Soft Computing Letters 2 (2020) 100006

Contents lists available at ScienceDirect

Soft Computing Letters

journal homepage: www.elsevier.com/locate/socl

Application of Swarm Robotic System in a Dynamic Environment using Cohort Intelligence

Priya Bannur[a,*], Purvi Gujarathi[a], Karthik Jain[a], Anand J Kulkarni[b]

[a] School of Computer Engineering & Technology, MIT World Peace University, Pune, 411 038, India
[b] Symbiosis Center for Research and Innovation, Symbiosis International (Deemed University), Pune, 412 115, India

ARTICLE INFO

Keywords:
Cohort Intelligence
Search and Rescue
Swarm Robotics
Metaheuristics
Artificial Intelligence

ABSTRACT

The nature inspired Swarm Intelligence has laid the foundation for many eclectic applications. This work considers a solution to one such application of Search and Rescue operation, based on Cohort Intelligence (CI) methodology which aims at modelling the behaviour of candidates based on the interaction amongst each other to achieve a common goal. Every candidate improves its own behaviour by observing all other candidates in the cohort. This method results in the refinement of the performance of the entire system. The research done so far in this application using CI is associated with robots deployed in a static alien establishment. However, this assumption is not suitable in real-life scenarios. In this paper, the obstacle avoidance and path planning of a swarm of robots was implemented while considering a dynamic alien establishment. The problem of robots occasionally getting stuck in the non-convex obstacles has also been solved using a perturbation technique. The following test cases were implemented - No Obstacle Case (NOC), Stationary Obstacles Case (SOC), Single Dynamic Obstacle Case (SDOC), Multiple Dynamic Obstacles Case with Same Velocity (MDOC-SV) and Multiple Dynamic Obstacles Case with Different Velocities (MDOC-DV).

1. Introduction

Artificial Intelligence (AI) has been instrumental in almost every sector of human life. Some of the AI driven and benefited fields are Agriculture [8], Transportation [1], Healthcare [10], Manufacturing [14], Military Guidance and Surveillance [17], and many more. Importantly, Robotics is the most significantly affected field so far. For example, Assistive Intelligent Robotic Wheelchairs [9], Collaborative Smart Drones [3], Marine Environment Monitoring [7], Autonomous Robotic Surgery [2], Assistive Swarm Robots for Fire-fighters [13] and Robot-assisted Urban Search and Rescue [4] are few of the examples of intelligent robotic systems.

This paper considers an application of a robotic system in search and rescue operations. Path planning and obstacle avoidance is a major concern in this domain. Some of the remarkable research done in this field includes a velocity-based motion planning technique where sensors were used to avoid collisions with the obstacles [16], a new algorithm based on the Simultaneous Replanning concept was introduced by Biswas et al. [5], a Fuzzy Ant Colony Optimization methodology was proposed [18] that uses ultrasonic transducers for obstacle detection and a swarm robotic method for dynamic obstacle avoidance called Self-organizing migrating algorithm was developed by Diep et al. [6].

Recently, an application inspired by the role of AI-based robots in the rescue operations was implemented in Self-organizing Multi-Agent Cooperative Robotic System [15] using the Cohort Intelligence (CI) methodology developed by Kulkarni et al. [11]. This implementation considered robots deployed in an alien establishment with obstacle(s). The robots in the arena try to reach the target by interacting and following the behavior of other robots in order to improve their own behavior and eventually of the entire cohort. This alien establishment was considered to be of static nature. Whenever the robot encountered any obstacle it moved along the boundary of the obstacle. All the obstacles were of convex nature. However, these assumptions are not close to real-life scenarios. This paper proposes a more realistic approach to this application using the same CI methodology. The path planning and obstacle avoidance of a swarm of robots were implemented while considering a dynamic environment. The obstacles in the environment can move in a random direction with a certain velocity. The robots do not collide with the obstacle(s), rather they maintain a safe distance with the obstacle(s) while moving in the arena. The problem of robots getting stuck in the nonconvex obstacles has also been resolved using a perturbation technique. Furthermore, this application was implemented with two variations in the context of candidate following one another, the roulette wheel approach where candidate to be followed is selected based on roulette wheel selection methodology and the follow median approach

[*] Corresponding author.
E-mail addresses: priyabannurs@gmail.com (P. Bannur), anand.kulkarni@sitpune.edu.in (A.J. Kulkarni).

https://doi.org/10.1016/j.socl.2020.100006
Received 11 April 2020; Received in revised form 3 July 2020; Accepted 30 July 2020

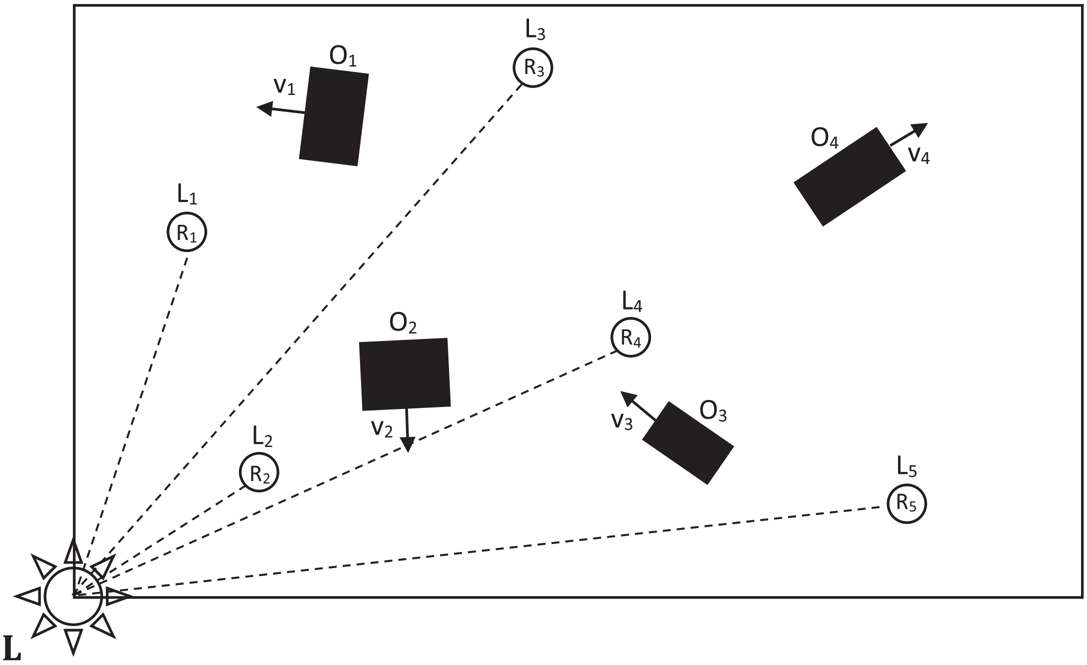


Fig. 1. A representation of the arena with robots and obstacles

where candidate to be followed is selected based on the median probability to reach the target. [12].

The remainder of this paper is structured as follows: Section 2 describes the mathematical formulation and experimental setup of the application. The validation of this implementation is presented in Section 3 by solving five test cases: No Obstacle Case (NOC), Stationary Obstacles Case (SOC), Single Dynamic Obstacle Case (SDOC), Multiple Dynamic Obstacles Case with Same Velocity (MDOC-SV) and Multiple Dynamic Obstacles Case with Different Velocities (MDOC-DV). The conclusion and a note on future directions are presented in Section 4 of this paper.

2. CI Framework for Swarm of Robots in Dynamic Environment

This endeavor is based on CI methodology which aims at modeling the behavior of the swarm of robots in a dynamic alien establishment. The nature of this approach is collaborative where all robots are self-adaptive and inter-dependent amongst one another. They cooperate with one another to achieve a common goal. A perturbation technique is devised to help the robots come out of the non-convex obstacle space. This resilience results in the refinement of the overall performance of the system.

Consider an arena of dimension $l \times b$ having robots and dynamic obstacles moving randomly (Fig. 1). The activity of robots, as well as the obstacles, is restricted within the limits of the arena. Assume the target of the system is a single light source $\boldsymbol{L}$. The goal of every robot R_i is to reach the light source $\boldsymbol{L}$ from their initial position without colliding with one another as well as with the obstacle(s). Every robot is assumed to have light and proximity sensors. Step length distance s is the distance with which every robot maneuvers in the arena in each iteration. The entire system is assumed to be on the same plane.

n	number of robots	
m	number of obstacles	
R_i	i^{th} robot	$i = 1, \dots, n$
L_i	light Intensity of i^{th} robot	$i = 1, \dots, n$
d_i	Euclidean distance of i^{th} robot from the light source	$i = 1, \dots, n$
p_i	probability of i^{th} robot	$i = 1, \dots, n$
O_j	j^{th} obstacle	$j = 1, \dots, m$
v_j	velocity of j^{th} obstacle	$j = 1, \dots, m$
s	step length distance	

The objective of this proposition is to maximize the light intensity sensed by each robot as follows,

$$Maximize \sum_{i=1}^{n} L_i \tag{1}$$

According to the inverse square law, L_i is inversely proportional to the square of the distance from the source,

$$L_i \propto 1/\left(d_i\right)^2 \tag{2}$$

The probability of selecting a robot to follow is calculated as follows,

$$p_i = L_i / \sum_{i=1}^{n} L_i \tag{3}$$

Initially, the configuration (position of robots as well as the obstacles, the velocity of obstacles v_i and the step length distance s) is random and the maximum number of iterations is k_{max}.

Step 1: The probability of each robot to reach the light source is calculated using eq. (3). In roulette wheel approach, every robot generates a random number ∈ [0, 1) that decides which robot to follow. Whereas, in follow median approach every candidate follows the candidate which has the median probability.

Step 2: New positions of the robots are generated step length distance away, in the direction of the robot to be followed. If any robot follows itself, the direction of movement is chosen at random. If any obstacle is encountered, the robot moves by keeping a safe distance from the obstacle.

Step 3: If any robot gets stuck in the non-convex space of obstacles for a significant number of iterations, then the robot is perturbed with some distance. An illustration of the same is shown in Fig. 2. Here, R_4 is perturbed to a new position R_4' as it was stuck in the non-convex space created by obstacles O_2 and O_3.

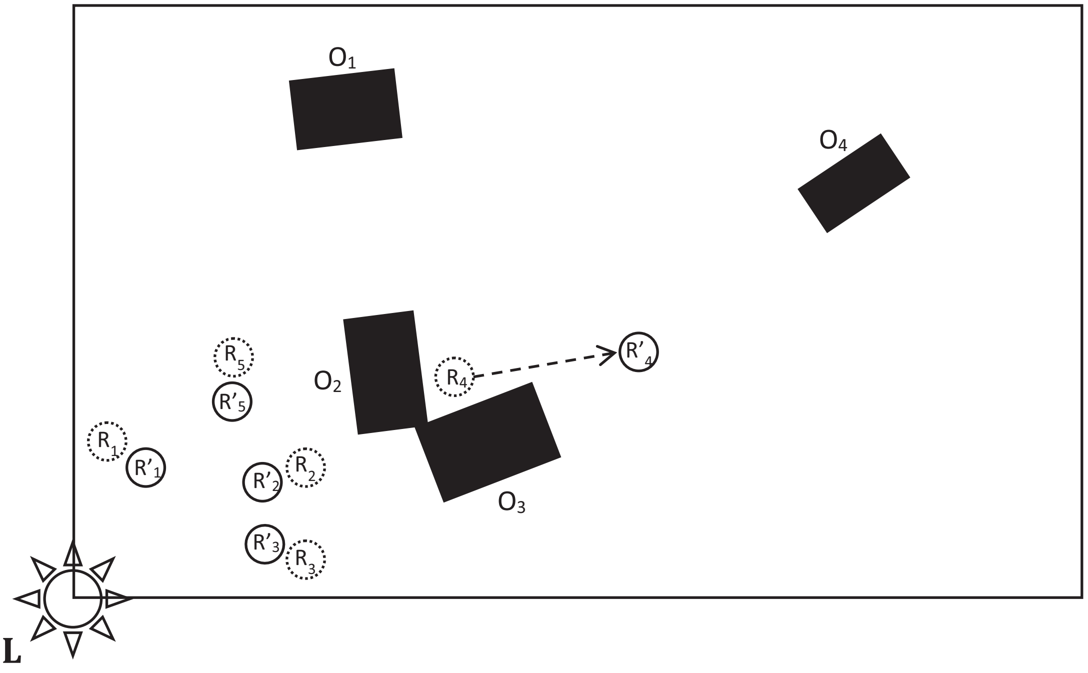


Fig. 2. An illustration of perturbation of robot

Step 4: The steps 1 to 3 are repeated until either the target is reached, or the maximum number of iterations is reached. The detailed flowchart of the procedure is presented in Fig. 3.

3. Numerical experiments and discussions

The algorithm was coded in Python 3.7 on Anaconda 4.8 platform and the simulations were run on Windows 10 operating system with Intel Core i5 2.5 GHz processor speed with 8 GB RAM. The following test cases were considered: No Obstacle Case (NOC), Stationary Obstacles Case (SOC), Single Dynamic Obstacle Case (SDOC), Multiple Dynamic Obstacles Case with Same Velocity (MDOC-SV) and Multiple Dynamic Obstacles Case with Different Velocities (MDOC-DV). For the roulette wheel approach as well as the follow median approach, every case was solved 30 times with 5 different initial configurations for each case. In every case, 5 robots were located randomly in a 5×5 arena. In addition, every robot maneuvers in the arena with step length (1 step = 0.14 units).

The NOC and SOC are illustrated in Fig. 4(a) and Fig. 4(b) respectively. The mean total time was observed to be significantly less than that of the same cases solved by Roychowdhury et al. [15]. Whenever a robot senses an obstacle in its path it diverts itself maintaining a safe distance from the obstacle without colliding with it. While simulating this case, a certain anomaly was encountered which restricted the movement of a robot due to nonconvex orientation of obstacles. To overcome this, a perturbation technique was devised which reconfigures its position. One such illustration is shown in Fig. 4(c). Similarly, the SDOC, MDOC-SV and MDOC-DV are illustrated in the Fig. 4(d), Fig. 4(e) and Fig. 4(f) respectively.

The results of execution of all five cases are presented in Table 1. The relative difference shown in Table 1 is used to gauge the performance of the robots in each case. It is the absolute difference of the mean of total traveled distance with respect to the mean of initial distance of the robots from the light source. As evident in Table 1, the execution time of both the approaches was comparable, however, the mean of total traveled distance is significantly less in the follow median approach than in the roulette wheel approach. This is due to the fact that the follow median approach avoids getting stuck in the local minima and simultaneously improves the solution, while in the roulette wheel selection approach there is some possibility that the robots may

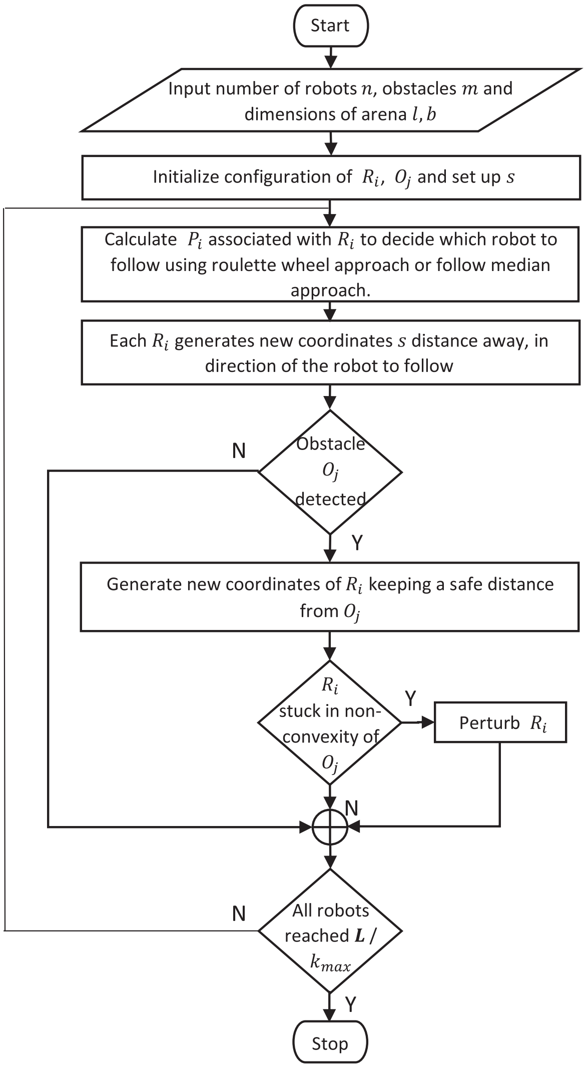
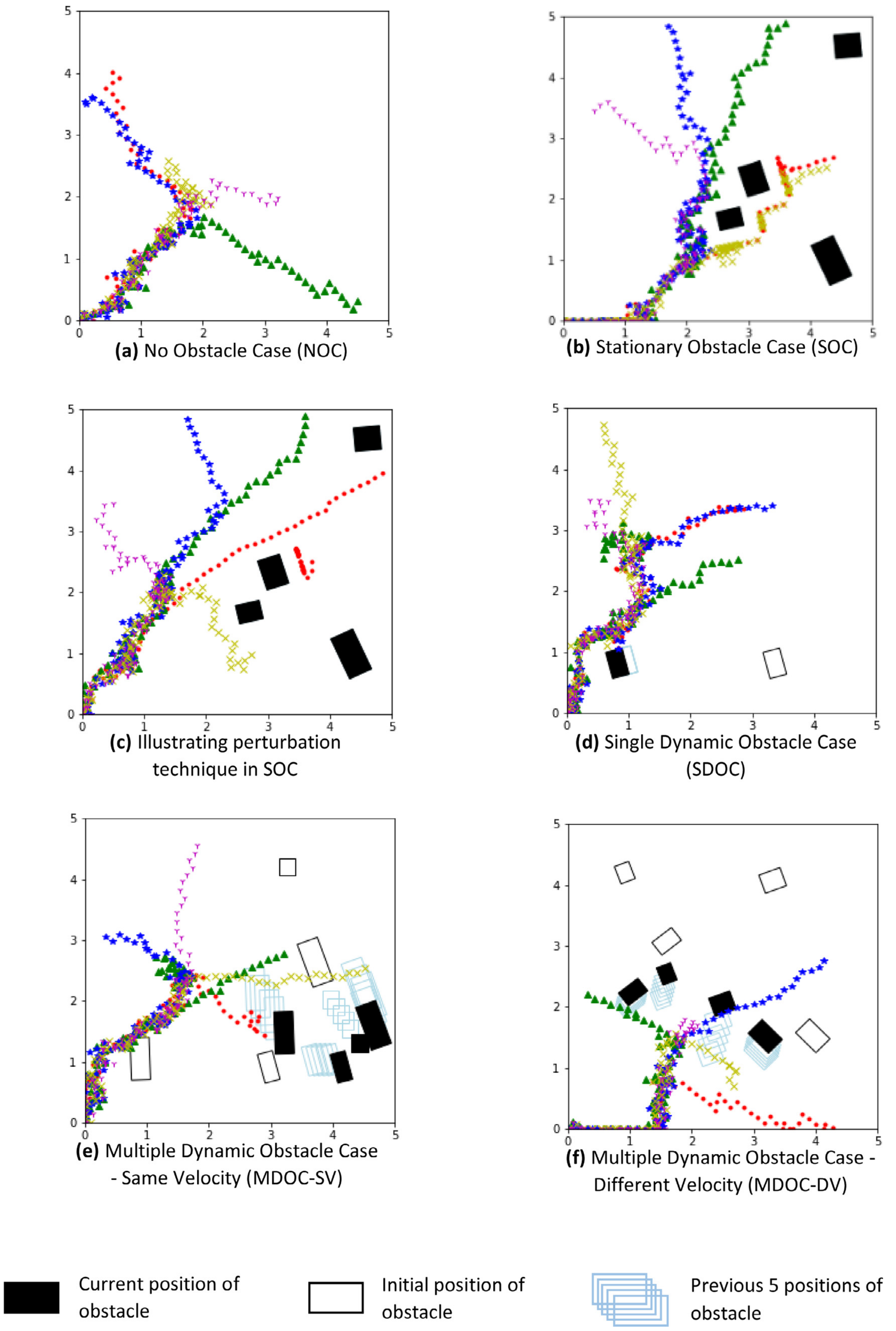

Start
Input number of robots n, obstacles m and dimensions of arena l, b
Initialize configuration of R_i, O_j and set up s
Calculate P_i associated with R_i to decide which robot to follow using roulette wheel approach or follow median approach.
Each R_i generates new coordinates s distance away, in direction of the robot to follow
N
Obstacle O_j detected
Y
Generate new coordinates of R_i keeping a safe distance from O_j
R_i stuck in non-convexity of O_j
Y
Perturb R_i
N
N
All robots reached $\boldsymbol{L}$ / k_{max}
Y
Stop

Fig. 3. Flowchart of the algorithm



(a) No Obstacle Case (NOC)

(b) Stationary Obstacle Case (SOC)

(c) Illustrating perturbation technique in SOC

(d) Single Dynamic Obstacle Case (SDOC)

(e) Multiple Dynamic Obstacle Case - Same Velocity (MDOC-SV)

(f) Multiple Dynamic Obstacle Case - Different Velocity (MDOC-DV)

Fig. 4. Graphical representation of various cases by roulette wheel approach

Table 1
Performance of CI for different cases.

			Roulette wheel approach					Follow median approach				
Cases	Config.	Mean of Initial Distance	Mean of Total Time (seconds)	Standard Deviation of Time (seconds)	Mean of Total Traveled Distance	Standard Deviation of Traveled Distance	Relative Difference of Traveled Distance	Mean of Total Time (seconds)	Standard Deviation of Time (seconds)	Mean of Total Traveled Distance	Standard Deviation of Traveled Distance	Relative Difference of Traveled Distance
NOC	1	3.42	0.02	0.01	11.58	2.30	2.39	0.02	0.00	5.94	0.00	0.74
	2	3.76	0.02	0.01	12.90	2.56	2.43	0.04	0.00	10.32	0.00	1.74
	3	3.10	0.01	0.00	6.10	1.28	0.97	0.05	0.00	12.02	0.00	2.88
	4	3.47	0.02	0.01	13.10	3.18	2.78	0.04	0.00	9.90	0.00	1.85
	5	3.74	0.02	0.01	7.64	0.93	1.04	0.06	0.00	13.29	0.00	2.55
SOC	1	3.43	0.03	0.01	10.13	2.08	1.95	0.02	0.00	5.09	0.00	0.48
	2	4.45	0.10	0.03	30.17	8.44	5.78	0.06	0.00	7.49	0.00	0.68
	3	4.41	0.05	0.02	16.49	4.19	2.74	0.03	0.00	6.51	0.00	0.48
	4	3.25	0.05	0.02	12.47	4.89	2.84	0.03	0.00	5.80	0.00	0.78
	5	5.69	0.08	0.03	25.71	6.38	3.52	0.08	0.00	13.29	0.00	1.34
SDOC	1	3.84	0.05	0.03	10.25	3.79	1.67	0.05	0.02	5.94	0.56	0.80
	2	2.32	0.03	0.01	6.10	0.69	1.63	0.03	0.01	5.23	0.00	0.34
	3	4.04	0.06	0.02	19.08	6.72	3.72	0.04	0.03	5.43	0.33	0.44
	4	2.66	0.03	0.01	9.56	2.75	2.59	0.03	0.01	4.50	0.27	0.70
	5	4.07	0.02	0.01	15.09	3.67	2.71	0.04	0.01	5.91	0.15	0.43
MDOC-SV	1	3.62	0.04	0.02	9.58	3.81	1.65	0.05	0.02	5.92	0.11	0.53
	2	3.87	0.06	0.02	15.96	4.53	3.12	0.06	0.01	6.41	0.48	0.59
	3	4.03	0.05	0.02	12.92	3.54	2.21	0.06	0.02	6.74	0.38	0.67
	4	2.75	0.06	0.02	8.37	1.84	2.04	0.03	0.00	4.50	0.30	0.64
	5	4.29	0.14	0.10	15.55	8.68	2.62	0.05	0.00	6.26	0.07	0.46
MDOC-DV	1	2.88	0.04	0.01	6.42	1.32	1.23	0.15	0.04	70.71	0.00	23.55
	2	3.95	0.04	0.04	10.43	5.85	1.64	0.05	0.01	6.15	0.55	0.56
	3	3.39	0.06	0.02	9.99	1.67	1.95	0.06	0.01	5.30	0.08	0.56
	4	2.81	0.06	0.02	7.52	1.32	1.68	0.27	0.03	70.71	0.00	24.16
	5	3.28	0.05	0.03	8.93	3.95	1.72	0.04	0.02	5.16	0.62	0.57

follow the worse performing robots. In the roulette wheel approach, all robots successfully reached the light source in all configurations, whereas in two configurations of MDOC-DV using follow median approach the robots failed to do so. It was observed that, in all the cases, the standard deviation of time is not significant, hence this system proves to be adaptive to any initial configuration. There is no deviation of traveled distance for the NOC and SDOC using the follow median approach as the environment in these cases is of static nature and hence the robots follow the same median robot for all trials without any exception.

4. Conclusion and Future Directions

The application of Swarm robotics in Search and Rescue operation was implemented using CI methodology with two different approaches: the roulette wheel selection approach and the follow median approach. The same application was implemented by Roychowdhury et al. [15], considering an ideal scenario of static nature of environment, which is rarely observed. Hence, to make it closer to real-life scenario, the entire system was considered to be dynamic in nature. This implementation was successfully validated for following independent static test cases like No Obstacle Case (NOC), Stationary Obstacles Case (SOC), as well as dynamic test cases like Single Dynamic Obstacle Case (SDOC), Multiple Dynamic Obstacles Case with Same Velocity (MDOC-SV) and Multiple Dynamic Obstacles Case with Different Velocities (MDOC-DV). The tests were conducted by arbitrarily positioning the robots and obstacles which can move randomly in the arena. This implementation was tested on non-convex orientation of obstacles and a new perturbation technique was devised for the same.

Swarm robotics considered in dynamic nature can have a wide range of applications. This paper shows the implementation of an application considering a single plane while more complex real-life problems can also be solved using this methodology considering the robots and obstacles in different planes. Furthermore, this work uses a perturbation technique to overcome the problem of robots getting stuck in the non-convex region formed due to dynamic configuration of obstacles. A better approach can be designed for this by making use of multiple proximity sensors on the robots or by making use of computer vision technology. However, these approaches may increase the heftiness which needs to be worked upon. This work can have strong potential in implementing applications like assistive swarm-robotics in healthcare, aerial swarm-robotics and aquatic swarm-robotics. Authors intend to apply this dynamic CI approach in such fields.

References

[1] R. Abduljabbar, H. Dia, S. Liyanage, S.A. Bagloee, Applications of artificial intelligence in transport: an overview, Sustainability 11 (1) (2019) 189.

[2] I. Andras, E. Mazzone, F.W.B. van Leeuwen, G. De Naeyer, M.N. van Oosterom, S. Beato, T. Buckle, S. O'Sullivan, P.J. van Leeuwen, A. Beulens, N. Crisan, F. D'Hondt, P. Schatteman, H. van Der Poel, P. Dell'Oglio, A. Mottrie, Artificial intelligence and robotics: a combination that is changing the operating room, World J. Urol. (2019).

[3] S.H. Alsamhi, O. Ma, M.S. Ansari, F.A. Almalki, Survey on collaborative smart drones and internet of things for improving smartness of smart cities, IEEE Access 7 (2019) 128125–128152.

[4] J.G. Blitch, Artificial intelligence technologies for robot assisted urban search and rescue, Expert Syst. Appl. 11 (2) (1996) 109–124.

[5] S. Biswas, S.G. Anavatti, M.A. Garatt, Obstacle avoidance for multi-agent path planning based on vectorized particle swarm optimization", in: G. Leu, H. Singh, S. Elsayed (Eds.), Intelligent and Evolutionary System, Proceedings in Adaptation, Learning and Optimization, 8, Springer, Cham, 2017, pp. 61–74.

[6] Q.B. Diep, I. Zelinka, R. Senkerik, "An algorithm for swarm robot to avoid multiple dynamic obstacles and to catch the moving target, in: International Conference on Artificial Intelligence and Soft Computing, 2019, 2019, pp. 666–675.

[7] M. Duarte, J. Gomes, V. Costa, T. Rodrigues, F. Silva, V. Lobo, M.M. Marques, S.M. Oliveria, A.L. Christensen, Application of swarm robotics systems to marine environmental monitoring", in: Proceedings of the OCEANS 2016 - Shanghai, Shanghai, 2016, pp. 1–8. 10-13 April 2016.

[8] N.C. Eli-Chukwu, Applications of artificial intelligence in agriculture: a review, Eng., Technol. Appl. Sci. Res. 9 (4) (2019) 4377–4383.

[9] I.P. Ktistakis, N.G. Bourbakis, Assistive Intelligent Robotic Wheelchairs, IEEE Potentials 36 (1) (2017) 10–13.

[10] A.J. Kulkarni, M.F. Baki, B.A. Chaouch, Application of the Cohort-intelligence optimization method to three selected combinatorial optimization problems, Eur. J. Oper. Res. 250 (2) (2016) 427–447.

[11] A.J. Kulkarni, I.P. Durugkar, M. Kumar, Cohort intelligence: a self supervised learning behavior, in: Proceedings of the 2013 IEEE International Conference on Systems, Man, and Cybernetics, Manchester, 2013, pp. 1396–1400. 13-16 October 2013.

[12] N.S. Patankar, A.J. Kulkarni, in: Variations of cohort intelligence, 22, Soft Computing, Springer, 2017, pp. 1731–1747.

[13] J. Penders, L. Alboul, U. Witkowski, A. Naghsh, J. Saez-Pons, S. Herbrechtsmeier, M. El-Habbal, A Robot Swarm Assisting a Human Fire-Fighter, Adv. Robot. 25 (1-2) (2011) 93–117.

[14] C. Renzi, F. Leali, M. Cavazzuti, A.O. Andrisano, A review on artificial intelligence applications to the optimal design of dedicated and reconfigurable manufacturing systems, Int. J. Adv. Manuf. Technol. 72 (1-4) (2014) 403–418.

[15] P. Roychowdhury, S. Mehra, R. Devarakonda, P. Shrivastava, S. Basu, A.J. Kulkarni, "A Self-organizing multi-agent cooperative robotic system: an application of cohort intelligence algorithm, in: A.J. Kulkarni, P.K. Singh, S.C. Satapathy, K.A. Husseinzadeh, K. Tai (Eds.), Socio-cultural Inspired Metaheuristics, Studies in Computational Intelligence, 828, Springer, 2019, pp. 121–135.

[16] C. Stachniss, W. Burgard, "An integrated approach to goal-directed obstacle avoidance under dynamic constraints for dynamic environments, in: IEEE/RSJ International Conference on Intelligent Robots and Systems, Lausanne, Switzerland, 2002, pp. 508–513. 30 September-4 October 2002.

[17] P. Svenmarck, L. Luotsinen, M. Nilsson, J. Schubert, "Possibilities and challenges for artificial intelligence in military applications", in: Proceedings of the NATO Big Data and Artificial Intelligence for Military Decision Making Specialists' Meeting, Neuilly-sur-Seine, France, 2018, pp. 1–16. 30 May-1 June 2018.

[18] C.T. Yen, M.F. Cheng, A study of fuzzy control with ant colony algorithm used in mobile robot for shortest path planning and obstacle avoidance, Microsyst. Technol. 24 (1) (2016) 125–135.