

# Space Exploration of Multi-agent Robotics via Genetic Algorithm

T. Ting, Kaiyu Wan, Ka Man, Sanghyuk Lee

► **To cite this version:**

T. Ting, Kaiyu Wan, Ka Man, Sanghyuk Lee. Space Exploration of Multi-agent Robotics via Genetic Algorithm. 9th International Conference on Network and Parallel Computing (NPC), Sep 2012, Gwangju, South Korea. pp.500-507, 10.1007/978-3-642-35606-3\_59 . hal-01551312

**HAL Id: hal-01551312**

**<https://hal.inria.fr/hal-01551312>**

Submitted on 30 Jun 2017

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



# Space Exploration of Multi-Agent Robotics via Genetic Algorithm

T.O. Ting<sup>1,\*</sup>, Kaiyu Wan<sup>2</sup>, K.L. Man<sup>2</sup>, Sanghyuk Lee<sup>1</sup>

<sup>1</sup>Dept. Electrical and Electronic Eng and <sup>2</sup>Dept. Computer Science and Software Eng,  
Xi'an Jiaotong-Liverpool University, Suzhou, China.

{toting, kaiyu.wan, ka.man, sanghyuk.lee}@xjtlu.edu.cn

**Abstract.** Robots play an important role in space exploration whereby the presence of human is almost impossible in some environments. Instead of using a robot, we incorporate a group of robots working together to achieve the definitive goal. Evolutionary algorithm, namely Genetic Algorithm is applied in the multi-agent robotics for space exploration. Hereby, the core focus of this paper is to study the effect of crossover rate upon the convergence of the exploration. As from our results, choosing the right parameter value is crucial for optimal coverage of the potential area.

**Keywords:** Genetic Algorithm, Multi-agent Robotics, Space Exploration

## 1 Introduction

An agent can be defined as a computational system that tries to fulfill a set of goals in a complex, dynamic environment [1]. It has sensors which provide information on the surrounding environment, and its actions are controlled by the agent's actuators. An agent can take upon many forms, depending upon its application and environment. Important properties that an agent possesses are autonomy, reactivity and proactiveness. Additional properties include being able to describe information states, deliberative states and motivational states. An agent is called autonomous if operates completely autonomously – that is, if it decides itself how to relate its sensory data to motor commands in such a way that its goals are attended to successfully [2]. An agent is said to be adaptive if it is able to improve its goal-achieving competence over time. Autonomous agents constitute a part of study in artificial intelligence.

Recently, growing interest has been shown in systems composing several interacting autonomous agents, instead of a single agent. They can be applicable in domains that cannot be handled by centralized systems. Nowadays, a solid platform of computer and network technology is available for realizing complex multi-agent systems. By having multiple robots cooperating, there are many advantages compared to using a single robot. The robots have a larger range of task domains, which can be seen through the wide range of applications discussed later. If the robots exchange infor-

mation with each other whenever they sense each other, they can localize themselves faster and more accurately. As there are many robots performing a single task to achieve a common goal, redundancy is introduced. Thus, it has higher fault tolerance than using a single-robot system. Being more robust, it can accomplish tasks more efficiently. As an example, in the year 2003, a mission called Beagle 2, attempting to explore planet Mars and collect soil samples from the planet. A spacecraft was sent with a single robot in it. This is a centralized system where the robot is controlled via wireless communication. The mission failed because the robot's antenna was damaged during landing, thus, no signal is received from the robot. Thus, if a decentralized multiple-robot system was used instead, this problem would not have occurred, since the robots can still perform work without one member. The only downfall is a slower speed as communications between agents take up an amount of precious time.

With multiple-robot systems, more robots would be needed and produced. This will greatly reduce the economic cost of a single robot, making it more marketable. Also, with multiple robots at a few different places, a wider area can be covered. This introduces distributed sensing and action, where the system can sense at one place and performs an action at another place. An example of a natural multi-agent society is an ant colony. Within this colony, you can see many different tasks being carried out by many ants, working together to achieve a single goal. You see as many as a few hundred ants carrying big chunks of food back to their nest, or attacking a predator much larger than themselves, for example a praying mantis. This is an inspiration for multi-agent robotic systems researches, knowing that this field yields many opportunities and benefits, as many applications unfold.

## **2 Evolutionary Robotics**

One critical goal in multi-agent robotic systems development is to design a distributed control infrastructure for the robots to perform their tasks over a problem-solving period without human supervision. They must be capable of dealing with dynamic changes occurring over time, such as unpredictable changes in environment, or even variations in their own performance capabilities. To improve a robot's likelihood of survival in a changing environment, learning, evolution and adaptation are important. They help by inducing appropriate competition or cooperation with other agents/robots. Learning is a strategy for an agent to adapt to its environment. Adaptation refers to an agent's learning by making adjustments with respect to its environment. Evolution is considered as a strategy for a population of agents to adapt to the environment [1].

When there is an environmental change or an unpredictable event occurs, an agent, through its experience, attempts its best response action. By evaluating how well that action responded, it learns something. If the evaluation result is bad, which means that the response action doesn't really work well, the agent will most likely avoid using that action again for that particular event / change or something similar to it. On the other hand, if the evaluation results is good, when that event / change occurs the next time around, the agent will more likely respond with the previous action. And so, the learning process continues, and the agent attempts to adapt to the environment. From the learning process, new behaviors or processes can be generated.

For living creatures, evolution is a process of selective reproduction and substitution. This process takes a long time, and spans over generations. This means that animals need to mate in order to evolve. But, robots can't mate. Instead, in multi-agent robotics, the agents evolve its controllers – also known as its brain or processor – via evolutionary algorithms, which is discussed in the following section.

### **3 Genetic Algorithm**

Genetic Algorithm (GA) is adaptive method that may be used to solve search and optimization problems [3, 4]. It applies the nature's law of survival of the fittest that only allows the organisms that best adapt to the environment to live. It may also be regarded as 'a hill-climbing search method that finds near-optimal solutions by subjecting a population of points in a search space to a set of biologically inspired operators.

#### **3.1 Representing Hypotheses in GA**

A chromosome can be represented by anything in a universal set – colored strings, decimal numbers, or even as simple as bit strings of zeros and ones. Hypotheses can have many descriptions. For example, we take the attribute "weather". Some possible values are sunny, rainy, cloudy, etc. We also can use bit strings to represent these values, but it is quite complex to do so. For example, we can take 010 to represent cloudy, 011 to represent any two possible values of cloudy or rainy, 111 to represent any weather condition, etc. If the second attribute is the football field condition, and the values range from dry to wet, we can represent the field condition in 4 levels. In binary string, we can take 00 to represent very dry, to 11 to represent very wet. To make computation easy for genetic operations such as mutation and crossover, hypotheses are usually described by bit strings.

#### **3.2 GA for Multi-Agent Robotics**

By applying evolutionary computation for solving optimization problems, robotics is one of the fields in which researchers have found many applications, ranging from control strategy synthesis to geometric motion planning. For motion planning, a chromosome represents a path that consists of straight-line segments. A path may or may not be feasible. The initial population of chromosomes is a group of paths. Chromosomes are evaluated and selected according to their fitness. Genetic operators are used for possible improvement. In this case, the fitness function is used to evaluate the feasibility, length, smoothness, clearness of a path [1]. Three commonly used genetic operators are selection, crossover and mutation. In selection, candidates with higher fitness have higher chances to survive in the next generation. Crossover recombines two parent paths into two new paths whereas mutation tunes the node coordinates in a feasible path for sharp adjustment.

### 3.3 Fitness Function

In behavior evolution, the fitness function used to select the best next location consists of two terms: one is the general fitness, and another is called special fitness. The general fitness term encourages the group robots to explore the potential field in new, less confident regions, and at the same time avoid repeating the work of other robots. It is defined as:

$$s_g = \prod_{i=1}^m \left\{ (1 - \max\{w_i^{f_k}\}) \prod_{j=1}^{m_e} \sqrt{d_{ij} - R_1} \right\} \quad (1)$$

where  $\max\{w_i^{f_k}\}$  denotes the maximum confidence weight corresponding to the location of robot  $i$ .  $m$  denotes the number of robots that are grouped together during one evolutionary movement step, according to a special fitness term. When the special fitness term is concerned with spatial diffusion,  $m$  becomes  $m_d$ .  $m_e$  denotes the number of robots that do not belong to  $m$ , and have just selected and executed their next motion strategies.  $d_{ij}$  refers to the distance between robots  $i$  and  $j$ , which is greater than a predefined distance threshold,  $R_1$ . In addition to the general fitness, we also define two special fitness terms corresponding to the criteria of multi-robot spatial diffusion and area coverage, respectively. They are:

1. *Spatial diffusion:*

$$s_1 = \prod_{i=1}^{m_d-1} \prod_{j=i+1}^{m_d} \sqrt{d_{ij} - R_2} \quad (2)$$

2. *Area coverage:*

$$s_2 = \frac{\sqrt{\Delta V}}{\prod_{i=1}^{m_c} \zeta_i} \quad (3)$$

where  $m_d$  denotes the number of spatially diffusing robots whose inter-distances  $d_{ij}$  are greater than the threshold  $R_2$ .  $\Delta V$  denotes the total number of locations visited by a group of  $m_c$  area-covering robots, based on their selected motion directions.  $\zeta_i$  denotes a significant proximity distance between robot  $i$  and other robots in the environment. Having defined the general and special fitness terms, the complete fitness function used in the evolution of group robots can be given as follows:

$$F = \begin{cases} s_g \cdot s_1, & \text{for spatially diffusing robots} \\ s_g \cdot s_2, & \text{for area covering robots} \end{cases} \quad (4)$$

Thus, given a certain stimulus, a robot first applies some genetic operations to a population of chromosomes representing possible next locations.

## 4 Simulation and Results

This simulation aims to study the effect of the different values of crossover probability on the efficiency of the system. The relevant toolbox is downloaded from [6]. The environment of exploration space is as depicted in Fig. 1 with relevant true potential field map shown in Fig. 2. The size is 300 x 300 pixels. It contains three medium-sized obstacles – two oval-shaped and one rectangular. Six robots are used in this experiment. Their initial positions are at the bottom left corner. These can be seen as small dots in Fig. 1. As suggested by most GA researchers [7, 8, 9, 10], the probabilities of crossover which are used in this experiment range from 0.5 to 0.9. The other simulation parameters are given in Table 1.

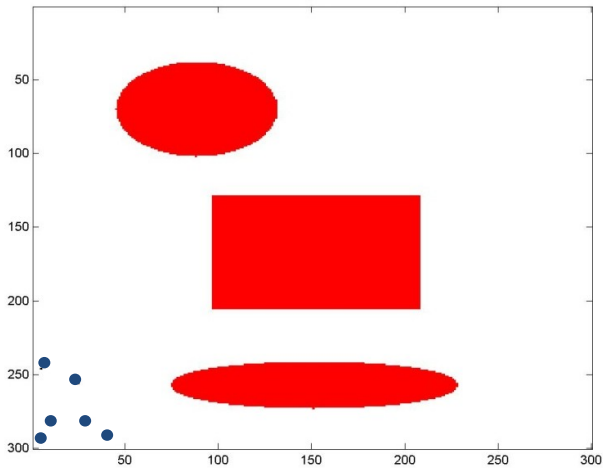


Fig. 1. The environment used for Simulation

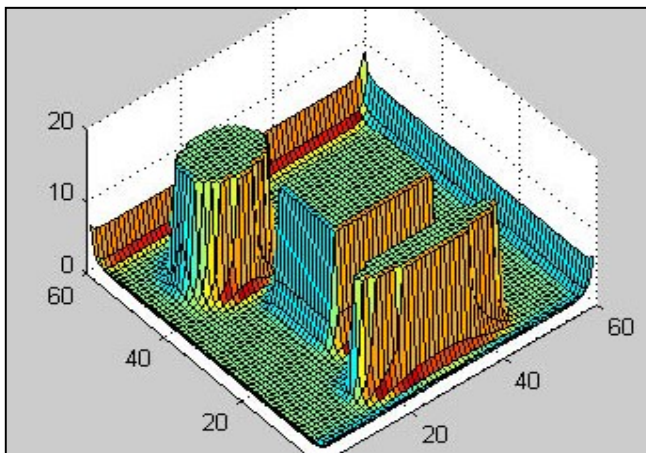


Fig. 2. True potential field map for simulation

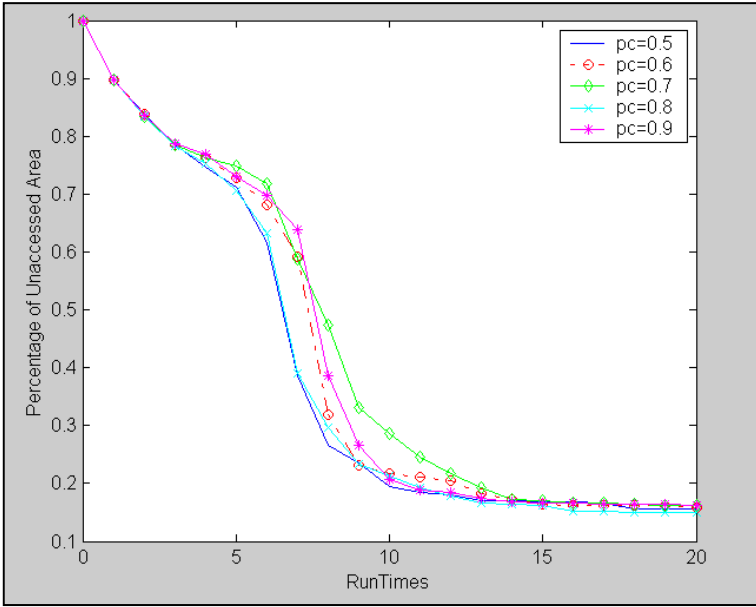
**Table 1.** Parameters used in the simulation

Parameter	Symbol	Unit	Value
Number of robots	$M$		6
Sensory section	$N$		16
Environment size		grid	300x300
Map resolution		grid	5
Map locations			60x60
Maximum movement step	$d_m$	location	7
Robot initial locations		location	(5, 5) (10, 40) (20, 10) (20, 30) (45, 25) (55, 5)
Behavior vector increment	$\psi$		0.2
Chromosome length	$(2L) \cdot M$	bit	96
Population size	$P$		120
Generations per step	$G$		48
Crossover probability	$p_c$		0.5-0.9
Mutation probability	$p_m$		0.1/0.05/0.005
Number of run steps			20

The mutation rate is already adaptively regulated according Equation (5) below.

$$p_M = \begin{cases} 0.1, & \text{if } 1 \leq \text{generation} < 40 \\ 0.05, & \text{if } 40 \leq \text{generation} < 80 \\ 0.005, & \text{if } 80 \leq \text{generation} < 120 \end{cases} \quad (5)$$

The percentage of the unaccessed environment is monitored throughout the experiment. This is shown in the graph of Fig. 3. The lower the percentage of unaccessed area, the better it is. It is clear from the graph that the system works best when  $p_c = 0.8$ . Also,  $p_c = 0.5$  yields almost the same result. The system is worst at  $p_c = 0.7$ . After RunTimes = 14, the percentage of unaccessed area for all crossover probabilities converges to about 17%. This value refers to the area covered by the obstacle. Thus the exploration of the map converges at approximately 14 run steps. The value of crossover rate do not affect much on this map, as it may be too small to see apparent results, and there are not enough obstacles. Simulations can be done using bigger maps in the future, but will take a much longer time. For one simulation of a single crossover probability value, it took 16 minutes on this 300×300 pixels monochrome map.



**Fig. 3.** Percentage of unaccessed area in the simulation

## 5 Conclusions

Behavior-based robotics, collective robotics, and evolutionary robotics have been inspired by biology, ethnology, sociology and other related fields. The three approaches are interrelated. They all aim at generating complex, adaptive and goal-driven group behaviors from simple local interactions among individuals in multi-agent robotic systems. However, they also produce different forms of autonomy, adaptability, task complexity, and intelligence in multi-agent systems. For example, although behavior based approaches are robust for many task environments, they are not necessarily adaptive. An evolutionary system can, on the other hand, improve the adaptability to the changes in a dynamic environment.

Multi-agent robotic system design is challenging because the performance of such a system depends significantly on issues that arise from the interactions between robots. Distributed approaches are appealing due to their properties of scalability and reliability. Learning, evolution, and adaptation are three fundamental characteristics of individuals in a multi-agent robotic system. Behavior-based robotics, collective robotics, and evolutionary robotics have offered useful models and approaches for cooperative robot control in the multi-robot domain.

Different techniques of reinforcement learning, genetic algorithms, artificial life, immune systems, probabilistic approaches, and multi-agent planning can be helpful in the design of cooperative controllers for multiple robots. Several important issues in multi-agent robotics, such as self-organization, multi-agent planning and control, coevolution, emergent behavior, reactive behavior, heterogenous/homogenous design,



multi-agent simulation, and behavior dynamics need to be addressed. The ultimate aim is to synthesize complex group behaviors from simple social interactions among individuals based on simple strategies.

As a conclusion, the evolutionary multi-agent robotics has a great potential in space exploration. The core of our evolutionary algorithm applied is Genetic Algorithm. The reactive movement of a robot is represented by a chromosome in GA. We analyzed the effect of the crucial parameter in GA, namely the crossover rate. The simulation results show that different crossover rate will produce different convergence curves. The major drawback in area exploration using Multi-Agent robots is the slow speed. One computation of  $300 \times 300$  pixel monochrome map will take around 16 minutes using 6 robots, 16 sensory sections. Our future works will be focusing on speed reduction alongside efficient exploration of the search area.

## 6 References

1. J.M. Liu and J-B Wu. *Multi-Agent Robotic Systems*. CRC Press LLC, Florida, 2001.
2. M.J. Mataric. *Autonomous Robots*. 1997. Referred by J.M. Liu and J.B. Wu (2001).
3. Y.H. Song, G.S. Wang, P.T. Wang, and A.T. Johns. Environmental/Economic Dispatch Using Fuzzy Logic Controlled Genetic Algorithms. *IEEE Proceedings on Generation, Transmission and Distribution*, 144(4), 377-382, 1997.
4. O. Khatib. Real-time Obstacle Avoidance for Manipulators and Mobile Robots. *International Journal of Robotics Research*, 5(1), 90-98, 1996.
5. J.M. Liu, J-B Wu, and D.A. Maluf. *Evolutionary Self-Organization of an Artificial Potential Field Map with a Group of Autonomous Robots*. IEEE, 1999.
6. Multi-Agent Robotics Matlab Toolbox.  
<http://www.crcpress.com/product/catno/2288>
7. Y.S. Yun and M. Gen. Performance Analysis of Adaptive Genetic Algorithms with Fuzzy Logic and Heuristics. *Fuzzy Optimization and Decision Making*, 2, 161-175, 2003.
8. L.K. Mak, Y.S. Wong, and X.X. Wang. An Adaptive Genetic Algorithm for Manufacturing Cell Formation. *International Journal of Manufacturing Technology*, 16, 491-497, 2000.
9. M. Srinivas and L.M. Patnaik. Adaptive Probabilities of Crossover and Mutation in Genetic Algorithms. *IEEE Transaction on Systems, Man and Cybernetics*, 24(4), 656-667, 1994.
10. Q.H. Wu, Y.J. Cao, and J.Y. Wen. Optimal Reactive Power Dispatch Using an Adaptive Genetic Algorithm. *Electrical Power & Energy Systems*, 20(8), 563-569, 1998.