Available online at www.sciencedirect.com

ScienceDirect

IFAC PapersOnLine 54-4 (2021) 1–6

Distributed Particle Swarm Optimization for Multi-Robot System in Search and Rescue Operations

David Paez * **Juan P. Romero** * **Brian Noriega** *
Gustavo A. Cardona ** **Juan M. Calderon** ***

* *Department of Electronic Engineering, Universidad Santo Tomas, Colombia (e-mail: {david.paezr, juanpromeroc, briannoriega}@usantotomas.edu.co)*
** *Department of Electrical and Electronic Engineering, Universidad Nacional de Colombia, Colombia. (e-mail: gacardonac@unal.edu.co)*
*** *Department of Computer Science and Engineering, Bethune Cookman University, Daytona FL, (e-mail: calderonj@cookman.edu), Universidad Santo Tomás, Bogotá, Colombia (e-mail: juancalderon@usantotomas.edu.co)*

Abstract: In this paper, we tackle the problem of finding victims in a search and rescue environment, taking into account that the terrain in a disaster is complex and dangerous for rescuers to traverse. Furthermore, time is crucial when saving lives from a disaster considering all of these challenges, a solution is proposed by using a cooperative robotics team, which speeds up the process of searching for survivors and avoids risking additional lives. This article focuses on the navigation of a swarm of robots that can avoid collisions with obstacles that can be either static or dynamic, and locate victims. The method we employed to navigate in the environment is based on a DPSO Distributed Particle Swarm Optimization where each particle swarm represents a single robot. We show the interaction between swarms, and we make use of artificial potential functions for collision avoidance and for attraction to victims. We perform several simulation experiments to test the navigation algorithm, avoiding obstacles, and finding victims. These experiments are carried out in different environments, varying the number of victims and also the size and number of obstacles. The results show how the algorithm allows the group to avoid obstacles and find possible victims, all experiments are implemented using a combination of Python with V-Rep.

Keywords: Robot Navigation, Autonomous Mobile Robots, Distributed-PSO, Distributed navigation, Bio-inspired systems.

1. INTRODUCTION

There are many different types of calamities that result from natural causes such as fires, earthquakes, river or weather disasters. These are occurring with an increasing frequency today. Recent studies have shown (Thomas and López, 2015), that the incidence of natural disasters has almost tripled in the last decades. Also, chemical risks, nuclear hazards, risks of armed confrontation endanger human life and property safety. It is worth mentioning that some of these are also caused by humans.

When such disasters occur the rescue of victims is imperative. Rescuing victims in time makes a great difference to save as many people as possible (Casper and Murphy, 2003). However, structure conditions after a disaster occur make it difficult to proceed and seek victims. Additionally, it can be dangerous for rescuers to look for victims when they ignore much of the crucial information in the terrain or remaining risks in the area. Rescue is critical work due to this is time-limited, it is dangerous and complex. Generating the necessity of having really trained and specialized rescuer teams which will reduce the risk of adding more victims to the event.

Researchers have been agreeing that the robotics field has great potential in disaster areas to maximize the number of survivors. Some of the specialized robots have shown great performance in assisting humans in the framework of a varied task (Cardona et al., 2021). However, in missions where robots can be affected easily as in search and rescue, where the risks could make robots drop out is worthy to have multi-robot systems, which proportionate robustness to the mission since if one robot fails or drop-out task satisfaction is not compromised if there are more robots than cover the mission (Cardona and Calderon, 2019). What was previously done by humans and trained dogs are now being taken over by robots which reduces the risk to human and animal lives. Generally, rescue robots are operated remotely because their operation requires advanced intelligence in the sense that tasks could be very specialized and in presence of disturbances, perturbations, and uncertainties, which are difficult to handle by robots. The purpose of this is to collect and process information,

10.1016/j.ifacol.2021.10.001

keeping search and rescue personnel safe avoiding further causalities (Dubé et al., 2016).

2. RELATED WORK

One of the recent techniques to achieve this cooperative behavior between robots is bio-inspired systems, which are based on animals, such as birds, ants and bees (Gordon, 2010). These animals exhibit applicable behaviors, such as localization, recognition, and the capability to explore large areas by cooperation when searching for a specific target. These techniques give alternatives for robots to be applied in multiple fields such as the military, agriculture, industry, and surveillance to mention just a few of them Mahmud et al. (2020). The focus of this paper is the application of particle swarm optimization (PSO) Huang and Dun (2008), in a multi-robot frame used in search and rescue environments. These robots maintain a formation while seeking victims and acquiring their location. Robots must avoid static or dynamic obstacles that might appear in the environment, we consider other robots as obstacles as well. The use of swarm robotics for disaster area exploration facilitates and speeds up rescue missions, making SLAM simultaneous localization and mapping faster and safer for rescuers.

Several PSO algorithms have been proposed and developed to achieve desired behaviors in Search and Rescue operations using multi-robot systems. In Couceiro et al. (2011b), exploration algorithms for multiple robot systems based on the PSO theory are proposed. This approach is based on the idea of each particle represents a robot of the multi-agent system. Generally, the multi-robot system is fully represented using a simple swarm, following the indications of the classic algorithm approach. Then, in Couceiro et al. (2011a) an improved version of the same algorithm is proposed using Fractional Order Equations with a Darwinian approach of the classic PSO, where some improvements are achieved looking for the avoidance of local minima. Within this group of works, finally, in Kumar et al. (2017) the software implementation of the Search and Rescue operations system is carried out using the Darwinian approach. Although these works have a similar purpose and are based on the same technique, we differ from them in the way they represent robots since we use a totally multi-swarm system. Approaches in which a simple robot is represented by a swarm and therefore a multi-robot system leads to the creation of a multi-swarm system is developed in Ayari and Bouamama (2017). Although the system is multi-swarm, the simulations presented do not face the problem of multiple swarms colliding in the case of converging on the same local or global minimum. Unlike our approach where we propose repulsive forces between the different swarms to avoid robot collisions. In Abbas and Abdulsaheb (2016) a multi-robot system is presented but using a multiobjective approach where robots are represented by each particle of a singular swarm. Finally in Bouamama (2010) is introduced Distributed Particle Swarm Optimization, where the multi-agent system is represented by a multi-swarm, but each agent is in charge of a subspace and the proposal is generalized as a multi-optimization distributed system using subspaces. The idea of subspaces generates a huge difference with our proposal because our entire multi swarm system works in the same space. We propose some variations so that this multi-agent system can interact within the same universe.

In the present document, we introduce multi-Robot system navigation based on a Multi-PSO approach. In this research, we are presenting four different contributions focused on allowing the navigation of a multi-robot system in a disaster zone and aimed to search for human victims in the same scenario. First, every agent of the multi-robot system is represented by a complete and independent particle swarm where every particle of the swarm represents the same robot. Second, given a single robot is represented by a swarm of particles, the localization of the physical robot is determined by the average of the particle positions belonging to the same swarm that represents the robot. Third, in a different approach than the classical PSO theory, we include a repulsion vector among particles that make up the same swarm given a minimal limit distance. The objective of this repulsion force is to avoid the convergence of the particles at the same point in contrast to the classical PSO approach. Conversely, we are looking for the exploration of the near areas to the robot through keeping a minimal distance among particles of the same swarm. Finally, the fourth contribution is the inclusion of a repulsion vector among different swarms. This repulsion vector is defined between the central position of a swarm and any particles that makes up the rest of the swarms. This repulsion vector aims to keep a minimum distance among robots given without this approach is highly probable the collision among robots given the convergence of the different swarms to the same position.

3. PROBLEM STATEMENT

We consider a search and rescue environment $S \in \mathcal{R}^3$ with a set of obstacles $S_o \subset S$, we denote S_c as areas that are clear to move for the robots. Note that $S_c \bigcup S_o = S$, we do consider obstacles as static places where robots cannot traverse, also other robots in the environment are considered as dynamic obstacles. Static obstacles can be either damaged structures and objects that caused the robots to avoid those zones. We assume victims are located randomly in the environment and even where obstacles are all around, there are not victims over a fire or an obstacle as robots have an objective to avoid collisions this might cause robots to never find the victim. The victims are in unknown places for the robot, however, the robot is capable to sense victims when those are in the sensing range.

Even when the working space belongs to $\mathcal{R}^3$ we simplify the system as an environment on $\mathcal{R}^2$ since the team of robots we make use of are ground robotics platforms base and those cannot move in the z-axis. Then areas of obstacles and clear to move can be also computed as two-dimensional spaces.

The robots we employ in this work are ground base, the well-known differential drive robots, which configuration space belongs to $SE(2)$ meaning we have two axes in which we can move and an angle to control the direction of where the robot is pointing to. We make use of the kinematic model of a differential robot as in Siegwart et al. (2011)

$$\begin{aligned}\dot{v}_x &= \frac{1}{2} r_R \dot{\varphi}_R + \frac{1}{2} r_L \dot{\varphi}_L,\\ \dot{v}_y &= 0,\\ \dot{\theta} &= \frac{1}{2l} r_R \dot{\varphi_R} - \frac{1}{2l} r_L \dot{\varphi_L},\end{aligned} \tag{1}$$

where r_R and r_L are the radius of the Right and left wheels, φ_R and φ_L are the angular velocities of right and left wheels respectively, l is the distance from the center of mass to the center of the wheel, $\dot{v}_x$, $\dot{v}_y$, and $\dot{\theta}$ are linear velocities in x-axis, y-axis, and yaw-angle, respectively. However, every robot can be modeled as a single integrator dynamic as shown in Bullo et al. (2009) as,

$$\dot{x}_i = u_i, \tag{2}$$

where x_i are the position vector of each robot in the environment and u_i is the control signal for each of the robots, note $i \in \mathcal{N}$ where $\mathcal{N}$ is the set of ground robots.

We assume all of the robots are able to communicate their location to other robots working in the environment. Also, each robot is able to locally sense the environment and acquire local information of where the obstacles, other robots, and victims are. The victims are detected and localized when those are in the robot sensing range, however, the robot does not know the victim localization before the exploration mission.

Then robots are able to navigate in the environment if they have proper information from it. Additionally, when a robot found a victim it is notified to others, notice robots are constantly navigating avoiding static and dynamic obstacles while trying to find victims in the environment.

Given a set of robots N in a search and rescue environment with static and dynamic obstacles, our goal is to find control inputs u_i to make the robots navigate in the environment through S_c zones, avoiding S_o areas and also other robots, until finding a victim. The approach we use to solve Problem 3 is through a Distributed Particle Swarm Optimization DPSO algorithm applied to a team of robots. We make use of artificial potential functions to create attraction forces to unknown areas and victims when those victims are detected. In the same way, we use repulsion forces to avoid collisions wither with obstacles and other robots. The description of the navigation control algorithm is then shown in the next section.

4. CONTROL DESIGN

In this section, we show the procedure on how to obtain the control applied to each robot. First let us define the classical Particle Swarm Optimization, which is widely used and is defined below

$$\begin{aligned}v_{n+1} &= wv_n + c_1 r_1 [pB_n - p_n] + c_2 r_2 [gB_n - p_n],\\ p_{n+1} &= p_n + v_{n+1},\end{aligned} \tag{3}$$

where v_{n+1}, v_n are velocity of the each particle in the next iteration, and velocity in the current iteration respectively, $c_1,\ c_2$ are learning constants corresponding to the individual and group component, $r_1,\ r_2$ are random values between 0 and 1, w is the inertia constant computer for the velocity of the particle, $pB_n,\ gB_n$ are the positions corresponding to the best historical value of the particle and the best value of the swarm respectively. Finally, $p_{n+1},\ p_n$ is the calculation of the particle position in the next iteration, and the position in the current iteration, respectively.

PSO algorithm has been used in robot motion planning before, where each robot is a particle that belongs to the swarm (León et al., 2016). They navigate with other particles to find the optimum on a surface or objective function that can be constrained or not. However, the classic PSO algorithm can lead to poor performance, first, it is slower in the speed of convergence compared to other optimization methods. Also, the algorithm usually is trapped in the local minimum since most of the particles are trapped in the local minimum causing poor performance or even being incapable of finding the optimal value. Finally, the PSO algorithm does not consider collision avoidance between particles, which makes multiple particles stay very close falling sometimes all together into the same local minimal and leading to the same information over the objective function.

Instead of the classical Particle Swarm optimization algorithm, we modify it to work with multiple PSO running over the same environment and interact with each other.

Definition 4.1 (Robot Representation)**.** *Every robotic agent from the multi-robot system is represented by its own Particle Swarm, hence there are as many-particle swarms as there are robots in the system.*

Definition 4.2 (Localization Robot)**.** *The localization of every robot is defined by the position average of all particles from its representative swarm.*

Definition 4.3 (Particle Swarm Initialization)**.** *Every particle of the same swarm is randomly initialized around of central robot point using Gaussian distribution function $N(\mu, \sigma)$ with μ as robot localization point and σ as robot size.*

In contrast to the classic algorithm, we do not allow particles to collide between them. Also, we generate an artificial potential function that is in charge of interactions between different swarms, so not only particles of the same swarm avoid collisions between them but also they avoid collisions with particles from other swarms. The proposed approach for a PSO algorithm which is a combination of the classical one and the new additions according to our requirements is described below.

$$\begin{aligned}v_{n+1} &= wv_n + c_1 r_1 \|pB_n - p_n\| + c_2 r_2 \|gB_n - p_n\| +\\ &\quad c_3 r_3 Rp_n + c_4 r_4 Rs_n,\\ p_{n+1} &= p_n + v_{n+1},\end{aligned} \tag{4}$$

where $\|pB_n - p_n\|$ is the error between the historical best position of the swarm particle and its current position. $\|gB_n - p_n\|$ is the difference between the best position among all particles swarm and the position of the current particle calculation.$Rp_n,\ Rs_n$ are the new terms, where Rp_n is the sum of the repulsion force of the particle with respect to the particles from the same swarm and Rs_n is the sum of repulsion to the average position of the other swarms. $c_1,\ c_2,\ c_3,\ c_4$, are the learning constants in charge of the trade-off among the four different approaches (Individual, social, particle repulsion, and swarm repulsion) of particle velocity calculation. $r_1,\ r_2,\ r_3,\ r_4$ are random values between 0 and 1 with the aim to provide a stochastic balance among the velocity calculation approaches.

Let's define the Repulsion force as a Gaussian Function as

$$G_{p|s}(x) = \frac{1}{1 + |\frac{x-\mu}{\sigma}|^{2g}}, \tag{5}$$

where μ is the average value of the function in this case its value is 0 since it would indicate the position of the particle that is being analyzed. σ is the standard deviation of the function that indicates the minimum proximity distance, in which the repulsion factor is 50%, g is the gain of the function that indicates how abrupt is the variation of the repulsion that is exerted when it is very close to the minimum distance, x is the data to determine in the function, this is the Euclidean distance that exists between the particle that is being analyzed and the repulsion force corresponding to another particle or to the middle position of another swarm.

We define the repulsion between particles in the same swarm as follows

$$Rp_n = \sum_{i=0}^{k} \|G_p(p_i - p_n)\|, \tag{6}$$

where $\|G_p(p_i - p_n)\|$ is the artificial repulsion force as a function of distance, which exists between the current particle and the rest in the same swarm. It is modeled by 5, i is the particle index in the swarm at which this particle belongs, k is the number of particles present in the swarm, n is the index of the current particle.

In a similar way, we define an inter-swarms repulsion which is defined as follows and prevent the swarms to get close to each other

$$Rs_n = \sum_{i=0}^{j} \|G_s(gM_i - p_n)\|, \tag{7}$$

where $\|G_s(gM_i - p_n)\|$ is the artificial repulsion function which depends on the distance. This force is applied between the current particle and the average positions of the other swarms, modeled by 5, i refers to the index of each swarm, j is the number of swarms present in the environment, n is the index of the current particle.

5. SIMULATION RESULTS

In order to perform experiments for multi swarm decentralized robot navigation in search and rescue missions as previously depicted, the multi-agent model and environment were implemented using Python and VRep. Python environment was used to implement the proposed mathematical model and evaluate the performance of its features in different cases. VRep was used in connection to Python with the aim to observe the behavior of the proposed model in 3D environments using the virtual model of the physical robotics platforms as Pioneer 3DX. The main objective of the use of VRep as a simulation environment is to validate the decentralized multi-robot navigation using distributed multi-swarm systems in realistic disaster scenarios.

Four experiments were performed, the first two experiments were performed on python with the aim to evaluate the performance of this work contribution related to particle and swarm repulsion forces respectively. The last two experiments correspond to the verification of the multi-robot behavior in a multi obstacle area with one and three human victims respectively. These last two experiments are performed using VRep in connection to Python.

5.1 A single Robot and a single Goal

This is the first case shown in this group of experiments, it corresponds to a single robot navigating through a multi obstacle area in search of a victim. The objective of this experiment is to show how a single particle swarm guides a single robot through the area and depicts how the particle repulsion forces keep distances among them. Additionally, the particles guide the robot avoiding obstacles and reaching the Goal as shown in figure 1. For this case, the victim and the open area (Goal) are localized together.

The figure 1 depicts the initial and final robot position with cyan and magenta dots respectively, the initial and final position of every particle with green and orange dots, and particle trajectories in blue lines. Additionally in a box, the particle position in a trajectory midway shown in magenta, and the repulsion forces among particles with green arrows are indicated. This box shows how the distance among particles is held.

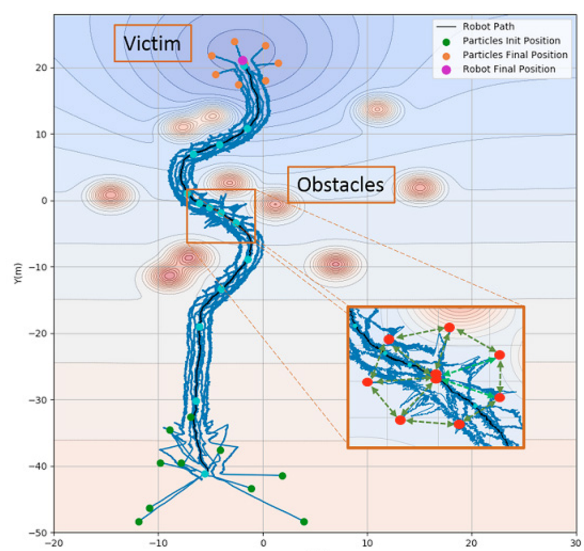

Fig. 1. Single robot going to the goal.

Figure 2 shows the minimal distance among particles in blue and the minimal distance allowed in magenta, 1 meter in this experiment. This figure depicts the minimal distance among particles through the 85 epochs. Any epoch is composed of 100 iterations. As an analysis result, figure 2 shows that the minimal distance was never reached by the inter-particles distance.

5.2 Multiple Robots in a Dynamic Formation

This experiment has the objective to observe the performance of the inter-swarm repulsion forces. One of the contributions of this paper is based on the application of the repulsion forces inter-swarms which aim to prevent collisions among robots. This experiment is based on forcing the robots to pass through a confined space in order to reach the victim. The obstacles are localized shaping a bottleneck and reducing the space on the road to the victim, those obstacles are trees as shown in figure

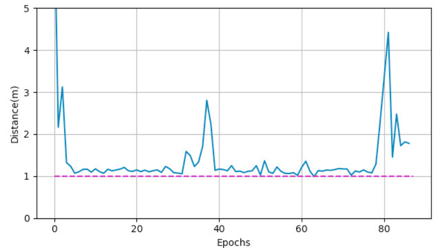

Fig. 2. Minimum distance among particles of the swarm.

3. Additionally, figure 3 shows the robots addressing the victim.

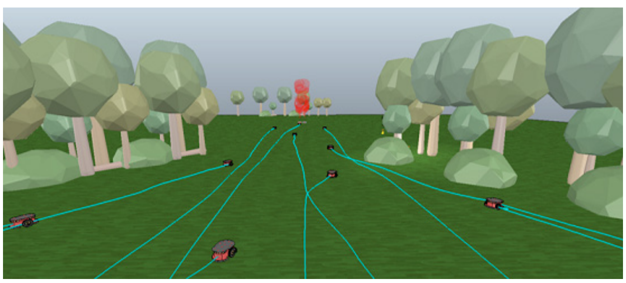

Fig. 3. Multiple robots system.

Figure 4 depicts the path followed by each robot in their way to the victim. The cyan dots are the initial position of every robot, the magenta dots are the final position, and the green dots show an intermediate position of the robots through the way. The green dots depict how the robot formation is modified by the robots in a way to avoid collisions and pass through the bottleneck and reach the victim. This experiment shows how the repulsion forces among swarms interact keeping a minimal distance among robots and allowing them to modify the formation dynamically. Even the final position (magenta dots) of the robots show how the robots keep the distance among them and at the same time they are localized around the victim.

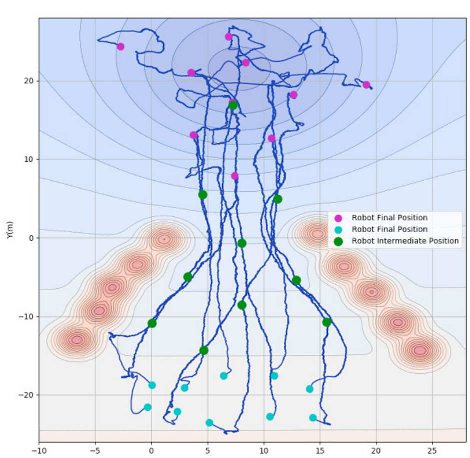

Fig. 4. Multi-robots path.

5.3 Multiple Robots and one victim

This experiment aims to test the proposed system in a virtual simulation environment. In this case, a multi-robot system composed of 26 robots that navigate in an environment with the presence of one victim and trees was implemented. Additionally, every robot is represented by a swarm of 50 particles which represent a total of 1300 particles, because of it the results of this experiment are shown using just the virtual environment as shown in figure 5. The blue lines depict the path followed by every robot. The simulation shows how the 26 robots team navigates through the trees looking for a victim. It is also observed how one group of 5 robots found a victim and navigate surrounding it and keeping a distance between them as highlighted by a yellow dotted circle in figure 5. This behavior is a result of the inter-swarms repulsion force because this force is avoiding robots converge at the same point, for this case that point would be the victim. Additionally, other collective behavior is emerging as the task assignment. The robots that detected the victim repels the other robots conditioning them to keep going straight and looking for more victims or at least to inspect the rest of the field as highlighted with white dotted circles in figure 5.

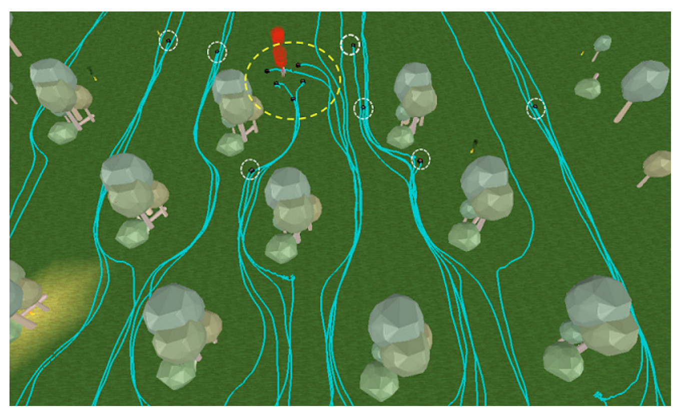

Fig. 5. Robot swarm navigation in a out-doors search and rescue environment - single victim.

5.4 Multiple Robots and multiple victims

This is the last experiment performed where a multi-robot system composed of 26 robots and three victims is simulated in a virtual environment, as depicted by figure 6. As expected the robots reached the three victims by different robot subgroups. As explained previously, the inter-swarm repulsion forces keep the distance among robots who found the victims as highlighted with the three yellow dotted circles. Given the repulsion forces, once a robot group finds the first victim the rest of the robots are repelled, forced to keep navigating, and looking for more possible victims. The experiment shows how the robots can navigate looking for victims and avoiding obstacles at the same time, even preventing crashes to other robots from the same group. Although the three victims were found, there are more robots who keep navigation and looking for more victims as highlighted with small white dotted circles in figure 6. This experiment evidences that the proposed method prevented the robots from colliding with each other even when more than one robot had reached the victim. The repulsive inter-swarm forces work well by preventing the robots from colliding. The repulsive forces within the swarm help the particles explore sub-optimal space but can help find better or alternate paths for the robot. At the end of this experiment, the three victims were found by seventeen in groups of nine, four, and four robots respectively, demonstrating that the robots effectively repel each other without sacrificing the search for human targets.

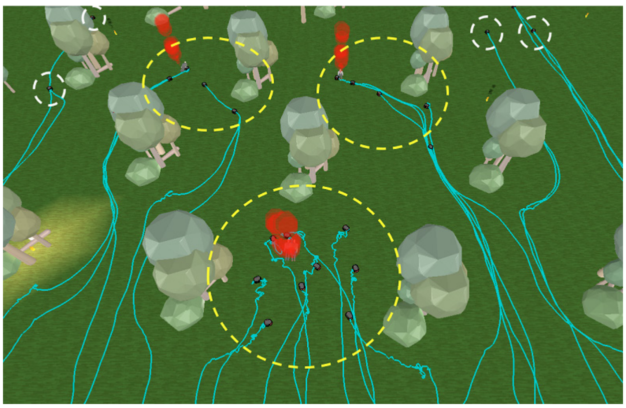

Fig. 6. Robot swarm navigation in a out-doors search and rescue environment - multiple victims.

6. CONCLUSIONS AND FUTURE WORK

We propose a D-PSO Distributed Particle Swarm Optimization algorithm in a multi-robot platform which allows the motion planning for robots to avoid dangerous areas and obstacles. In addition, we maintaining effective communication between the robots through the knowledge of their nearby environment. Either the collision between robots and obstacles is tackled based on artificial potential functions. In the same way, we manage to impose attractive forces on unknown areas and potential victim locations. DPSO here allows the swarm to expand in the environment and due to the interaction with other swarms, it helps the robots to escape from local minimum and to find alternative paths through disaster scenarios to reach an optimum. Furthermore, our modified DPSO algorithm, includes repulsion not only between particles but also between swarms, generating they explore in a wider manner environment. Note that here each swarm controls a single robot where the robot position is defined by the average of the particle swarm positions. In future work, we plan to study more the interaction between multiple swarms that can cooperate to deal with common goals in presence of adversarial agents.

REFERENCES

Abbas, N.H. and Abdulsaheb, J.A. (2016). An adaptive multi-objective particle swarm optimization algorithm for multi-robot path planning. *Journal of Engineering*, 22(7), 164–181.

Ayari, A. and Bouamama, S. (2017). A new multiple robot path planning algorithm: dynamic distributed particle swarm optimization. *Robotics and biomimetics*, 4(1), 1–15.

Bouamama, S. (2010). A new distributed particle swarm optimization algorithm for constraint reasoning. In *International Conference on Knowledge-Based and Intelligent Information and Engineering Systems*, 312–321. Springer.

Bullo, F., Cortes, J., and Martinez, S. (2009). *Distributed control of robotic networks: a mathematical approach to motion coordination algorithms*, volume 27. Princeton University Press.

Cardona, G.A. and Calderon, J.M. (2019). Robot swarm navigation and victim detection using rendezvous consensus in search and rescue operations. *Applied Sciences*, 9(8), 1702.

Cardona, G.A., Ramirez-Rugeles, J., Mojica-Nava, E., and Calderon, J.M. (2021). Visual victim detection and quadrotor-swarm coordination control in search and rescue environment. *International Journal of Electrical & Computer Engineering (2088-8708)*, 11(3).

Casper, J. and Murphy, R.R. (2003). Human-robot interactions during the robot-assisted urban search and rescue response at the world trade center. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 33(3), 367–385.

Couceiro, M.S., Ferreira, N., and Tenreiro Machado, J. (2011a). Fractional order darwinian particle swarm optimization. In *Symposium on fractional signals and systems*, 127–136.

Couceiro, M.S., Rocha, R.P., and Ferreira, N.M. (2011b). A novel multi-robot exploration approach based on particle swarm optimization algorithms. In *2011 IEEE International Symposium on Safety, Security, and Rescue Robotics*, 327–332. IEEE.

Dubé, R., Gawel, A., Cadena, C., Siegwart, R., Freda, L., and Gianni, M. (2016). 3d localization, mapping and path planning for search and rescue operations. In *2016 IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR)*, 272–273. IEEE.

Gordon, D.M. (2010). *Ant encounters: interaction networks and colony behavior*, volume 1. Princeton University Press.

Huang, C.L. and Dun, J.F. (2008). A distributed pso–svm hybrid system with feature selection and parameter optimization. *Applied soft computing*, 8(4), 1381–1391.

Kumar, A.S., Manikutty, G., Bhavani, R.R., and Couceiro, M.S. (2017). Search and rescue operations using robotic darwinian particle swarm optimization. In *2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, 1839–1843. IEEE.

León, J., Cardona, G.A., Botello, A., and Calderón, J.M. (2016). Robot swarms theory applicable to seek and rescue operation. In *International Conference on Intelligent Systems Design and Applications*, 1061–1070. Springer.

Mahmud, M.S.A., Abidin, M.S.Z., Emmanuel, A.A., and Hasan, H.S. (2020). Robotics and automation in agriculture: Present and future applications. *Applications of Modelling and Simulation*, 4, 130–140.

Siegwart, R., Nourbakhsh, I.R., and Scaramuzza, D. (2011). *Introduction to autonomous mobile robots.*

Thomas, V. and López, R. (2015). Global increase in climate-related disasters. *Asian Development Bank Economics Working Paper Series*, (466).