A Fuzzy-Logic Based Chaos GA for Cooperative Foraging of Multi-Robots in Unknown Environments.

Article *in* International Journal of Robotics and Automation · January 2012

DOI: 10.2316/Journal.206.2012.1.206-3553 · Source: DBLP

2 authors:

Jianjun Ni
Hohai University

Simon X. Yang
University of Guelph

A FUZZY-LOGIC BASED CHAOS GA FOR COOPERATIVE FORAGING OF MULTI-ROBOTS IN UNKNOWN ENVIRONMENTS

Jianjun Ni* and Simon X. Yang**

Abstract

This paper investigates the foraging of multiple robots in completely unknown environments. The onboard robot sensor information and expert knowledge of foraging are used to forage the targets. The foraging problem in this paper is defined as a searching task, where the robots cooperate to find and reach all the targets in an efficient way. A novel fuzzy-logic based chaos genetic algorithm (FCGA) is proposed for target foraging in unknown environments. The fuzzy logic is used to avoid the disorder of the robot movement and reduce the search time when there is no information about the targets or the information density around the robots is the same. The chaos genetic algorithm enables the robots find the targets efficiently. In the proposed approach, the robot motion can be dynamically adjusted to guarantee that all the targets can be found, even in some difficult situations such as targets are at some locations difficult to find or obstacles are linked together. The proposed approach is capable of dealing with uncertainties, *e.g.*, some robots break down. In comparison to the pure chaos genetic algorithm (PCGA) and the random-search approach, experimental results show that the proposed approach is more efficient in foraging all the targets.

Key Words

Foraging problem, fuzzy logic, chaos genetic algorithm, multi-robot system, unknown environment

1. Introduction

Multi-robot systems have been the subject of much research since the 1970s for various tasks. The coordination of multi-robot systems has recently attracted more and more attention, because a mobile robot team can complete a task more rapidly and more efficiently than an individual robot alone [1]–[4]. After years of development, both the hardware and software have been greatly advanced, which make it possible for a multi-robot system to work in unknown and dynamic environments [5]–[7].

The research of multi-robot systems includes localization, task allocation, collision avoidance, path planning, hardware realization, and so on [8]–[11]. Task allocation is an essential issue for multi-robot systems and this problem has been proven as an NP-complete problem [12]–[14]. Currently, there are three main issues for task allocation in multi-robot systems, namely, hunting, scheduling, and foraging [15]–[17]. In the hunting task, Cao *et al.* [18] proposed a distributed control approach to multi-robot hunting tasks in unknown environments. Ma *et al.* [16] proposed a multi-robot coordinated hunting strategy with dynamic alliance. In the scheduling task, Shah and Meng [19] proposed a communication-efficient dynamic task scheduling algorithm for a heterogeneous multi-robot system. Kim *et al.* [20] proposed a hybrid intelligent agent-based scheduling and control system architecture for an actual industrial warehouse order picking problem. The foraging problem is a type of task allocation, which usually exists in a multi-robot system and is different with the task of hunting and scheduling. It can be described as follows: there are some targets in an environment, and some robots should find these targets while minimizing time or cost. In this problem, Tian *et al.* [14] proposed a multi-robot task allocation for fire-disaster response based on reinforcement learning. In the paper, two task allocation algorithms based on reinforcement learning are given out, one is a non-cooperation learning algorithm and the other one is cooperation learning algorithm. The simulation results of the paper show that the learning algorithm designed for multi-robot fire-fighting task is effective. Wegner and Anderson [17] described an approach to multi-robot control for the environments that focused on combining the limited abilities of modern autonomous control systems together with human control, and implemented this approach in a rescue domain. Zhu and Yang [21] proposed

* College of Computer and Information, Hohai University, Changzhou, Jiangsu Province, P.R. China 213022; e-mail: nijj@hhuc.edu.cn

** Advanced Robotics and Intelligent Systems (ARIS) Laboratory, School of Engineering, University of Guelph, Guelph, Ontario, Canada N1G 2W1; e-mail: syang@uoguelph.ca

Recommended by Prof. H. Hagras

(DOI: 10.2316/Journal.206.2012.1.206-3553)

a self-organizing map based multi-agent architecture for multi-robot systems. It is capable of controlling a group of mobile robots to complete multiple tasks simultaneously.

There is much research on various problems of multi-robot task allocation. Archibald and Frost [22] studied the formation initialization problem. Lerman *et al.* [23] analyzed the dynamic task allocation of multi-robot systems. Feddema *et al.* [24] focused on how decentralized control theory could be used to analyze the control of multiple cooperative robotic vehicles. Casper and Murphy [25] studied the human–robot interactions during the robot-assisted urban search and rescue response. Much research on task allocation focused on the algorithms, but few considered real world problems. Zhu and Yang [26] used a neural network approach to deal with the dynamic task assignment of multi-robots. Although this method considered the problems of the changing of tasks, the flow diagram of the proposed algorithm shows that the target location must be input to the network at every step, so the approach used known information of the targets. Arsie *et al.* [27] consider a class of dynamic vehicle routing problems, in which a number of mobile agents in the plane must visit target points generated over time by a stochastic process. Although the target points are generated dynamically, the authors suppose that the targets broadcast their positions to the whole environment and each agent is able to receive the signal broadcasted by each target. However, in the actual world, sometimes the location information is not known. Similar problems also exist in other literature. For example, Tian *et al.* [14] used the Q-learning algorithm to deal with the task allocation in the fire-disaster response. The nature of Q-learning algorithm decides its work flow this way: a lot of robots move randomly at first, with more and more experiments, the Q-value table is set up, then the robots can move directed by this table to find the targets at a minimum time or cost. However, if it is an actual fire-disaster response, it would never have time to try again and again. The same problems exist in the ant colony algorithm and the bee-inspired algorithm of the multi-robot system to deal with the foraging problem [28], [29].

In this paper, the foraging problem near the real world is studied, where the locations of targets are unknown; and the robots can use only the onboard sensor information to search for the targets. The foraging problem studied in this paper is different from the general path planning in multi-robots where the locations of targets are often known and there is no cooperation among the robots [30], [31]. A novel fuzzy-logic based chaos genetic algorithm (FCGA) is proposed and experiments are conducted in various situations. The experimental results show the efficiency of the proposed approach in comparison to the pure chaos genetic algorithm (PCGA) and random-search approach used in the foraging task. The robots can automatically find all the targets along a better trajectory than other approaches and avoid obstacles whenever the situations are changed, such as the targets are at some places difficult to find, the obstacles are linked together, or the robots start from the same location. The proposed approach can handle uncertainties, *e.g.*, some robots break down. Because most parameters are self-adaptive, the number of parameters need to be set by the designer in the proposed approach is small. These parameters are not very sensitive, which can be in a wide range.

The paper is organized as follows. In Section 2, the problem statement is given and a new model is described. Section 3 presents the proposed FCGA. The experiments at various situations are given in Section 4. Section 5 discusses the sensitivity of the parameters in details. At last, the conclusion is given in Section 6.

2. The Problem Statement

In this paper, the foraging problem of a multi-robot team in unknown environments is studied. The problem is defined as follows. (1) The robots have no knowledge about the environments and the locations of targets. The robots just know the total number of the targets to forage. (2) The robots are labelled as r_i, $i = 1, 2, \ldots, n$, which belong to a robot team $\Re$. (3) There is a set of targets, $\Im = \{t_1, t_2, \ldots, t_m\}$, in a bounded environment. The target locations are unknown in the whole process. (4) The targets have some information, which can be sensed by the robot in a limited area (*e.g.*, the infrared radiation of the heat source, the radiation of the radioactive source, or the odour of the odourous source). (5) The foraging process can be described as follows: at the beginning, a foraging task is given to a robot team $\Re$. This task can be denoted by $\Omega = (n, m)$, where n is the number of robots in the team $\Re$, m is the number of targets need to be found. The robots begin to search the targets in the environment. When the robots arrive at the location of one target, this target is reached and is removed from the workspace. After all the targets are found, the task is completed.

The density of the target information in the workspace around the target t_i is given by:

$$I_t(t_i, p_e) = \begin{cases} A_t, & \text{if } D(p_{t_i}, p_e) <= 1 \\ \frac{A_t}{D(p_{t_i}, pe)}, & \text{if } 1 < D(p_{t_i}, p_e) <= R_t \\ 0, & \text{if } D(p_{t_i}, p_e) >= R_t \end{cases} \quad (1)$$

$$D(p_{t_i}, p_e) = \sqrt{(x_{t_i} - x_{p_e})^2 + (y_{t_i} - y_{p_e})^2} \quad (2)$$

where $p_e = (x_{p_e}, y_{p_e})$ and $p_{t_i} = (x_{t_i}, y_{t_i})$ are the coordinates of location p_e and target t_i, respectively; A_t is the largest density of the target information; R_t represents the radius of the target information; and function $D(p_1, p_2)$ defines the distance between locations p_1 and p_2.

The problem definition is near the foraging problem in the real world, such as fire disaster response, and searching for radioactive objects or odourous sources. The solution may be very meaningful in some real fields [14], [25]. An example of this problem is shown in Fig. 1. In this example, there are two targets denoted by solid triangles; three robots denoted by solid circles; and four obstacles denoted by solid squares. The range of the target information is denoted by many concentric circles, and the biggest radius of these concentric circles is R_t.

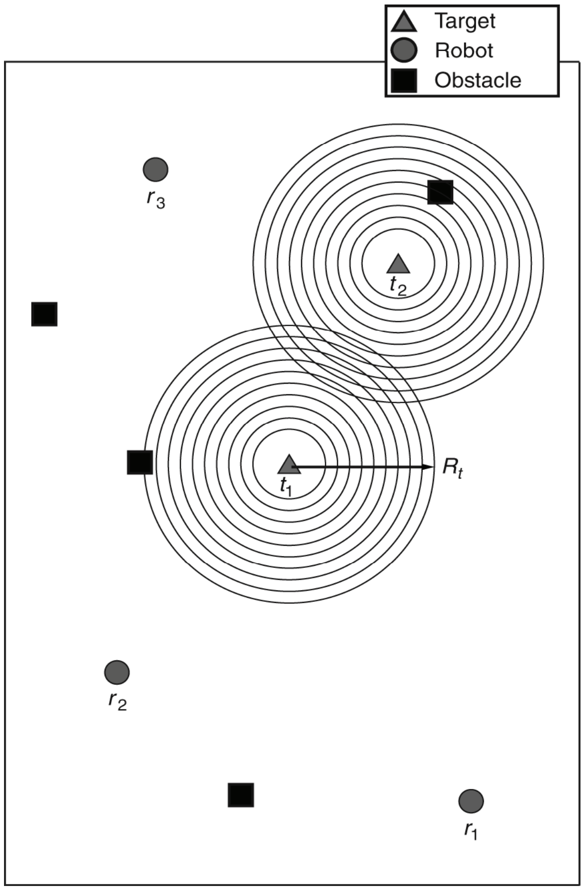

Figure 1. A simple example of the foraging problem.

The task in this paper is similar to those tasks described in some related literature. For example, Lerman *et al.* [23] described a task scenario. In their scenario, there are two types of tasks in an environment. There are a lot of robots in the environment. The robot chooses to change its state, or the type of task it is assigned to execute, with probabilities given by transition functions. The goal of this paper is to investigate how incomplete knowledge of the environment (through local observations), as well as the dynamically changing environment affects task allocation. Meng *et al.* [32] set up a scenario. In their scenario, there are some targets distributed randomly in the environment. There is a homogeneous multi-robot system. The aim of it is to evenly distribute the robots to the tasks. A modified particle swarm optimization algorithm was proposed, to calculate the real-time velocity of the robots.

However, there are some differences between the literature and the problem studied in this paper. There are two basic problems that need to be resolved in the task studied. The first one is how to do the task efficiently, when there is no target information or the information density around the robots is the same. The other one is how to coordinate these robots efficiently as a robotic team.

3. The Proposed Approach

In this study, all the targets need to be found efficiently in term of search time. It is a typical optimization problem, so the genetic algorithm (GA) could be used [33]–[35].

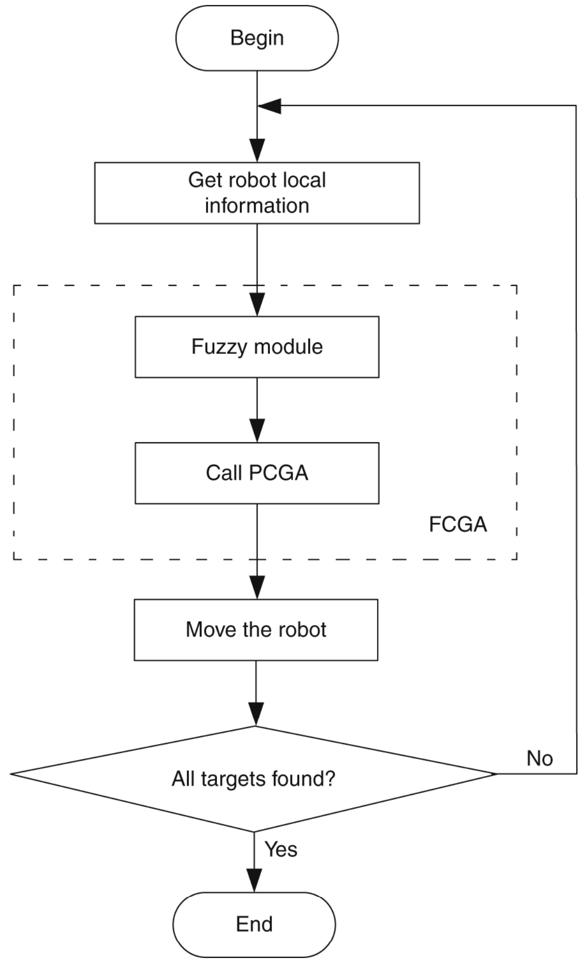

Figure 2. Flow diagram of the proposed approach.

However, the global information of the targets is unknown in this study. In some cases, the result may not be good if using the pure GA alone. All the information and knowledge should be used to direct the robot movement, which could be realized by fuzzy logic [36]–[38]. In this paper, a FCGA approach is presented. The basic idea of the proposed approach is that a PCGA is used as the basic method to search for the targets and avoid obstacles in the least possible time, and the fuzzy logic is used to improve the performance of PCGA. The diagram of the proposed approach is shown in Fig. 2.

3.1 Pure Chaos Genetic Algorithm

A lot of work has been done on the problem of foraging for targets in minimal time by GA approaches. However, many of these applications are based on the hypothesis that the locations of targets are known, *e.g.*, Gong *et al.* [34] proposed a GA based combinatorial auction algorithm for multi-robot cooperative hunting, the fitness function of Gong *et al.*'s algorithm is as follows:

$$f = \sum_{j=1}^{m} \sum_{i=1}^{n} D(p_i, p_j) \tag{3}$$

where p_i denotes the location of the hunter; p_j denotes the location of the prey key point; $D(p_i, p_j)$ is the distance

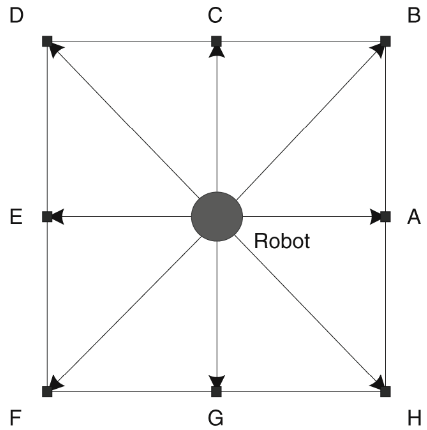

Figure 3. The moving directions of the robot.

from the hunter to the prey key point. In the fitness function of Gong *et al.*'s algorithm, the location of the prey is known before the action of the robots. However, the target (prey) locations in this paper are unknown. The algorithm proposed in this paper is introduced as follows.

3.1.1 The Chromosome of the Robot Movement

For simplification without loss generality, the robot can move in eight directions only (see Fig. 3). The robot can only move between the discrete locations in the environment, and {A, B, ..., H} are the next possible locations of the robot.

The direction of a single robot can be coded as a 3-bit binary number, if the number of robots is n, the whole length of the code (chromosome) is $n \times 3$. For example, if there are three robots in the workspace, one chromosome for the robot directions is $C = \{d_1, d_2, d_3\} = \{1\ 0\ 0, 0\ 1\ 0, 1\ 0\ 1\}$, where C denotes the chromosome and $d_i, i = 1, 2, 3$, is the gene (the robot direction) of the chromosome C. This chromosome means that the direction of the robot r_1 is 180°, r_2 is 90°, and r_3 is 225°.

3.1.2 The Fitness Function

The fitness function is the key to the GA-based method in multi-robot task allocation. Although the target locations are unknown, the onboard sensor information from all the robots in the system can be used to construct the fitness function:

$$f = \sum_{i=1}^{n} I(d_i), \quad d_i \in C \tag{4}$$

where $I(d_i)$ is a function to calculate the information density at the next location by the direction d_i, which can be detected by the robot r_i. Assume $p_{r_i} = (x_{r_i}, y_{r_i})$ is the next location of the robot r_i, then

$$I(d_i) = \sum_{j=1}^{m} I_t(t_j, p_{r_i}) - w_1 \sum_{k=1}^{o} O(o_k, p_{r_i}) \tag{5}$$

where $I_t(t_j, p_{r_i})$ is the information density of the target t_j at the location p_{r_i}, which can be sensed by the robot and the value of it can be calculated by (1); w_1 is the parameter to adjust the effect of targets and obstacles; o is the number of obstacles found by the robots, and $O(o_k, p_{r_i})$ are used to judge whether there are any obstacles in the location p_{r_i}. The last part in (5) is used to avoid the obstacles.

$$O(o_k, p_{r_i}) = \begin{cases} 1, & \text{if obstacle } o_k \text{ is at } p_{r_i} \\ 0, & \text{otherwise} \end{cases} \tag{6}$$

3.1.3 The Chaos Genetic Algorithm

Because the chromosome is coded in binary, the selection, crossover and mutation operation are realized easily. The general method is adopted in this paper [39]. To improve the performance of GA, some chaos operation was input to GA [40]. In this paper, a chaos operation is used to avoid the premature convergence problem of GA. The following rule is used to judge whether there is a premature convergence:

$$\begin{aligned} &\text{IF } average(fitness)_t - average(fitness)_{t-1} < \varepsilon \\ &\text{AND } average(fitness)_{t-1} - average(fitness)_{t-2} < \varepsilon \\ &\text{THEN } flag\_premature = 1 \end{aligned} \tag{7}$$

where $average(fitness)_t$ is the average fitness of the population at time t; ε is a very little positive number; *flag_premature* is the flag to indicate whether there is a premature convergence or not. If a premature convergence happens, the chaos operation is used to update the chromosome. In this paper, the following chaos model is used:

$$\alpha = \beta\alpha(1 - \alpha) \tag{8}$$

where α is the chaos variable and $0 \le \alpha \le 1$; β is the attractor. Some new chromosomes can be obtained through the following equation:

$$C_j^{new} = \zeta_j C_j^{old} + \rho\alpha_j(1 - \zeta_j) \tag{9}$$

where C_j^{new} and C_j^{old} denote the new and the old chromosome of the population, respectively; j denotes the sequence number of the chromosome in the population. α_j is the chaos variable produced by (8); $0 \le \zeta_j \le 1$ is a random number; ρ denotes the max code of the chromosome, and $\rho = (2^{3n} - 1)$ in this paper.

3.2 The Proposed Fuzzy-Logic Based Chaos Genetic Algorithm

When there is no information about the targets or the information density is the same around the robots, the robots would move randomly in this situation by general methods. However, if memory and knowledge could be used very well, the robot can move in a better direction, combined with the information detected by onboard sensors. There is a lot of knowledge on successful searches in the real world, such as searching by the partition of the search area, and searching by the priority of targets. In this paper, some concepts are proposed and fuzzy logic is used to realize the functions of such knowledge.

Dispersion degree: This concept is used to judge whether the robots go together too much. Here, the fuzzy parameter u_1 is used to denote it. Assume that there are n robots in the area, the area is a $W \times H$ rectangle, with width W and height H. The Gauss function is used to calculate the value of dispersion degree [41]:

$$\overline{D}_t = \frac{2!(n-2)!}{n!} \sum_{i=1}^{n} \sum_{j=i+1}^{n} D(p_{r_i}, p_{r_j}) \tag{10}$$

$$\delta_t = \frac{\overline{D}_t}{\sqrt{(H^2 + W^2)}} \tag{11}$$

$$u_1 = e^{-\frac{1}{2}(\frac{\delta_t - c_t}{\sigma_t})^2} \tag{12}$$

where $\overline{D}_t$ represents the average distance between the robots in the robot team at time t; δ_t is the dispersion; c_t denotes the centre of Gauss function; and σ_t determines the width of Gauss function. The Gauss function is fixed completely by the parameter c_t and σ_t. Here, a self-adaptive method is used to calculate the two parameters:

$$\sigma_t = \frac{1}{2}\{\max(\delta_k) - \min(\delta_k)\} \tag{13}$$

$$c_t = \frac{1}{l} \sum_{k=t-l+1}^{t} \delta_k \tag{14}$$

where l is the length of the robot memory, which depends on the hardware of the robot.

Homodromous degree: This concept is used to judge whether all the robots move at the same direction. Here, the fuzzy parameter u_2 is used to denote it. Assume that there are n robots in the area, and the directions of all these robots are $\{d_1, d_2, \ldots, d_n\}$, then the value of homodromous degree can be calculated by:

$$u_2 = \frac{\frac{2!(n-2)!}{n!} \sum_{i=1}^{n} \sum_{j=i+1}^{n} \mathrm{abs}(d_i - d_j)}{h - 1} \tag{15}$$

where h is the number of possible directions of the robot movement, which is 8 in this paper; and abs() is the absolute value function.

District-difference degree: This concept is used to judge whether all the robots stay in the same area. The fuzzy parameter u_3 is used to denote it. In the actual search task, the environment is usually divided into different parts based on the number of targets and search resources. The environment is divided into N_d parts $\{\mathrm{P}_1, \mathrm{P}_2, \ldots, \mathrm{P}_{N_d}\}$ in this paper, where $N_d \le n$ and is set as from 2 to 9 always. The value of district-difference degree can be calculated by:

$$u_3 = \frac{n}{\max\left\{\sum_{j=1}^{n} P(i, r_j), i = 1, 2, \ldots, N_d\right\}} \tag{16}$$

where $P(i, r_j)$ is the function to judge whether the robot r_j is in the ith part of the environment.

If the robots already stay in the same area, another fuzzy logic needs to be used to make some of them go out this area, the fuzzy parameter u_4 is imposed to denote it:

$$u_4 = \frac{1}{\sum_{i=1}^{n} D(p_{r_i}, p_0)} \tag{17}$$

where p_0 is the centre location of the environment.

The function (5) in the fitness function (4) can be changed by these fuzzy parameters:

$$I(d_i) = u_1 u_2 u_3 \left[\sum_{j=1}^{m} I_t(t_j, p_{r_i}) - w_1 \sum_{k=1}^{o} O(o_k, p_{r_i}) \right] + w_2 u_4 \tag{18}$$

where w_2 is a constant. By the modified fitness function (see (4) and (18)), the cooperative behaviour emerges naturally.

The work flow of the proposed FCGA approach is summarized as follows: (1) Initialize the population for the directions of robots; (2) Calculate the fuzzy parameters u_1, u_2, u_3, and u_4 of each chromosome, if the robot team is directed by this chromosome; (3) Calculate the fitness of each chromosome; (4) Judge whether there is a premature convergence in the population and deal with it; (5) Get a new population by the selection, crossover and mutation operation; (6) If $t < T$, Go to step (2); (7) Output the chromosome with maximum fitness in the population; (8) Decode the chromosome and update the locations of robots. The pseudo code of the proposed FCGA in this paper is introduced in Fig. 4.

4. Experiment Studies

To demonstrate the effectiveness of the proposed approach for cooperative foraging of multi-robots in unknown environments, five experiments are conducted that were coded in MATLAB. The parameters in all of the experiments are the same (see Table 1), where the environment is a 20×30 ($W = 20$, $H = 30$) rectangle and is divided into four parts.

4.1 Targets at Random Locations

In this experiment, the locations of obstacles, targets, and robots are random. There are three robots, six targets and six obstacles in this experiment. Figure 5 is one of the experiment results based on the proposed approach. Figure 5(a) is of the initial locations of targets, obstacles and robots. The circles are robots, the triangles are targets and the squares are obstacles. Figure 5(b) and (c) is the robot locations after 10 and 20 steps, respectively. Figure 5(d) is the final trajectories of robots based on the proposed approach, where the hollow circles denote the start locations and the solid circles denote the end locations of robots, respectively. Figure 6(a)–(c) is the information

```
////////////        Pseudo code of the FCGA        /////////

Initialization ( );        % Initialize the parameters and the population;
Do {
    Fuzzy _ module ( );      % Calculate the fuzzy parameters of each chromosome;
    Cal _ fitness ( );       % Calculate the fitness of each chromosome;
    t = t + 1;
    Judge _ premature ( );   % Judge whether there is a premature in the population;
    If flag _ premature = =1 Then
        Chaos _ dealwith ( );          % Chaos operator;
    Endif
    Selection ( );                     % Selection operator;
    Crossover ( );                     % Crossover operator;
    Mutation ( );                      % Mutation operator;
    Get _ population ( );              % Get a new population;
} While ( t < T )          % T is the maximum repeat time;
Output ( );                % Output the chromosome with maximum fitness;
Decode ( );                % Decode the chromosome;
Update ( );                % Update the locations of robots;
```

Figure 4. Pseudo code of the proposed FCGA.

Table 1
Some Parameters of the Proposed Approach

Parameters	Value	Remarks
A_t	50	The largest density of the target information
R_t	10	Radius of the target information
S_p	20	Size of the chaos GA population
P_c	0.8	Crossover probability of the chaos GA
P_m	0.2	Mutation probability of the chaos GA
T	40	The maximum repeat time of the chaos GA

density of the targets at initial status, after 10 and 20 steps, respectively.

The trajectories of robots in Fig. 5 show that the robots can avoid the obstacles automatically and find all the targets efficiently. For example, the robot r_1 in the area ‘P$_1$’ moved towards t_2 and t_3 in the area ‘P$_2$’ first (see Fig. 5(b)), and moved back towards t_1 because the robot r_2 is searching in the area ‘P$_2$’ (see Fig. 5(c)), then the whole search time is reduced (see Fig. 5(d)). So the task could be completed efficiently, although the local trajectory of one robot is not straight toward one target. To show the advantages of the proposed approach, it is compared with PCGA and random-search method. Because a lot of things are random, every experiment was conducted 10 times. There are 651 locations in the whole environment. If the method of exhaustive search (such as the standard lawn mower method) is used, the number of steps needed is 217, when there are three robots. So the maximum time of the experiment is set as 217. If the targets can not be found before this time, the experiments stop and the task is considered failed. The results of the search time are shown in Fig. 7 and Table 2. The results show that the proposed approach has good ability to find the targets. The average search time and the standard deviation is less than the other two approaches. The standard deviation of the other two approaches is higher, which shows that the result fluctuation of the other two approaches is higher than the proposed approach.

4.2 Targets Located at Locations Difficult to Find

In the foraging problem, if the distribution of the targets is very uneven and some targets are at the corners or the borderline, the robots tend to move to the place where the density of the target information is very high, when PCGA or a random-search method is used. This situation

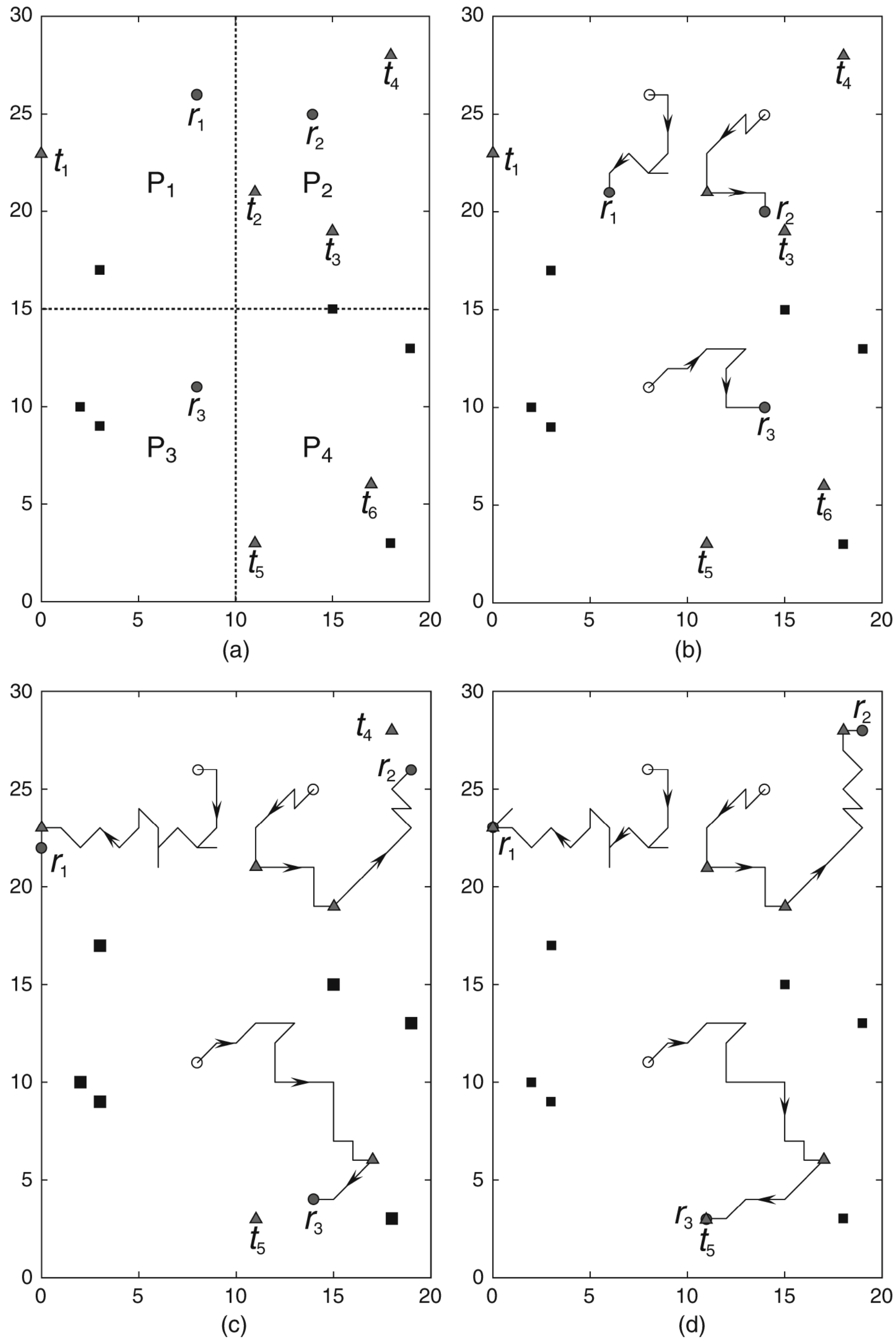

Figure 5. Robot trajectories by the proposed FCGA approach, where the locations of targets are random: (a) the initial status; (b) after 10 steps; (c) after 20 steps; (d) the final trajectories.

may lead to much more search time for other targets. In the worst case, some targets would never be found due to the difficult situation. To test the proposed approach in the difficult situation, this experiment is conducted. The locations of obstacles and robots are random, but the targets are at some locations that are difficult to find and are fixed in the whole process of this experiment. Figure 8(a) is one of the distribution of the targets, robots, and obstacles. It shows that three targets are at the cut line, their locations are (5,15), (10,15), and (10,25); one target is at the borderline of the area, its location is (20,10), and one target is at the corner, its location is (20,30). There aren't any targets in the area 'P_1', 'P_2', and 'P_4'. The information density of the targets will be very low and difficult to find, when most of the targets are found and removed from the environment (such as the situation in Fig. 6(c)). Figure 8(b)–(d) is of the final trajectories of robots based on the proposed FCGA, PCGA, and a random method, respectively, where the initial distribution is the same as Fig. 8(a) to have comparability.

The results in Fig. 8 show that the fuzzy logic becomes more important to direct the search process in this

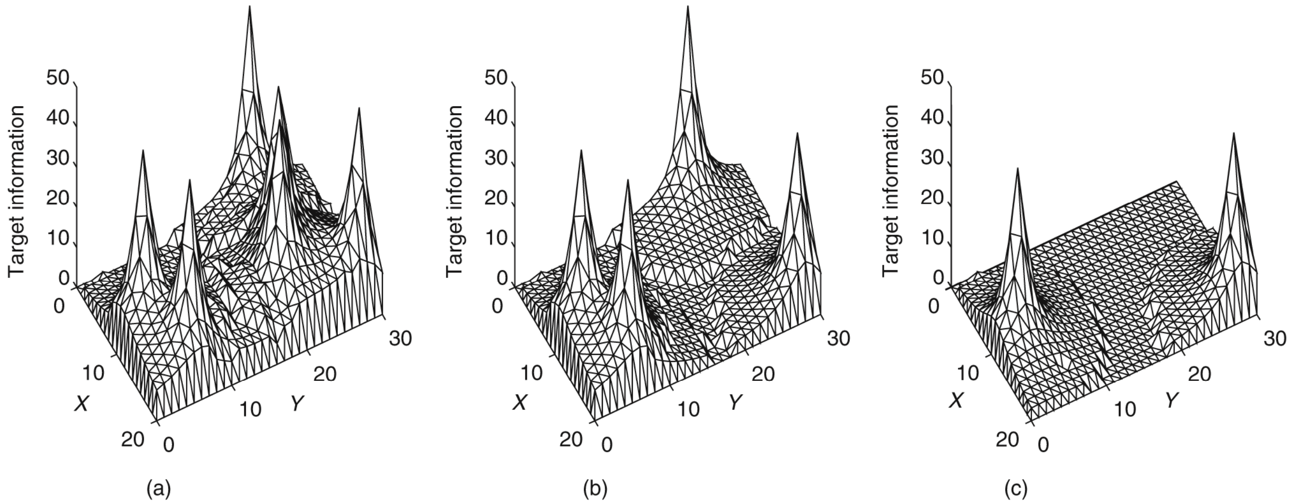

Figure 6. Target information density of one experiment: (a) the initial status; (b) after 10 steps; (c) after 20 steps.

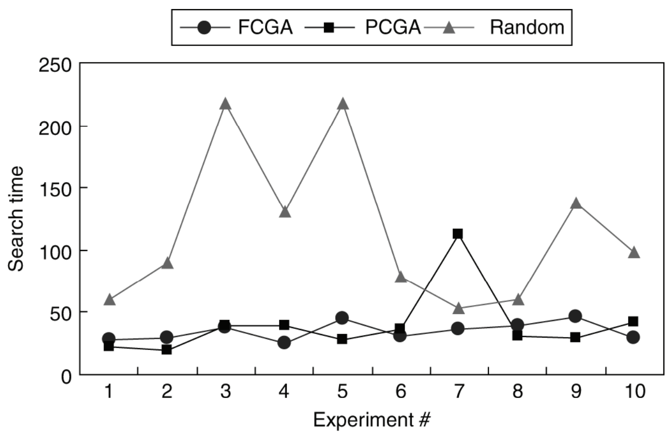

Figure 7. Search time of 10 repetitions with random target locations.

Table 2
Results of 10 Repetitions with Random Target Locations

Search Time	FCGA	PCGA	Random
The max value	45	112	138
The min value	28	19	53
The average value	34.7	40.0	88.5
The standard deviation	7.24	26.40	32.39
The failed times	0	0	2

situation. The proposed approach can direct the robots to reach all the targets at some better trajectories (see Fig. 8(b)). The final trajectories of robots based on PCGA become disorderly near the end (see Fig. 8(c)). To prove the effectiveness of the proposed approach in this situation, this experiment is repeated 10 times. The results of the search time are shown in Fig. 9 and Table 3. Compared with the experiment that the locations of targets are random (see Table 2), the search time is increased and it fails two times in the experiments based on PCGA approach; the average search time of the proposed FCGA approach increases too, but it does not fail in any of the 10 repetitions.

4.3 Robots Start from the Same Location

To further test the proposed approach in the applications near to real world, where the robots always start from the same location, this experiment is conducted. The locations of obstacles and targets are random, but the robots start from the same location, here the location is (0, 0) (see Fig. 10(a)). Figure 10(b)–(d) is the final trajectories of robots based on the proposed FCGA approach, PCGA, and a random method, respectively, where the initial distribution is the same as Fig. 10(a) to have comparability. The final trajectories show that the robots directed by the proposed approach can separate automatically at first and can go to different parts of the environment. The other two approaches do not have this performance (see Fig. 10(b)–(d)). The results of 10 repetitions of the experiment regarding search time are shown in Fig. 11 and Table 4, which have proved the efficiency of the proposed approach in this situation.

4.4 Some Robots Break Down

The proposed approach is robust. It can deal with the situation of unexpected events (such as the robots would break down, including communication loss and motion failure), and it does not need any added changes to the proposed approach. Figure 12 shows this kind of situation. At the beginning, there are six robots and six targets (see Fig. 12(a)), after 10 steps, one robot breaks down, and another robot breaks down too after 20 steps (see Fig. 12(b) and (c)). The final trajectories of the robot team (see Fig. 12(d)) show that the proposed approach can work satisfactorily in the case of unexpected events.

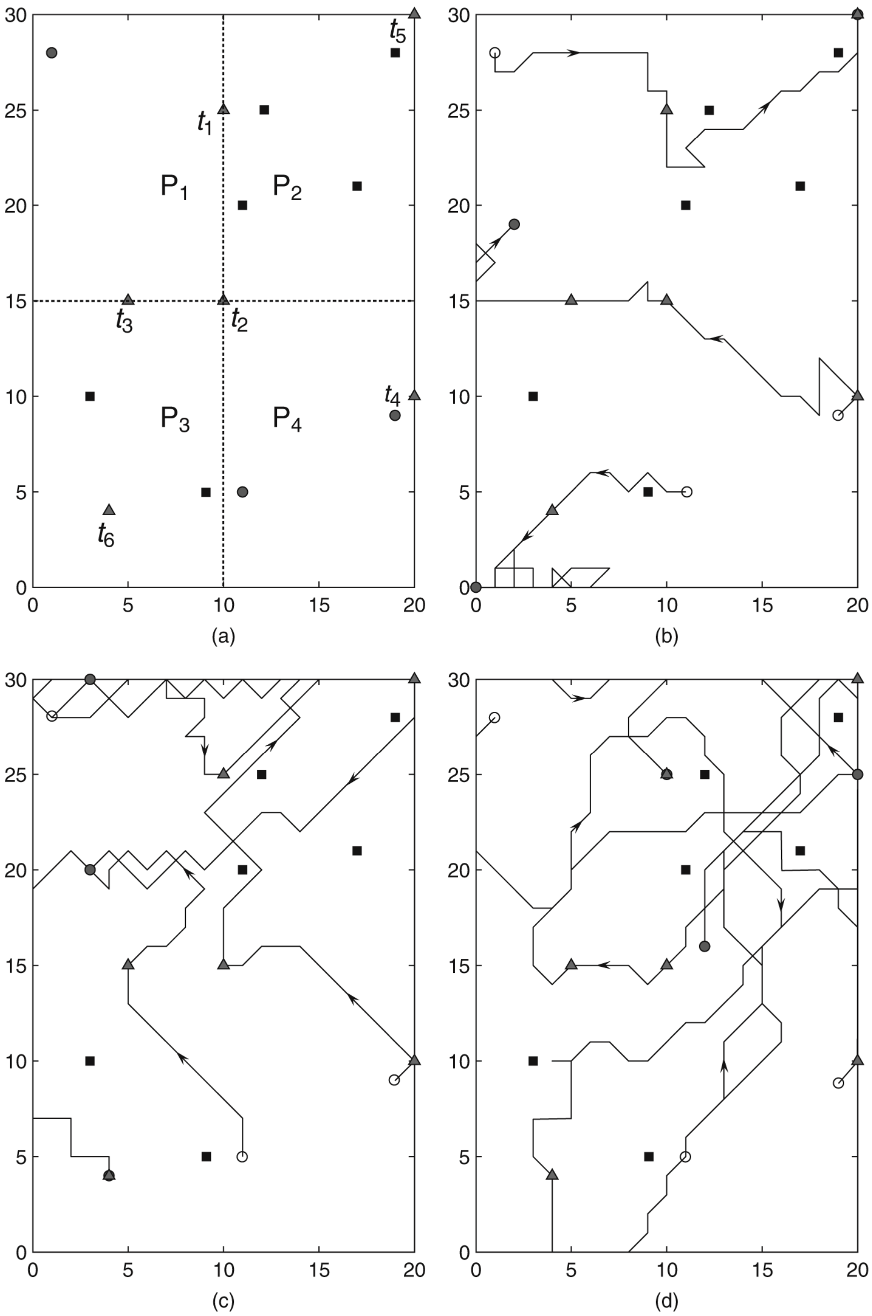

Figure 8. The initial status and final trajectories of robots, where targets located at locations difficult to find: (a) the initial status; (b) the trajectories based on FCGA; (c) based on PCGA; (d) based on a random method.

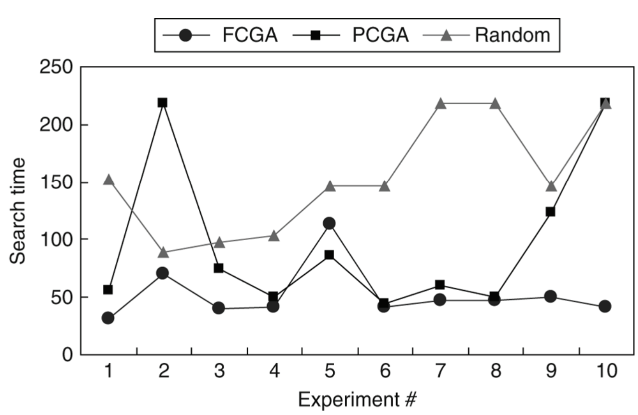

Figure 9. Search time of 10 repetitions where targets are at locations that are difficult to find.

Table 3
Results of 10 Repetitions Where Targets are at Locations that are Difficult to Find

Search Time	FCGA	PCGA	Random
The max value	113	124	153
The min value	32	44	89
The average value	52.5	68.3	126.0
The standard deviation	23.46	26.53	27.88
The failed times	0	2	3

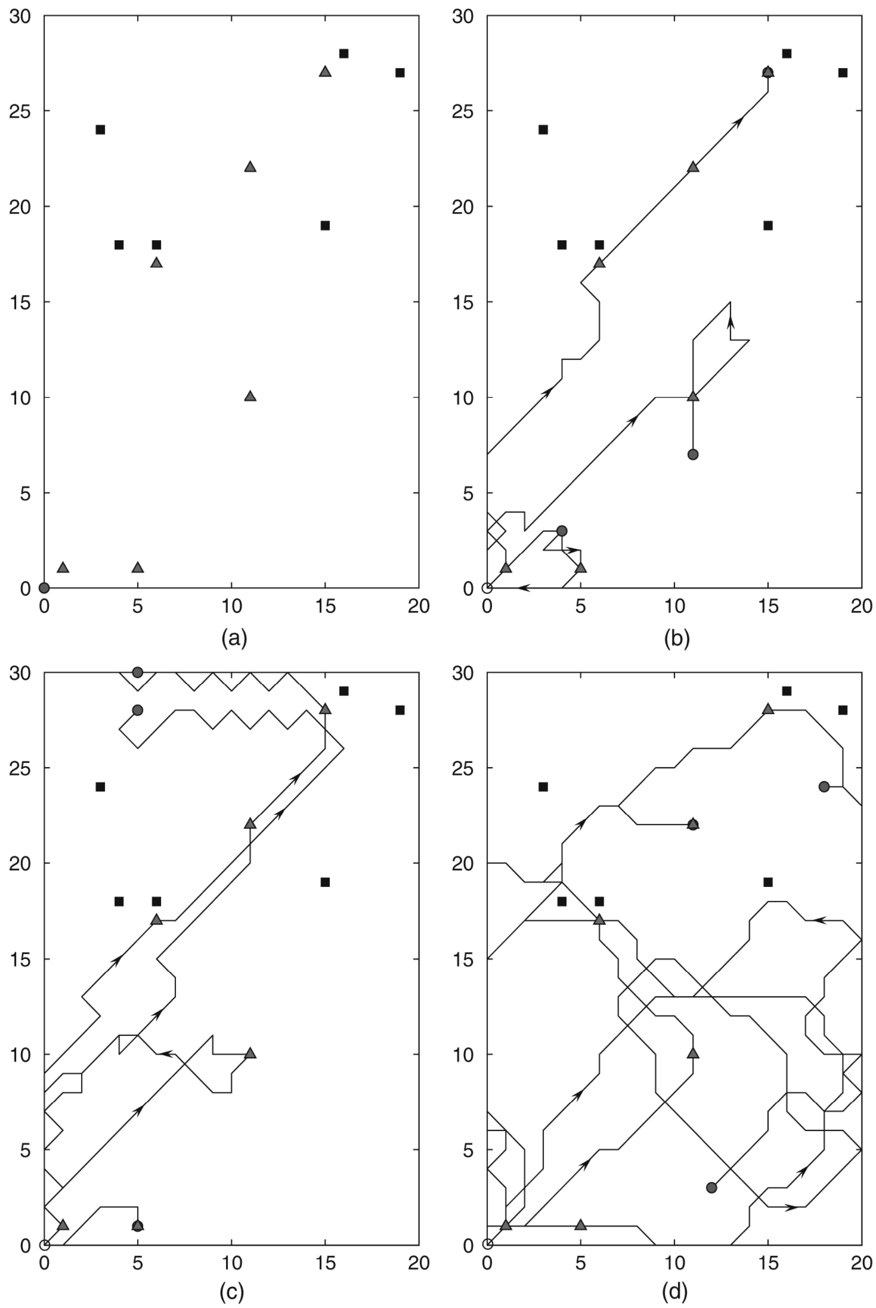

Figure 10. The initial status and final trajectories of robots with the same starting location: (a) the initial status; (b) the trajectories based on FCGA; (c) based on PCGA; (d) based on a random method.

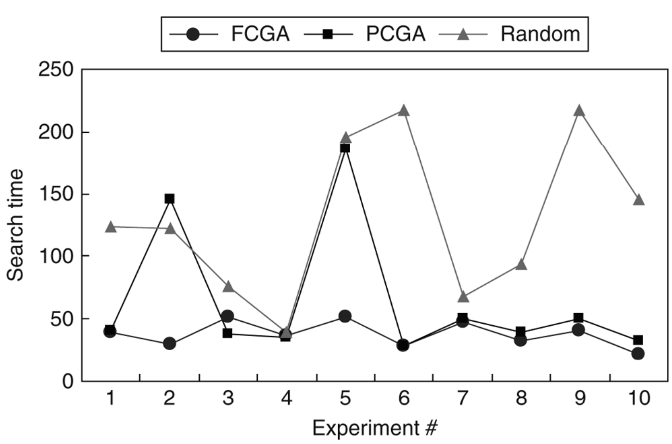

Figure 11. Search time of the experiment, where the robots start from the same location.

Table 4
Results with the Robots Starting at the Same Location

Search Time	FCGA	PCGA	Random
The max value	51	186	196
The min value	22	28	39
The average value	38.0	64.4	108.1
The standard deviation	9.92	54.60	49.60
The failed times	0	0	2

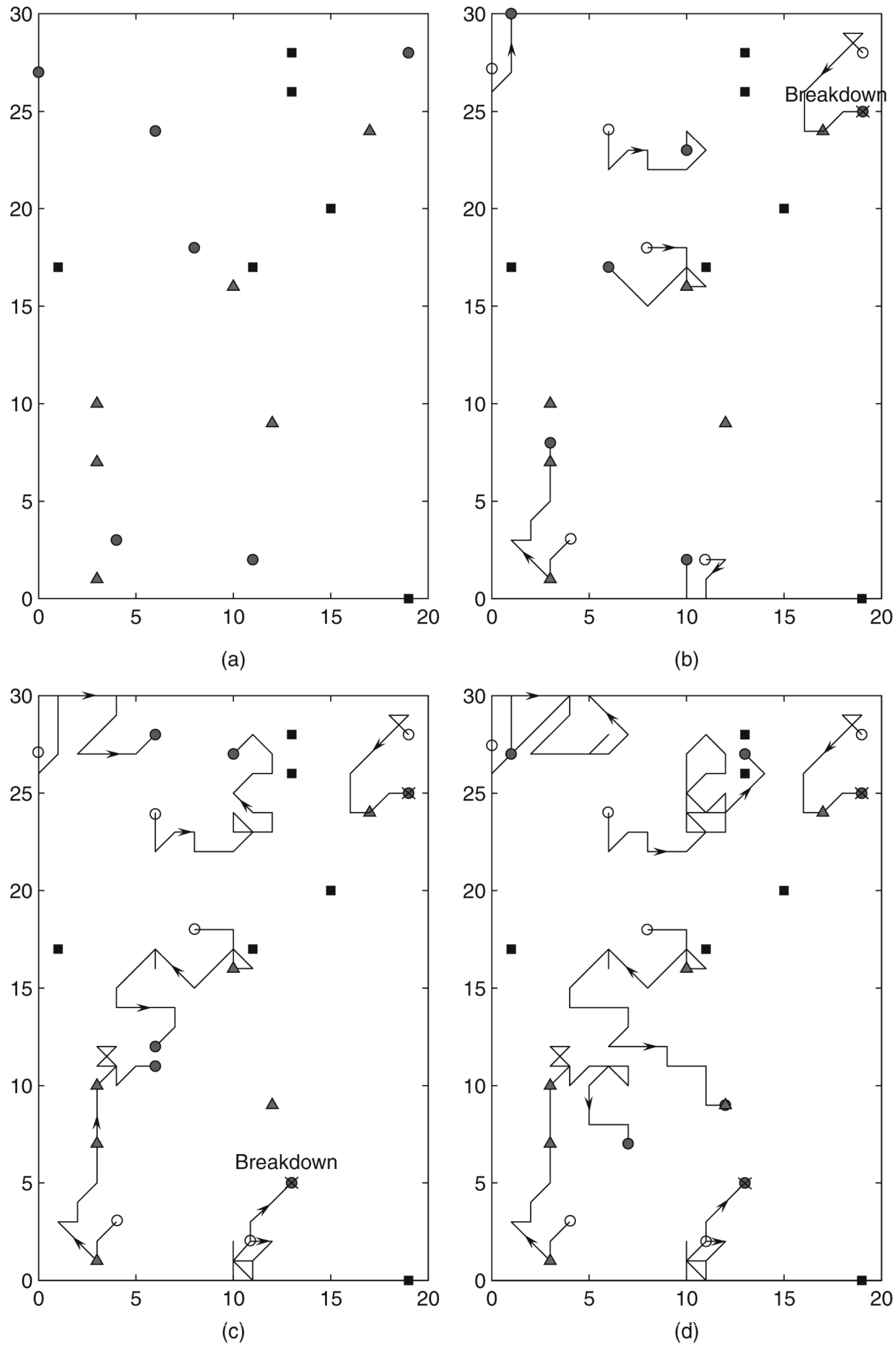

Figure 12. Experiment in an environment with sudden change: (a) the initial status; (b) one robot breaks down; (c) another robot breaks down; (d) the final trajectories.

4.5 Obstacles are Linked Together

To further test the performance of the proposed approach, an experiment in an environment where the obstacles are linked together is conducted. The obstacles are linked with L- and U-shapes in this experiment (see Fig. 13). Figure 13(a) is the initial distribution of the robots, targets, and obstacles, where some targets are surrounded by the obstacles. Figure 13(b) and (c) is the trajectories of the robot team after 10 and 20 steps, respectively. Figure 13(d) is the final trajectories of the robot team.

The results show that the robot team can find all the targets efficiently in the case of linked obstacles.

5. Discussion

The results of the five experiments in Section 4 show that the proposed approach can cooperatively forage for targets

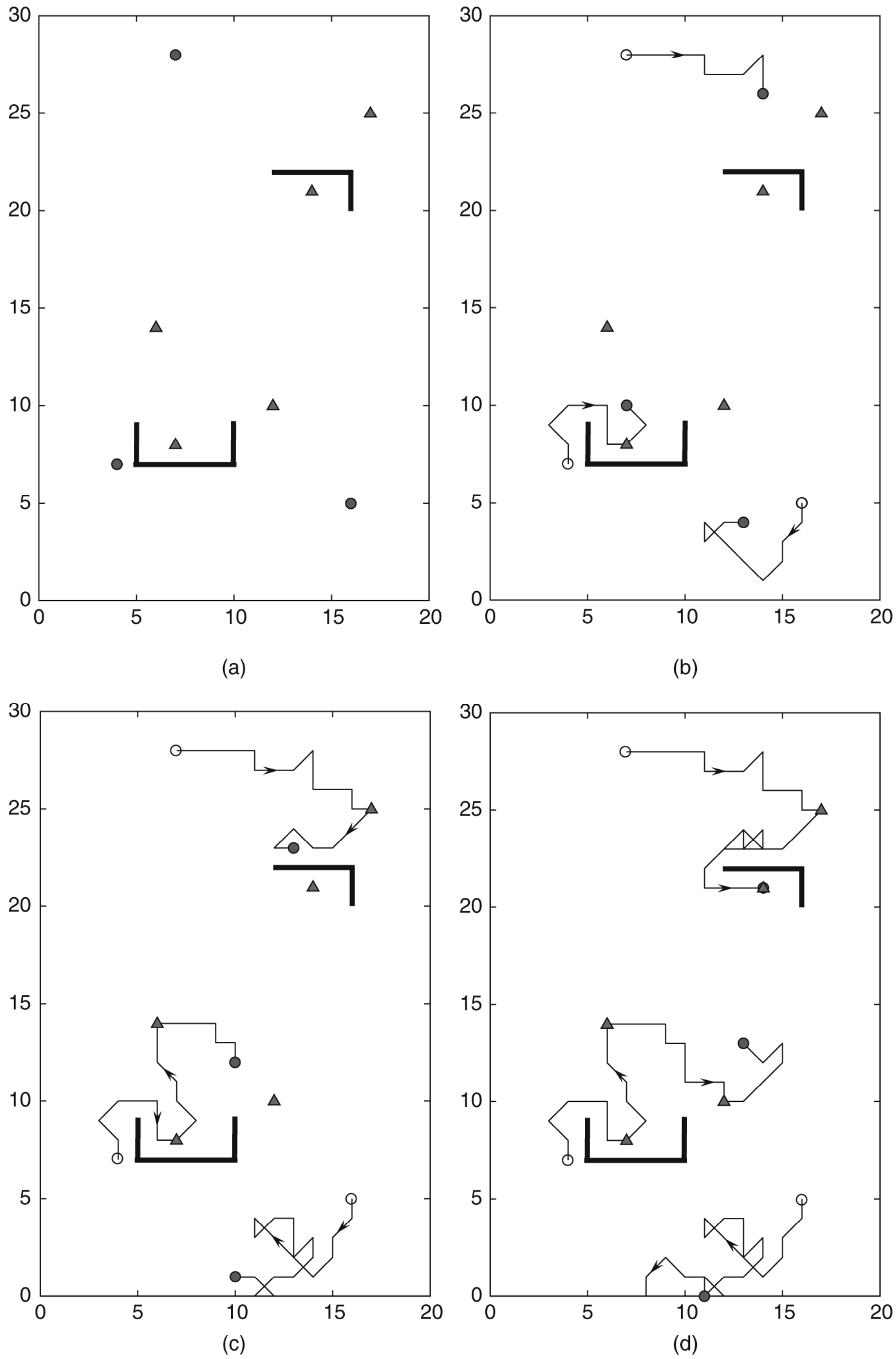

Figure 13. Experiment in an environment with linked obstacles: (a) the initial status; (b) after 10 steps; (c) after 20 steps; (d) the final trajectories.

under various situations. The parameter sensitivity of the proposed approach is discussed in this section.

At first, the parameters of the study scenarios are discussed. There are some important parameters of this scenario, including the area of the environment, the radius of the target information (R_t), and the number of robots in the team. These parameters would be different with the reality application. The fitness function of the FCGA (see (4) and (18)) shows that these parameters will not affect the performance of the proposed approach, although the search time may change with these parameters, for example, the search time will increase with the increasing of the area and will decrease with the increasing of the radius of the target information (R_t). To test the adaptivity of the proposed approach, some experimental studies are conducted. Figure 14 is the results in one experiment carried out under the same conditions of the scenario in Section 4, except the environment area is a 40×40 square.

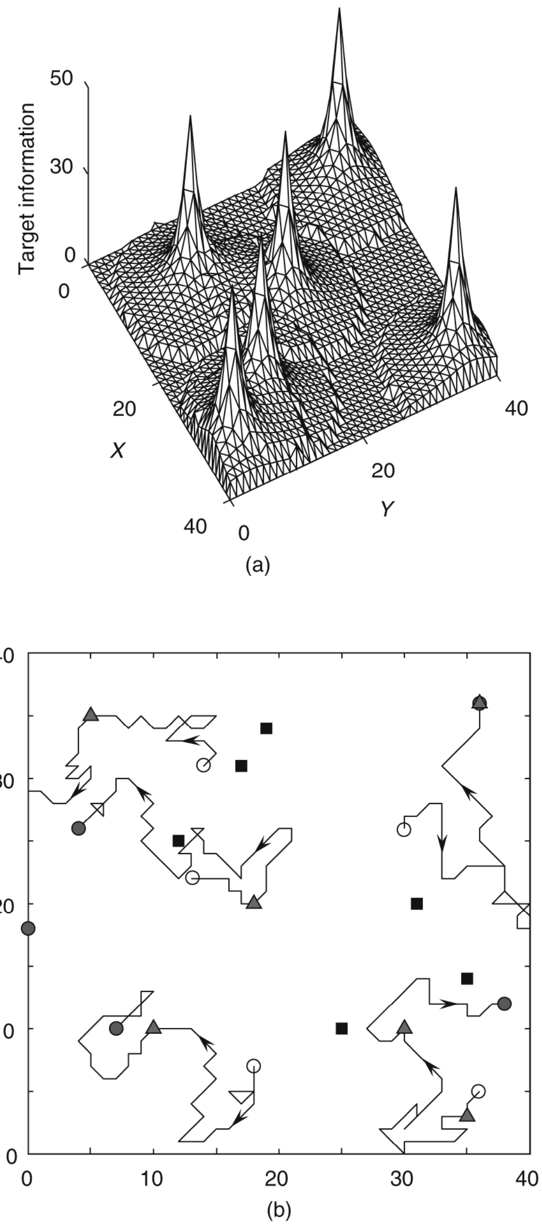

Figure 14. The results of one experiment in a 40×40 environment: (a) the initial distribution of the target information; (b) the final trajectories of robots.

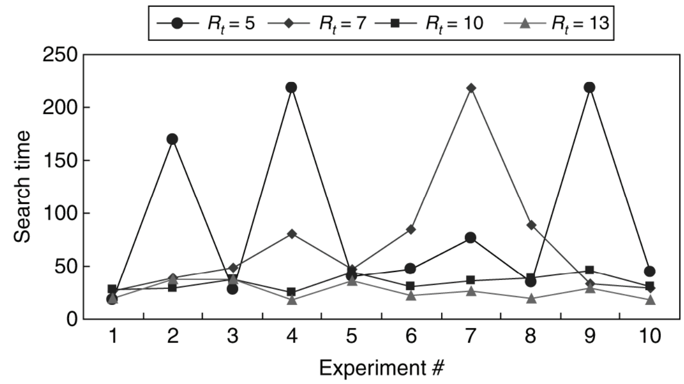

Figure 15. Results of 10 repetitions with different R_t.

Figure 14(a) is the initial distribution of the target information and (b) is the final trajectories of robots in this experiment. Figure 15 is the result of one experiment, in which the effect of different R_t was tested (the results of the first experiment in Section 4 are used as reference, where $R_t = 10$). Figure 16 is the initial distribution of the target information in the experiments with $R_t = 5$, 7, and 13, respectively. The initial distribution of the target information in the experiment with $R_t = 10$ can be seen in Fig. 6(a). The target information distribution in Figs. 14(a) and 16(a) show that the effect of the area of the environment and the radius of the target information is similar. The average search time of 10 repetitions with $R_t = 5$, 7, 10, and 13 are 57.8, 51.1, 34.7, and 26.3, respectively, which prove the deduction above. Figure 17 is the result of one experiment that was carried out under various number of robots and targets: (a) having 6 robots and 6 targets; (b) having 6 robots and 8 targets; (c) having 7 robots and 10 targets; (d) having 8 robots and 5 targets. The results of these experiments show that the proposed approach is capable of dealing with different conditions of the scenario.

Because the sensitivity of the parameters of the PCGA (such as the size of the chaos GA population S_p, the crossover probability P_c, and the mutation probability P_m) are discussed in a lot of literature, only the parameters different from those literature are discussed in this section. Equations (4)–(18) show that there are some additional parameters in the proposed approach: u_1, u_2, u_3, u_4, w_1, and w_2. Equations (12)–(17) show that u_1, u_2, u_3, and u_4 are changed adaptively with the parameters of the scenario, so the sensitivity of them need not be discussed. Here, only w_1 and w_2 are discussed in detail.

The parameter w_1 is used to adjust the motion of the target foraging and the obstacle avoidance. It is related to the largest density of the target information (see the parameter A_t of the scenario in (1), which is set as 50 units in this paper). It is necessary that $w_1 >= mA_t$ to avoid the obstacles, otherwise the robots maybe collide with the obstacle when the target is very near the obstacle. Because w_2 is the parameter to avoid a lot robots staying in the same area, its priority is less than both the target foraging and the obstacle avoidance, so $w_1 >= mA_t > w_2$ is needed. To illustrate the influence of w_1 and w_2 to the proposed approach, some experiments are performed (the results of the first experiment in Section 4 are used as reference, where $w_1 = 6 \times 50 = 300$, and $w_2 = 100$). The other parameters are the same as the experiment introduced at Section 4. The results of these experiments are shown in Fig. 18.

The additional parameters in the proposed approach are very important and should be set with the actual foraging tasks. For example, the parameter w_1 and w_2 should be changed when the number of robots and the largest density of the target information are different. These parameters can be chosen in a very large range of value, as shown in Figs. 14–18. In real conditions, the

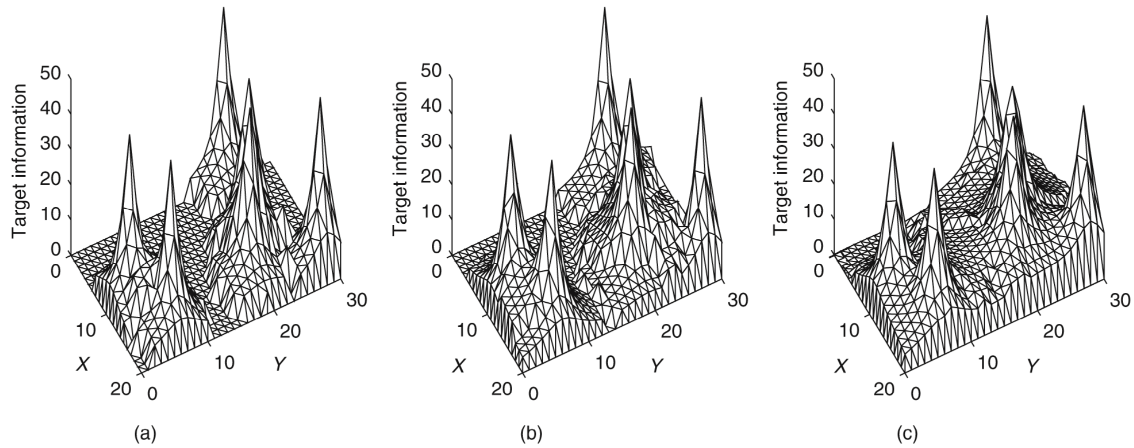

Figure 16. The initial distribution of the target information at different R_t: (a) 5; (b) 7; (c) 13.

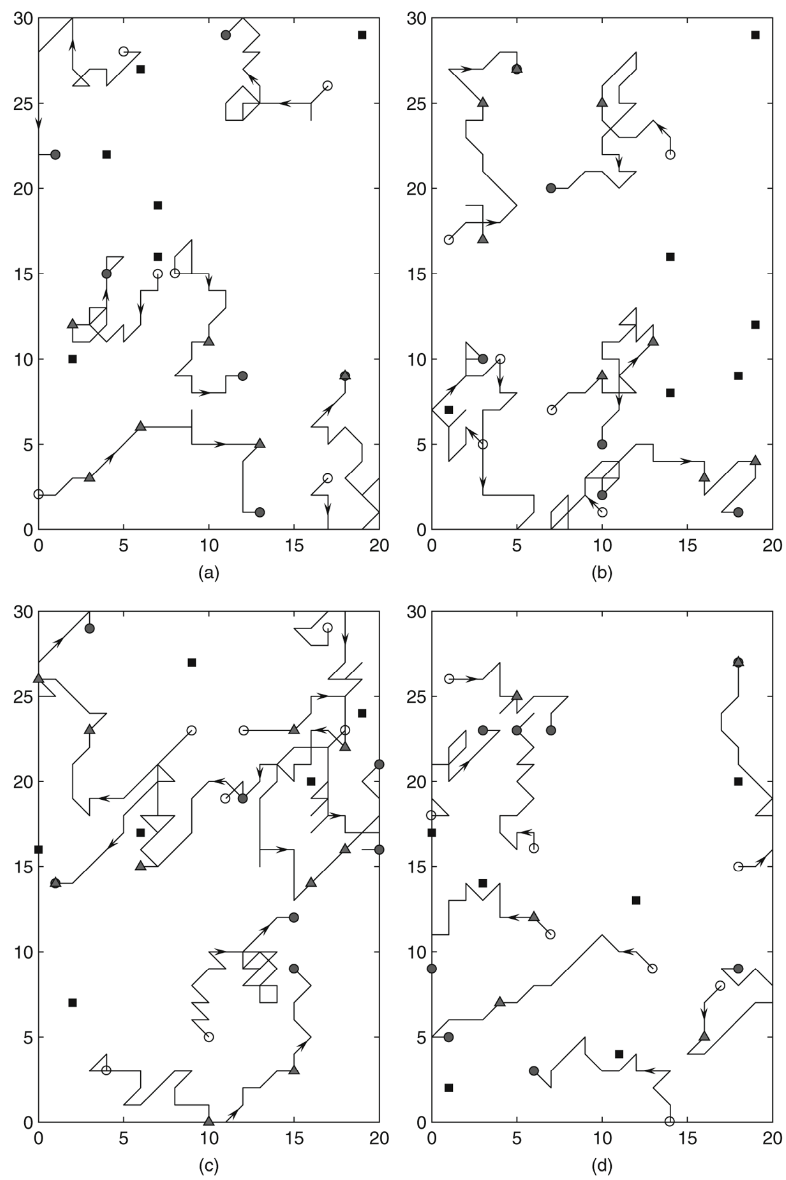

Figure 17. Final trajectories of robots with different number of robots and targets: (a) having 6 robots and 6 targets; (b) 6 robots and 8 targets; (c) 7 robots and 10 targets; (d) 8 robots and 5 targets.

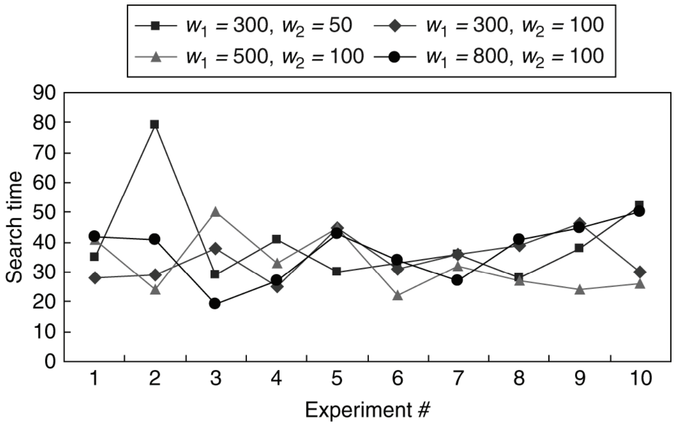

Figure 18. Results of 10 repetitions with different w_1 and w_2.

proposed approach is possibly subjected to several restrictions that can turn it less usable, such as the location and the communication problem of the robots, and the discontinuities of the target information. These issues have been addressed in recent literature [8, 42].

6. Conclusions

Dynamic cooperative task allocation of multi-robot system is investigated. A novel FCGA is proposed. The proposed approach can deal with various situations autonomously and find all the targets efficiently. Integrated with fuzzy logic, the proposed approach has several interesting features and advantages. It can automatically disperse to different areas when all the robots are grouped closely or start from the same location. In addition, it can deal with uncertainties. Parameters of the proposed approach change adaptively with the environment and task, and can be chosen in a wide range. The feasibility and efficiency of the proposed approach are discussed and illustrated through experiment studies. The proposed approach is applicable of dynamically allocating tasks in various unknown environments such as the fire disaster response and the radioactive object searching.

Acknowledgements

The authors would like to thank the Hohai University Innovation Foundation (XZX/09B002-02) and the Open Fund of Jiangsu Key Laboratory of Power Transmission & Distribution Equipment Technology (2010JSSPD02) for their support of this paper. The authors would also like to thank the editors and the reviewers for their helpful comments.

References

[1] R.M. Murray, Recent research in cooperative control of multi-vehicle systems, *Transactions of the ASME Journal of Dynamic Systems, Measurement and Control*, *129*(5), 2007, 571–583.

[2] H. Duman and H. Hu, United we stand, divided we fall: Team formation in multiple robot applications, *International Journal of Robotics and Automation*, *16*(4), 2001, 153–161.

[3] S. Sariel, T. Balch, and N. Erdogan, Naval mine countermeasure missions, *IEEE Robotics and Automation Magazine*, *15*(1), 2008, 45–52.

[4] A. Farinelli, L. Iocchi, and D. Nardi, Multirobot systems: A classification focused on coordination, *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, *34*(5), 2004, 2015–2028.

[5] M.J. Matarić, G.S. Sukhatme, and E.H. Østergaard, Multi-robot task allocation in uncertain environments, *Autonomous Robots*, *14*(2–3), 2003, 255–263.

[6] H. Ghenniwa, J. Eze, and W. Shen, Physical robot agents: Coordinated intelligent and rational agents for collaborative robots, *International Journal of Robotics and Automation*, *21*(2), 2006, 73–80.

[7] H. Li, S.X. Yang, and M.L. Seto, Neural-network-based path planning for a multirobot system with moving obstacles, *IEEE Transactions on Systems, Man, and Cybernetics, Part C*, *39*(4), 2009, 410–419.

[8] K. Tanaka and E. Kondo, A scalable localization algorithm for high dimensional features and multi robot systems, *Proc. IEEE International Conference on Networking, Sensing and Control*, Sanya, China, 2008, 920–925.

[9] K.S. Kwok, B.J. Driessen, C.A. Phillips, and C.A. Tovey, Analyzing the multiple-target-multiple-agent scenario using optimal assignment algorithms, *Journal of Intelligent and Robotic Systems: Theory and Applications*, *35*(1), 2002, 111–122.

[10] S.X. Yang and M. Meng, Neural network approaches to dynamic collision-free robot trajectory generation, *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, *31*(3), 2001, 302–318.

[11] S.K. Chalup, C.L. Murch, and M.J. Quinlan, Machine learning with AIBO robots in the four-legged league of RoboCup, *IEEE Transactions on Systems, Man, and Cybernetics, Part C*, *37*(3), 2007, 297–310.

[12] B.P. Gerkey and M.J. Matari, A formal analysis and taxonomy of task allocation in multi-robot systems, *The International Journal of Robotics Research*, *23*(9), 2004, 939–954.

[13] S.L. Smith and F. Bullo, Monotonic target assignment for robotic networks, *IEEE Transactions on Automatic Control*, *54*(9), 2009, 2042–2057.

[14] Y. Tian, M. Yang, X. Qi, and Y. Yang, Multi-robot task allocation for fire-disaster response based on reinforcement learning, *Proc. International Conference on Machine Learning and Cybernetics*, Baoding, China, 2009, 2312–2317.

[15] R. Akkiraju, P. Keskinocak, S. Murthy, and F. Wu, An agent-based approach for scheduling multiple machines, *Applied Intelligence*, *14*(2), 2001, 135–144.

[16] Y. Ma, Z. Cao, X. Dong, C. Zhou, and M. Tan, A multi-robot coordinated hunting strategy with dynamic alliance, *Proc. 2009 Chinese Control and Decision Conference*, Guilin, China, 2009, 2338–2342.

[17] R. Wegner and J. Anderson, Agent-based support for balancing teleoperation and autonomy in urban search and rescue, *International Journal of Robotics and Automation*, *21*(2), 2006, 120–127.

[18] Z. Cao, M. Tan, L. Li, N. Gu, and S. Wang, Cooperative hunting by distributed mobile robots based on local interaction, *IEEE Transactions on Robotics*, *22*(2), 2006, 403–407.

[19] K. Shah and Y. Meng, Communication-efficient dynamic task scheduling for heterogeneous multi-robot systems, *Proc. IEEE International Symposium on Computational Intelligence in Robotics and Automation*, Jacksonville, FI, USA, 2007, 230–235.

[20] B. Kim, S.S. Heragu, R.J. Graves, and A.S. Onge, A hybrid scheduling and control system architecture for warehouse management, *IEEE Transactions on Robotics and Automation*, *19*(6), 2003, 991–1001.

[21] A. Zhu and S.X. Yang, A som-based multi-agent architecture for multirobot systems, *International Journal of Robotics and Automation*, *21*(2), 2006, 91–99.

[22] J.K. Archibald and R.L. Frost, A decentralized approach to multi-robot formation initialization, *International Journal of Robotics and Automation*, *22*(4), 2007, 304–312.

[23] K. Lerman, C. Jones, A. Galstyan, and M.J. Matarti, Analysis of dynamic task allocation in multi-robot systems, *International Journal of Robotics Research*, *25*(3), 2006, 225–241.

[24] J.T. Feddema, C. Lewis, and D.A. Schoenwald, Decentralized control of cooperative robotic vehicles: Theory and application, *IEEE Transactions on Robotics and Automation*, *18*(5), 2002, 852–864.

[25] J. Casper and R.R. Murphy, 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*, *33*(3), 2003, 367–385.

[26] A. Zhu and S.X. Yang, A neural network approach to dynamic task assignment of multi-robots, *IEEE Transactions on Neural Network*, *17*(5), 2006, 1278–1287.

[27] A. Arsie, K. Savla, and E. Frazzoli, Efficient routing algorithms for multiple vehicles with no explicit communications, *IEEE Transactions on Automatic Control*, *54*(10), 2009, 2302–2317.

[28] T. Zheng and L.Yang, Optimal ant colony algorithm based multi-robot task allocation and processing sequence scheduling, *Proc. 7th World Congress on Intelligent Control and Automation*, Chongqing, China, 2008, 5687–5692.

[29] N. Lemmens, S. De Jong, K. Tuyls, and A. Nowe, Bee behaviour in multi-agent systems (a bee foraging algorithm), *Proc. 7th European Symposium on Adaptive and Learning Agents and Multi-Agent Systems*, Maastricht, Netherlands, 2007, 145–156.

[30] I. Hassanzadeh and S. Sadigh, Path planning for a mobile robot using fuzzy logic controller tuned by GA, *Proc. 2009 6th International Symposium on Mechatronics and its Applications (ISMA09)*, Sharjah, United Arab Emirates, 2009, 1–5.

[31] H. Kawanaka, T. Yoshikawa, and S. Tsuruoka, Acquisition of fuzzy control rules for a mobile robot using genetic algorithm, *Proc. 6th International Workshop on Advanced Motion Control. Proceedings (Cat. No. 00TH8494)*, Nagoya, Japan, 2000, 507–512.

[32] Y. Meng and J. Gan, Self-adaptive distributed multi-task allocation in a multi-robot system, *Proc. IEEE Congress on Evolutionary Computation*, Hong Kong, China, 2008, 398–404.

[33] J.M. Wilson, A genetic algorithm for the generalised assignment problem, *Journal of the Operational Research Society*, *48*(8), 1997, 804–809.

[34] J. Gong, J. Qi, G. Xiong, H. Chen, and W. Huang, A GA based combinatorial auction algorithm for multi-robot cooperative hunting, *Proc. 2007 International Conference on Computational Intelligence and Security*, Harbin, China, 2007, 137–141.

[35] A. Loredo-Flores, E.J. González-Galván, J.J. Cervantes-Sánchez, and A. Martínez-Soto, Optimization of industrial, vision-based, intuitively generated robot point-allocating tasks using genetic algorithms, *IEEE Transactions on Systems, Man, and Cybernetics, Part C*, *38*(4), 2008, 600–608.

[36] A.M. Tehrani, M.S. Kamel, and A.M. Khamis, Fuzzy reinforcement learning for embedded soccer agents in a multi-agent context, *International Journal of Robotics and Automation*, *21*(2), 2006, 110–119.

[37] M. Mucientes, R. Iglesias, C.V. Regueiro, A. Bugarín, P. Cariñena, and S. Barro, Fuzzy temporal rules for mobile robot guidance in dynamic environments, *IEEE Transactions on Systems, Man, and Cybernetics, Part C*, *31*(3), 2001, 391–398.

[38] H. Bezine, H.N. Derbel, and A.M. Alimi, Fuzzy control of robot manipulators: some issues on design and rule base size reduction, *Engineering Applications of Artificial Intelligence*, *15*(5), 2002, 401–416.

[39] M. Zhou and S. Sun, *Theory and application of genetic algorithm* (Beijing: National Defense Industry Press in China, 1996).

[40] M. Gao, J. Xu, J. Tian, and H. Wu, Path planning for mobile robot based on chaos genetic algorithm, *Proc. Fourth International Conference on Natural Computation*, Jinan, China, 2008, 409–413.

[41] J.R. Jang, C. Sun, and E. Mizutani, *Neuro-Fuzzy AND Soft Computing: A computational approach to learning and machine intelligence* (New York: Prentice Hall, 1997).

[42] G.A. Caro, F. Ducatelle, and L.M. Gambardella, Wireless communications for distributed navigation in robot swarms, *Proc. Applications of Evolutionary Computing, EvoWorkshops 2009*, Tubingen, Germany, 2009, 21–30.

Biographies

Jianjun Ni received his Ph.D. degree in School of Information and Electrical Engineering from China University of Mining and Technology, Xuzhou, China, in 2005. Currently he is an Associate Professor of College of Computer and Information at Hohai University, China. He was a Visiting Professor with the Advanced Robotics and Intelligent Systems (ARIS) Laboratory at the University of Guelph in Canada from November 2009 to October 2010. He has published over 20 papers in related international conferences and journals. His research interests include fuzzy systems, neural networks, robotics, machine intelligence, and multi-agent system.

Simon X. Yang received the B.Sc. degree in engineering physics from Beijing University, China in 1987, the first of two M.Sc. degrees in biophysics from Chinese Academy of Sciences, Beijing, China, in 1990, the second M.Sc. degree in electrical engineering from the University of Houston, USA, in 1996, and the Ph.D. degree in electrical and computer engineering from the University of Alberta, Edmonton, Canada in 1999. Dr. Yang joined the School of Engineering at the University of Guelph, Canada, in 1999. Currently he is a Professor and the Head of the Advanced Robotics & Intelligent Systems (ARIS) Laboratory at the University of Guelph in Canada. His research interests include intelligent systems, robotics, sensors and multi-sensor fusion, wireless sensor networks, control systems, soft computing, and computational neuroscience.