EAI Endorsed Transactions
on Internet of Things

Research Article

EAI.EU

Improvised multi-robot cooperation strategy for hunting a dynamic target

Oussama Hamed[1,*] and Mohamed Hamlich[1]

[1] ISSIEE Laboratory ENSAM, University of Hassan II, Morocco

Abstract

Hunting moving targets with random motions and behavior is a challenge for robotic systems, and it occupies a significant position in the research on coordination and cooperation in multi-robot systems. Cooperative hunting between robots is used in a wide range of fields such as industry, military, rescue and other fields. The aim of this paper is to present a new strategy for target hunting by means of a cooperative multi-robot system in two-dimensional space, especially moving targets with random and unexpected behavior. The strategy is inspired from the behavior of wolves in hunting and Wolf Swarm Algorithm, as it summarizes the roles of robots in the system in three roles: the leader wolf, the antagonist wolf, and the follower wolf. This diversity of roles contributed to improve the convergence performance of the algorithm and reduce significantly the pursuit time. The validity of this strategy is supported by computer simulations.

Keywords: Multi-Robot System (MRS), hunting, path planning, multiagent cooperation.

Received on 16 January 2021, accepted on 28 January 2021, published on 08 February 2021

doi: 10.4108/eai.8-2-2021.168691

1. Introduction

The multi-robot cooperation and coordination becomes a topic of primary importance in industry [1], military [2], research, etc. The use of the multi robot system (MRS) with coordinated and cooperated robots facilitates executing and solving complex tasks, that are difficult to solve with a single robot [3]. The key to optimize the overall global performance, is the manner to coordinate and cooperate between all the robots in the MRS, by orientating the behavior of each robot.

Hunting moving targets with random movements and behavior is a challenge for robotic systems [4]. This hunting process requires cooperation between robots [5], in order to chase the target, and constantly adapt to dynamic changes in target behavior. Furthermore, the hunting and pursuit tasks need analyzing of the situation and information available in real time by all the robots within the system, and sharing useful information between them [6]. Hence, this is a multidisciplinary task which requires knowledge in robotics, optimization algorithms, real-time dynamic path planning [7], wireless communication [8], management of MRS [9], etc. Therefore, it is a research and optimization problem; as it is a problem in order to deal with dynamic information that is constantly changing, and to find the goal through cooperation and coordination between robots, as well as trapping the target with the minimum time and resources possible. Although this problem has been significantly solicitous, it has not yet been solved.

In view of this issue, researchers developed various methods and technics to reach the target location and hunt it. To conduct the cooperative hunting by multi-robot in unknown and dynamic environments, Jianjun Ni and Simon X. Yang proposed a novel approach based on a bioinspired neural network for the real time cooperative hunting by multi-robot in unknown and changing environments [10]. Hiraoki Yamaguchi uses the feedback control law to coordinate the motion of multiple robots, and each robot has a formation vector to control the formations so that they capture a target by forming troop formations

[*] Corresponding author. Email: oussamahamed11@gmail.com

[11]. There are other methods that have been inspired from optimization algorithms [12–15]; such as Jim Pugh and Alcherio Martinoli that presented a multi-search algorithm, where they modify the Particle Swarm Optimization algorithm [16] to mimic the multi-robot search process [17]. Xialolin Luan and Yutting Sun proposed Wolf Swarm Algorithm [18], which is for encircling a target with underwater robots. This algorithm is built on the basis of three behaviors in the hunting process that are (preying, wandering, and summoning).

In this work, we present a new strategy inspired from Wolf Swarm Algorithm and wolf hunting behavior. This strategy dedicated to multi-robot systems to pursue and hunt a single dynamic target with an unknown behavior. The swarm contains three kinds of wolves, the leader, the antagonist and the followers. These three roles are temporary and constantly renewed according to the status of the wolf and the prey. The cognitive behavior of the antagonist and the followers, reduces the research space of the solution, prevents the convergence of the robots in a wrong position while the prey is free from hunters, and helps to find quickly the target.

The remainder of the paper is constructed as follows. In Section II, we are going to present the proposed strategy and explain the roles of the three types of robots. Section III is reserved for the results of the simulation and the discussion. Finally, Section VI concludes the paper and suggests some directions for future work.

2. The proposed strategy

The hunting strategy relies on three roles within the swarm. These three roles aid in finding and hunting the target using cooperative competitive behavior. The hunting process is performed by N robots in a two-dimensional space. The position of each robot R_r where r = (1,…,N) is expressed as $R_r = [R_{rx}, R_{ry}]$, where R_{rx} and R_{ry} are the variables to be optimized. The robots (wolves) try to get closer to the target (prey) by perceiving its odor concentration. The odor concentration is expressed by $Ø(R_r, T)$, where T is the target vector position and $Ø$: $IR^n \rightarrow IR$ is the fitness function to be maximized over the n-variable vector R_r. To calculate the distance between two vectors of the robot and target in a n-dimensional real vector space, the taxicab distance is applied as following:

$$d_1(R_r,T) = ||R_r,T||_1 = \sum_{j=1}^{n} |R_{r,j}-T_j| \qquad (1)$$

The following subsections describe the strategy and the algorithm of the swarm, as well a detailed explanation of the leader, antagonist and follower roles.

2.1. Leader

The leader is distinguished from the rest of the swarm by being the closest wolf to the prey, and it is the one who has an idea of where the prey is located. The leader is determined by comparing the prey odor concentration between all wolves, and the wolf who can detect the highest odor concentration takes the role of the leader. Hence, $\exists R^l$ $Ø(R^l, T) \geq Ø(Rr, T)$ where r = (1,…N). For each iteration, the fitness function value of the leader is compared with all the other wolves. Therefore, four scenarios are possible in this case. If the leader wolf has the highest fitness function value, it preserves its role as a leader. The same result if the leader wolf shares the highest fitness function value with one or more wolves. The wolf loses its leader position if another wolf has a higher fitness function value. In the case that several wolves have the same fitness function value, and no one of them is the leader, then the new leader is chosen from them randomly. In the whole swarm, it exists only one leader; it is the only who is allowed to summon the followers to capture the prey.

2.2. Antagonist

The antagonist wolf does not carry out the leader's orders, and it is always seeking for the leader title by searching for prey in his own way. The number of Antagonist wolves A_wolves is a random integer from the interval [(N-1)/(λ +1) , (N-1)/λ], where λ = [1, N/2] is the antagonism proportion factor. The antagonist wolf has its own cognitive way of searching for prey in the search space. Instead of looking in all directions for prey (Fig.1), the antagonist wolf listens to the leader wolf's howling when it summons its follower wolves, then estimates the position of the prey. Therefore, this procedure allows the wolf to limit the directions in which to search for prey.

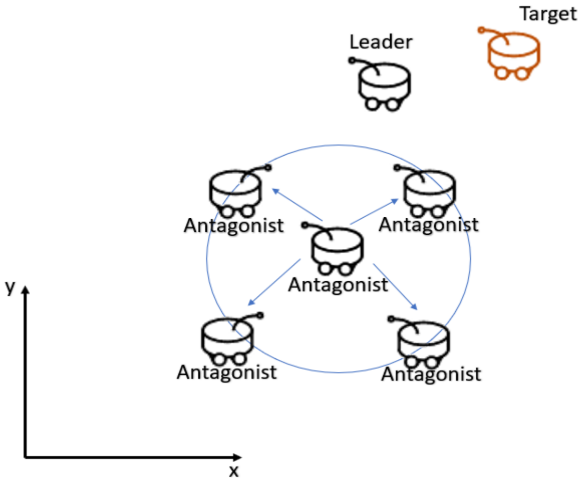

Figure 1. The antagonist looks for the target in all directions

From the position of the leader wolf R^l, the antagonist R_k^a can predict the potential positions of the target. Based on potential target positions, the seeking area is reduced to a specific area instead of searching everywhere in all directions (Fig.2).

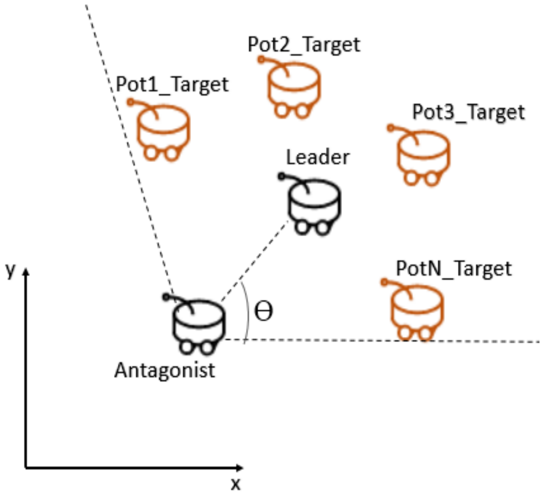

Figure 2. The antagonist limits the area seeking for the target

The angle Θ is the angle between the horizontal vector from (R^a_{kx} , R^a_{ky}) to (R^a_{kx}+ 1 , R^a_{ky}) and the vector (R^a_{kx} , R^a_{ky}) to (R^l_x,R^l_y).

$$\Theta = 2\arctan \frac{R^l_y - R^a_{ky}}{\sqrt{(R^l_x - R^a_{kx}+1)^2+(R^l_y - R^a_{ky})^2} + R^l_x - R^a_{kx} + 1} \qquad (2)$$

Where R^a_{ky} and R^a_{kx} are the coordinates of the antagonist wolf R^a_k with $1 \leq k \leq$ A_wolves, and the angle Θ value is by radian.

$$\Theta' = \begin{cases} \Theta, & \Theta \in [0, \pi] \\ \Theta + 2\pi, & \Theta \notin [0, \pi] \end{cases} \qquad (3)$$

The motion of the R^a_k is affected linearly by its current position and the potential positions of the target. Hence, the R^a_k control is expressed as follows:

$$\begin{pmatrix} \widetilde{R^a_{kx}} \\ \widetilde{R^a_{ky}} \end{pmatrix}_{\varphi} = \begin{pmatrix} R^a_{kx} \\ R^a_{ky} \end{pmatrix} + \begin{pmatrix} \beta_1 \\ \beta_2 \end{pmatrix} \circ \begin{pmatrix} \cos(\frac{2\pi\varphi}{\psi}) \\ \sin(\frac{2\pi\varphi}{\psi}) \end{pmatrix} \qquad (4)$$

Where $(\widetilde{R^a_{kx}} , \widetilde{R^a_{ky}})^t_{\varphi}$ is the temporary position of R^a_k in accordance with the integer φ which is the factor for determining the advancing direction and it takes a range of values as it is expressed in (5).

The vector β is the seeking vector which is expressed in equation (9). ψ is the amount of the global directions whither the R^a_k will seek for the prey.

$$\varphi = \left[\frac{\psi \text{ x } \Theta'}{2\pi} - l_1 \ , \frac{\psi \text{ x } \Theta'}{2\pi} + l_2 \right] \qquad (5)$$

l_1 and l_2 where $1 \leq l_1, l_2 \leq \psi$ are two integers for limiting seeking area. The benefit of using these two factors is to determine whether the hunting field is broad or narrow. As shown in Fig.3, a broad band field of hunting helps to get closer from the prey more accurately but it takes more time. While, using a narrow band gives a quick result but it is not accurate.

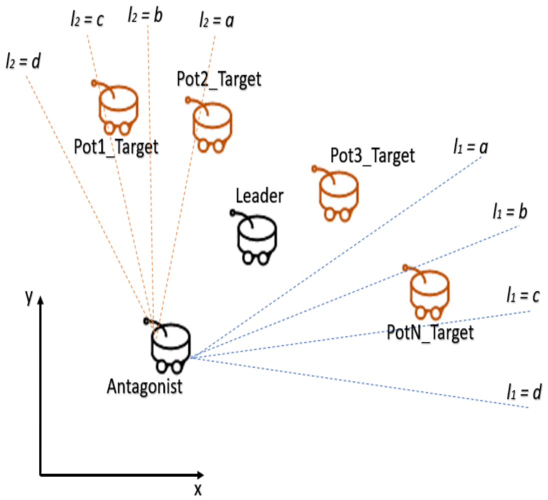

Figure 3. The antagonist limits the seeking area according to the value of the two factors l_1 and l_2, where the integers used in this example a < b < c < d define the range of target search.

The wolf R^a_k memorizes the position with the highest fitness function value, and it updates its position if $\max(\text{Ø}(\widetilde{R^a_{k\varphi}} , T)) > \text{Ø}(R^a_k, T)$. Hereafter, the antagonist wolf wins the leader position if $\text{Ø}(R^a_k, T) > \text{Ø}(R^l, T)$, and the leader becomes an antagonist.

2.3. Followers

Each follower wolf R^f_j with j = [1, N-A_wolves-1] performs two behaviors for hunting the prey.

Summoning behavior

When the leading wolf starts howling, the follower wolves go directly towards the leader, since it is the closest wolf to the prey. The new position of the wolf R^f_j is expressed as follows:

$$\begin{pmatrix} R^f_{jx}\ (i+1) \\ R^f_{jy}\ (i+1) \end{pmatrix} = \begin{pmatrix} R^f_{jx}\ (i) \\ R^f_{jy}\ (i) \end{pmatrix} + \begin{pmatrix} \alpha_1 \\ \alpha_2 \end{pmatrix} \Theta \begin{pmatrix} R^l_x - R^f_{jx}\ (i) \\ R^l_y - R^f_{jy}\ (i) \end{pmatrix} \Theta$$

$$\begin{pmatrix} \varepsilon_1 \\ \varepsilon_2 \end{pmatrix} \Theta \begin{pmatrix} \left|R^l_x - R^f_{jx}\ (i)\right|^{-1} \\ \left|R^l_y - R^f_{jy}\ (i)\right|^{-1} \end{pmatrix} \qquad (6)$$

Where α is the advancing vector which is expressed in equation (9), and i is the iteration index. The vector $(\varepsilon_1, \varepsilon_2)^t$ is the deviation vector where $\varepsilon_1, \varepsilon_2$ are random numbers in $0 < \varepsilon_1, \varepsilon_2 \leq 1$.
With the deviation vector, the follower wolf does not take the direct path toward the leader. Instead, its way deviates randomly according to the factors ε_1, ε_2. This deviation helps to get a wide coverage area by the followers. As shown in the Fig.4, the follower can be in any position in the colored area (ε_1 x α_1 x α_2 x $\varepsilon 2$), according to the value of the deviation and advancing factors. Whereas, if the prey takes a direction toward the colored area or enters into this area, it may find a follower which becomes then the closest wolf.

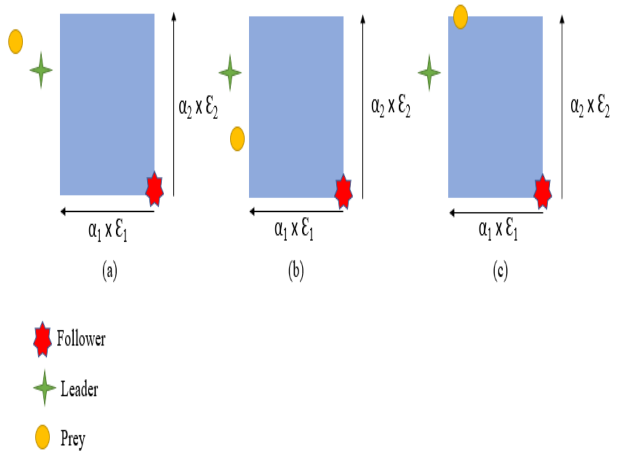

Figure 4. In the initial case (a), the follower advances toward the leader while its next position is in the colored area. (b) And (c) are two of the next cases possible.

The follower wolf changes its behavior to preying when the distance between it and the leader becomes smaller than $D_{conv.}$ The distance of convergence D_{conv} is described as:

$$\text{Dconv}=\left(\frac{1}{\max(xs)-\min(xs)}+\frac{1}{\max(ys)-\min(ys)}\right)^{-1} \text{x } \sigma \qquad (7)$$

Where **σ** =]0, 1] is the convergence factor. (xs, ys) are the boundaries of the search space.

Preying behavior

With this behavior, the follower wolves encircle and swoop the prey. Hence the new position of the wolf R^f_j is expressed as follows :

$$\begin{pmatrix} R^f_{jx}\ (i+1) \\ R^f_{jy}\ (i+1) \end{pmatrix} = \begin{pmatrix} R^f_{jx}\ (i) \\ R^f_{jy}\ (i) \end{pmatrix} + \begin{pmatrix} \gamma_1 \\ \gamma_2 \end{pmatrix} \Theta \begin{pmatrix} R^l_x - R^f_{jx}\ (i) \\ R^l_y - R^f_{jy}\ (i) \end{pmatrix} \Theta \begin{pmatrix} \tau_1 \\ \tau_2 \end{pmatrix} \qquad (8)$$

Where $(\tau_1, \tau_2)^t$ is the encirclement vector, whereas $0 \leq \tau_1, \tau_2 \leq 2 - i/\text{max\_Iteration}$, and γ is the preying vector (9). These two factors τ_1, τ_2 are for mimics the encirclement process of the wolves (Fig.5). This process prevents the escape of the prey. At each iteration, the follower wolf changes its position randomly within a determined area. The size of this area depends on the distance between the follower and the prey, the encirclement factors, and the preying factors.

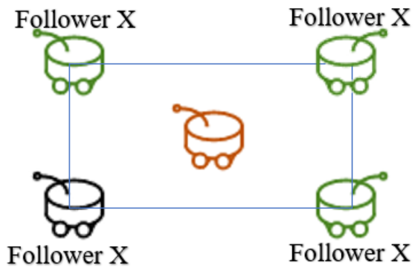

Figure 5. Encirclement process of prey for a single follower wolf

After encircling the prey, the wolves finish the hunting process by attacking the target. For the mathematical model of approaching the prey, the maximum value of the vector $(\varepsilon_1, \varepsilon_2)^t$ decreases linearly after each iteration from 2 to 1. Fig.6(a) shows that the area where the wolf can move in the earlier iteration is greater than the latest. Fig.6(b). While in the final iterations, the area is the smallest. Fig.6(c).

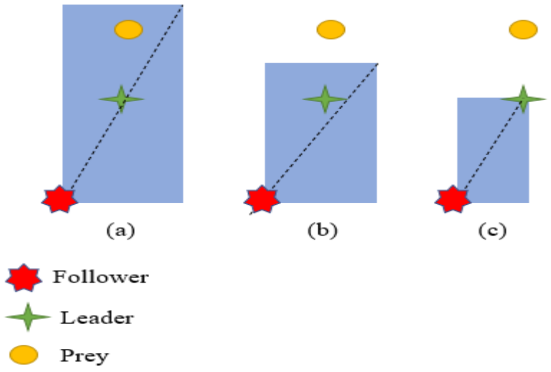

Figure 6. Encirclement and attacking the prey

Search process

The values of the vectors α, β and γ are proportional and expressed as follows:

$$\begin{pmatrix}\beta_1\\\beta_2\end{pmatrix} = \frac{4}{3}.\begin{pmatrix}\gamma_1\\\gamma_2\end{pmatrix} = \frac{1}{2}.\begin{pmatrix}\alpha_1\\\alpha_2\end{pmatrix} = \begin{pmatrix}\max(xs)\text{- }\min(xs)\\\max(ys)\text{- }\min(ys)\end{pmatrix}.\ S \quad (9)$$

The scalar S where 0 < S < 1 is the step factor which is related mainly to the search speed of the solution in the research space, and the characteristics of the robot and also the workspace.

3. Simulation results

In this section, we present simulation results to validate the proposed strategy of cooperative hunting operation. The simulation experiments are based on MATLAB Online R2020b.

Table 1. PARAMETER ASSUMPTIONS

Parameters	Meaning	Values
N	Total of wolves	4
λ	Antagonism proportion factor	2
S	Step factor	0.05
σ	Convergence factor	0.65
ψ	Seeking directions	8
l_1	Limit one of seeking area	1
l_2	Limit two of seeking area	3

In the following simulation process, we adopted three experiments for the different initial positions of the robots and the target, as well as random movements of the target that are randomly determined in the algorithm where P_x and P_y the coordinates of the target: $P_x(i+1) = P_x(i) + \mu_1$ and $P_y(i+1) = P_y(i) + \mu_2$. μ_1 and μ_2 are random numbers where $-1 \leq \mu_1,\mu_2 \leq 1$.

The multi robot system is consisted of four mobile robots, one of them is the antagonist (∇) and the others (X) are the followers and the leader. All the locations of these robots in addition to the target (O) are initialized randomly into a two-dimensional space 20x20. To catch the target, we need at least three wolves.

In Fig.7, Fig.8, Fig.9 we have initialised four robots including the robot who plays the role of the antagonist wolf.

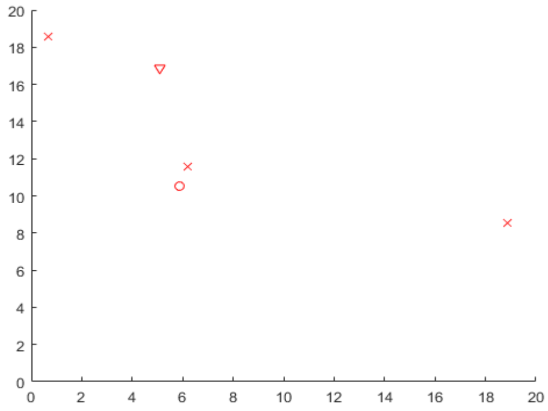

Figure 7. Initial positions of the robots and the target for the first experiment

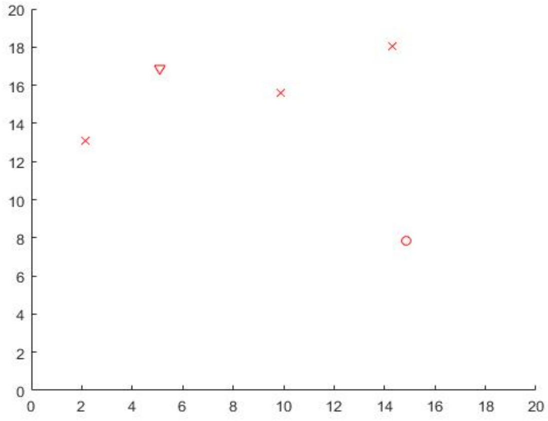

Figure 8. Initial positions of the robots and the target for the second experiment

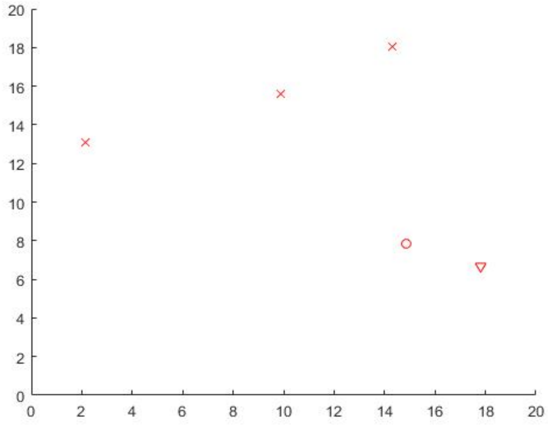

Figure 9. Initial positions of the robots and the target for the third experiment

For the Wolf Swarm Algorithm (WSA) There are no random parameters in the algorithm, where $(\tau_1,\tau_2)\ ^t$=$(\mathcal{E}_1,\mathcal{E}_2)^t = (1{,}1)^t$. Therefore, the result is unchangeable if the same experience is repeated with the same initial positions and the same behavior and movements of the target.

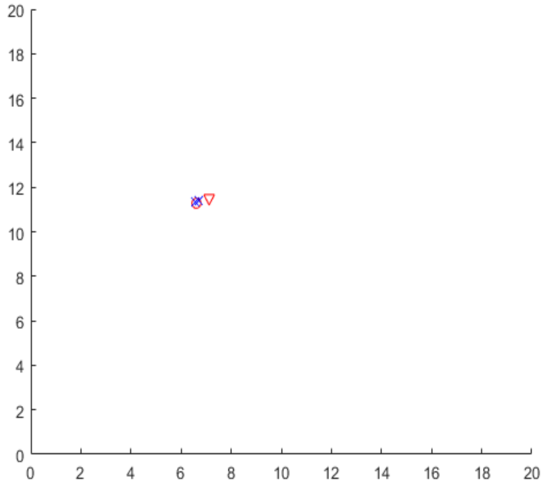

Figure 10. Robots hunt the target for the first experiment using WSA

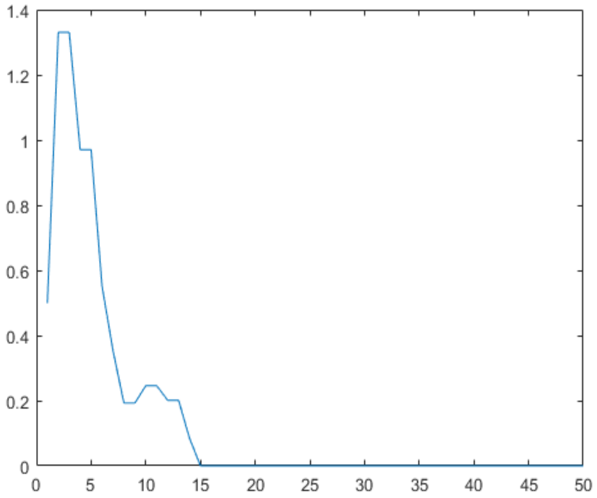

Figure 11. Best cost function progression for the first experiment using WSA

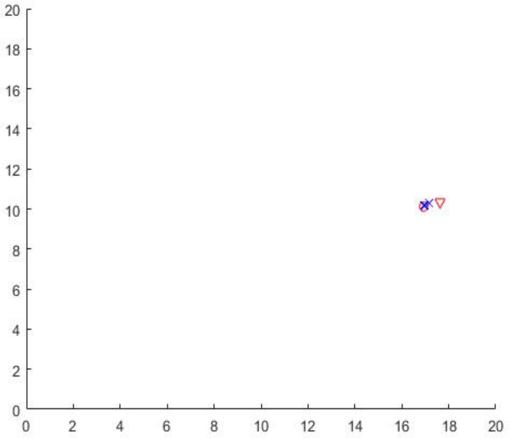

Figure 12. Robots hunt the target for the second experiment using WSA

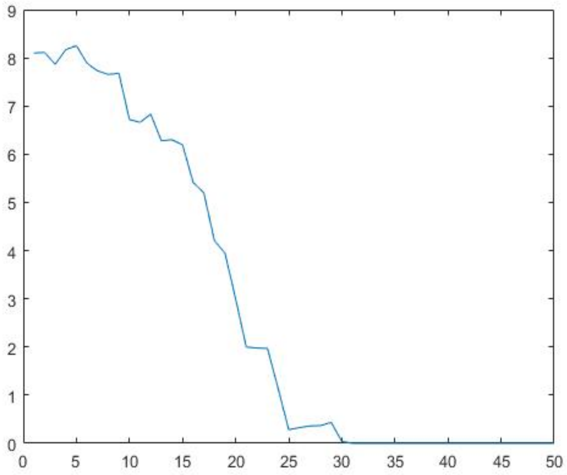

Figure 13. Best cost function progression for the second experiment using WSA

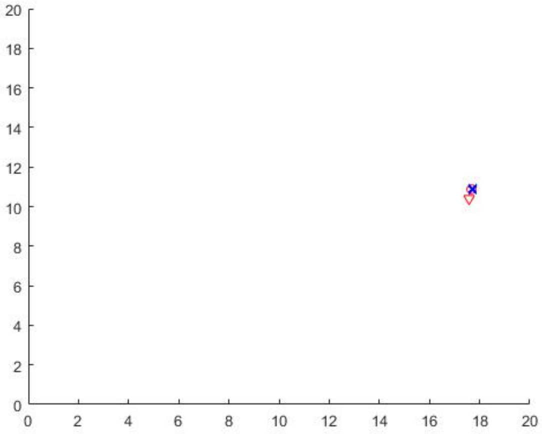

Figure 14. Robots hunt the target for the third experiment using WSA

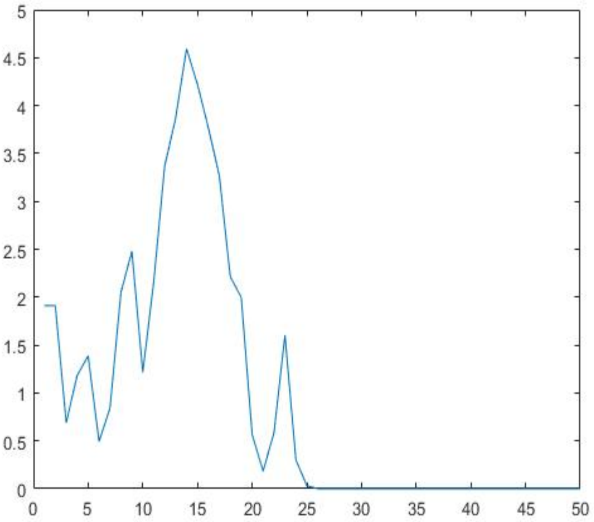

Figure 15. Best cost function progression for the third experiment using WSA

We have performed the same experiences with respect the same initial positions and the target behavior, but at this time the values of deviation and encirclement factors are random as expressed previously. The use of random factors contributed to obtaining different results each time we have repeated the same experiment. In order to differentiate and clarify the role of both factors separately, we first perform the experiments with encirclement vector $(\tau_1, \tau_2)^t=(1,1)^t$, and random values of deviation factors as mentioned in the section of summoning behavior. The three experiments were repeated fifty times for each experiment, and the results in the table. 2, table.3 and table.4 show the rate of times the prey has been caught in each number or set of numbers of iterations.

Table 2. Results of the first experiment for $(\tau_1,\tau_2)^t=(1,1)^t$ and random ε

Iteration	1-7	8	9	10	11	12-19	20-29	30-40	>40
Rate	0.02	0.14	0.16	0.24	0.12	0.2	0.06	0.02	0.04

Table 3. Results of the second experiment for $(\tau_1,\tau_2)^t=(1,1)^t$ and random ε

Iteration	1-10	11-24	25	26	27	28	29	30-40	>40
Rate	0	0.22	0.06	0.12	0.18	0.1	0.12	0.1	0.1

Table 4. Results of the third experiment for $(\tau_1,\tau_2)^t=(1,1)^t$ and random ε

Iteration	1-10	11-20	21	22	23	24	25-29	30 - 40	>40
Rate	0	0.24	0.12	0.18	0.14	0.12	0.1	0.04	0.06

These results show that despite the use of deviation factors that are random numbers, hunting by using these factors has given better results than (WSA) in most attempts. As shown in the table 2, the hunting process for the first experiment has been accomplished in 10 iterations 12 times from 50 attempts, which is a percent of 24%. table 3 shows that the hunting process for the second experiment has been accomplished in 27 iterations 9 times from 50 attempts, which is a percent of 18%, and 11 times with a percent of 22% for the iterations between 11 and 24. For the third experiment, the hunting process has been accomplished in 22 iterations 9 times from 50 attempts, which is a percent of 18% as shown in the table 4. In the figures Fig.16 – Fig.21, we expose the results of the hunt and the best cost function for a sample of attempts that are performed in 10 iterations for the first experiment, 27 iterations for the second experiment, and 22 iterations for the third experiment.

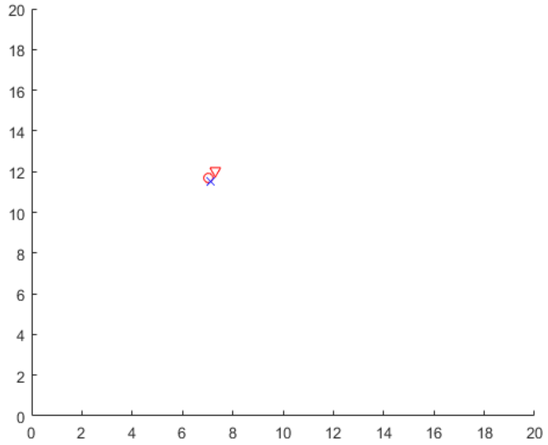

Figure 16. Robots hunt the target for the first experiment for $(\tau_1,\tau_2)^t=(1,1)^t$ and random Ɛ

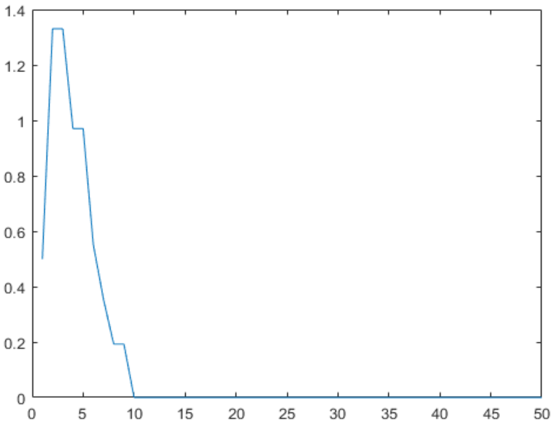

Figure 17. Best cost function progression for the first experiment for $(\tau_1,\tau_2)^t=(1,1)^t$ and random $\mathcal{E}$

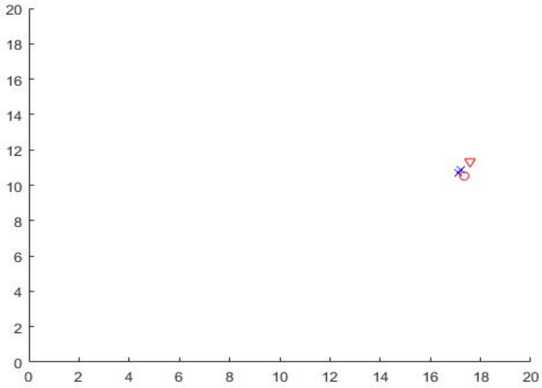

Figure 18. Robots hunt the target for the second experiment for $(\tau_1,\tau_2)^t=(1,1)^t$ and random $\mathcal{E}$

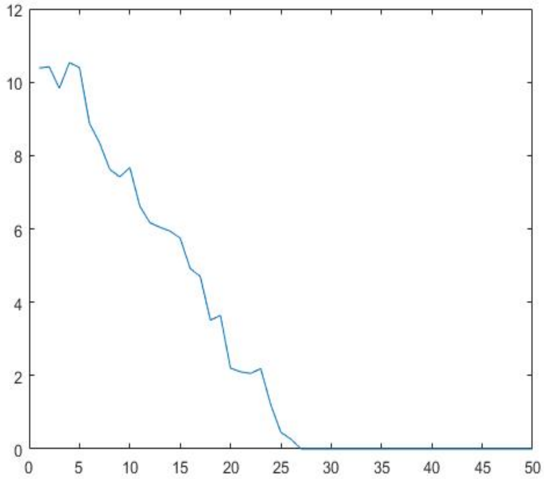

Figure 19. Best cost function progression for the second experiment for $(\tau_1,\tau_2)^t=(1,1)^t$ and random $\mathcal{E}$

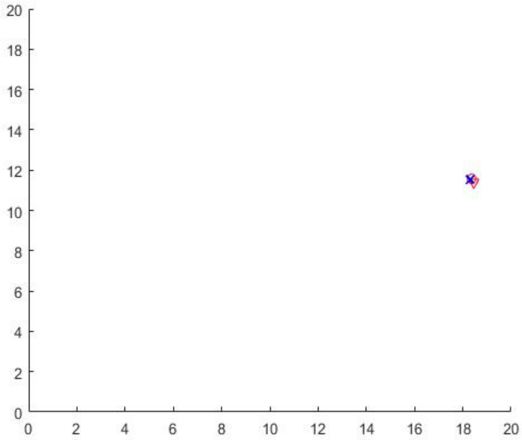

Figure 20. Robots hunt the target for the third experiment for $(\tau_1,\tau_2)^t=(1,1)^t$ and random $\mathcal{E}$

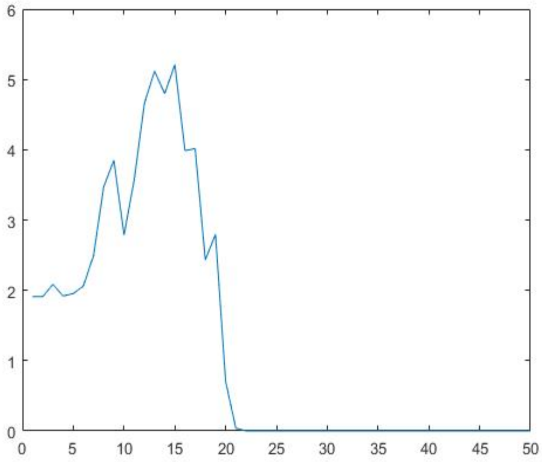

Figure 21. Best cost function progression for the third experiment for $(\tau_1,\tau_2)^t=(1,1)^t$ and random $\mathcal{E}$

Adding the deviation vector to the behavior of following robots contributed to the hunting of prey in the fewest number of iterations compared to Wolf Swarm Algorithm. Notwithstanding the foregoing, the results were not always good due to the use of random factors. Getting good results or bad results depend mainly on the random movement of the target, and the fact that we have hunter robots that depend on random deviation factors helps, with a large percentage, to track prey from at least one of these robots.

At this time, the three experiments were repeated again fifty times for each experiment, and the values of the two vectors $(\tau_1,\tau_2)^t$ and $(\mathcal{E}_1,\mathcal{E}_2)^t$ are calculated according to the previous sections related with summoning behavior and preying behavior. The results are in the in the tables table 5, table 6 and table 7.

Table 5. Results of the first experiment for random ε and τ

Iteration	1-6	7	8	9	10	11 - 19	20 - 29	30 - 40	>40
rate	0.02	0.1	0.18	0.22	0.18	0.1	0.06	0.02	0.12

Table 6. Results of the second experiment for random ε and τ

Iteration	1-10	11-23	24	25	26	27	28-29	30-40	>40
rate	0	0.24	0.12	0.16	0.18	0.12	0.02	0.02	0.14

Table 7. Results of the third experiment for random ε and τ

Iteration	1-10	11-19	20	21	22	23	24-29	30 -40	>40
rate	0	0.26	0.14	0.2	0.16	0.1	0.02	0	0.12

The use of the encirclement vector $(\tau_1, \tau_2)^t$ as mentioned in the section of preying behavior has improved the result of the hunt compared to WSA and using $(\tau_1, \tau_2)^t$=$(1,1)^t$ in the experiments. As shown in the table 5, the hunting process for the first experiment has been accomplished in 9 iterations 11 times from 50 attempts, which is a percent of 22%. Table 6 shows that the hunting process for the second experiment has been accomplished in 26 iterations 9 times from 50 attempts, which is a percent of 18%. For the third experiment, the hunting process has been accomplished in 21 iterations 10 times from 50 attempts, which is a percent of 20% as shown in the table 7.

Although adding the encirclement vector has led to improved results by reducing the number of iterations needed to capture the prey, the number of times the prey was caught after 40 iterations has been increased. The reason for this increase is the use of the random factor to capture the prey, so that the wolf may sometimes need a longer time in order to catch the prey even though it is close to it.

In the figures Fig.22 – Fig.27, we expose the results of the hunt and the best cost function for a sample of attempts that are performed in 9 iterations for the first experiment, 26 for the second experiment, and 21 iterations for the third experiment.

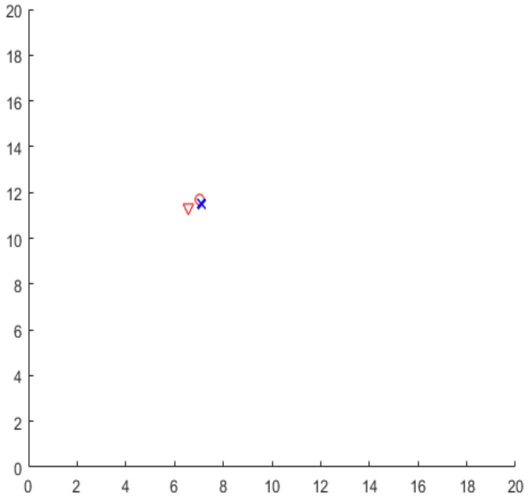

Figure 22. Robots hunt the target for the first experiment for random ε and τ

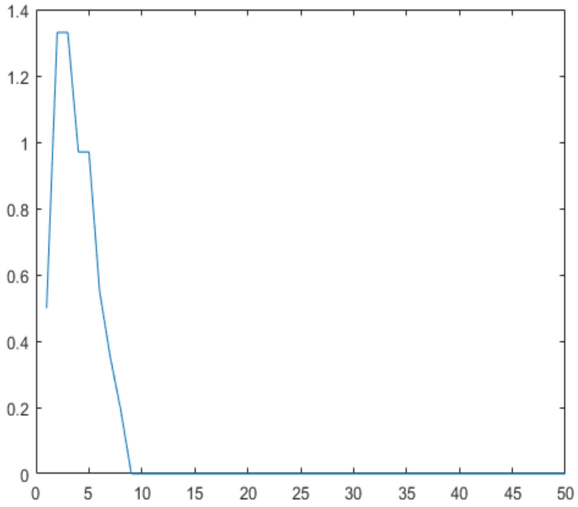

Figure 23. Best cost function progression for the first experiment for random ε and τ

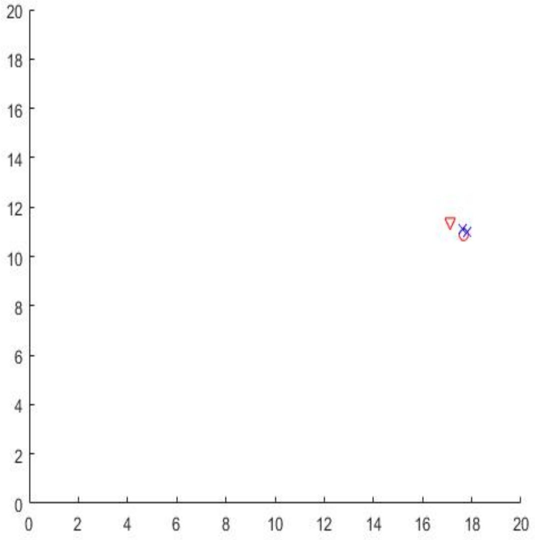

Figure 24. Robots hunt the target for the second experiment for random ε and τ

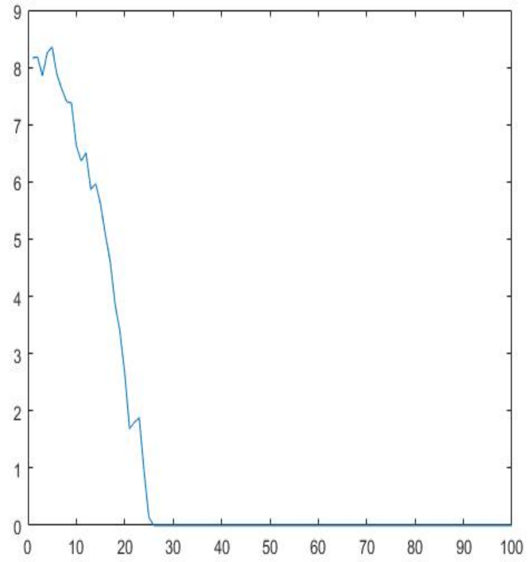

Figure 25. Best cost function progression for the second experiment for random ε and τ

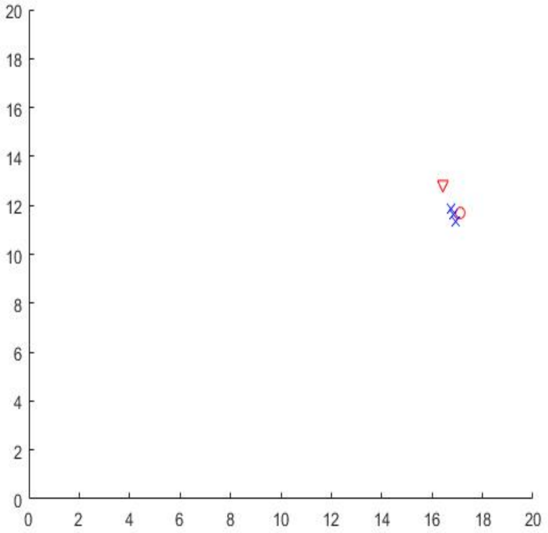

Figure 26. Robots hunt the target for the third experiment for random ε and τ

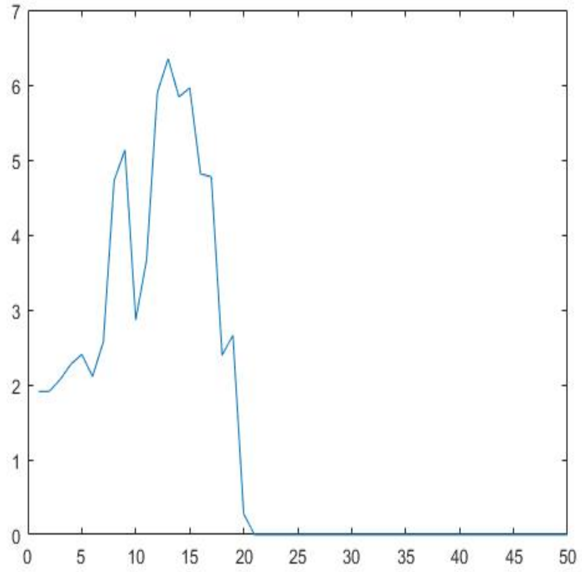

Figure 27. Best cost function progression for the third experiment for random ε and τ

The improvement was not only limited to the number of attempts at which the prey was captured, but also the execution time. Using the antagonist wolf strategy to compete with the wolf leader for leadership was of greater benefit. As this wolf's method of tracking prey enables it to reach the prey faster than searching in all directions, as is the case with the WSA.

As shown in the table 8, we experimented the hunting operations in different conditions while varying the total number of directions in each experiment, then compared the execution time of the WSA and the proposed strategy. The execution time for the Wolf Swarm Algorithm depends mainly on the total number of directions and the number of iterations the hunting process has taken. For the proposed strategy, the execution time does not depend on the total number of directions, in which the number of directions that a wolf will search for the prey is determined by the two factors l_1 and l_2. In these experiments, the limit one of seeking area $l_1 = 1$ and the limit two of seeking area $l_2 = 3$, where the robot looks for the target in five directions.

Table 8. Comparison between execution time in ms for vice leader wolf in WSA and the antagonist wolf for the proposed strategy for l1 =1 and l2 = 3.

Total directions	6	8	16	24	32	64
WSA	0.2475	0.2242	0.2420	0.3885	0.3519	0.4214
Proposed strategy	0.1475	0.1515	0.1368	0.1256	0.1318	0.1384

It is worth noting that reducing the total number of directions does not necessarily mean reducing the execution time. In fact, in some experiments, the execution time was less when we increased the total number of directions. This can be interpreted as the total of the number of directions is proportional to the accuracy of the research, the greater number of directions, the greater the accuracy. Also, the accuracy of the search helps in finding the target in fewer iterations.

4. Conclusion and future work

This paper presents an improvised strategy of multi robot cooperation for hunting a dynamic target, which is inspired from wolves behavior during the hunt process. This cooperative work is achieved by three types of robots that are the leader, the followers, and the antagonists. Each type of robot has its own cognitive behavior, which enables the optimum solution to be found. Hence, in this work, we improve the convergence performance and reduce significantly the pursuit time. The validity of this strategy is supported by computer simulations.

However, some limitations are worth noting. Although our strategy provided desirable results and was supported by numerical simulation, the existence of obstacles within the research space were not taken into account.

Future work should therefore include follow-up work designed to hunt the target on the premise of avoiding obstacles without reducing the global performance.

References

[1] L. Zaidi, M. Sahnoun, B. Bettayeb, “Task allocation based on shared resource constraint for multi-robot systems in manufacturing industry” IFAC-PapersOnLine,Volume 52, Issue 13, 2019, Pages 2020-2025, ISSN 2405-8963.

[2] A. Gautam and S. Mohan, "A review of research in multi-robot systems," 2012 IEEE 7th International Conference on Industrial and Information Systems (ICIIS), Chennai, 2012, pp. 1-5, doi: 10.1109/ICIInfS.2012.6304778.

[3] Sung Jin Yoo, Bong Seok Park, Connectivity preservation and collision avoidance in networked nonholonomic multi-robot formation systems: Unified error transformation strategy, Automatica, Volume 103, 2019, Pages 274-281, ISSN 0005-1098.

[4] Cao, Zhiqiang & Tan, M. & Li, Lei & Gu, Nong & Wang, Shuo. (2006). Cooperative Hunting by Distributed Mobile Robots Based on Local Interaction. Robotics, IEEE Transactions on. 22. 402 - 406. 10.1109/TRO.2006.862495.

[5] Y. Ma, Z. Cao, Xiang Dong, C. Zhou and M. Tan, "A multi-robot coordinated hunting strategy with dynamic alliance," 2009 Chinese Control and Decision Conference, Guilin, 2009, pp. 2338-2342, doi: 10.1109/CCDC.2009.5192770.

[6] Vidal, René & Shakernia, Omid & Kim, H. Jin & Shim, David & Sastry, Shankar. (2002). Probabilistic pursuit-evasion games: Theory, implementation, and experimental evaluation. Robotics and Automation, IEEE Transactions on. 18. 662 - 669. 10.1109/TRA.2002.804040.

[7] Bae, Hyansu; Kim, Gidong; Kim, Jonguk; Qian, Dianwei; Lee, Sukgyu. 2019. "Multi-Robot Path Planning Method Using Reinforcement Learning." Appl. Sci. 9, no. 15: 3057.

[8] Danilo Tardioli, Ramviyas Parasuraman, Petter Ögren, Pound: A multi-master ROS node for reducing delay and jitter in wireless multi-robot networks, Robotics and Autonomous Systems, Volume 111, 2019, Pages 73-87, ISSN 0921-8890.

[9] Serebrenny V., Shereuzhev M. (2020) Dependence of Dynamics of Multi-robot System on Control Architecture. In: Kravets A. (eds) Robotics: Industry 4.0 Issues & New Intelligent Control Paradigms. Studies in Systems, Decision and Control, vol 272. Springer, Cham. https://doi.org/10.1007/978-3-030-37841-7_10

[10] J.J. Ni, S.X. Yang, “Bioinspired neural network for real-time cooperative hunting by multirobots in unkown environnements”, IEEE Trans. Neural Netw. 22 (12) (2011) 2062-2077.

[11] H. Yamaguchi, “A cooperative Hunting Behavior by Mobile Robot Troops,” Proc.ieee Int.conf.robotics Autom., vol. 18, no.9, pp. 931-940, 1998.

[12] Milad Nazarahari, Esmaeel Khanmirza, Samira Doostie, Multi-objective multi-robot path planning in continuous environment using an enhanced genetic algorithm, Expert Systems with Applications, Volume 115, 2019, Pages 106-120, ISSN 0957-4174.

[13] Qilin Feng, Canxin Zhang, Jingyu Lu, Hao Cai, Zhilong Chen, Yibin Yang, Fei Li, Xianting Li, Source localization in dynamic indoor environments with natural ventilation: An experimental study of a particle swarm optimization-based multi-robot olfaction method, Building and Environment, Volume 161, 2019, 106228, ISSN 0360-1323.

[14] Yibin Yang, Boyuan Zhang, Qilin Feng, Hao Cai, Mingrui Jiang, Kang Zhou, Fei Li, Shichao Liu, Xianting Li, Towards locating time-varying indoor particle sources: Development of two multi-robot olfaction methods based on

whale optimization algorithm, Building and Environment, Volume 166, 2019, 106413, ISSN 0360-1323.

[15] K. Albina and S. G. Lee, "Hybrid Stochastic Exploration Using Grey Wolf Optimizer and Coordinated Multi-Robot Exploration Algorithms," in IEEE Access, vol. 7, pp. 14246-14255, 2019, doi: 10.1109/ACCESS.2019.2894524.

[16] J. Kennedy and R. Eberhart, "Particle swarm optimization," Proceedings of ICNN'95 - International Conference on Neural Networks, Perth, WA, Australia, 1995, pp. 1942-1948 vol.4, doi: 10.1109/ICNN.1995.488968.

[17] Jim Pugh and Alcherio Martinoli, “Inspiring and Modeling Multi-Robot Search with Particle Swarm Optimization”, In Swarm Intelligence Symposium, 2007. SIS 2007. IEEE, pages 332-339. IEEE, 2007.

[18] Xiaolin Luan and Yuting Sun, “Research on Coperative Encirclement Strategy of Multiple Underwater Robots based on Wolf Swarm Algorithm”, 2020 J.Phys. :Conf. Ser. 1570 012017