Journal of Theoretical and Applied Information Technology
15th November 2019. Vol.97. No 21

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

VICTIM LOCALIZATION USING DARWINIAN IMPERIALIST COMPETITIVE ALGORITHM

AHMAD AL_QEREM

[1]Computer Science Dept, Zarqa University, Jordan

E-mail: ahmad_qerm@zu.edu.jo

ABSTRACT

Grid search for the victim in unknown environments using a swarm of robots tend to be not practical. Many research has been done using evolutionary algorithms for victim localization. The search problem is modelled as an optimization problem. The convergence to the optimal solution is one of the key factors in such a problem. Due to the harsh and unpredictable nature of the environment, cooperative navigation and victim identification in these environments are incredibly challenging to achieve. The obstacle avoidance, collision of robots and intra communication between them are still needed to assist the victim localization process in unpredictable situations. In this study, the Robotics Darwinian Imperialist Competitive Algorithm (RDICA) algorithm has been used for victim localization, to avoid a local optima problem, the exploration facility using do-revolution feature has been used. According to multiple experiments that conducted under different scenarios, distinctive results have been revealed. The average of crashed robots significantly reduced. Moreover, a 7.3 and 6.93% achieved improvement over the well-known state of the arts DPSO and RDPSO algorithms respectively while preserving the comparable number of iterations adequate to gain these results.

Keywords: *Imperialist Competitive Algorithm Victim Localization Swarm Robotics Do-Revolution*

1. INTRODUCTION

The number of victims s caused by natural disasters, wars or attacks has risen rapidly over the past few years. You can notice that on a daily basis dozens of innocent people have suffered from such disasters. In addition, other humans usually risk their lives in order to rescue them (i.e. civilian defenders). However, robots replaced humankind in search and rescue tasks to reduce the number of victim s. However, this solution required intervention by human beings, monitoring and remote control, while it cannot be used in certain situations, especially as people may not be very aware of the environment. As a result, non-human solutions were proposed that would interact with groups of robots to reach the victim without any external instructions[1]. This new approach was called swarm robotics.

The use of one of three types of algorithms, namely non-heuristic [2] algorithms, heuristic [3] and meta-heuristic [4], developed Swarm robotics. The algorithms in current swarm robotics solutions are less precise and not so fast, based on comparisons in literature reviews. Swam robots thus need a long time to define the position of the victim s and reach them, which could risk their lives. Moreover, in case of fire and sinking, these robots sometimes have problems to locate victim s in a short space of time. In past efforts, an accurate and effective robotic swarm has not been achieved [5]. The precise and efficient use of these robots in order to save people's lives is thus essential.

There are many bearable swarming robots applications. This involves miniaturization duties such as sensor functions that spread through micro machines or into human bodies [6] (Nano robots, macrobiology). One of the most promising uses of swarm robotics is in fatality rescue missions. The existence of liveliness through infraroad sensors may be detected through various bulk robotic swarms to areas where people cannot securely access. The swarm robots, on the other hand, can be adapted to tasks that require cheap designs, such as mining or farming search functions [7]. In particular, military robots ' squadrons can form a separate army[8]. A team of separate boats, able to take offensive actions and conduct themselves, have been tested by the US Navy. Boats are carefree and can be equipped for the demoralization or dismantling of galloping

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

vessels. The majority of efforts were focused on small tools. However, Harvard demonstrated in 2014 that it was the largest single squadron of 1,024 robots ever. [9]. A large number of other applications can be solved by swarms of small air cars that are currently under considerable research. [9] Existing systems, such a Shooting Star, are capable of managing hundreds of small air vehicles within the external area [9] with GNSS systems (e.g. GPS) or even by installing them with in-flight tracking systems[11], when GPS is available, as compared to the pilot studies of aerospace robots using precise motion detector systems. The auto-monitoring, [9] plume tracking[10] and small gonorrhea surveys of small air vehicles have already been tested. Cooperative fleets of land vehicles and drones, including targeted application for cooperative environment monitorings, [10] convoy protection, [11] and moving local goals and tracking, were subject to a number of actions. [12].

The Robotic Squadron is a highly organized, self-regulating robot system characterized by high repetition. Local devices and robots can not access global information are sensor and communication capabilities in the robot. The collective demeanor of the robot squad shows the interaction of each individual robot with its analogue and with the ambience. The robot swarm typically consists of homogenous robots, although heterogenic robot swarm examples exist Swarm Robotics is a multiple robots process as a system which comprises a large number of simple automation [13]. The desired collective behavior should result from robotic-environmental interactions. This process has emerged in the field of artificial swarm intelligence, also biological insect studies and various natural areas, in which the Swarm's behavior takes place [14]. Searching for robot swarm is a study of automation designing, their physical structure and dominant demeanors. It is an inspirational but not only the emerging behaviours, the swarm intelligence of gregarious insects. A wide range of sophisticated squadron situations is created by relatively simple individual rules. The principal component is the connection between the members of the team that creates a series of constant comments. The demeanors of the squadron cover permanently change of individuals in cooperation with others, as well as the demeanors of the entire cohort [15, 16]. Distinguished from common, distributed, robotic systems, the squadron can be confirmed on a large number of robots, and it promotes scalability [17]. For instance, wireless transmission systems such as radio frequency can achieve this domestic connection [18]. Minimization and cost are the main obstacles to the establishment of large groups of robots. Thus, simplicity must be reinforced with every robot that stimulates rather than every single piece the intelligent swarm of the group.

2. META-HEURISTIC ALGOITHMS

There are several research projects dealing with the problem of finding a proper design for swarm robotics to improve their efficiency and improve their search and rescue capability. This section examines some of the main works on this issue. The authors of [19] have addressed the deficiencies (Swarm Optimization (PSO) article and GSA). Algorithms for gravitational search). An Improved Particle Swarm Optimization (IPSO) has been embedded with motion mechanism of Improved Gravitational Search Algorithm (IGSA). The proposed hybridizations IPSO IGSA keep up the effective harmony among exploitation and exploration. Their method proposed was three steps. First, there are two methods for collectively hybridizing PSO and GSA with serial mode hybridization, i.e. GSA / PSO output as GSA / PSO input. The hybridization results then depend on PSO and GSA winning results. Each item in the hybridisation system eventually updates the function with each PSO tempo and GSA acceleration contribution. The improvement of the hybrid IPSO-IGSA approach is higher than the common path deviation. That is why it suggests that the hybrid regulations proposed provide the robots with adequate path planning and that they almost did not lose their way. The results also found that the robot's increase in the cost of speed offers the possibility of achieving a minimum of levels of the victim characteristic. In addition, with some robots growing, the number of tilts increases and the number of iterations increases the walking time. Nevertheless, the complexity of the hybrid algorithm proposed is less than the previous ones.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

The authors of [20], on the other hand, investigated the limits of PSO and different optimisation algorithms in the best local partner. In the same way, the preventive approaches to preventing obstacles and communication issues have not been taken into account when in the virtual (simulation) environment PSO and Darwinian Particle Swarm Optimization (DPSO) algorithms are not equal due to available obstacles in the planned pathways. In the end, the authors suggested extending PSO and RDPSO in order to deal with this issue, taking into account obstacle avoidance. They are referred to respectively as RPSO (Robot PSO) and RDPSO (Robot DPSO), in the fields of multi-robot structures, in such a way to undertake these promising organic systems. In order to improve the rules (RDPSO), the Ultrasonic Sensor has adapted the PSO and DPSO parameters to include a new parameter. The set of regulations has elements in this method first (search for gadgets (e.g. robots). Then there are several rounds, steps or iterations in the set of policies. Following that, the PSO and DPSO adopt the previous set of guidelines. Finally, the set of regulations is customized by adding a new parameter that takes its cost from the Ultrasonic Sensor. 33 robots were randomly deployed in a 30x30-sized search region using up to 6 swarms with 250 up to 350 iterations in order to validate this proposed method. The results showed that the robots were trapped in a good neighborhood during the preliminary test. However, growing the sort of robot will enhance this behavior and could grow the charge characteristic solution decreases. In contrast, the enormous form of iteration booms the response to the value of the function. Findings also confirmed that RDPSO performs better than RPSO and PSO algorithms. Similarly, improving this consent rate in order to respond to perfect and correct prediction after a vast variety of different iterations of the collective final consequences of the RDPSO algorithms. The endorsement that the RDPSO algorithms examine an unknown situation.

In a work proposed by Michael[21]. The chain of a semi-Markov. The predictive version of their microscopic counterpart using a simulation experiment method can be used in comparison. This observation is also an initial effort to develop a predictive version which is likely to take RDPSO dynamics and estimate the overall standard performance of robots. The result of the cease test is obtained through MRSIM, a robotic simulator based on a MATLAB. RDPSO's total efficiency depends on the parameter selection. The simulation results show that the predictive model can see universal RDPSO performance with small inconsistencies, and is thus regarded as a reliable means of integrating automated swarm. This fundamental relevance as you plan on the robot group general execution under RDPSO, without relying on a simulation or any other type of tentative evaluation. This also includes an improvement [5]by the same authors to read the RDPSO communications technology structure and features in addition to depicting the dynamics of the data pool shared by partners. The method of communication between the different robots does not matter. The optimization of communication between robots is used because the main step towards the upward implementation of RDPSO is taken. Fully based on the RDPSO standards, the ad hoc reactive routing protocol on-demand vector remoteness protocol reduces overhead message exchange within robotic swarms. The methodology proposed effectively reduces overhead communication, thus improving the quantification and the relevance of the RDPSO algorithm. It consisted of a collective exploration of a 20-10-meter scenario in which the cognitive response of robots was turned into the light that was tormented in its current position. The rationalized RDPSO is currently suitable to decrease to around 20 %. Notwithstanding the way in which the disparities among the overhead relocation and the desired broad spectrum of springs to convey one plot were currently not significant in smaller robot offices, it was significantly less for swars of 15 robots. The creators maintain that behind the essential unequivocal dispatch within the RDPSO, this paper provides a thorough explanation that gives the first step on that path. The strategies and techniques needed in the field of swarm robotics are urgently needed to improve the management of collective tasks through the use of robots. It's miles feasible to perform the design of such obligations by way of thinking about it as a group of easier behaviors called subtasks. All robots are required to perform the same task.

In this study, however, [22] the technology proposed by the researchers is based entirely on

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

wave algorithms. The Wave Swarm is a common place method for managing a series of subtasks which represent the collective navigation, which in swarm robotics is a key challenge. In this work, they have used the Wave Swarm algorithm. This algorithm was used to sequentially perform subtasks and to enable more complicated behaviors to be performed by using the message approach between swarm participants. 3 subtasks, namely recruitment, alignment and movement are performed for controlled navigation of the robot swarms / cluster. The wavelet algorithms, an allotted class of algorithm paintings called initiators, send messages to their respective neighbors in order to transmit and receive statistical data. They then start sending messages to the neighboring neighbors. As far as robot swarms are concerned, it is crucial that each robot does the same subtask immediately, regardless of whether some robots are delayed. Wave Algorithms are designated to indicate the performance of subtasks, creating a reliance on each robot during the neighborhood performance of the subtasks. Collective navigation can be defined because there is no lack of swarm members in the coordination movement of a robot group from a point to another in the environment. This move can be completed with a group of robots or with the right definition. The communication between adjacent robots should be kept completely unconnected in every other case.

On the other hand, the artificial fish swarm and the swarm algorithm for Artificial Bee Colony are subject to sensitive searching. Algorithms have some disadvantages, including long walking time, a mistaken look, which impedes the excessive dimensional and complex optimisation of characteristics. The authors of [23] proposed a new parallel hybrid swarm study according to multi-center clusters which fully implements the efficiency of clusters to maximize speed and precision.

They have designed a clever set of rules (Artificial Fish Swarm and Artificial Bee Colony) for hybrid swarms. Some approaches to the description of the sensitive serial rule set of the hybrid swarm were started to expand this hybrid algorithm. Firstly, the reverse gain in knowledge of the mechanism to initially randomly distribute a precondition for every person, calculating the health price of the situation for every man or woman. The authors then decided whether or not state-of - the-art algebra was reached. When reached, the placement of the man or woman problem with the nice health value is in two swarms, after which the reverse swarm interchanges with the lowest answer. If not, stop executing. If not. Random behavior is achieved if after individual leadership extremes, the most beneficial price is not continuously changed.

The rules are finished, and the artificial fish swarm and synthetic bee colony swarm producing the top-rated solutions in the local area. After this Swarms are allegedly dispensed into two companies, one carrying out the Dynamic weigh Artificial Fish Swarm algorithm DN-AFS set of rules and the other carrying out the Random Perturbation Artificial Bee Colony RP-ABC algorithm, in accordance with the preliminary answer obtained. Each individual's fitness value shall be calculated to parent whether the new fitness value calculated for two companies is higher than the contemporary fitness value.. Alternatives are selected if they are far advanced. If not, the modern health fee is preserved and the best price is compared and exchanged for sure iterations between groups. To the assessment of whether the maximum cycle range is reached, the quality response will be recorded.

For years, one un-married robot or multi-robots / swarm robots have been studied as difficulty with area insurances[24, 25]. This problem is also an important factor in synchronized location and mapping. The basic venture in swarm robots is the design of an assigned control and coordination mechanism in the target search and trapping process. In order to solve the impediment, the researchers in [26] proposed a decentralized manipulation of swarm robotics guidelines for target and trapping by bacterial chemotax. First, in line with main robotics ' positions on the target location a neighbourhood-coordinating device is set up. The used chemo-taxis algorithm for the microorganism was defined via the objective under the guidance of the gradient records. Chemotaxis is employed to clear the hassles of the dispensed controlled swarm robots by treating each robotic as a bacterial mechanism of the bacteria. The objective of this microorganism is to

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

find a better awareness of nutrients in the area. Duration of the search for a higher awareness of vitamins can detect the objectives which can be sensed by robots.

In order to verificate this algorathm the coefficient a=0.1/2 with a sensing range of 0.8 was initiated in 26 robots on a 300X300 square area. The results have shown that the BC ruleset is efficient and can cover places within certain steps and replenish grids, i.e., by using more robots, its miles faster to cover the places.

To check this algorithm, 26 robots had been initiated in a 300×300 square area wherein the coefficient a=0.1/2 and the sensing range of 0.8. The outcomes indicated that the BC set of rules is efficient and turned into able to cover the location inside certain steps and re-cowl the grids, that is, its miles faster to cover the place by using the use of more robots.

Another issue is the death of the essential equipment, or that fire could cause additional victims and actuality during catastrophes or earthquakes. There are currently no means and system that can correctly and quickly discover the places of the victims. This increases the number of causes of death and injured persons. Unfortunately, the detection of disaster areas causes both a postponement and an evacuation process in most of these catastrophic conditions. As a result, people are relying more on the use of robots for search and rescue work. A few heterogeneous robots are therefore developed to look and rescue robots from two complementary viewpoints [27]. They are rapid, accurate and can quickly locate the victim s, which could be entirely based on Heterogeneous Discoverage: the use of Voronoi regulations. While the allocation of robots responsibilities is based on optimization that reduces the cost functionality.

In [27], heterogeneous DisCoverage proposed was modified to confirm the heterogeneous case on the basis of earlier research.

This boundary set of regulations track the weighted center of a certain area (each robotic has a Voronoi tesseling), which allows the robot network to determine why the weight function is near the boundary of the map. It is made by means of the robot after several border paths. The disaster site is a non-convex zone, where the direct calculation of the Euclidean distance can be mistaken. This defines the distance between the two factors in an immediate line after which the rate of change is measured. (e.g. Shortest direction border). To maintain flexibility and connectivity.

In view of task optimization, the authors have defined a technique to assign robotics the most important challenge whenever they are. This project is adaptable to various circumstances. In addition, the author has described a cost feature which can be reduced, mainly by time. Consequently, any task that can achieve the price decrease value will probably be considered an ideal challenge. A few duties along with delivery: and evacuation. 10 people randomly allocated within the correct half and a random reputation (4 people could have been suffering, 3 victim s could not go around and 2 deaths) with the intention of validating the proposed method. Scheduled at five km / h speed. The authors have shown that the robots have behaved exactly as the rules say when simulating the proposed technology.

On the basis of criteria, the number of robots in their experiments used, dimensions of the area of application, flexibility of algorithms to enable robots to move flexibly with no constraints (i.e. communication constraints). Table 1 presents comparisons between them following the review of the related works. In addition to important factors determining the algorithm's efficiency. For example, the number of robots reaching the victim and the time (delay) required.

Table1. Comparison Between Meta-Heuristic Algorithm.

Used algorithm	Number of robots	Area	Flexibility	Average delay	Number of robots reached victim	Communication
RDPSO	15	20×10 m	Flexible	360 Sec	3	ZigBee802.15.4 wireless protocol
Algorithm1 Wave Swarm	36	6 × 6 cm	Not flexible	720 Sec	No results were shown	-
Novel hybrid algorithm (IPSO-IGSA)	6	232.79 cm	Not flexible	610.221Sec	2	Motorola 68331 25MHz microprocessor
A Bacteria Chemotaxis Algorithm for Target Search and Trapping	26	300×300 cm	Flexible	470Sec	15	-

3. LOCALIZATION ALGORITHM

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

The ICA is part of the swarm intelligence hypothesis, which comes from the field of human social advancement. It was first acquainted with non - stop improvement problems in 2007. However, since late, it has been widely linked to discreet advancement issues [28].

In the proposed algorithm we did not use the penalty-reward rules used by the RDPSO algorithm, which its basic work during the search in the environment, at some point the robots can not improve the cost function, so the algorithm works by swarming the robot and cutting communication with other robots. The proposed algorithm solved the problem of local best by the Do Revolution which was not used in other algorithms (RDPSO and DPSO), which works while searching for robots in the environment, the algorithm minimizes the cost function, at some stage the algorithm stops the functioning of the minimization function and the robots think that it has reached the victim which it did not, then the robots think it has reached the victim .

In the modified algorithm we have used Mean Square Error as a cost function we used the following equations along both x and y axis's

$$\boldsymbol{MSE} = \frac{1}{n}\sum_{i=1}^{n}(Y_{i-\hat{Y}_i}) \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (1)$$

$$\boldsymbol{MSE} = \frac{1}{n}\sum_{i=1}^{n}(x_{i-\hat{x}_i}) \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (2)$$

Where: $\hat{x}_i$ and $\hat{y}_i$ Represents the coordinates of both robots and the victim .

x_i and y_i: In order to calculate these values, random numbers are to be used. If the answer was less than 0.5 then the first cell in the row will be filled by 0, and if the answer was larger than 0.5 then the first cell in the row will be filled by 1. Then calculate the values resulting from the equation of mean Square error and continue the process for each robot in the environment until the end of the variables X and Y see Fig. 1. We have also used the shortest path functions which make each robot identifies its position in the environment and how far it is from other robots, this help robot restricts the area to determine the victim 's location, where, when communication between robots exists, and each robot searches in a place, the other robots shall look in other places which help therefore, in reaching the victim at the shortest time possible.

```
clc
clear all
N=20; % Number of particles
Max_iteration=30; % Maximum number of iterations
nVar=10;% Number of Decision Variables
load('v1.mat');
x1= xy0(1); % position
y1= xy0(2); % position
 CostFunction=@(x,x1) ICAc1(x,x1);
 CostFunctiony=@(y,y1) ICAc2(y,y1);
for i=1:N
    for j=1:nVar
        if rand<=0.5
            X(j,i)=0;
            Y(j,i)=0;
        else
            X(j,i)=1;
            Y(j,i)=1;
        end
    end
end
      for i=1:N %Calculate all the objective values first
%       X(:,i)'
      Fitness(1,i)=CostFunction(X(:,i)',x1)
%       Y(:,i)'
      Fitnessy(1,i)=CostFunctiony(Y(:,i)',y1)
        end
```

Figure 1: Mathlab Code For Calculating The Fitness Value

If the answer was close to zero or zero, that means that the robot is very close to the victim or has reached the victim . In this case, when the robot reaches a predetermined point, we use the two-point equation to calculate the distance between the robot and the victim . So, we can know how many steps the robot can take to reach the victim . The second case involves the robot moving away from the victim if the answer to the equation was close to 1 or 1. As for the first step of the robots '

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

movement, the equivalence probability methods are used to calculate the next region's position as follows:

$$p_{swarm}=| \ E_{I/MAX_{1\le J\le N_{strongest\ swarm}}} \ E_J \ | \quad ...(4)$$

Wheres : p_{swarm}: the probability of next position

The robot that has the lowest cost function than the other robots, is considered to be in the best position. To calculate the best position we use the following equation:

$$p^{Robot} = p^R + \beta * \xi * (p^{Strogest\ swarm} - p^R)$$

Where: p^{Robot}: new position of robot, β: is aparameter with a default value, ξ: is a matrix of 1 × n whose values are random values between 0 and 1, $p^{Strogest\ swarm}$: Position of the strongest swarm, p^R: Current Position of robot.

Accordingly, we change the place of the robots from swarm to another powerful swarm of robots to reach the best solution (reach the victim) and then calculate the total cost for each swarm as follow:

$$T.C._n = Cost(swarm_n) + \xi mean\{Cost(robots\ of\ swarm_n)\} \quad .....(6)$$

We add, obstacle avoidance and communication in order to adapt the ICA algorithm to work efficiently in a robot environment.

$$f(x) = cost(x_i) + \xi_1 cost\ (xg_i) + \xi_2 cost(xm_i) + \xi_3 cost(xo_i) \quad .....(3)$$

Where:

x_i : is the current position of the robot.

xg_i : is the best solution obtained by an entire swarm so far the begging of iteration until this time. xm_i : is the communication protocol used by this robot which the received signal from the entail.

xo_i : is the value of the obstacle avoidance sensor in our case ultrasonic sensor.

ξ_1, ξ_2, ξ_3 : random values constant between 0 and 1.

The advantage of using the shortest path equation is that each robot knows its position on all robots and this facilitates the research procedure so that the robot knows that this searching area has been conducted and moves to search elsewhere and when the robot approaches the victim , distance equation is used and the number of steps and distances are recorded. The divide between thex, y-plane. In the Cartesian direction, if there are two points in Euclidean n-space where p= s (p1, p2,..., pn) and q= s[q1, q2,..., qn], at that point there is a separation of p from p to q, or q to p of Pythagorean formula[29]:

$$d(p,q) = d(q,p) = \sqrt{(q_1 - p_1)^2 - (q_2 - p_2)^2 + \cdots + (q_n - p_n)^2} \quad .....$$

The result of the mean square error equation was used in determining the location of the victim . If the result was close to 0 that means the robot is close to the victim , and if the result was close to 1, that means the robot is getting away from the victim .

4. THE MRSIM (MULTI-ROBOT SIMULATOR)

This section includes detailed MRSim (Multi - Robotic Simulation) explanations, the way in which it is functioning, the virtual cartograph, the co - ordinating system, the cinematic model and the sensory system and the complete dissection and evaluation of test results.

This section contains a detailed explanation of the MRSim (Multi-Robot Simulator) and the way of its operation, virtual map, coordinate system, kinematic model, and sensory system, as well as, a complete dissection of the experimental results and evaluation.

The MRSim (Multi-Robot Simulator) is an extension to the independent SIMROBOT toolbox, established in 2001 for MatLab 5. It is used in virtual environments to simulate the behavior of many mobile robots. Compared to its predecessor SIMROBOT, MRSim offers two significant contributions.

It is suitable for multi-robot applications. Despite the remarkable characteristics of SIMROBOT for mobile robotics, it has numerous inconveniences for multi-robot applications. MRSim was therefore mainly built to enable users to create multi-robot applications like hop communications. Each MRSim robot can be fitted with several simulated sensors, and can be controlled just like SIMROBOT with its own control algorithm.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

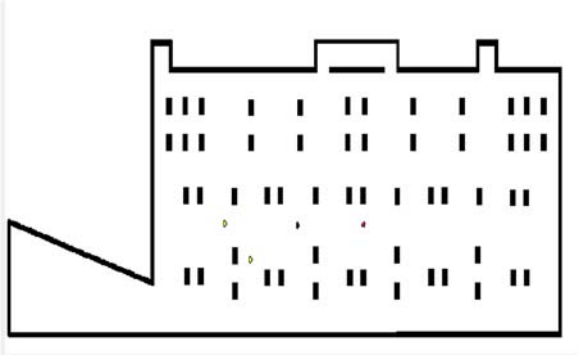

Figure 2. Mrsim Swarm Robotics Environment

A sample environment in which a swarm of robots perform their tasks is shown in Fig 2. The white areas are empty and the black objects are hindrances. The black and yellow robots are the victim s and the red triangle. The users are manually positioned. The victim 's location is not known to either the simulator or the algorithms. When carrying out the approach proposed, robots should be in a position to reach the victim in a shorter time.

Use any Graphics Editor (e.g. Paint) to create a new map in the virtual setting to save the picture as a 1-bit (black / white) bit map. The points drawn are shown as strong barriers (walls). The virtual environment map is represented internally by the uint8 matrix (unsigned 8-bit integer). Consequently, the maximum value of each element of the matrix is 255, the minimum value being 0.

To create a new map of the virtual environment, use any graphics editor (e.g., Paint), which can save the image as a 1-bit (black/white) bitmap (usually known as a monochromatic image). The drawn points will be represented as solid obstacles (walls). The virtual environment map is internally represented by *uint8* (unsigned 8-bit integer) matrix. The maximum value of each of the matrix' elements is therefore 255, and the minimum is 0.

The customizations of the coordinate system in robots are shown in Fig. 3. The robot is in default mode, i.e. (x, y) = (0, 0) and the address = θ = 0 °. Besides solid barriers, robots are also detected by sensors. This is guaranteed by placing each robot in the default environment map as a constraint at each step. In addition, the sensor allows distinguishing different robots based on their

Virtual sensors placed on the robot are given in this coordinate system. When you create a user-defined shape for a robot, it may be useful to monitor that number.

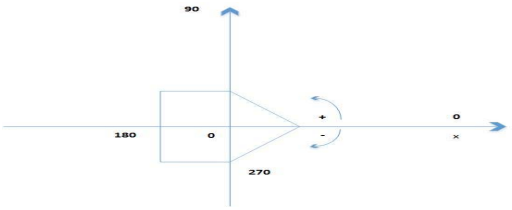

Figure 3. The coordinate system of the robot Kinematic Model

The emulator presumes wheeled robots, i.e. Robots with different driving systems (a traditional wheelchair model). This type of chassis offers two degrees of freedom of movement (DOF) with two traditional non-steering wheels.

The identification of obstacles in the emulation depends on whether there is a non-zero element (indicating a constraint or a robot) at a particular location in the matrix (default environment map).

Emulated sensors use Bresenham's algorithm [30]. This routine was written in C and converted to a dynamically linked library (*DLL*) - (the *mex* file)

The scanning angle overlaps with the rays. A number of rays are given by the resolution parameter. The coordinates of dots on these rays are calculated using the algorithm already mentioned. If the beam collides with a block (i.e., If there is a non-zero element in the matrix), the end point of that beam is stored and a new beam calculation is initiated. For the ultrasonic sensor, the nearest point is selected and its distance is returned with the obstacle number (see table 2). If there are no obstructions within the sensor range, the value 999999 is returned. For the laser scanner, distances are returned to all obstructions detected within their range and accuracy with block numbers.

unique identifier number (ID). This number is generated by the EDITOR when adding a new robot. When the robot is deleted, the robots are remembered in the list. If a robot collides with

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

another obstacle or robot, it stops, that is, its flag crashed is set to 1.

Table 2. Coefficients values of different algorithms

Coefficients	RDICA	RDPSO	DPSO
fract	0.632	0.632	0.6
pc	0.8	0.8	0.25
ps	0.04	0.04	0.03
pobs	0.2	0.2	1.2
pcomm	0.4	0.4	0.65
RangeComm	200	200	150

Where: fract : fractional coefficient, pc: cognitive weight, Ps: social weight, pobs: obstacle susceptibility weight, pcomm: communication constraint weight and RangeComm: maximum communication range.

5. EXPERIMENTAL RESULTS

In this part of the study, a number of experiments were carried out in the robotics environment. The first experiment consisted of 5 robots and one victim , the second experiment consisted of 10 robots and one victim , the third experiment consisted of 15 robots and a victim , the fourth experiment consists of 20 robots and one victim , and the fifth experiment consists of 25 robots and one victim on the software Matlab R2017b and the simulator user is MRSim_vbeta. In terms of user ready specifications were INTEL (R) CORE (TM) i3-4005U CPU@ 1.7GHz 1.70GHz

The operating system used is Windows 8.1 Enterprise 64 bit, and the screen resolution is 1366*768. The screen resolution is very important in our work because it varies from one device to another. The area of our experiments on the simulator was of (26.33*9.21 cm) wide, noteing that there were 57 obsticals counted around 30% of the total area, and we have considered 750 simulated trials for the all algorithms under study, considering 50 trails for each scenario, and the cost function that has been used was the mean square error in addition to, shortest path equation, human dedction sensor and altra sonic sensor, as well as, we have used Mobile Ad Hoc Network (MANET) as a communication protocol. We set a condition if the distance between the robot and the victim 50 points or less (each 28.346 point equals to 1 cm), this process is recorded by the arrival of the robot to the victim , and this distance varies according to the screen resolution and extent of the sensor used (150 degrees).

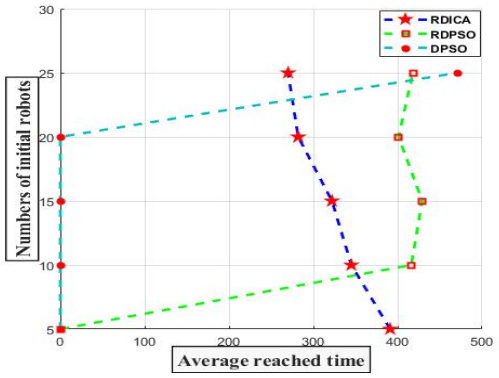

Figure 4. The number of initial robots and the average reached the time in RDICA, RDPSO and DPSO algorithms.

Fig 4. Represents the number of robots in the five experiments, the average reached time in each of the RDICA, RDPSO and DPSO algorithms. For the proposed RDICA algorithm, the average reached the time in the five experiments was 322 Sec, while, it was 416 Sec for the RDPSO, and 471 Sec for the DPSO. Results show that the average reached the time of robots to the victim of the RDICA algorithm is less than the

RDPSO and DPSO algorithms.

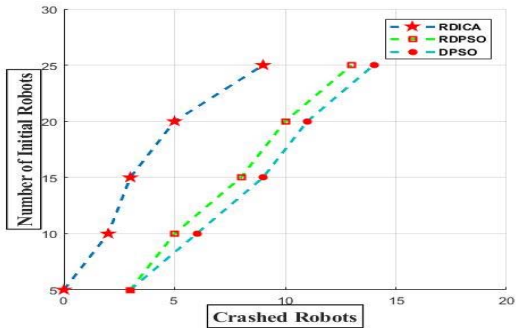

Figure 5. The Number Of Initial Robots And The Crashed Robots In RDICA, RDPSO And DPSO Algorithms.

Fig 5. Represents the number of robots in the five experiments and the number of crashed robots in each of the RDICA, RDPSO and DPSO algorithms. For the proposed RDICA algorithm, the average of crashed robots in the five experiments was 22%, while it was off 53% RDPSO algorithm, and 58.2% in DPSO

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

algorithm. Results show that the crashed robots in the RDICA algorithm are less than the RDPSO and DPSO algorithms.

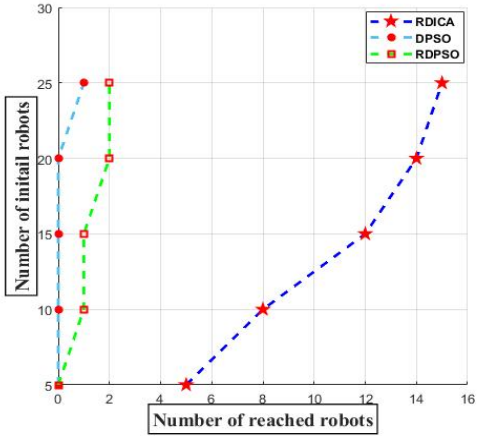

Figure 6. The number of initial robots and the number of reached robots in RDICA, RDPSO and DPSO algorithms.

Fig 6. Represents the number of robots in the five experiments and the arrival of robots in each experiment for the RDICA, RDPSO and DPSO algorithms. For the proposed RDICA algorithm, the average number of arrived robots in the five experiments was 78%, while it was 6.93% in RDPSO algorithm, and 0.8% in DPSO algorithm. The results show that the number of arrived robots during the search for the victim in RDICA algorithm is more than the RDPSO and DPSO algorithms.

6. DISCUSSION OF RESULTS

Five experiments were carried out, each experiment containing different numbers of robots (5,10,15,20,25) in order to determine the best algorithm in which we can locate the victim s at the highest accuracy, and thus, get as many robots as possible to reach victim s at the shortest time and at the lowest crash rate. It is clear from the figures that the RDICA algorithm had an average arrival rate in the five experiments 78%, while it was 6.93% in the RDPSO algorithm and 0.8% in the DPSO algorithm. The main reason for the arrival of more robots in the RDICA algorithm is the high accuracy ratio, which was extracted after comparing 10 algorithms on 12 benchmarks and on a search area - the dimensions (10,20,30). The higher the accuracy rate, the highest of determining the location of the victim and the larger number of arrivals. The crash rate in the RDICA algorithm was 22%, while 53% in the RDPSO algorithm and 58.2% in the DPSO algorithm. The main reason is that the RDICA algorithm has used the Do Revolution feature, which helps the robot avoid the local minima problem that other algorithms have suffered. During the process of search for victim algorithm operates the minimization of the cost function, in some stage of search the algorithm stops the work of minimization of the cost function and the robot would think that it's arrived the victim , the Do Revolution feature work on moving the robot from its place to a new place, if the new place is better the robot moves to it because this place inevitably leads to the victim and if otherwise, the robot remains in the original place.

The average reached time in the RDICA algorithm was the least with 322 sec, where it was 416 sec in the RDPSO algorithm and 471 sec in the DPSO algorithm, and this ratio was measured based on the number of arrivals only, this ratio calculated by using the shortest path equations, including Euclidean Distance, which helps robots to locate other robots in the environment, so that each robot sends a message contains many details, including the position of the robot, this helps robots that, this area has been searched and robot moves to search elsewhere, this work leads to reach the victim in the shortest time possible.

7. CONCLUSIONS

To help solve the problem of search and rescue using swarm robotics, we have studied several algorithms that contribute to solve this problem, the RDPSO and DPSO Algorithms, which found that they suffer from several problems, including local minima, which is one of the most important factors affecting the optimization algorithm, in which it illusions the robot in a certain stage that it reached the desired goal while it did not, this problem affects the searching robots in the robotics environment, robots cannot be able to determine the location of the victim accurately, due to low accuracy rate. Algorithms also suffer from the collision problem, which may be caused by either the robot has smashed with an obstacle or with another robot in the environment, in addition to the problem of speed of robot in getting access to the victim .

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Based on the discussion of results and experiments conducted by using (ICA) algorithm, which gives the optimal solution and leads to the place of the victim in order to reach an optimal algorithm that contributes in solving the problems that algorithms suffer from in robotics environment, and after comparing several different algorithms, we have found out that the RDICA algorithm that is adopted in this study contributed in solving these problems by increasing the number of reached robots, which was at an average of 78%, while it was 6.93% in the RDPSO algorithm and 0.8% in the DPSO algorithm. Also, the average of crashed robots was reduced and the victim 's location was determined at a higher accuracy, where it was 22% in the RDICA algorithm, while it was off 53% in the RDPSO algorithm, and 58.2% in the DPSO algorithm. In addition to the reduction of the average reached time, which was 322 Steps in the RDICA algorithm, 416 steps in the RDPSO algorithm and 471 steps in the DPSO algorithm.

We have also used a feature called Do Revolution, through this feature we have been able to prevent the robot from falling into the problem of local minima and increase the accuracy in locating the victim that RDPSO and DPSO algorithm has suffered from, which has reflected in the robotics environment, as when robots searching using these two algorithms, cannot determine the place of the victim precisely because of the problem of local minima, as well as, its low accuracy ratio. We also concluded that, after the implementation of several experiments, the number of five robots is the best number of searches that we must adhere to because the algorithm with this number eliminated the problem of the collision, where the arrival rate was 95%.

REFERENCES

[1] V. Trianni, Evolutionary swarm robotics: evolving self-organising behaviours in groups of autonomous robots vol. 108: Springer, 2008.
[2] N. S. Rao, S. Kareti, W. Shi, and S. S. Iyengar, "Robot navigation in unknown terrains: Introductory survey of non-heuristic algorithms," Citeseer1993.
[3] J. Pearl, "Heuristics: intelligent search strategies for computer problem solving," 1984.
[4] L. Bianchi, M. Dorigo, L. M. Gambardella, and W. J. Gutjahr, "A survey on metaheuristics for stochastic combinatorial optimization," Natural Computing, vol. 8, pp. 239-287, 2009.
[5] M. S. Couceiro, A. Fernandes, R. P. Rocha, and N. M. Ferreira, "Understanding the communication complexity of the robotic Darwinian PSO," Robotica, vol. 33, pp. 157-180, 2015.
[6] H. Mandviya, A. Patil, T. Nanavare, S. Bhoite, S. M. Ramteke, and P. ABMSP's APCOER, "3D HCI: An Human Computer Interface System using 3D Leap Motion Camera," International Journal of Engineering Science, vol. 10647, 2017.
[7] M. Rubenstein, C. Ahler, and R. Nagpal, "Kilobot: A low cost scalable robot system for collective behaviors," in Robotics and Automation (ICRA), 2012 IEEE International Conference on, 2012, pp. 3293-3298.
[8] G. E. Marchant, B. Allenby, R. Arkin, E. T. Barrett, J. Borenstein, L. M. Gaudet, et al., "International governance of autonomous military robots," 2011.
[9] P. Kanade, H. Somani, V. Motwani, and S. Tiwari, "Distributed Computing for Co-Operative Swarms," International Journal of Engineering and Management Research (IJEMR), vol. 5, pp. 138-142, 2015.
[10] M. Saska, J. Langr, and L. Přeučil, "Plume tracking by a self-stabilized group of micro aerial vehicles," in International Workshop on Modelling and Simulation for Autonomous Systems, 2014, pp. 44-55.
[11] M. Saska, V. Vonásek, T. Krajník, and L. Přeučil, "Coordination and navigation of heterogeneous MAV–UGV formations localized by a 'hawk-eye'-like approach under a model predictive control scheme," The International Journal of Robotics Research, vol. 33, pp. 1393-1412, 2014.
[12] H. Kwon and D. J. Pack, "A robust mobile target localization method for cooperative unmanned aerial vehicles using sensor fusion quality," Journal of Intelligent & Robotic Systems, vol. 65, pp. 479-493, 2012.
[13] H. Hamann, Swarm Robotics: A Formal Approach: Springer, 2018.
[14] G. Beni and J. Wang, "Swarm intelligence in cellular robotic systems," in Robots and Biological Systems: Towards a New Bionics?, ed: Springer, 1993, pp. 703-712.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

[15] H. Gintis, "Clash of the Titans: A Review of Edward O. Wilson, The Social Conquest of Earth (WW Norton, 2012)," 2012.
[16] R. Bouffanais, Design and control of swarm dynamics: Springer, 2016.
[17] H. Hamann, Y. Khaluf, J. Botev, M. Divband Soorati, E. Ferrante, O. Kosak, et al., "Hybrid societies: challenges and perspectives in the design of collective behavior in self-organizing systems," Frontiers in Robotics and AI, vol. 3, p. 14, 2016.
[18] S. Kernbach, Handbook of collective robotics: fundamentals and challenges: CRC Press, 2013.
[19] P. Das, H. S. Behera, and B. K. Panigrahi, "A hybridization of an improved particle swarm optimization and gravitational search algorithm for multi-robot path planning," Swarm and Evolutionary Computation, vol. 28, pp. 14-28, 2016.
[20] M. S. Couceiro, R. P. Rocha, and N. M. Ferreira, "A novel multi-robot exploration approach based on particle swarm optimization algorithms," in Safety, Security, and Rescue Robotics (SSRR), 2011 IEEE International Symposium on, 2011, pp. 327-332.
[21] M. S. Couceiro, R. P. Rocha, and F. M. Martins, "Towards a predictive model of an evolutionary swarm robotics algorithm," in Evolutionary Computation (CEC), 2015 IEEE Congress on, 2015, pp. 2090-2096.
[22] L. S. Junior and N. Nedjah, "Efficient Strategy for Collective Navigation Control in Swarm Robotics," Procedia Computer Science, vol. 80, pp. 814-823, 2016.
[23] W. Li, Y. Bi, X. Zhu, C.-a. Yuan, and X.-b. Zhang, "Hybrid swarm intelligent parallel algorithm research based on multi-core clusters," Microprocessors and Microsystems, vol. 47, pp. 151-160, 2016.
[24] Y. Gabriely and E. Rimon, "Spanning-tree based coverage of continuous areas by a mobile robot," Annals of mathematics and artificial intelligence, vol. 31, pp. 77-98, 2001.
[25] Y. Stergiopoulos and A. Tzes, "Spatially distributed area coverage optimisation in mobile robotic networks with arbitrary convex anisotropic patterns," Automatica, vol. 49, pp. 232-237, 2013.
[26] B. Yang, Y. Ding, Y. Jin, and K. Hao, "Self-organized swarm robot for target search and trapping inspired by bacterial chemotaxis," Robotics and Autonomous Systems, vol. 72, pp. 83-92, 2015.
[27] D. Yanguas-Rojas, G. A. Cardona, J. Ramirez-Rugeles, and E. Mojica-Nava, "Victim s search, identification, and evacuation with heterogeneous robot networks for search and rescue," in Automatic Control (CCAC), 2017 IEEE 3rd Colombian Conference on, 2017, pp. 1-6.
[28] E. Atashpaz-Gargari and C. Lucas, "Imperialist competitive algorithm: an algorithm for optimization inspired by imperialistic competition," in Evolutionary computation, 2007. CEC 2007. IEEE Congress on, 2007, pp. 4661-4667.
[29] H. Anton, Elementary Linear Algebra, Binder Ready Version: John Wiley & Sons, 2013.
[30] X. Wu and J. G. Rokne, "Double-step incremental generation of lines and circles," Computer Vision, Graphics, and Image Processing, vol. 37, pp. 331-344, 1987.