Static Drone Placement by Elephant Herding Optimization Algorithm

Ivana Strumberger, Nebojsa Bacanin, Slavisa Tomic, Marko Beko, and Milan Tuba, *Member, IEEE*

***Abstract*—Optimal placement of drones is a very challenging problem and it belongs to the group of hard optimization problems for which swarm intelligence algorithms were successfully applied. This paper presents an implementation of the recent elephant herding optimization algorithm for solving the static drone location problem. The objective of the model applied in this paper is to establish monitoring of all targets with the least possible number of drones. In empirical tests we used two problem instances: one with 30 uniformly distributed targets, and one with 30 clustered targets. The simulation results show that the elephant herding optimization algorithm performs well in covering targets for both instances of the problem, especially considering the number of drones that were deployed.**

Key-words – Elephant herding optimization algorithm, metaheuristic optimization, static drone location problem, swarm intelligence.

I. Introduction

With the emerging of low energy consumption machines, light materials and high performance processing devices, the usage and applications of flexible flying drones has increased. Drones can be used in variety of applications such as vehicle tracking, traffic management, fire detection, military operations [1], etc. where inspite of different terrains of monitoring a reliable system that can obtain proper and relevant data is required.

Technical structure of the drone depends on the manufacturer but what makes the drones useful are electrical motors and camera that is used to observe targets in the flying zone. Targets that are monitored are considered as points which can be static or mobile, depending on the type such as machines, plants, animals, people, climate, etc. In recent years, numerous studies consider drone placement when drones are used as aerial wireless base stations for cellular network [2]. Similarly to anchor nodes targeting unknown nodes in wireless sensor network, drones deployment must be done in a way to cover multiple targets, where each target must be covered by at least one drone [3], [4]. The objective of the model used in this paper is to monitor all the targets with the least possible number of drones.

The optimal placement of drones is a very challenging problem and it belongs to the group of hard optimization problems (NP-Hard) [5]. For solving such kind of problems, classical, deterministic methods cannot obtain acceptable results within reasonable time. Instead, metaheuristic methods have been proven as successful optimizers in obtaining satisfying results when tackling such hard optimization. Swarm intelligence is the subset of metaheuristics algorithms and in the literature many papers can be found that show successful implementations of these methods on practical, real-life NP-hard problems [6], [7], [8], [9].

In this paper, we propose the elephant herding optimization (EHO) algorithm adopted for solving static drone location problem. EHO algorithm, proposed by Wang et al. in 2015 for global unconstrained optimization [10] is a rather new approach that belongs to the group of swarm intelligence metaheuristics.

There have been only few implementations of the EHO that can be found in the literature. In [11] EHO was tested on standard benchmark problems. EHO was also applied to the support vector machine parameters tuning [12], multilevel image thresholding [13], and other practical problems [14].

The rest of the paper is organized as follows. After Introduction, a mathematical formulation of static drone location problem is presented in Section II. Section III presents EHO algorithm adopted for drone placement problem. Experimental results, along with discussion are given in Section IV. Finally, remarks and conclusion are given in Section V.

II. Mathematical Formulation of Drone Location Problem

In this section, a mathematical formulation of the static drone location problem (SDLP) is presented. We used similar formulation as in [15].

The flying region of the drone is represented as a rectangular two-dimensional terrain with length x_{max} and width y_{max}. Each drone is defined by its position coordinates (x, y) and radius r. Targets are static and within the terrain of dimension of $x_{max} \times y_{max}$. Let U denote the set of available drones and

This research is supported by the Ministry of Education, Science and Technological Development of Republic of Serbia, Grant No. III-44006.

Ivana Strumberger, Graduate School of Computer Science, Megatrend University, Bulevar umetnosti 29, 11070 Belgrade, Serbia (e-mail: istrumberger@naisbitt.edu.rs).

Nebojsa Bacanin, Graduate School of Computer Science, Megatrend University, Bulevar umetnosti 29, 11070 Belgrade, Serbia (e-mail: nbacanin@naisbitt.edu.rs).

Slavisa Tomic, Institute for Systems and Robotics (ISR), Instituto Superior Tecnico (IST), Av. Rovisco Pais 1, Lisbon 1049-001, Portugal (email: stomic@isr.ist.utl.pt).

Marko Beko, Universidade Lusofona de Humanidades e Tecnologias, Campo Grande 376, 1749-024 Lisboa, Portugal, and CTS/UNINOVA, Campus da FCT/UNL, Monte de Caparica, 2829-516 Caparica, Portugal (e-mail: beko.marko@ulusofona.pt, mbeko@uninova.pt).

Milan Tuba (corresponding author), Department of Technical Sciences, State University of Novi Pazar, Vuka Karadzica bb, Novi Pazar, Serbia (e-mail: tuba@ieee.org).

978-1-5386-3073-0/17/ $31.00© 2017 IEEE

let T indicate the set of targets to be monitored. Each target $t_i \in T$ is defined by its coordinates (X_{t_i}, Y_{t_i}).

If we assume that a drone u with monitoring radius r^u is positioned at coordinates (x_u, y_u), and that there is a target t_i, the distance $D_{t_i}^{x_u,y_u}$ between u and t_i can be defined as:

$$D_{t_i}^{x_u,y_u} = \sqrt{(X_{t_i} - x_u)^2 + (Y_{t_i} - y_u)^2} \tag{1}$$

Each drone u with radius r^u has a visibility θ which exemplifies a disk in the plane. There are two main issues that have to be taken into consideration when mathematically representing drone coverage of targets. First, coordinates (x_u, y_u) must be determined, where each drone $u \in U$ with radius r^u should be located in order to monitor the targets. Second, given the location (x_u, y_u) of the drone $u \in U$ with radius r^u, it has to be determined which target $t_i \in T$ is monitored by a drone $u \in U$. The mathematical representation of these two assessments can be represented as decision variables [15]:

$$\delta_{xy}^u = \begin{cases} 1, & \text{if the drone } u \text{ is located at } (x,y) \\ 0, & \text{otherwise} \end{cases} \tag{2}$$

and

$$\gamma_{t_i}^u = \begin{cases} 1, & \text{if the target } t_i \text{ is observed by the drone } u \\ 0, & \text{otherwise} \end{cases} \tag{3}$$

The objective of the model used in this paper is to monitor all the targets with the least possible number of drones. The model can be formulated as follows [15]:

$$\min f(\delta) = \sum_{(x,y)} \sum_{u \in U} \delta_{xy}^u \tag{4}$$

s.t.

$$\sum_{x,y} \delta_{xy}^u \leq 1 \quad \forall u \in U \tag{5}$$

$$\gamma_{t_i}^u \leq \sum_{(x,y)} \delta_{xy}^u \left(\frac{r^u}{D_{t_i}^{uxy}} \right) \quad \forall u \in U, t_i \in T \tag{6}$$

$$\sum_{u \in U} \gamma_{t_i}^u \geq 1 \quad \forall t_i \in T \tag{7}$$

$$\delta_{xy}^u \in \{0,1\}, \ \forall (x,y), \ 1 \leq x \leq x_{max} \tag{8}$$

$$1 \leq y \leq y_{max}, \ u \in U \tag{9}$$

$$\gamma_{t_i}^u \in \{0,1\}, \ \forall t_i \in T, u \in U \tag{10}$$

The form of objective function as in Eq. (4) refers to minimization of number of employed drones. Constraint (5) ensures that the drone u is positioned in at most one location. Condition (6) is used to set the value of decision variable $\gamma_{t_i}^u$. If the radius of the drone u is less than the distance between target t_i and the drone u, variable $\gamma_{t_i}^u$ takes a value of 0, and vice-versa. Constraint (7) specifies that each target t_i is monitored by at least one drone. Finally, constraints (8) - (10) specify the domain of the variables.

III. Elephant herding optimization algorithm

Elephant herding optimization algorithm (EHO) was proposed by Wang et al. [10], for solving global optimization tasks. Inspired by the social skills and structural independence of the elephants in herds, the authors have formulated a general purpose heuristic search. The idea behind this formulation is associated with the elephants coexisting together in groups (clans), under the leadership of a matriarch. This habitation is composed of females and calves, where the oldest female is often chosen to be matriarch. Changes in clans are coming from male calves, when they reach their full growth, they leave the clan to live independently, but they still communicate through low frequency vibrations.

Behavior of the elephants can be divided into two different social environments [10]: elephants living in clans under the influence and leadership of a matriarch; male elephants separating from clan to live on their own, but still communicate. These social environments can be modeled as updating and separating operators. In the EHO algorithm, each solution (elephant) in each clan is updated by its current position and matriarch through updating operator. After that, through the separating operator, the population diversity is enhanced at the later generations of the algorithm.

Each potential solution in the algorithm is represented as an integer number vector with the dimension of $3M$, where M is the number of used drones. Each drone is described by two coordinates of the drone's $2D$ position, and the third component represents drone's coverage radius r. As mentioned in Section II, we implemented model where each drone and each target are static.

Updating operator in the EHO algorithm is defined as follows. Population of potential problem solutions is divided into n clans. Each solution j in the clan ci alters its position by the influence of the matriarch ci which has the best fitness value in generation [10]:

$$x_{new,ci,j} = x_{ci,j} + \alpha \times (x_{best,ci} - x_{ci,j}) \times r \tag{11}$$

where $x_{new,ci,j}$ indicates the new position of the solution j in clan ci, $x_{ci,j}$ represents the old position of the individual j in the clan ci, and $x_{best,ci}$ is the best solution in the clan ci found so far. Parameter $\alpha \in [0, 1]$ is a scale factor that designates the influence of matriarch ci on $x_{ci,j}$, while $r \in [0, 1]$ is random variable with uniform distribution.

The fittest solution in each clan ci is updated using the following expression [10]:

$$x_{new,ci,j} = \beta \times x_{center,ci} \tag{12}$$

where $\beta \in [0, 1]$ denotes the influence factor of the $x_{center,ci}$ on the updated individual. Center of the clan ci for d-th dimension problem can be calculated as [10]:

$$x_{center,ci,d} = \frac{1}{n_c i} \times \sum_{j=1}^{n_d} x_{ci,j,d} \tag{13}$$

where $1 \leq d \leq D$ represents the d-th dimension, D is the total dimension of the search space, and $n_c i$ indicates the number of solutions in clan ci.

The separating operator in EHO algorithm is applied to the individual with the worst fitness and can be modeled at each generation as [10]:

$$x_{worst,ci} = x_{min} + (x_{max} - x_{min} + 1) \times rand \qquad (14)$$

where x_{max} and x_{max} represent the upper and lower bound of the position of the individual, $x_{worst,ci}$ indicates the individual with the worst fitness in clan ci, and $rand \in [0, 1]$ is a random number chosen by uniform distribution.

In the initialization phase of the algorithm, the population is divided into n clans. Each solution in each clan is generated using Eq. 14.Also, at this phase, the fitness value is calculated for each individual j in the clan ci.

Pseudo-code of the EHO algorithm for constrained optimization is given in Algorithm 1.

Algorithm 1 Pseudo-code of EHO algorithm for constrained problems

Initialization. Generate individuals; divide population into n clans; calculate fitness for each individual;set generation counter $t = 1$ and maximum generation $MaxGen$.
while $t < MaxGen$ **do**
  Sort all solutions according to their fitness
  for all clans ci **do**
    for all solution j in the clan ci **do**
      Update $x_{ci,j}$ and generate $x_{new,ci,j}$ using Eq. 11
      Select and retain better solution between $x_{ci,j}$ and $x_{new,ci,j}$ using Debs rule
      Update $x_{best,ci}$ and generate $x_{new,ci,j}$ using Eq. 12
      Select and retain better solution between $x_{best,ci}$ and $x_{new,ci,j}$
    end for
  end for
  for all clans ci in the population **do**
    Replace the worst solution in clan ci using Eq. 14
  end for
  Evaluate population and calculate fitness
end while
return the best solution among all clans

IV. Empirical tests and discussion

In this Section, we show settings of the algorithm parameters, empirical tests and results, as well as discussion.

In empirical tests we used two problem instances: one with 30 uniformly distributed targets, and one with 30 cluster distributed targets. Instance with uniformly distributed targets is harder to solve. All tests were performed in the working domain which was a square of the size 100 m by 100 m. For all drones in the population we set r to 15 m, similar as in [15]. Test instances were generated using pseudo-random number generator.

It should be noted that the metric units and the nature of the targets are not relevant, since they are used only for experimental purposes.

We adjusted EHO algorithm parameters as follows: number of clans $n = 5$, number of solutions in each clan $n_{ci} = 10$, and maximum generations number $MaxGen = 800$, which yields the total number of 40,000 objective function evaluations. Additionally, we set values for scale factors α and β to 0.5 and 0.1, respectively. Same values for n, n_{ci}, α and β parameters were used in [10].

We developed our own software framework for testing purposes in Visual Studio 2017 with .NET Framework 4.7. Software has three main components: EHO algorithm, problem set component for generating instances, and component for visual representation of the obtained results. We tested our algorithm in 30 independent runs, each starting from different pseudo-random number seed. All tests were performed on Intel CoreTM i7-4770HQ processor @2.4GHz with 32GB of RAM memory.

Experimental results for 30 clustered and 30 randomly distributed targets are shown in Tables I and II, respectively. In tables, we show the obtained results for different number of employed drones and for absolute coverage of targets, coverage of targets in percentile values, and execution time of the EHO algorithm. We also show best and mean results obtained in 30 independent runs of the algorithm.

TABLE I: Experimental results for drone placement in clustered targets scenario

Results for 30 clustered targets

Drone No.		Targets Cov.	Targets Cov. %	Exec.Time (s)
1	Best	9	30%	2.3
	Mean	9	30%	2.3
2	Best	16	53.3%	3.2
	Mean	16	53.3%	3.2
3	Best	23	76.6%	5.5
	Mean	23	76.6%	5.5
4	Best	30	100%	7.9
	Mean	30	100%	7.9

Despite the fact that all targets cannot be covered by only one drone, for experimental purposes, we conducted tests with lower number of drones to analyze behavior of the EHO algorithm. In clustered targets scenario, targets are grouped into four clusters, where three clusters have 7 tags and one cluster has 9 tags. As expected, with one employed drone, only one cluster was covered. In this case, the algorithm performs well covering the cluster with the most tags (9 tags). This scenario is shown on the left side of Figure 1.

As can be seen in Table I, with 2 and 3 drones, 16 and 23 targets are covered. As expected, the algorithm managed to cover all targets with optimal number of 4 drones. This scenario is depicted on the right side of Figure 1.

In clustered targets experiments, best and mean values are equal, and the algorithm manages to cover optimal number of targets with different drone numbers.

From the Table II, where we show results for 30 uniformly distributed targets, it can be concluded that the EHO estab-

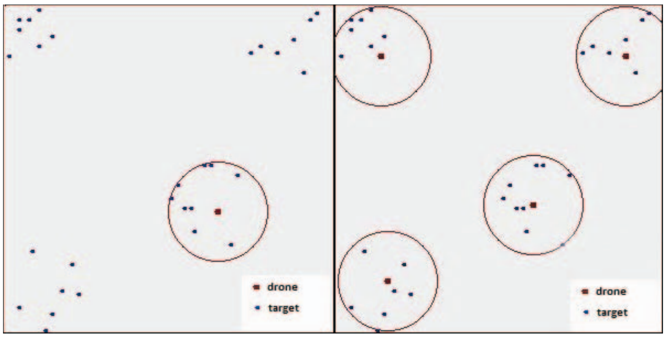

Fig. 1: Examples with one drone (left), and four drones (right) in clustered target set

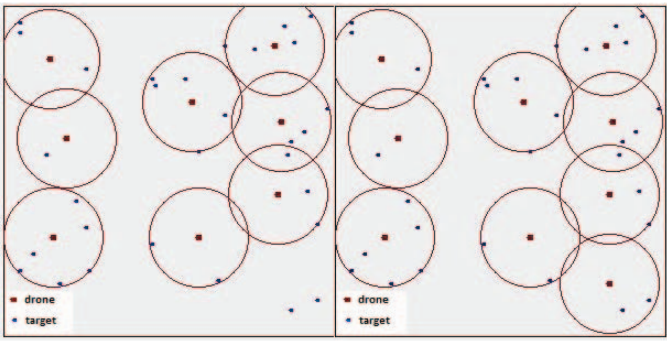

Fig. 2: Examples with eight drones (left), and nine drones (right) in random target set

lishes coverage of all targets, with 9 drones, which is the optimal number of drones for this particular test instance. Scenario with 9 drones is showen on the right side of Figure 2.

TABLE II: Experimental results for drone placement in random targets scenario

Results for 30 random targets				
Drone No.		**Targets Cov.**	**Targets Cov. %**	**Exec.Time (s)**
1	Best	6	20%	3.5
	Mean	5	16.6%	3.5
2	Best	11	36.6%	5.3
	Mean	10	33.3%	5.3
3	Best	15	50%	7.1
	Mean	13	76.6%	7.1
4	Best	18	60%	8.8
	Mean	16	53.3%	8.8
5	Best	21	70%	11.3
	Mean	20	66.6%	11.3
6	Best	24	80%	15.9
	Mean	22	73.3%	15.9
7	Best	26	86%	19.7
	Mean	23	76.6%	19.7
8	Best	28	93%	25.5
	Mean	27	90%	25.5
9	Best	30	100%	33.4
	Mean	29	96.6%	33.4

In the case of randomly distributed targets, best and mean values are different, as can be seen from Table II. Due to the difficulty of this test instance, in some runs the algorithm was unable to cover the largest possible number of targets with available drones. Thus, we ran additional tests, where we increased the number of generations of algorithm's execution to 1,200, and in this case, best and mean values were equal.

V. Conclusion

In this paper we have introduced the elephant herding optimization (EHO) algorithm adopted for solving static drone location problem (SDLP). EHO is relatively new swarm intelligence metaheuristics and it was not tested on similar problems before. We tested the EHO algorithm on instances with 30 clustered and 30 randomly deployed targets. According to the presented experimental study, it can be concluded that EHO has potential in dealing with these kinds of problems.

References

[1] H. Chen, X.-M. Wang, and Y. Li, "A survey of autonomous control for UAV", in *Proceedings of the 09 International Conference on A Artificial Intelligence and Computational Intelligence (AICI 2009)*. IEEE, November 2009, pp. 267–271.

[2] R. I. Bor-Yaliniz, A. El-Keyi, and H. Yanikomeroglu, "Efficient 3-d placement of an aerial base station in next generation cellular networks", in *2016 IEEE International Conference on Communications (ICC)*. IEEE, 2016, pp. 1–5.

[3] E. Tuba, M. Tuba, and D. Simian, "Wireless sensor network coverage problem using modified fireworks algorithm", in *International Wireless Communications and Mobile Computing Conference (IWCMC)*. IEEE, 2016, pp. 696–701.

[4] J. Lyu, Y. Zeng, R. Zhang, and T. J. Lim, "Placement optimization of UAV-mounted mobile base stations", *IEEE Communications Letters*, vol. 21, no. 3, pp. 604–607, 2017.

[5] M. Younis and K. Akkaya, "Strategies and techniques for node placement in wireless sensor networks: A survey", *Ad Hoc Networks*, vol. 6, no. 4, pp. 621–655, June 2008.

[6] N. Bacanin, M. Tuba, and I. Strumberger, "RFID network planning by ABC algorithm hybridized with heuristic for initial number and locations of readers", in *Proceedings of the IEEE International Conference on Modeling and Simulation*, March 2015, pp. 39–44.

[7] I. Strumberger, N. Bacanin, and M. Tuba, "Enhanced firefly algorithm for constrained numerical optimization", in *Proceedings of the IEEE International Congress on Evolutionary Computation (CEC 2017)*, June 2017, pp. 2120–2127.

[8] E. Tuba, M. Tuba, and E. Dolicanin, "Adjusted fireworks algorithm applied to retinal image registration", *Studies in Informatics and Control*, vol. 26, no. 1, pp. 33–42, 2017.

[9] E. Tuba, M. Tuba, and M. Beko, "Node localization in ad hoc wireless sensor networks using fireworks algorithm", in *5th International Conference on Multimedia Computing and Systems (ICMCS)*. IEEE, 2016, pp. 223–229.

[10] G.-G. Wang, S. Deb, and L. dos S. Coelho, "Elephant herding optimization", in *Proceedings of the 2015 3rd International Symposium on Computational and Business Intelligence (ISCBI)*, 2015, pp. 1–5.

[11] G.-G. Wang, S. Deb, X.-Z. Gao, and L. dos S. Coelho, "A new metaheuristic optimisation algorithm motivated by elephant herding behaviour", *International Journal of Bio-Inspired Computation*, vol. 8, no. 6, pp. 394–409, January 2017.

[12] E. Tuba and Z. Stanimirovic, "Elephant herding optimization algorithm for support vector machine parameters tuning", in *Proceedings of the 2017 International Conference on Electronics, Computers and Artificial Intelligence (ECAI)*, June 2017, pp. 1–5.

[13] E. Tuba, A. Alihodzic, and M. Tuba, "Multilevel image thresholding using elephant herding optimization algorithm", in *Proceedings of 14th International Conference on the Engineering of Modern Electric Systems (EMES)*, June 2017, pp. 240–243.

[14] S. Gupta, V. P. Singh, S. P. Singh, T. Prakash, and N. S. Rathore, "Elephant herding optimization based PID controller tuning", *International Journal of Advanced Technology and Engineering Exploration*, vol. 3, no. 24, pp. 194–198, January 2016.

[15] D. Zorbas, T. Razafindralambo, D. P. P. Luigi, and F. Guerriero, "Energy eficient mobile target tracking using flying drones", in *Procedia Computer Science, International Conference on Ambient Systems, Networks and Technologies (ANT)*, vol. 19, 2013, pp. 80–87.