

# RoboCupRescue 2021 TDP Agent Simulation MRL (Iran)

Mahdi Chatri<sup>1</sup>, AmirHoseein Moshfeghi<sup>1</sup>

<sup>1</sup>Islamic Azad University of Qazvin, Electrical, IT & Computer Sciences Department, Mechatronics Research Laboratory,  
Qazvin, Iran

{chatrimahdi, amirh.moshfeghi}@gmail.com

**Abstract.** This paper presents recent developments and research studies of the MRL-RSL team for the upcoming Robocup 2021 event. Last year, we mainly focused on the police force, ambulance team, fire brigade agents' strategies, and improving algorithms such as path planning and clustering, making the MRL-RSL team champion of Robocup 2019 Rescue Agent Simulation league. This year, we aim to improve search strategy through an Artificial bee colony (ABC) search algorithm.

**Keywords:** Robocup, Rescue Simulation, Strategy, Search

## 1 Introduction

In the 2019 Robocup Competition, we focused on the fire brigade and ambulance team strategists and implementing some mentioned approaches. Also, this year, we focus on the search strategies, then trying to improve them. Due to the recent changes on ADF applied to some agents and strategies, we need to change the search algorithm besides the optimized task allocation of the fire brigade agents, so they would not search for fire or try to distinguish fire. This year, we tend to optimize our agents' coordination and do our best to fit into new changes of ADF as our last time approach; MRL uses various methods for searching, like random search. However, we released that these methods can be more powerful if we combine them with a new algorithm called IABC [2]. In the following sections, we introduce our new search strategy briefly.

## 2 Modules

### 2.1 Clustering

For the clustering task, a clustering method based on the previous year has been applied to the system based on k-means algorithm[1].

#### 2.1.1 Map clustering

In the map clustering task, an extended form of the K-means algorithm has been implemented to help the decision-making system on every agent. The similarity level in a usual way in two-dimensional space is considered based on the Euclidean distance. However, here we use the combination of the density of points and the significance of buildings and the probability of citizens and agents in each region strategy, then assign with Hungarian algorithm.

### 3 Strategy

#### 3.1 Ambulance

Proportional to the number of available agents in the scenario and creating partitions, we assign each agent to a partition, and agents began searching and rescuing their partitions. However, sometimes even though assignments are very efficient, the agents have to travel a long way to get to their partition (Figure.1), which reduces their utilization factor and efficiency. To avoid this, we have added a new strategy called "Sticky Move," based on potential field motion planning. If an agent detects a civilian closer than a specific threshold to the path towards the target partition, the agent tries to rescue the civilian before heading towards the target partition. However, many ambulances are trying to rescue the same civilian; to avoid such conditions, we utilize an algorithm based on the Monte Carlo method [6] to estimate the remaining time until a civilian dies and therefore calculate the optimum number of ambulances needed to rescue that civilian. We assign only as many as-needed ambulances to rescue tasks and let the rest go and reach their target partition.

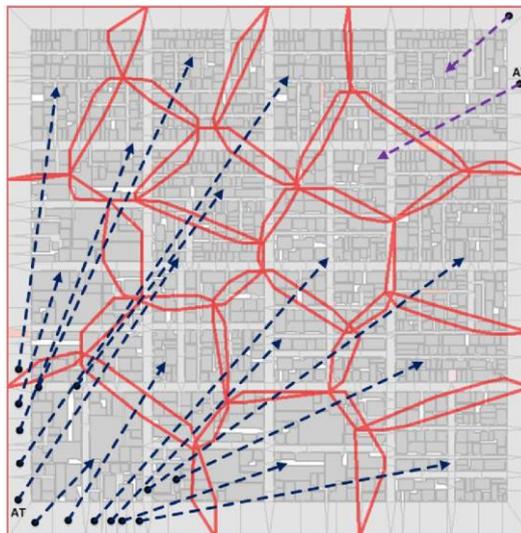


Figure 1. Aggregation of agents in specific zones forces them to travel a long way to get to their partition

#### 3.2 Police force

Since police agents directly impact the other agent's performance, most MRL attention was on police agents' coordination and cooperation with other RoboCup 2016. One of the problems we are facing from the results, a new straightforward method dentate blockades that trap agents. This problem reduces the agent's performance drastically.

In RoboCup 2014, we used a guideline-based strategy to cover the police force challenge. In this strategy, police agents smoothly clear the blocked roads alongside the guidelines without any residues. As a result, the agents can move faster and without getting stuck.

Another improvement is prioritizing police clear tasks based on 1) traffic; 2)the importance of the buildings and the importance of the partition. It makes a satisfactory improvement on police and other agents due to the changing of priorities. We have also changed our strategy to unlock trapped agents with higher priority.

### 3.3 Fire Brigade

Based on the gained experience from the previous RoboCup, we believe that the behavior of the fire brigade agents can improve through a simple yet effective idea inspired by the Artificial Bee Colony (ABC) optimization algorithm. However, fire brigades do not search for fire, and they do not have to extinguish fire anymore. Their primary focus is to rescue civilians, and we committed some changes to make this problem. In the following sub-section, we briefly recall the ABC algorithm, showing how we apply a modified version of this algorithm to help fire brigade agents.

#### 3.3.1 Artificial Bee Colony (ABC)

A set of honey bees, called a swarm, can accomplish tasks through social cooperation. In the ABC algorithm [4], there are three types of bees: employed bees, onlooker bees, and scout bees. The employed bees search for food around the food source in their memory; meanwhile, they share the information of these food sources with the onlooker bees. The onlooker bees tend to select good food sources from those found by the employed bees. The food source with higher quality (fitness) will significantly be selected by the onlooker bees rather than lower quality. The scout bees have been translated from a few employed bees, which abandon their food sources and search for new ones.

#### 3.3.2 Optimized search method based on ABC

The standard ABC shows slow convergence speed because of its poor exploitation ability [2]. To tackle this problem, many several different ABC variants have been proposed. In GABC, the information of  $Gbest$  is introduced into the solution search equation. It is expected that the  $Gbest$  can guide the search and enhance the exploitation. In GABC, the improved solution search model is defined by [4].

$$v_{i,j} = x_{i,j} + \phi_{i,j} \cdot (x_{i,j} - x_{k,j}) + \psi_{i,j} \cdot (Gbest_j - x_{i,j}) \quad (1)$$

Where  $j \in [1, D]$  is a randomly selects dimensional index. The parameter of  $X_k$  is a random solution for  $k \in [1, N]$  and the weight of  $\phi_{i,j}$  assings randomly between  $-1.0$  and  $1.0$ .  $Gbest$  is the global best solution found so far. The weight  $\psi_{i,j}$  is a generated random number in the range  $[0, C]$ , where  $C$  is a pre defined value. By the suggestions [4],  $C = 1.5$  is a good setting.

In GABC, the employed bees use Eq. (1) to search for new solutions. The onlooker bees compute the probability of each solution and select some better solutions to conduct a further search. Like the employed bees, the onlooker bees also use the same model Eq. (1) to search for new solutions. For the onlooker bees, the selected solutions have good fitness values. If we use the same model to generate new solutions, the search efficiency will not be high.

In our new approach, we try to construct two search phases: exploration and exploitation. The employed bees focus on exploration, and the onlooker bees concentrate on exploitation. To implement our approach, we design a new solution search model for the exploitation phase (the onlooker bees) [2].

$$v_{i,j} = a_1 \cdot x_{i,j} + a_2 \cdot Gbest + a_3 \cdot (x_{k,j} - x_{i,j}) \quad (2)$$

where  $x_i$  is the current solution,  $Gbest$  is the global best solution found so far, and  $x_k$  is a randomly selected solution ( $i \neq k$ ). The weights  $a_1$ ,  $a_2$  and  $a_3$  are three random values and  $a_1 + a_2 + a_3 = 1$ . We firstly generate three random numbers  $a_1$ ,  $a_2$  and  $a_3$  in the range  $[0, 1]$ . Then, we use Eqs. (3), (4), and (5) to generate new weights.

$$a_1 = \frac{a_1}{a_1 + a_2 + a_3} \quad (3)$$

$$a_2 = \frac{a_2}{a_1 + a_2 + a_3} \quad (4)$$

$$a_3 = \frac{a_3}{a_1 + a_2 + a_3} \quad (5)$$

In [5], Akay and Karaboga pointed out that changing only one dimension of parent solutions in ABC may slow convergence speed. A new parameter  $MR$  was introduced to control the number of updating dimensions [5]. In this paper, the  $MR$  parameter has been employed in our approach. For each food source (solution), we check all dimensions  $j$ , and  $j = 1, 2, \dots, D$ . If  $rand_j$  then update  $v_{i,j}$ . According to Eq. (2), then the new search model is rewritten by:

$$v_{i,j} = \begin{cases} v_{i,j}, & \text{if } rand_j < MR \\ v_{i,j}, & \text{otherwise} \end{cases} \quad (6)$$

where  $MR$  is a predefined probability parameter,  $rand_j$  is a random value generated in the range  $[0,1]$  for the  $j$ th dimension, and  $v_{i,j}$  is defined by Eq. (2).

The main processes of our new approach IABC are given below :

*Step 1.* Randomly generate  $N$  solutions, and compute their fitness values. Set  $FES = N$  ( $FES$  is the number of fitness evaluations).

*Step 2.* For each solution  $X_i$ , a new solution  $V_i$  is generated by Eq. (1). Compute the fitness value of  $V_i$  and  $FES = FES + 1$ . The better one between  $X_i$  with  $V_i$  is used as the new  $X_i$ . If  $f(V_i) < f(X_i)$ , then  $trial_i = 0$ ; otherwise  $trial_i = trial_i + 1$ .

*Step 3.* Calculate the probability  $p_i$  according to the following equation:

$$p_i = \frac{fit_i}{\sum_{i=1}^N fit_i} \quad (7)$$

if  $X_i$  is selected, then a new solution  $V_i$  is created by Eq. (6). Compute the fitness value of  $V_i$  and  $FES = FES + 1$ . The better one between  $X_i$  with  $V_i$  is used as the new  $X_i$ . If  $f(V_i) < f(X_i)$ , then  $trial_i = 0$ ; otherwise  $trial_i = trial_i + 1$ .

*Step 4.* If  $\max\{trial_i\} > limit$ , the solution  $X_i$  is updated by the following equation.

$$x_{i,j} = low_j + rand_j \cdot (up_j - low_j) \quad (8)$$

where  $[low_j, up_j]$  is the constraint box, and  $rand_j$  is a random value generated in the range  $[0,1]$ .

*Step 5.* Set  $t = t + 1$ . If  $FES < MaxFES$ , then go to Step 2; otherwise, stop the algorithm ( $MaxFES$  is the maximum value of  $FES$ ).

Compared to the standard ABC, IABC only changes the search model of onlookers bees. So, both IABC and ABC have the same computational time complexity.

### 3.3.3 Fire brigades' task allocation

We will use this algorithm for assigning tasks to the fire brigade agents, the same as the ABC algorithm. As mentioned above, we have three types of bees in the ABC algorithm: onlookers, employed, and scouts. For this purpose, we assume Fire brigade agents same as these three types. Here onlookers are responsible for deploying scouts and employed agents.

We use a random number to spread these types of agents into their clusters. For example, we consider a cluster with ten fire brigades. We divide these agents into two sections, and scouts are the fire brigades; they are currently free and do not have any tasks, so according to our top section, these agents do a random search till they have to assign a task. Whenever a task comes, these agents become employed because the new task is essential. When a scout bee is searching its cluster and suddenly finds a civilian in a building or hear a voice, it stops its scouting job like searching, and it would become employed, then it tries to rescue a civilian after the job is accomplished, it can be a scout again and search its cluster till another prioritized task assigned to it. If the civilians are accessible also communication be is available, then they share the information with the onlooker. Otherwise, they will go to the refuge as an information

exchange center. The agent that is in the refuge will share the information with other agents. Then agents can be obtained as a source of information for the better decision and choose the better target.

## **4 New implementation of ambulance due to ADF changes**

As we mentioned before, we are implementing police force agents for technical challenge competition; here we describe some details about our implementation.

### **4.1 Tactics**

In ADF there is a new concept called Tactics that has a specific definition for each type of agent. We have implemented the tactics for police force agents in a class called TacticsAmbulance. This class includes initialize, precompute, resume, preparation, and think methods that each are responsible for a specific task like decision making, routing, clustering.

### **4.2 Algorithm Module**

#### **4.2.1 Path planning**

Using a path planner is essential for finding optimum paths in environments like rescue simulation platform. According to the provided sample, the ADF has provided one implementation for path planning as an example; we have implemented our own cached path planning algorithm.

#### **4.2.2 Clustering**

In scenarios that include vast segments of urban environment running, some algorithms are very resource-consuming and inefficient, so we need to use clustering algorithms to simplify the problem and run our algorithms more efficiently. We are using a K-means algorithm in the algorithm, clustering directory and enhancing it to satisfy our needs.

### **4.3 Action**

The Ambulance behavior changes based on having a target or not. For example, if an ambulance does not find any civilian, it will start searching the environment until it finds a civilian to rescue. Based on these behaviors, the civilian has two main functionalities, load and search.

### **4.4 Search**

If an ambulance agent does not find any civilian on the map, it will start the searching task. There is a class in the extraction directory that also includes the calculation method, which contains an optimum search algorithm for civilian search

## **5 Communications**

Communication has an essential role in helping an agent to take efficient actions. The world model of each agent is updated by communication and the information it is obtaining from its view. In each world model, all information that is not obtained from the agent's view is provided by communication. Consequently, this means that most information in an agent world model relies upon communication. An agent selects its actions based on the information from its world model only, making them utterly dependent on it. Given that there is a limit to the actions an agent can select from only the updated information that it sees, the additional information updated by communication becomes crucial. Thus, by improving the content of this communication, an agent can take more effective action. For the former method, each of the agents unified much information and judged it. In this way, the agent can make a quick and correct decision. However, this method has a low point. It is cooperation between agents using

communication. Conventionally, we implemented only the efficient action in the unit of an agent by using communication. In this method, however, to satisfy agent's demands, we make another agent cover those demands occasionally.

## 6 Results

As shown in Table 1, performance analysis has prepared with this year against our last project to prove our progress rather than previous years.

**Table 1.** Comparison of MRL new strategy and MRL latest implementations

	Paris 2	Berlin 2	Istanbul 3	VC 3
MRL 2019	73.20	146.68	220.50	135.41
MRL 2021	74	148	222.9	136.30

## 7 Conclusions

Based on the last changes on rules of this year, we made to the fire brigade agent's strategy and search: Fire brigade agents can work better by spreading wisely into the map, and change type depends on the situation for better decision making, their task assignment improved in a significant way, but there are some problems in communication that we are currently considering, besides we are doing our best to solve these minimal problems.

## 8 References

- [1]. Karypis, M.S.G., Kumar, V. and Steinbach, M., 2000, May. A comparison of document clustering techniques. In *TextMining Workshop at KDD2000 (May 2000)*.
- [2]. Cao, Y., Lu, Y., Pan, X. and Sun, N., 2019. An improved global best guided artificial bee colony algorithm for continuous optimization problems. *Cluster computing*, 22(2), pp.3011-3019.
- [3]. Karaboga, D., 2005. An idea based on honey bee swarm for numerical optimization (Vol. 200, pp. 1-10). *Technical report-tr06, Erciyes university, engineering faculty, computer engineering department*.
- [4]. Zhu, G. and Kwong, S., 2010. Gbest-guided artificial bee colony algorithm for numerical function optimization. *Applied mathematics and computation*, 217(7), pp.3166-3173.
- [5]. Akay, B. and Karaboga, D., 2012. A modified artificial bee colony algorithm for real-parameter optimization. *Information sciences*, 192, pp.120-142.
- [6]. Kroese, D. P.; Brereton, T.; Taimre, T.; Botev, Z. I. (2014). "Why the Monte Carlo method is so important today". *WIREs Comput Stat* 6: 386–392.