A novel bio-inspired algorithm for hunting in multi robot scenario

Poorva Agrawal[1], Himanshu Agrawal[2], Vidyasagar Potdar[3]
[1,2]Symbiosis Institute of Technology, Symbiosis International Deemed University, India
[3]Curtin Business School, Curtin University, Australia

Article Info

Article history:

Received Jan 20, 2019
Revised Apr 22, 2019
Accepted May 13, 2019

Keywords:

Alliance
Bio-inspired algorithm
Cooperative hunting
Multi robots

ABSTRACT

In a multi-robot scenario, cooperative hunting is a key issue when a group of robots are hunting for evader/evaders and when the location of the evader is continuously changing. Cooperative hunting is addressed in this paper by proposing a novel bio inspired Corner Dragging Algorithm (CDA). Corner Dragging Algorithm operates by making an alliance of robots that drag the evader towards any one of the four corners; whichever is closest to the evader. Different shapes of obstacles are avoided during this pursuit. While developing the Corner Dragging Algorithm, we analyze the shortcomings and advantages of some of the existing algorithms including dynamic alliance and formation construction algorithm and incorporate these changes in our design to achieve improved results. Performance of the algorithm is evaluated on the basis of simulation in MATLAB.

Corresponding Author:

Poorva Agrawal,
Symbiosis Institute of Technology,
Symbiosis International Deemed University,
Maharashtra, Pune 412115, India.
Email: poorva.agrawal@sitpune.edu.in

1. INTRODUCTION

In a multi robotic environment, multi-robots work cooperatively to complete missing critical tasks such as naval mine mission [1], catching invaders under surveillance areas [2], troop formation & threat assessment [3], hunting in ocean water [4] replace humans in dangerous environments or manufacturing processes [1], biomedical sciences [5], resemble humans in appearance, behavior and cognition for specific applications [6-8] etc.

This cooperative behavior of multi robots includes various tasks such as path planning [9], collision avoidance [10], target search [11] proper coordination among robots [6, 12], localization of the path [1, 13], proper hardware realization [14], task allocation [6] etc. These cooperative behavior traits [15] and coordination activities amongst robots are biologically inspired by specimens of wild life such as those in bats and fire flies [16-17] when they communicate with each other.

This paper focuses on the problem of multi robot hunting for catching evaders. An evader is an opponent who has to be caught but attempts to escape. The hunting problem deals with multiple robots that work cooperatively in order to catch the evaders [18]. For example, in line of control where human life could potentially be in danger; multi-robot teams can be used to catch the enemy.

This problem is a challenging problem because it encompasses numerous other sub-problems such as task allocation by the temporary commander who sees the evader first [19], searching for evaders [18, 20], localizing the positions of evaders [20], detecting and avoiding collisions that come in the way of robots when catching the evaders [20] etc.

The most challenging aspect in this hunting problem is to catch an evader who has higher degree of intelligence (e.g. high range sensors) compared to the multi-robots who are attempting to catch them [20].

Further, the path of the evader is not predefined and can be irregular and unpredictable. Hence these multi robots have to be extremely adaptive [21] and robust enough to fulfill the task of hunting these evaders efficiently as well as in a timely manner.

This paper proposes a modification to the dynamic alliance and strategy formation algorithm [20] termed as the *Corner Dragging Algorithm (CDA)* for the hunting problem. The strategy formation algorithm uses four robots to catch one evader by approaching from all four directions. In comparison, the novelty of CDA is that it drags the evader to a corner and uses only two robots to catch the evader instead of four. CDA limits the evader’s escape routes to minimal. It intelligently coordinates two robots to drag an evader to the corner before catching it. Simulation result shows the corner dragging algorithm supersedes the dynamic alliance and formation construction algorithm [20] in terms of the performance of hardware resources required, time required to catch the evader and related communication overhead on each robot.

The rest of the paper is organized as follows; Section 2 describes the related research work. Section 3 outlines the problem statement. Section 4 explains the proposed approach. The experiments and results are discussed in Section 5. Results analysis and its discussion in Section 6 and the conclusion and future scope are provided in Section 7.

2. RELATED WORK

This section presents the relevant research work in the area of multi robot systems. Multi robot systems demonstrate a joint collaborative behavior directed towards achieving a goal that has a common interest [22]. Here robots work together on a task that creates a progressive result such as increasing performance or saving time [21]. Coupled with bio-inspired algorithms, multi robot cooperation performs these tasks even more efficiently. Bio inspired algorithms solve many real-world problems by integrating various techniques including fuzzy logic [23], particle swarm optimization and hybrid models [24]. We now elaborate the relevant literature in this context.

2.1. Multi-Robots Hunting Tasks and Approaches

Ni and Yang have used neural networks based shunting [20, 25-26] for multi-robot hunting. Multi robot hunting approaches can be used to catch a single evader [2, 27] or multiple evaders [20, 28-29].

Xu et al. [27] effectively addressed the problem of one to one hunting task i.e. the task of a single hunter to catch a single prey. The authors here proposed a TAO-MTP algorithm (Tracking Automatic Optimization Moving - Target Pursuit). This algorithm uses a queue to store the trajectory of the prey, which help other robots to track the path. However this algorithm only works well when the speed of the hunter is more than that of prey.

Ma et al. proposed a dynamic alliance of multi robots for hunting [30]. This dynamic alliance attempted to catch more than one evader with an objective of decreasing the total hunting time. Yamaguchi used holonomic mobile robots to hunt a target using troop formations [2]. The system used local position feedback and a reactive control framework to form a troop.

Kachroo et al. focused on herding problem [31]. The dog sheep herding problem is considered, where sheep is the passive entity depending on dog’s position. Dynamic programming was used to solve this problem. The authors proposed an algorithm based on Dijkstra’s shortest path solution and direct dynamic programming solution algorithms. The direct dynamic programming solution showed superiority over Dijkstra’s shortest path algorithm in terms of complexity.

Yannakakis et al. focused on supervised and unsupervised learning mechanisms to address the multi robot coordination problem [32-34]. Particle Swarm Optimization approaches are now discussed. Zebing Wang proposed self-organizing cooperative hunting task [35] by robotic swarm. Each robotic swarm could detect the angle of target in motion. As human objects moved through the detection region, it made the proposed approach effective and feasible.

Zhang et al. [36] designed a self-organizing method for swarm robots using simplified virtual-force model for decomposition of hunting behavior in cluttered environment. This control method is used that takes location of target as an input and then decides the motion of swarm robots. Andreas Kolling et al. [37] present the human swarm interaction survey. Huang et al. [38] presented the cooperative hunting behavior mathematically. On the basis of decomposition of hunting behavior, the loose preference rule is considered as an interaction among individuals and the targets for ideal hunting.

Grid modeling [39] is used by swarm robots in a hunting task. To detect hunting points by control strategy and then using particle swarm optimization, the authors found optimal moving paths. Zebing Wang’s self-organizing cooperative hunting approach is different from the grid modeling approach because it finds optimal path and therefore could reach to the target in less time. Nighot et al. [40] used swarm intelligence in hunting process and focused on flexibility, robustness and self-organizing capability with the help of swarm intelligence.

Cooperative game based approaches are now discussed. Li et al. [41] proposed a game theory approach for multi-robot hunting target problem. The authors focused on coordination among robot by using two feasible searching strategies namely roaming search and regional search. The authors used non formation movement in hunting.

Krishna et al. [42] proposed many potential field functions using method of Lyapunov formation to avoid inter-robot collision, inter-formation and obstacle collision and attractions towards target. Chen Wang et al. [43] handled the problem to solve hunting activities in dynamic environment. Based on quick surrounding direction and quick capture direction, they obtained optimal path.

2.2. Task Allocation

Task allocation is way of assigning, subdividing or choosing any task of the problem. The issue with task allocation is to enable proper communication among team members. Hunt et al. [44] were inspired from cognitive behavior of animals for cooperation and coordination. The decentralized approach for groups or team to adapt changing task was inspired by ant and bees. The proposed algorithm improved the way for handling the task complexity and task dependencies. Yuan et al. [45] used contract net protocol task allocation problem in multi robot. They proposed a method that decides a successful bidder using neural network. The algorithm consensus based bundle [46] is extended and named as consensus basing grouping algorithm by Hunt et al. [44]. The results are improved on the basis of task allocation. In this algorithm the human involvement is reduced. Xiang [47] proposed a combined biological inspired self-organizing map and a velocity synthesis algorithm. The self-organizing map neural network is used by assigning a team of AUVs [47-49] to achieve multiple target locations. These algorithm when combined works well in task assignment and path planning in different scenarios. Drew et al. [50] used bounty hunters and bail bondsmen for multi-agent task allocation. The task is finished by bounty hunters and then they collect their bounties. Self-organizing map approach for task assignment in a 3-D environment helped Xin et al. [51] to dynamically plan the path and complete the task.

2.3. Target Detection & Tracking

Target detection is an ability to recognise and sense the target. Once the position is known, then target needs to be tracked. Cyril and Simon [11] classified the target management problems in detail. The authors focussed on the two separate problems i.e. target detection and target tracking. Lin [52] proposed a framework for coordination of multi-robots. The approach used here is leader, follower and region based. The follower follows the leader and surrounds the leader in designated region to stay free which reduces redundancy. Authors in [53] applied a vision system in mobile robots to track the path and signs on the road.

2.4. Coalition Formation

Coalition Formation is based on forming a coalition of robots. He et al. [54] proposed tracking algorithm and artificial potential field. The authors used virtual force for robots based on distance and velocity. Task allocation becomes issue when a coalition is formed by the robots. Jose et al. proposed a framework that led to solve the problem of interference that is when two robots want to access the same point. The model used is support vector regression along with double round auction [54]. Zhang et al. proposed a strategy of leader follower using distributed Kalman filter [55]. The path of robots involved in pursuing is not fixed. Therefore, the formation changes accordingly based on decentralized partially observable Markov decision processes. Sun et al. [56] proposed model fixed the problem of position assignment and traditional artificial force model by using formation control method. This strategy reduced the time taken to reach the target. Yingbai Hu et al. [57] used hand eye coordinated system to implement number of tasks like tracking, recognizing and grasping of an object. The author used motion fusion approach improved the response time.

2.5. Strategies for Hunting

There are various strategies that can be used for hunting. Chen et al. [58] analyzed that one of the way can be using angle made by robot at first go on the basis of the evader called as objective robot in the paper. This strategy reduced the hunting time. An et al. [59] proposed a hunting strategy using partition global search strategy, which depends on multi-robotic fish's group size and different conditions of search target. Xialong et al. [60] proposed a control strategy named Multi-Input Multi-Output (MIMO) system. The complexity of the system is reflected by its strong coupling. Sariel et al. [1] focused on mine countermeasure missions on the sea floor. The mission was to find and seize the stockpiles before their deployments. The authors here did not focus on combining coverage and detection strategies together.

Hongwen Ma et al. [3] proposed a distributed adaptive robust protocol. With the use of Lyapunov - Krasovskii functional and Young's inequality, the time delays are eliminated in multi agent systems making a robust system.

2.6. Contribution

Our contribution in this paper is as follows:

a) We propose a Corner Dragging Algorithm. Corner dragging algorithm helps in dragging an evader to any one of the four corners. This includes defining proper coordination between two robots and to drag an evader in the corner before robots catch it.

b) Simulation shows superiority of corner dragging algorithm over dynamic alliance and formation construction algorithm [20] in terms of performance of hardware resources used, time and related communication overhead on each robot.

3. PROBLEM STATEMENT

In this paper, we study the problem of multi-robot hunting where the environment in which the robots catch the evader is unknown, which means that the multi-robots neither have any information about the evader nor the environment. This problem reflects a real time scenario where no prior information exists with the robots. The only information that the multiple robots have is the information about the area of the environment where they have to search and catch the evaders.

Here, the robots are labeled as $r_{i\,,} i = 1,2,..,n$, which belongs to a robot team Ω. Each robot is an Omni-directional robot with 360° visual capability. The robots can also communicate with each other; recognize each other and the evaders too.

The evaders are labeled as $e_j j = 1,2,\dots,m$. Each evader is also an omni-directional evader with 360° visual capability. The evaders move randomly when it does not see any robot or obstacle at first. As the evaders see any robot it tries to escape and avoid the obstacles.

As discussed in [20], the hunting task begins by a robot team Ω. The task can be denoted by $T = \{N_e, N_r, A_s\}$, where N_e is the number of evaders, N_r is the number of robots to catch the evader and A_s is the area of searching space. The robot now begins to search for any potential evader. Any robot that first sees the evader becomes the temporary commander C_t and broadcasts the position (P) of evader to all other robots of the team. The commander creates a unique ID to identify the robots which broadcasted to other robots of the team. The robots in the team will pursue the evader i.e. the robots will start moving towards the evader. If the condition to catch the evader is satisfied, that is if the distances between robots and evader is such that the evader cannot escape, the robots then are in catch stage and the evader is caught. This ends with the hunting task of a team.

4. PROPOSED APPROACH

The real time hunting task has two major issues that need to be solved. The first one is to efficiently hunt the evaders using cooperative behavior. The second one is to reduce the communication burden while hunting [20]. To achieve proper hunting task by multi robots, the robots need to be well coordinated as the evader is intelligent enough to escape.

In the proposed approach we have focused on reducing the communication burden on the master robot. The proposed approach uses the bio- inspired algorithm, Corner Dragging Algorithm with which the numbers of robots are reduced to two as compared to four in [20]. The alliance is formed so that the evaders are dragged towards the corner. In the modified algorithm, there is an improvement in the total number of resources used, time required to catch the evader and the communication burden on master robot shown in Table 1. We would now like to define some flags that are used to describe the proposed approach.

4.1. Flags

Flags are used to know the status of each robot while hunting. In general, there are four stages, such as searching, pursuing, catching and broken [20]. This flag is represented using $f_1(r_i)$, where r_i is robot. Likewise we have also used the flag for each evader e_j as$f_2(e_{\,j})$. This flag shows the status of the evader as being unknown, known or caught.

$$f_1(r_i) = \begin{cases} 0, & brokenstage \\ 1, & searchingstage \\ 2, & pusuingstage \\ 3, & catchingstage \end{cases} \qquad f_2(e_j) = \begin{cases} 0, & unknown \\ 1, & known \\ 2, & caught \\ \end{cases}$$

Table 1. Pseudo-Code of Corner Dragging Algorithm

Step	Algorithm
1	Initialize the status flags of all the robots f_(r_i)=1;
2	The robot that finds the evader firsts becomes r_1= master, f_{r_i}=2
3	Calculate $r_i = \sqrt{(x_{r_i} - C_{1_{jk}})^2 + (y_{r_i} - C_{2_{jk}})^2}$
	$C_{1_{jk}} = \pm X$ and $C_{2_{jk}} = \pm Y$ for all j = {1, 2} and k= {1, 2}.
	% The corner that is nearest to robot is selected. It is represented by $corn(x, y)$. %
4	Selected corner, $corn = \min(r_i)$
5	If $corn_x = -$X (x coordinate of nearest corner), then,
6	$k_1 = +ve$
7	Else $k_1 = -ve$
8	End
9	Calculate $dist\left(p_{r_1}, p_{e_j}\right)$
	% the distances of robot r_1 form evader is calculated, in which the evader is dragged towards the corner in the direction of x axis.
	If $corn_y = -$Y then
	$k_2 = +ve$
	Else $k_2 = -ve$
	End
	Calculate

4.2. Strategy Of Dynamic Alliance

Hunting task begins with the searching process. Both the evaders and the robots move randomly. As soon as the robot sees the evader, that robot becomes the temporary commander or master. This master knows the position of the evader and therefore it can calculate the distance of each robot with respect to the evader. The master robot requests the status flag and position of the other robots. Then the pursue state begins. The distance from evader to robot is calculated in such a way that the evader is dragged towards the corner and can be caught only by the two robots one is the master and the other is the follower.

The master robot also needs to find the nearest corner from it and then drag the evader towards that corner. This helps in saving the energy and time needed to catch the evader.

The explanation of the complete hunting task is given below: Assuming the field of interest is a square platform. The robot r_1 is the master and r_2 is the follower. Let |X| and |Y| are highest values of co-ordinates of the field of interest. Then the distance of robot from all four corners can be calculated by the following equation

$$r_i = \sqrt{(x_{r_i} - C_{1_{jk}})^2 + (y_{r_i} - C_{2_{jk}})^2} \tag{1}$$

Out of these four distances, the corner that is nearest to robot has to be selected. It is represented by $corn\ (x, y)$. Therefore, selected corner, $corn = \min(r_i)$. The distances of robot r_1 from evader is calculated by the following equation, in which the evader is dragged towards the corner in the direction of x axis.

$$dist\left(p_{r_1}, p_{e_j}\right) = \sqrt{(x_{r_1} - x_{e_j} \pm k_1)^2 + (y_{r_1} - y_{e_j})^2} \tag{2}$$

The distances of robot r_2 from evader is calculated by the following equation, in which the evader is dragged towards the corner in the direction of y axis.

$$dist\left(p_{r_2}, p_{e_j}\right) = \sqrt{(x_{r_2} - x_{e_j})^2 + (y_{r_2} - y_{e_j} \pm k_2)^2} \tag{3}$$

Once the master robot knows the position and corner where the evader has to be caught, the pursue stage begins. If the flag value of the robot is 1, i.e. the status of the robot is in searching stage. At the pursue stage, the robots coordinate with each other and move towards the evader. The evader is caught in the corner using corner dragging algorithm.

4.3. Coordination/Cooperation Dragging Strategy

We adopt a simple and novel algorithm to catch the evader by dragging the evader towards the border so that the chance for evader to escape decreases. There are only two robots in a team who works in a cooperative and well-coordinated form in order to catch the evader. So, the robot who is temporary commander broadcasts the message and the other robot receives. Therefore, the communication burden on master robot is reduced to 66.67%. The communication is between two robots only where one is the temporary commander and the other is the slave whereas in [20], the temporary commander communicates with other three robots.

4.4. Strategy of Formation

After the pursue stage is over the catch stage begins. In the catch stage the robots come closer to evader and then it becomes impossible for the evader to escape as the evader is covered from all sides. Here the hunting tasks end if all the evaders in the field of interest are caught.

5. SIMULATION AND RESULTS

Performance evaluation of the proposed corner dragging algorithm is simulated in MATLAB in SimIam robot simulator. In this simulation a team of robots are required to hunt evaders within a given environment.

The robots are represented using $\Omega = \{r_{i\,,} i = 1,2,..,n\}$ and the evaders are represented using $e_{j\,,}\ j = 1,2, \ldots, m$. We have considered several practical cases such as static vs dynamic environments, robots at different locations and gathered relevant results. The parameters used in these simulations are shown in Table 2. When conducting these simulations, we have considered the following assumptions: 1) Assumption 1: The velocity of the evader is less than that of robot. If the evader increases the speed after he sees the robot, the robot will also increase the speed more than evaders speed. So, always the speed of robot is more than that of evader. 2) Assumption 2: The total hunting time is measured in seconds and not in steps. The step length is a relative value. It accounts for the steps of all robots in a team and not just one. The parameters and assumptions used throughout these simulations are the same.

Table 2. Hunting Task Parameters

Parameters	Value	Remarks
R_s	2	The range of onboard sensors (m)
N_c	2	Number of robots to catch one evader
v_r	0.5	Velocity of robot (m/s)
v_e	0.3	Velocity of evader (m/s)

5.1. Static Environment with one Evader

This is the first simulation that we conducted using the proposed CDA. In this simulation experiment, we have considered a case of one evader caught by only two robots in the given environment along with some obstacles. The robots drag the evader towards the nearest corner and catch it. The area is 20 x 20 (m^2). As reported in [20], we have replicated the worst case scenario in simulation by positioning the evader in the top center of the environment i.e. (10, 20). The two robots are positioned at the two corners of the environment i.e. (0, 0) and (20, 0). Figure 2 shows the locations of the robots, evaders and obstacles. The Khepera robot with red color is the evader and those with the black are the robots that catch the evaders. There are four obstacles in the field of interest. At the search stage all of them move randomly and during the pursue stage the robots drag the evader to corner and finally catch the evader.

Figure 1(a) shows the initial position of robots. Figure 1(b) shows the initial positions of robots and evader in the simulator. Figure 1(c) shows trajectory while robots are catching the evader. Figure 1(d) shows the final positions of the robot and evader in the hunting process. Figure 1(e) shows the graph of final trajectories of the robots and the evader when the evader is caught. We have randomly used the angles of robots to reflect the worst case scenario and hence not adopted the approach as used in [20].

The results from Figure 1 demonstrate that the evader has certain degree of intelligence because when it finds an obstacle or a robot it changes the direction rapidly. However, even with this level of intelligence the robots can finally capture the evader efficiently. The results show that the pursuit stage begins at 25th second and takes 34 seconds. The catch stage begins at the 35th second and takes 5 seconds. The cooperation between the two robots can be seen in Figure 1(c), (d) and (e).

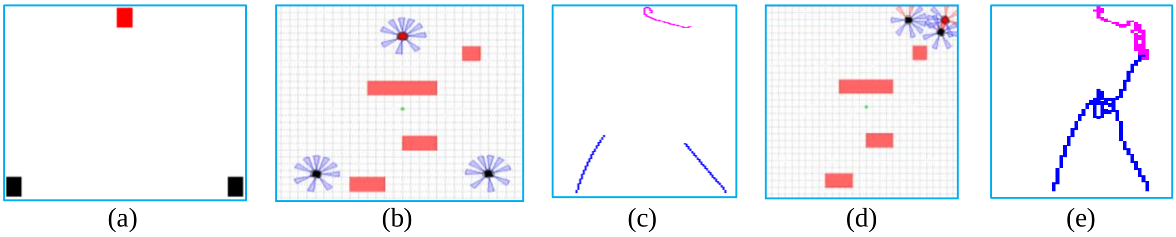

Figure 1. (a) Initial position of robots and the evader; (b) Initial positions of robots and evader in the simulator. The robot with a red dot is the evader and the robots with a black dot are the ones catching the evader; (c) Trajectory while robots are catching the evader; (d) Final positions in simulator where evader is caught; (e) Final trajectories where robots catch evader

5.2. Multiple Evader

This simulation deals with the hunting experiment for multiple evaders. In this experiment we have considered a case of two evaders caught by four robots in the given environment along with some obstacles. The two robots form a team to catch one evader and the other team of robots catches the other evader. The nearest robots from the evaders form the teams. The robots drag the evader at corner and catch the evader. The area of hunting is same as used in [20] as shown in Figure 2.

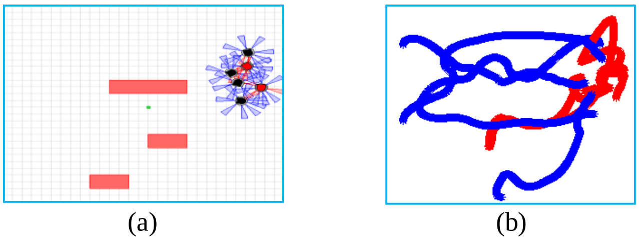

Figure 2. (a) Final positions in simulator where evaders are caught. The robots with the red dots are the evaders and the robots with a black dot are the ones catching the evader; 2(b) Final trajectories where robots catch evaders

5.3. Robot Failure

This simulation helps to test the robustness aspect in multi robot system. This proposed approach simulates when some robots fail. The reason for failure of robots can be physical failure of sensors or power system or it could be communication breakdown. There are two stages where robot failure might occur.

One is maturity stage and other is at infant stage. Maturity stage is where the robots are in a pursuing or catching stage and infant stage is where the robots are in searching stage. We have considered the case where the robot fails at the maturity stage because the robots here cooperate and are in a team. We then catch the evader with remaining robots shown in Figure 3. The robot in the bottom right corner failed at 12.19 seconds, after which another robot joined the team to complete the hunting task.

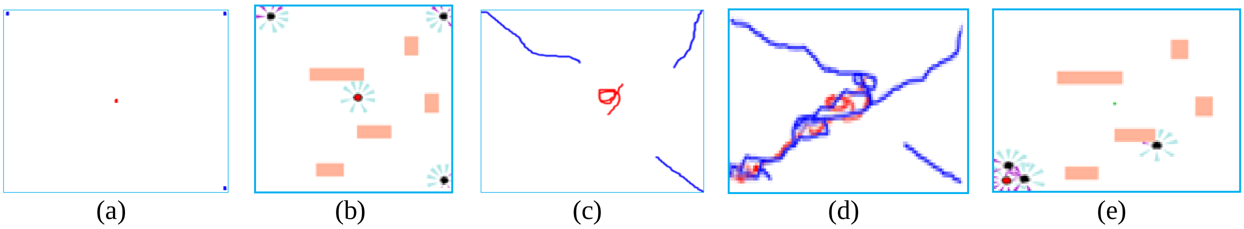

Figure 3. (a) Initial position of robots and evader; (b) Initial position of robots and evader in simulator. The robot with a red dot is the evader and the robots with a black dot are the ones catching the evader; (c) Trajectory while catching evader; (d) Final trajectories of robots and evader in simulator; (e) Final position of robots and evader in simulator

5.4. Obstacles with different shapes

To further test the performance of the corner dragging algorithm, we simulated an experiment where the environment had different shapes of obstacles as shown in Figure 4. Rest the environment is same with two robots and 1 evader. The robots could successfully avoid the obstacles of different shapes and drag the evader and catch it in corner. Figure 4(b) and 4(c). Figure 4(a) and 4(d) shows the initial trajectory and final trajectory of the robots.

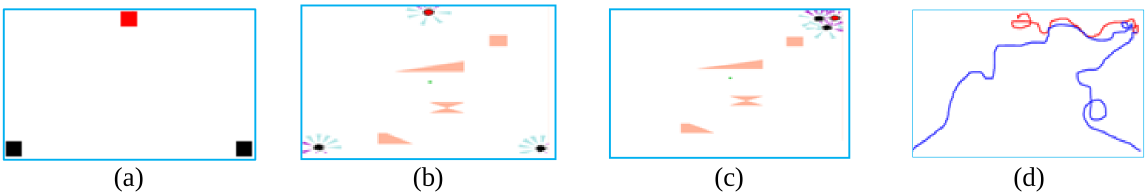
(a) (b) (c) (d)

Figure 4. (a) Initial position of robots & evader; (b & c) Initial & final position of robots &evader in simulator; (d) Final trajectories of robots & evader

6. RESULTS ANALYSIS AND DISCUSSION

In this section we discuss the results that were observed from different simulation described in Section 5.

6.1. Static Environment with one Evader

With the use of corner dragging algorithm, there is improvement in the result when compared with the previous work done in [20]. The numbers of resources are reduced so there is very less communication between robots. As the master robot interacts with only other robot, the overhead reduces significantly. Also as the evader sees fixed boundary from two sides, the evader is caught easily. But in case if the evader is surrounded by four robots, the evader tries to escape, which in turn takes more time to catch. The comparative results are shown in the Table 3, 4 and 5. The hardware cost is reduced by 50%.

Table 3. Hardware Resources Used

Algorithm	Robot	Evader
Dynamic Alliance and Strategy of Formation [20]	4	1
Corner Dragging Algorithm	2	1

Table 4. Time Needed in Hunting Task

Algorithm	Robot
Dynamic Alliance and strategy of formation [20]	75 seconds
Corner Dragging Algorithm	38 seconds

Table 5. Communication on Robot

Algorithm	Master Robot	Slave Evader
Dynamic Alliance and strategy of formation [20]	One to three communication	One to one communication
Corner Dragging Algorithm	One to one communication	One to one communication

6.2. Robot Failure

In order to test the robustness of CDA, the experiments are simulated. This proposed approach simulates when some robots fail in maturity state. The team of robot is formed and the robots are pursuing the evader. At this point of time if robots fails, the hunting process should not suffer. The robots can successfully catch the evader with remaining number of robots using corner dragging algorithm. As the number of resources are reduced in corner dragging algorithm, the communication overhead on master robot is also reduced.

6.3. Obstacles with Different Shapes

In this scenario of obstacles with different shapes, the corner dragging algorithm is performing well. The experiment is conducted while having different shapes of obstacles like square, triangle and polygon. These shapes are used in the environment so as to reflect the real world objects. In a case of any real life hunting problem, the robots may come across any type of hurdles on the way. This can be one of those which we considered. Tables 6 and 7 show the results in comparison to [20].

Table 6. Hardware Resources Used

Algorithm	Robot	Evader
Dynamic Alliance and Strategy of Formation [20]	4	1
Corner Dragging Algorithm	2	1

Table 7. Time Needed in Hunting Task

Algorithm	Time to find	Time to pursue	Time to catch
Dynamic Alliance and Strategy of Formation [20]	18	34	5
Corner Dragging Algorithm	13.259 sec	4.211	2.5 sec

7. CONCLUSION AND FUTURE SCOPE

In this paper we studied a classical hunting problem using multi-robots and developed a modified bio - inspired algorithm known as Corner Dragging Algorithm for the hunting task with one evader and two robots with different positions of robots, evader and obstacles. In addition, we performed simulations on multiple evaders being hunted and robots' failure as well. Simulation experiments have shown an improvement in results in terms of reduced hardware resources, communication overhead and time needed for hunting. One possible extension of our work is to work on team of multiple robots and evaders and to apply it in different applications. Furthermore, a robustness analysis can also be conducted in future with multiple evaders and robots.

ACKNOWLEDGEMENT

The authors would like to thank the anonymous reviewers and the Associate Editor for their valuable feedback.

REFERENCES

[1] S. Sariel, et al., "Naval mine countermeasure missions," *IEEERobot and Automation Magazine*, vol. 15, pp. 45-52 2008.

[2] H. Yamaguchi, "A cooperative hunting behavior by mobile robot troops," *International Journal on Robotics Research*, vol. 18, pp. 931-940, Sep 1999.

[3] H. Ma, et al., "Neural-Network-Based Distributed Adaptive Robust Control for a Class of Nonlinear Multiagent Systems with Time Delays and External Noises," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 46, pp. 750-758, 2015.

[4] R. Lv, et al., "A multi-AUV hunting algorithm with ocean current effect," *IEEE International Conference on Cyber Technology in Automation, Control, and Intelligent Systems (CYBER)*, pp. 869-874, 2015.

[5] Z. Li, et al., "Guest Editorial an Overview of Biomedical Robotics and Bio-Mechatronics Systems and Applications," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 46, pp. 869-874, Jun 2016.

[6] A. Farinelli, et al., "Multirobot systems: a classification focused on coordination," *IEEE Transaction on System Man and Cybernetics*, vol. 34, pp. 2015-2028, 2004.

[7] J. Saunders, et al., "Teach Me–Show Me—End-User Personalization of a Smart Home and Companion Robot," *IEEE Transactions on Human-Machine Systems*, vol. 46, pp. 27-40, 2015.

[8] H. Modares, et al., "Optimized Assistive Human–Robot Interaction Using Reinforcement Learning," *IEEE Transactions on Cybernetics*, vol. 46, pp. 655-667, 2016.

[9] P. Bhattacharajee, et al., "Multi-robot path-planning using artificial bee colony optimization algorithm," *IEEE Conference on Nature and biologically inspired computing*, pp. 219-224, 2011.

[10] P. Agrawal and H. Agrawal., "A Review on Multi Robot Cooperation using Bio Inspired Neural Networks," *International Journal on Soft Computing Mathematicsand Control (IJSCMC)*, vol. 2, pp. 15-24, Nov 2013.

[11] C. Robin and S. Lacroix, "Taxonomy on Multi-robot Target Detection and Tracking," *Multi-Agent Coordination in Robotic*, 2015.

[12] Z. Wang, et al., "A self–organizing cooperative hunting by robotic swarm based on particle swarm optimization localization," *International Journal of Bio- Inspired Computation*, vol 5, pp. 68-73, 2015.

[13] V. Odakura and A. H. R. Costa, "Cooperative multi-robot localization: using communication to reduce localization error," *in ICINCO*, pp. 88-93, 2005.

[14] R. Reeve and J. Hallam, "An analysis of neural models for walking control," *IEEE Transactions on Neural Networks*, vol. 16, pp. 733-742, 2005.

[15] GyuhoEoh, et al., "Multi-robot cooperative formation for overweight object transportation System Integration," *IEEE/SICE International Symposium*, 2011.

[16] X. S. Yang, "Firefly algorithms for multimodal optimization," *in Stochastic algorithms: foundations and applications*, Springer, pp. 169-178, 2009.

[17] X. S. Yang, "A new metaheuristic bat-inspired algorithm," in *Nature inspired cooperative strategies for optimization (NICSO)*, Springer, pp. 65-74, 2010.

[18] S. X. Yang and M. Q. Meng, "Real-time collision-free motion planning of a mobile robot using a neural dynamics-based approach," *IEEE Transactions on Neural Networks*, vol. 14, pp. 1541-1552, 2003.

[19] R. E. Montemerlo, et al., "Game theoretic control for robot teams," *Proceedings of the IEEE International Conference onRobotics and Automation (ICRA)*, pp. 1163-1169, 2005.

[20] J. Ni and S. X. Yang, "Bioinspired neural network for real-time cooperative hunting by multirobots in unknown environments," *IEEE Trasaction on Neural Networks*, vol. 22, pp. 2062-2077, 2011.

[21] D. P. Barnes and J. O. Gray, "Behaviour synthesis for co-operant mobile robot control," *International Conference on Control '91*, vol. 2, pp. 1135-1140, 1991.

[22] R. A. Brooks, "Interaction and intelligent behavior," Massachusetts Institute of Technology, ACM Digital Library, 1994.

[23] L. Sheremetov, et al., "Design of fuzzy knowledge based systems with multi-set confidence models," *IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pp. 290-295, 2011.

[24] S. B. Saad, et al., "Hybrid Coordination Strategy of a Group of Cooperating Autonomous Underwater Vehicles," IFAC-Papers Online – Elsevier, 2015.

[25] H. D. Patiño, et al., "Neural networks for advanced control of robot manipulators," *IEEE Transactions on Neural Networks*, vol. 13, pp. 343-354, 2002.

[26] W. He, et al., "Adaptive Neural Impedance Control of a Robotic Manipulator with Input Saturation," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 46, pp. 334-344, 2016.

[27] M. Xu, et al., "Moving target pursuit algorithm using improved tracking strategy," *IEEE Transactions on Computational Intelligence and AI in Games*, vol. 2, pp. 27-39, 2010.

[28] R. R. Brooks, et al., "Game and information theory analysis of electronic countermeasures in pursuit-evasion games," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 38, pp. 1281-1294, 2008.

[29] D. Zhu, et al., "Multi-AUV Hunting Algorithm Based on Bio-inspired Neural Network," in Unknown Environments, 2015.

[30] Y. Ma, et al., "A multi robot coordinated hunting strategy with dynamic alliance," *in Proc. Chinese Control and Decision Conference, Guilin, China*, pp. 2338-2342, Jun 2009.

[31] P. Kachroo, et al., "Dynamic programming solution for a class of pursuit evasion problems: The herding problem," *IEEE Transaction on Systems Man and Cybernetics*, vol. 31, pp. 35-41, Feb 2001.

[32] G. N. Yannakakis, et al., "Emerging cooperation with minimal effort: Rewarding over mimicking," *IEEE Transactions on Evolutionary Computing*, vol. 11, pp. 382-396, Jun 2007.

[33] X. Cao, et al., "AUV cooperative hunting algorithm based on bio-inspired neural network for path conflict state," *IEEE Conference on Information and Automation*, pp. 1821-1826, 2015.

[34] G. Huang, et al., "Semi-Supervised and Unsupervised Extreme Learning Machines," *IEEE Transactions on Cybernetics*, vol. 44, pp. 2405-2417, 2014.

[35] Z. Wang, et al., "A self–organising cooperative hunting by robotic swarm based on particle swarm optimisationlocalisation," *International Journal of Bio-Inspired Computation*, vol. 7, pp. 68-73, 2015.

[36] H. Zhang, et al., "Hunting in Unknown Environments with Dynamic Deforming Obstacles by Swarm Robots," *International Journal of Control and Automation*, vol. 8, pp. 385-406, 2015.

[37] A. Kolling, et al., 'Human Interaction with Robot Swarms: A Survey," vol. 46, pp. 9-26, 2015.

[38] T. Y. Huang, et al., "A self-organizing cooperative hunting by swarm robotic systems based on loose-preference rule," *Acta Automatica Sinica*, pp. 57-68, 2013.

[39] P. Yang, et al., "Swarm Robots hunting behavior based on particle swarm optimization," *Applied Mechanics and Materials*, vol. 328, pp. 187-192, 2013.

[40] M. K. Nighot, et al., "Multi-robot hunting based on swarm intelligence," *International Conference on Hybrid Intelligent Systems (HIS)*, pp. 203-206, 2012.

[41] J. Li, et al., "Coordinated multi-robot target hunting based on extended cooperative game," *Information and Automation*, pp. 216-221, 2015.

[42] K. Raghuwaiya, et al., "Cooperative Control of Multi-robot Systems with a Low-Degree Formation," *Advanced Computer and Communication engineering technology*, vol. 362, pp. 233-249.

[43] C. Wang, et al., "A new approach of multi-robot cooperative pursuit," *Chinese Control Conference (CCC)*, pp. 7252-7256, 2013.

[44] S. Hunt, et al., "A consensus-based grouping algorithm for multi-agent cooperative task allocation with complex requirements," *Cognitive computation*, vol. 6, pp. 338-350, 2014.

[45] Q. Yuan, et al., “Multi-robot task allocation using CNP combines with neural network,” *Neural Computing and Applications*, vol. 23, pp. 1909-1914, Dec 2013.

[46] S. J. Hunt, “Task allocation and consensus with groups of cooperating Unmanned Aerial Vehicles,” PhD thesis, 2014.

[47] X. Cao and D. Zhu, “Multi-AUV Task Assignment and Path Planning with Ocean Current Based on Biological Inspired Self-organizing Map and Velocity Synthesis Algorithm,” *Intelligent Automation & Soft Computing*, 2015.

[48] X. Cao and D. Zhu, “Multi-AUV underwater cooperative search algorithm based on biological inspired neuro dynamics model and velocity synthesis,” *Journal of Navigation*, vol. 68, pp. 1075-1087, Nov 2015.

[49] G. A. Kaminka, et al., “Adaptive multi-robot coordination: A game-theoretic perspective,” *IEEE International Conference in Robotics and Automation (ICRA)*, pp. 328-334, 2010.

[50] D. Wicke, et al., “Bounty hunters and multiagent task allocation,” *International Conference on Autonomous agents and multiagent system*, pp. 387-394, 2015.

[51] X. Yi, et al., “A bio-inspired approach to task assignment of multi-robots,” *IEEE symposium on Swarm Intelligence (SIS)*, pp. 1-5, 2014.

[52] C. W. Lin, et al., “Hierarchical Coordination for Multi-Robot Systems with Region-Based Tracking Control,” *International Journal of Automation and smart technology*, vol. 5, 2015.

[53] H. N. M. Shah, et al., “Sign Detection Vision Based Mobile Robot Platform,” *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 7.

[54] L. L. He and X. C. Lou, “A Multi-Input Feedback Control Algorithm for Formation Control,” *Journal of Theoretical and Applied Information Technology*, vol. 48, 2013.

[55] W. Zhang, et al., “Multi-agent pursuit with decision-making and formation control,” *Chinese Control Conference (CCC)*, pp. 7016-7022, 2013.

[56] X. Sun, et al., “Multi-agent formation control based on artificial force with exponential for,” *Intelligent Control and Automation*, pp. 3128-3133, 2014.

[57] Y. Hu, et al., “Development of Sensory-Motor Fusion-Based Manipulation and Grasping Control for a Robotic Hand-Eye System,” *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, pp. 1-12, May 2016.

[58] Y. Z. Chen, et al., “Hunting Strategy for Multi-mobile Robots System Based on Angle First,” *Journal of Beijing University of Technology*, May 2012.

[59] Y. An, et al., “Multiple Robotic Fish's Target Search and Cooperative Hunting Strategies,” *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 12, 2014.

[60] X. Li, et al., “A multi-input multi-output control strategy for intelligent nonholonomic robots,” *Control and Decision Conference (CCDC)*, pp. 4698-4703, 2015.