACKNOWLEDGMENTS

REFERENCES

[1] V. Atanassova, S. Fidanova, I. Popchev, P. Chountas, *Generalized Nets, ACO-Algorithms and Genetic Algorithm*. In: "Monte Carlo Methods and Applications", K.K. Sabelfeld, I. Dimov, De Gruyter, 39–46, 2012.

[2] T. Bektas, *The Multiple Traveling Salesman Problem: an Overview of Formulations and Solution Procedures*, Omega **34**(3), 209–219, 2006.

[3] L. Chen, K. Aihara, *Chaotic Simulated Annealing by a Neural Network Model with Transient Chaos*, Neural Networks **8**(6), 915–930, 1995.

[4] P. Dare, *Optimal Design of GPS Networks: Operational Procedures*, PhD Thesis, School of Surveying, University of East London, UK, 1995.

[5] P. Dare, H.A. Saleh, *GPS Network Design: Logistics Solution using Optimal and Near-Optimal Methods*, Journal of Geodesy **74**, 467–478, 2000.

[6] M. Dorigo, M. Birattari, *Ant Colony Optimization*. In: "Encyclopedia of Machine Learning", C. Sammut, G.I. Webb (Eds.), Springer, 36–39, 2010.

[7] K. Falconer, *Fractal Geometry: Mathematical Foundations and Applications*, Wiley, 400 pages, 2013.

[8] S. Fidanova, *Hybrid Heuristics Algorithms for GPS Surveying Problem*, Lecture Notes in Computer Science **4310**, Proceedings of the 6th International Conference on Numerical Methods and Applications, T. Boyanov, S. Dimova, K. Georgiev, G. Nikolov (Eds.), 239–248, 2007.

[9] S. Fidanova E. Alba, G. Molina, *Memetic Simulated Annealing for GPS Surveying Problem*, Lecture Notes in Computer Science **5434**, Proceedings of the 4th International Conference on Numerical Analysis and Its Applications, S. Margenov, L.G. Vulkov, J. Waśniewski (Eds.), 281–288, 2009.

[10] S. Fidanova, E. Alba, G. Molina, *Hybrid ACO Algorithm for the GPS Surveying Problem*, Lecture Notes in Computer Science **5910**, Proceedings of Large Scale Scientific Computing, I. Lirkov, S. Margenov, J. Waśniewski (Eds.), 318–325, 2010.

[11] B. Hofmann-Wellenhof, H. Lichtenegger, J. Collins, *Global Positioning System: Theory and Practice*, Springer, 326 pages, 1993.

[12] R. Horst, P.M. Pardalos, *Handbook of Global Optimization*, Springer, 879 pages, 1995.

[13] J.B. Kruskal, *On the Shortest Spanning Subtree of a Graph and the Traveling Salesman Problem*, Proceedings of the American Mathematical Society **7**(1), 48–50, 1956.

[14] A. Leick, *GPS Satellite Surveying*, 3rd edition, Wirley, 464 pages, 2004.

[15] L. Liberti, C. Lavor, N. Maculan, A. Mucherino, *Euclidean Distance Geometry and Applications*, SIAM Review **56**(1), 3–69, 2014.

[16] T.E. Malliavin, A. Mucherino, M. Nilges, *Distance Geometry in Structural Biology: New Perspectives*. In: "Distance Geometry: Theory, Methods and Applications", A. Mucherino, C. Lavor, L. Liberti, N. Maculan (Eds.), Springer, 329–350, 2013.

[17] A. Mucherino, O. Seref, *Modeling and Solving Real Life Global Optimization Problems with Meta-Heuristic Methods*. In: "Advances in Modeling Agricultural Systems", Springer Optimization and Its Applications **25**, P.J. Papajorgji, P.M. Pardalos (Eds.), 403–420, 2008.

[18] C.H. Papadimitriou, *The Euclidean Travelling Salesman Problem is NP-complete*, Theoretical Computer Science **4**(3), 237–244, 1977.

[19] M. Rani, R. Agarwal, *Generation of Fractals from Complex Logistic Map*, Chaos, Solitions and Fractals **42**, 447–452, 2009.

[20] H.A. Saleh, P. Dare, *Effective Heuristics for the GPS Survey Network of Malta: Simulated Annealing and Tabu Search Techniques*, Journal of Heuristics **7**, 533–549, 2001.

[21] H.A. Saleh, P. Dare, *Heuristic Methods for Designing a Global Positioning System Surveying Network in the Republic of Seychelles*, The Arabian Journal for Science and Engineering **26**(1B), 74–93, 2002.

[22] T. Stutzle, H.H. Hoos, *MAX-MIN Ant System*; In: "Future Generation Computer Systems", vol. 16, M. Dorigo, T. Stutzle, G. Di Caro (Eds.), 889–914, 2000.

[23] E-G. Talbi, *Metaheuristics: From Design to Implementation*, Wiley, 624 pages, 2009.

[24] P. Teunissen, A. Kleusberg, *GPS for Geodesy*, 2nd edition, Springer, 650 pages, 1998.

[25] P-F. Verhulst, *A Note on the Law of Population Growth*, Correspondence Mathematiques et Physiques **10**, 113–121, 1938 (in French).

[26] D. Yang, G. Li, G. Cheng, *On the Efficiency of Chaos Optimization Algorithms for Global Optimization*, Chaos, Solitions and Fractals **34**, 1366–1375, 2007.