Intelligent Automation and Soft Computing, Vol. 14, No. 1, pp. 61-72, 2008

MULTI-ROBOT HUNTING IN DYNAMIC ENVIRONMENTS

ZHIQIANG CAO[1], NONG GU[2], MIN TAN[1], SAEID NAHAVANDI[2], XIAOFENG MAO[3] AND ZHENYING GUAN[2]

[1]*Laboratory of Complex Systems and Intelligence Science*
Institute of Automation
Chinese Academy of Sciences
Beijing 100080, P. R. China

[2]*School of Engineering and Information Technology*
Deakin University
Geelong, VIC 3217, Australia

[3]*Mechanical Engineering Department*
Pennsylvania State University
University Park, PA 16802

E-mail: zqcao@compsys.ia.ac.cn, ng@deakin.edu.au, tan@compsys.ia.ac.cn, saeid.nahavandi@deakin.edu.au, xum102@psu.edu

ABSTRACT—This paper is concerned with multi-robot hunting in dynamic environments. A BCSLA approach is proposed to allow mobile robots to capture an intelligent evader. During the process of hunting, four states including *dispersion-random-search*, *surrounding*, *catch* and *prediction* are employed. In order to ensure each robot appropriate movement in each state, a series of strategies are developed in this paper. The *dispersion-search* strategy enables the robots to find the evader effectively. The *leader-adjusting* strategy aims to improve the hunting robots' response to environmental changes and the *outflank* strategy is proposed for the hunting robots to force the evader to enter a besieging circle. The *catch* strategy is designed for shrinking the besieging circle to catch the evader. The *predict* strategy allows the robots to predict the evader's position when they lose the tracking information about the evader. A novel collision-free motion strategy is also presented in this paper, which is called the *direction-optimization* strategy. To test the effect of cooperative hunting, the target to be captured owns a *safety-motion* strategy, which helps it to escape being captured. The computer simulations support the rationality of the approach.

Key Words: Multi-robot, hunting, besieging circle, dynamic environment

1. INTRODUCTION

The past decades have seen an increasing research interest in multiple robots [1]-[3] in a wide variety of fields, ranging from manufacturing to the military field. This resulted from some basic facts. Many tasks are inherently distributed, either in space, time, or functionality, which cannot be accomplished by a single robot. Moreover, even for some specific tasks that can be completed by a single robot, the quality of the solution may be improved if multiple robots are involved. Although utilizing multiple robots may bring the system properties of parallelism, robustness, and redundancy, the difficulties of tasks are actually increased if the robots are not properly organized. Some challenging issues should be considered carefully. In this paper, we focus on task modeling, cooperative strategies and collision avoidance in the context of multi-robot hunting.

The hunting task is a particular challenge due to its dynamic nature and unknown irregular motion of the evader (the target to be captured). So far, some progress has been made. Yamaguchi used a feedback-control law for coordinating the motion of multiple mobile robots to capture/enclose a target by troop formation [4]. A control method, named as linear autonomous system, was proposed to generate the shape of a group consisting of multiple robots to capture a target [5]. An approach based on multiple objective behavior coordination was presented by Pirjanian and Mataric to accomplish the cooperative target acquisition task [6]. The approach provides mechanisms for distributed command fusion across a group of robots to pursue multiple goals of multiple robots in parallel. Other related researches include the pursuit game [7], which is an important topic in distributed artificial intelligence and discusses two kinds of agents: predator (pursuer or hunter) and prey (evader or invader). The objective of this game is to make multiple predators pursue and finally capture the prey. In order to capture the prey, a metalevel coordination strategy to implement an adaptive organization has been proposed in [8]. An approach to learn cooperative behavior of agents was investigated in [9], which is based on classifying situations and the nearest-neighbor rule. In [10], two pursuit policies have been considered. The greedy policy requires each pursuer to approach the adjacent cell with the highest probability of containing an evader, while the global maximum policy intends to search the entire environment. The case of supervisory agents (such as a helicopter) which can estimate the evader's position has been investigated [10, 11]. In addition, reinforcement learning has also been considered to complete the task [12]. Another important issue should be considered concerning the intelligence of the prey. The prey may move randomly regardless of environmental change, on the other hand, it may escape by an appreciate strategy. The possible strategies for the prey range from simply moving in one direction over random moves to elaborate strategies like maximizing the distance from the nearest, several or all predators [9].

In a typical hunting task, multiple mobile robots with limited sensory capabilities cooperatively hunt a tricky evader of some intelligence in dynamic model-free environments. *Direct seizing* (DS) approach is a solution to the problem. Some positions, which we refer to as *catch* points below, are determined by one special robot named as the leader. After the leader is selected properly, these *catch* points can be located, which are very close to the detected evader. When most of the *catch* points are occupied by the robots, the evader cannot move any more. Each robot needs to decide which *catch* point is suitable for it. Usually, the approach with an adjusting leader (DSLA) has more adaptability to dynamic environments than that with a fixed leader (DSLF). The DSLA approach may obtain a good result in many environments, however, the *drag* problem, where the group shape is destroyed after a long chase, is intractable. This will increase the interference among the robots, which prolongs the completion time. To solve this problem, in this paper, an approach called *Besieging Circle Shrinking with Leader Adjusting* (BCSLA) is proposed. Multiple robots endeavor to distribute evenly on a besieging circle based on an adjustable leader and the detected evader before the system begins to shrink. Each robot has one appropriate position called a *besieging* point in the besieging circle. The distance between each *besieging* point and the evader is greater than that from any *catch* point to the evader. Therefore, the group shape may be better kept even after a long chase.

The paper is organized as follows. Section 2 presents the hunting behavior. Task modeling is introduced in section 3. Section 4 proposes a series of strategies for hunting robots, which enable them to catch the evader. Section 5 discusses the *safety-motion* strategy, which is propitious for the evader to escape. Simulation examples are given in section 6 and section 7 concludes the paper.

2. HUNTING BEHAVIOR

Let each robot be labeled as $R_i \in \aleph$, where $i = \{1..N\}$. Besides a common coordinate system W is defined, the robots will establish the polar coordinate systems Σ_{ci}, Σ_{eveni} and Σ_{Ri} (see Figure 1 [4]), which will be described in detail in section 4. The evader also needs to establish a coordinate system Σ_e for its motion control.

During the process of hunting, each hunting robot may acquire the positions of each other by communication or perception. When the evader is detected by at least one robot, the location of the evader is shared among the robots. In order to avoid any possible collisions, the robot adopts a range sensor model S_{range} to perceive the environment. In addition, the robots can recognize each

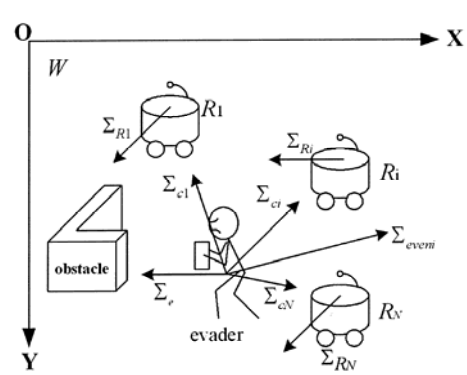


Figure 1. The coordinate systems [4].

other and the evader. Assume that the robots and the evader can see each other within the same range as that of S_{range}.

3. TASK MODELING

The hunting task, from the group's perspective, is modeled by M, which reflects the transition of states. Denote M by a quadruple $(E_s, E_c, Q_{st}, \sigma)$, where

- $E_s = \{e_{s1}, e_{s2}, ..., e_{sm}\}$ describing the set of perceptual events with m elements.
- $E_c = \{e_{c1}, e_{c2}, ..., e_{ck}\}$ is the collection of finite communication events with k elements.
- Q_{st} is a discrete state set with t elements, denoted by $\{q_1, q_2, ..., q_t\}$.
- σ is a mapping from one state to another state, expressed by $\sigma(q_s, E') = q_p \in Q_{st}$, where $q_s \in Q_{st}$, $E' \subset E$, $E_s \times E_c \to E = \{e_{sc1}, e_{sc2}, ..., e_{scn}\}$.

In the sequel, M is employed to model the whole hunting task.

The first problem for the system is to find the evader. A strategy for the robots to make formation and search the entire environment is feasible in consideration of limited sensor ranges. The strategy goes well when the robots search the environment, but the occasions that the robots chase the evader only from its one side often happen, which perhaps lead to a longer task execution time. The *dispersion-search* strategy appears to be a good substitute. Regardless of initial distribution of robots, it can result in an effective search. When one robot sees other robots, it keeps away from them, otherwise, it moves randomly. This searching stage will last if no one robot sees the evader (e_{sc1}). When the evader is detected (e_{sc2}), the pursuit begins. According to the BCSLA approach, the strategy to adopt is that after the evader enters a besieging circle formed by the robots, that is to say, the condition of shrinking the besieging circle is satisfied (e_{sc3}), the robots shrink the circle to capture the evader, or else (e_{sc4}), the robots will endeavor to besiege the evader firstly. Because of the complexity of the task and its environment, the robots may lose track of the evader. In this case, the system should make the decision based on the recorded evader position to find the evader again. If the evader is seen again within certain steps (e_{sc5}), say, $N_{prediction}$, the robots continue to pursue the evader, otherwise (e_{sc6}), they will re-search the environment. The above process is repeated until the task is completed. In addition, if one robot fails to broadcast the related information that is expected by other robots or the information is abnormal, the robot is considered to be abnormal. Under this circumstance, the system should adjust itself dynamically to adapt to unexpected changes.

Based on the above description, Q_{st} may contain four states: *dispersion-random-search* (q_1), *surrounding* (q_2), *catch* (q_3) and *prediction* (q_4). The possible state transitions are as follows.

$$\sigma(q_1, \{e_{sc2}, e_{sc4}\}) = q_2,\ \sigma(q_1, \{e_{sc2}, e_{sc3}\}) = q_3,\ \sigma(q_2, \{e_{sc2}, e_{sc3}\}) = q_3,$$
$$\sigma(q_2, \{e_{sc1}\}) = q_4,\ \sigma(q_4, \{e_{sc5}, e_{sc4}\}) = q_2,\ \sigma(q_4, \{e_{sc5}, e_{sc3}\}) = q_3,\quad \sigma(q_4, \{e_{sc6}\}) = q_1$$

4. ROBOT CONTROL STRATEGIES

Based on task modeling supported by the BCSLA approach, a series of strategies are deduced so that each robot can make its appropriate decision in each state.

4.1 Dispersion-search Strategy

The robot is in the *dispersion-random-search* state and endeavors to find the evader effectively. This strategy allows an individual robot to keep away from other robots when it sees them (Boolean variable b_1=1), and in other cases (b_1=0) it moves randomly, which is shown in equation 1. The direction $[x_r, y_r]^T$ will be sent to the *direction-optimization* strategy, which combines it with perceptual information to control the robot.

$$\begin{bmatrix} x_r \\ y_r \end{bmatrix} = \begin{cases} \begin{bmatrix} p_{cx} - p_{nearx} \\ p_{cy} - p_{neary} \end{bmatrix} / \sqrt{\left(p_{cx} - p_{nearx}\right)^2 + \left(p_{cy} - p_{neary}\right)^2} & b_1 = 1 \\ \begin{bmatrix} \cos(-\tau \cdot sig\rho) & -\sin(-\tau \cdot sig\rho) \\ \sin(-\tau \cdot sig\rho) & \cos(-\tau \cdot sig\rho) \end{bmatrix} \cdot \begin{bmatrix} x_d \\ y_d \end{bmatrix} & b_1 = 0 \end{cases} \tag{1}$$

where $\left[p_{cx}, p_{cy}\right]^T$, $\left[p_{nearx}, p_{neary}\right]^T$ are coordinates of the robot and another one nearest to it, respectively; $\left[x_d, y_d\right]^T$ refers to the robot's heading; τ is an angle randomly rotated; $\rho(\rho \in [0,1])$ is a random number and $sig\rho = \begin{cases} -1 & 0.5 > \rho \geq 0 \\ 1 & 1 \geq \rho \geq 0.5 \end{cases}$.

4.2 Leader-adjusting Strategy

As mentioned above, after the evader is detected, the system manages to form a besieging circle to capture it. Let r_s denote the circle's radius, which is influenced by the maximum sensing range $s_{\max}$ and the radius r of robot. When the conditions for the besieging circle to shrink are satisfied, the circle will shrink to one with a radius of r_c . No matter which state (*surrounding*, *catch* or *prediction*) the robot is in, the system always determines a robot as the leader for the purpose of ideal positions. The leader is meaningless when the system is in the *dispersion-random-search* state and fixed in the *catch* state.

We denote with $P_r(i)$ the positions of all normal robots, where $i = \{1..N\}$ and $N >= 3$. We denote with P_T , C_T the current position of the evader and its center, respectively. $\overrightarrow{P_T P_r(i)}$ provide the direction vectors from the evader to the center of each robot, where $i = \{1..N\}$. Before describing the *leader-adjusting* strategy, an angular matching algorithm and the conditions for the besieging circle to shrink are introduced. For the purpose of describing a common angular matching algorithm, a virtual variable r_{sc} is adopted instead of r_s or r_c . In actual decision-making, r_s and r_c will be chosen according to different states.

4.2.1 Angular Matching Algorithm ($AM(R_i, r_{sc})$)

When robot R_i ($R_i \in \aleph$) is regarded as the leader, each robot obtains its ideal motion position as follows:

step1_1: establish the polar coordinate system Σ_{ci} whose pole is C_T with the polar axis direction of $\overrightarrow{P_T P_r(i)}$.

step1_2: calculate the coordinates $P_c^{mi}(\lambda_m, \varphi_m)$ for all robots in Σ_{ci} where $\varphi_m \in [0, 2\pi)$ and $m = \{0..N-1\}$.

step1_3: generate the ideal motion positions of all robots. The positions are N points evenly distributed in a circle centered at C_T with a radius of r_{sc} . These points are defined as $P_d^{ni}(r_{sc}, \phi_n)$, where $\phi_n = \frac{2\pi}{N} \cdot n$ and $n = \{0..N-1\}$. Accordingly, these coordinates are $P_w^i(x_n, y_n)$ in common coordinate system W .

step1_4: each robot determines a proper ideal position. The robot with a larger φ_m selects the position with a larger ϕ_n .

4.2.2 Conditions for the Besieging Circle to Shrink

Conditions for the system to shrink the besieging circle are considered from two aspects: the angle constraint and the distance constraint. The evader has entered the besieging circle if and only if these two constraints are met simultaneously. Considering the leader R_i , the constraints are described as follows.

(1) Angle constraint

step2_1: when N is an even number, obtains $P_r(\min)$, which is $P_r(l)$ that makes the angle between $\overrightarrow{P_T P_r(i)}$ and $\overrightarrow{P_T P_r(l)}$ minimal, where $l = \{1..N\}$ and $l \neq i$. Then establishes a polar coordinate system Σ_{eveni} whose pole and polar axis direction are C_T and $\frac{\overrightarrow{P_T P_r(i)} + \overrightarrow{P_T P_r(\min)}}{2}$, respectively.

step2_2: If N is an odd number, Σ_{ci} is regarded as Σ_{eveni}.

step2_3: calculate the coordinates $P_b^{mi}(\lambda_m, \gamma_m)$ for all robots in Σ_{eveni}, where $\gamma_m \in [0, 2\pi)$ and $m = \{0..N-1\}$.

step2_4: When $\exists \gamma_k \in \left[\frac{\pi}{2}, \pi\right) \cap \exists \gamma_j \in \left[\pi, \frac{3\pi}{2}\right)$ occurs, where $k, j = \{0..N-1\}$, the angle constraint is satisfied.

(2) Distance constraint

step3_1: judge whether the angle constraint is satisfied. If it is, the robots obtain their ideal positions by calling $AM(R_i, r_c)$.

step3_2: calculate the distances between the current positions and corresponding ideal positions of robots.

step3_3: If each distance is less than a certain value, say, $k_d(s_{\max} - 3r)$, the distance constraint is satisfied, or else it means that one or more robots are far away from their ideal positions and the besieging circle is not allowed to shrink.

4.2.3 Leader Selection

If the leader is not appropriate for the current situation, it should resign its leadership to other robots for better group movement. Take the case in Figure 2 as an example. Assume that the robot of ID 1 is the leader of the system. In this case, the robots of ID 1, 3, 2, 5, 4 will move towards the positions 5, 1, 2, 3, 4 (expressed by sign ×), respectively. Such movement possibly confuses the system. If the robot of ID 2 is selected as the new leader, a good situation will arise and the robots of ID 1, 3, 2, 5, 4 will move toward positions d, e, a, b, c (expressed by sign +), respectively.

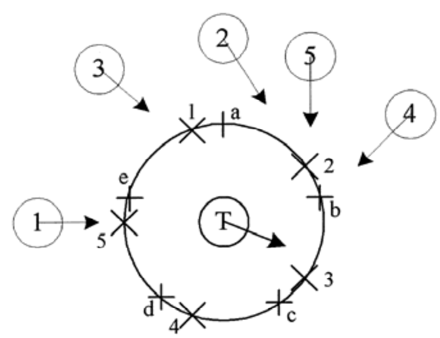


Figure 2. The sketch map of leader adjustment.

The process of leader selection is described as follows.

step4_1: determine all candidates. For $\forall R_i$, where $i = \{1..N\}$, the polar coordinates of all robots except R_i in Σ_{ci} are obtained. Let n_{rl} and n_{rr} denote the numbers of robots under $\varphi_m \in [0, \pi)$ and $\varphi_m \in [\pi, 2\pi)$, respectively and $n_{rl} + n_{rr} = N - 1$. If the condition $n_{rl}, n_{rr} \in \left[\mathrm{int}(\frac{N-1}{2}),\ \mathrm{int}(\frac{N-1}{2}) + 1\right]$ is satisfied, it shows that robot R_i divides all other robots into two subgroups with similar or the same number. The robot is considered to be a leader candidate.

step4_2: Each robot assumes each candidate as the leader to obtain all ideal positions of the robots and calculates the maximum value d_{lm} of the distances between these positions and corresponding current positions. In order to avoid switching the leader frequently, for the current leader, set $d_{lm} = k_{as} d_{lm} (k_{as} < 1)$. The value is regarded as the selection criterion for each candidate.

step4_3: The robot with a minimum d_{lm} becomes the new leader. When there is not a proper candidate, the leader remains unchanged unless it fails.

Occasionally, some robots perhaps choose different leaders or the same ideal positions. These conflicts can be resolved naturally with the motions of the robots.

4.3 Outflank Strategy

The *outflank* strategy is used when the robot is in the *surrounding* state to endeavor to make the evader enter the besieging circle. According to the selected leader R_s by the *leader-adjusting* strategy, each robot determines its ideal motion position by $AM(R_s, r_s)$.

4.4 Catch Strategy

The *catch* strategy is adopted when the robot is in the *catch* state and the robotic system may catch the evader by reducing the besieging radius. The robot can acquire its ideal position by calling $AM(R_s, r_c)$.

4.5 Predict Strategy

When the robots are in the *surrounding* state, they record the position of a reference robot finding the evader. Because the system possibly lose track of the evader, the motion prediction of the evader is meaningful. In this paper, the robots predict the evader's motion based on the recorded positions of the reference robot and the evader. This is termed as the robot being in the *prediction* state. Assume the evader escapes along the reverse direction from the evader to the reference robot and moves in the robot's maximum step size, the suppositional escaping position of the evader can be calculated. Then each robot acquires its ideal motion position as determined in the *outflank* strategy.

4.6 Direction-Optimization Strategy

As mentioned above, each robot acquires its ideal motion position or expected direction based on a certain task state without considering the obstacles near it. In order to approach the ideal movement in the real world, an effective collision-free motion strategy is essential. A *direction-optimization* strategy is proposed, by which each robot obtains its actual moving direction that has the least angle with respect to its ideal direction on the premise of the predetermined step size.

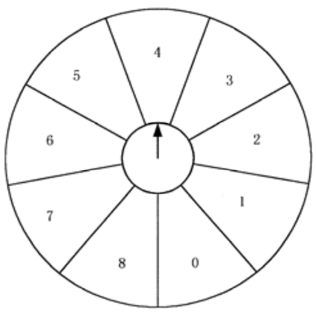


Figure 3. The layout of sensors.

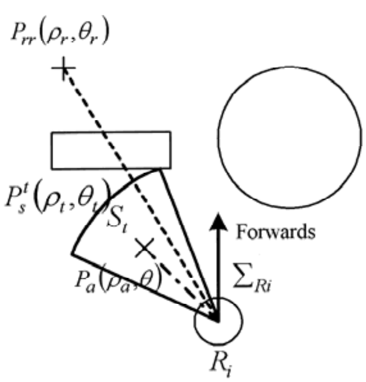


Figure 4. A sample of a robot environment.

The robot adopts the range sensor model S_{range} to perceive the environment and the detecting zone of each sensor is a sector. Figure 3 shows the layout of sensors whose numbers are assigned from 0 to 8 as starting from the reverse direction of the robot's heading, which is shown in arrow. The robot may know the presence or absence of other objects in each sector zone as well as the distance to them. For R_i, it establishes its polar coordinate system Σ_{Ri} (see Figure 4) whose pole and polar axis direction are its center and current moving direction, respectively. We denote with $P_{rr}(\rho_r, \theta_r)$ the coordinates of the ideal motion position in Σ_{Ri}. The coordinates of the detecting border of sensors S_t in Σ_{Ri} are $P_s^t(\rho_t, \theta_t)$, where $t = \{0..8\}$, ρ_t is the maximum sensing range when no obstacle is detected, otherwise, reading from S_t after the evader is considered, and $\theta_t \in \left[-\pi + \frac{2\pi}{9}t, -\pi + \frac{2\pi}{9}(t+1)\right]$. Only S_t is drawn as an example in Figure 4. Denote $P_a(\rho_a, \theta)$ as the coordinates of the next motion position in Σ_{Ri}, where ρ_a is the step size determined by the robot's current position, ideal position and maximum step size; θ is the angle it rotated. The goal is to seek θ within $[-\zeta_{r\max}, \zeta_{r\max}]$ of the current moving direction on the constraint of predetermined ρ_a such that the robot moves along the collision-free direction that has the least angle with respect to the ideal direction.

On the basis of sensory information, the distances $P_a P_s^t$ from P_a to the detecting border of each sensor should be greater than or equal to a safety distance D_{safe} determined by the velocity and r, namely,

$$P_a P_s{}^t \geq D_{safe}(t = 0,1,\ldots,8) \tag{2}$$

The final value of θ should satisfy equation (2) and make $|\theta - \theta_r|$ a minimum. Considering the t^{th} sensor, we have

$$\sqrt{(\rho_a \cos\theta(t) - \rho_t \cos\theta_t)^2 + (\rho_a \sin\theta(t) - \rho_t \sin\theta_t)^2} \geq D_{safe} \tag{3}$$

where $\theta(t)$ are the values of θ satisfying the condition of the t^{th} sensor in equation (2). From equation (3), it can be obtained that

$$\cos(\theta(t) - \theta_t) \leq \frac{\rho_a{}^2 + \rho_t{}^2 - D_{safe}{}^2}{2\rho_a \rho_t} = V \tag{4}$$

When $|\rho_a - \rho_t| \geq D_{safe}$ is satisfied, $\theta(t) \in [-\zeta_{r\max}, \zeta_{r\max}]$.

When $\rho_a + \rho_t < D_{safe}$ is satisfied, $\theta(t) \in \Phi$, the empty set.

When $\rho_a + \rho_t \geq D_{safe} \cap |\rho_a - \rho_t| < D_{safe}$ is satisfied, we have

$$\theta(t) - \theta_t \in [-2\pi + \arccos V, -\arccos V] \cup [\arccos V, 2\pi - \arccos V] \tag{5}$$

Any value within the range of θ_t should be suitable for equation (5), therefore,

$$\theta(t) \in \left\{ \left[\arccos V - \frac{25}{9}\pi + \frac{2\pi}{9}t, -\arccos V - \pi + \frac{2\pi}{9}t\right] \cup \right.$$

$$\left. \left[\arccos V - \frac{7}{9}\pi + \frac{2\pi}{9}t, -\arccos V + \pi + \frac{2\pi}{9}t\right] \right\} \cap [-\zeta_{r\max}, \zeta_{r\max}] \tag{6}$$

if $\arccos V \leq \frac{8\pi}{9}$ is satisfied, and $\theta(t) \in \Phi$, when $\arccos V > \frac{8\pi}{9}$ is satisfied.

The set of values of θ satisfying the conditions of all sensors is defined as Ω, which is the intersection of $\theta(t)(t = 0,1,\ldots,8)$. When Ω is not equal to the empty set, the most preferred value of θ, which is expressed by $\theta_{\min}$, can be obtained to make $|\theta - \theta_r|$ a minimum, or else, the proper θ cannot be found. In this case, the robot will turn right angle $\zeta_{r\max}$ without any change in position. To sum up, the next moving direction of the robot is defined as:

$$V_{direction} = \begin{bmatrix} \cos\beta & -\sin\beta \\ \sin\beta & \cos\beta \end{bmatrix} \cdot \begin{bmatrix} x_d \\ y_d \end{bmatrix} \tag{7}$$

$$\beta = \begin{cases} \theta_{\min} & \Omega \neq \Phi \\ -\zeta_{r\max} & \Omega = \Phi \end{cases} \tag{8}$$

5. STRATEGIES FOR THE EVADER

Assume that the evader adopts the same sensor model as that of individual robot. While the evader does not sense any robot or static obstacle, it moves randomly; otherwise, it should move along a safety direction. A *safety-motion* strategy is proposed. The evader establishes a polar coordinate system Σ_e whose pole is its center and the polar axis direction is its heading. We denote with $P_e^i(\rho_i, \theta_i)$ the coordinates of the detecting border of sensors S_e^i in Σ_e, where $i = \{0..8\}$ and ρ_i is reading from S_e^i when it senses any object, or else, the sensor is ignored and for convenience ρ_i is far greater than the maximum sensing range of the evader; $\theta_i \in \left[-\pi + \frac{2\pi}{9} i, -\pi + \frac{2\pi}{9}(i+1)\right]$. Based on the evader's current direction, Q (a multiple of 4) directions are generated and their set $\Im$ is depicted as follows.

$$\Im = \left\{ \zeta_q \middle| \zeta_q = -\pi + \frac{2q\pi}{Q} (q = 0,1,\ldots,Q-1) \right\} \tag{9}$$

The evader may move to the position $P_n^q(V_e, \zeta_q)$ on the premise of the predetermined step size V_e without any collisions when the distance from the position to the detecting border of each sensor should be greater than or equal to a safety distance L_{safe} influenced by the evader's velocity and radius, that is,

$$d_i(\zeta_q) = \min\left(P_n{}^q P_e{}^i\right) \geq L_{safe} (i = 0,1,\ldots,8) \tag{10}$$

When $\exists \zeta_q \in \Im$ satisfying equation (10), the evader is still capable of moving, otherwise, no feasible moving direction is available, which indicates that the evader is captured.

Let Ψ label the set of the directions within $\left[-\frac{\pi}{2}, \frac{\pi}{2}\right)$ of the current direction and $\Psi = \left\{ \zeta_q \middle| \zeta_q = -\pi + \frac{2q\pi}{Q}\left(q = \frac{Q}{4}, \frac{Q}{4}+1, \ldots, \frac{3Q}{4}-1\right)\right\}$. The *safety-motion* strategy is to select the best one ζ from all ζ_q satisfying equation (10) in the set Ψ, and the best value should make $dis(\zeta_q)$ maximum, thus,

$$dis(\zeta) = \max_{\zeta_q} dis(\zeta_q) = \max_{\zeta_q} \min\left(d_0(\zeta_q), d_1(\zeta_q), \ldots, d_8(\zeta_q)\right) \tag{11}$$

If ζ is found, the evader will rotate ζ with the step size V_e, or else, it only turns right $\frac{\pi}{2}$.

6. SIMULATIONS

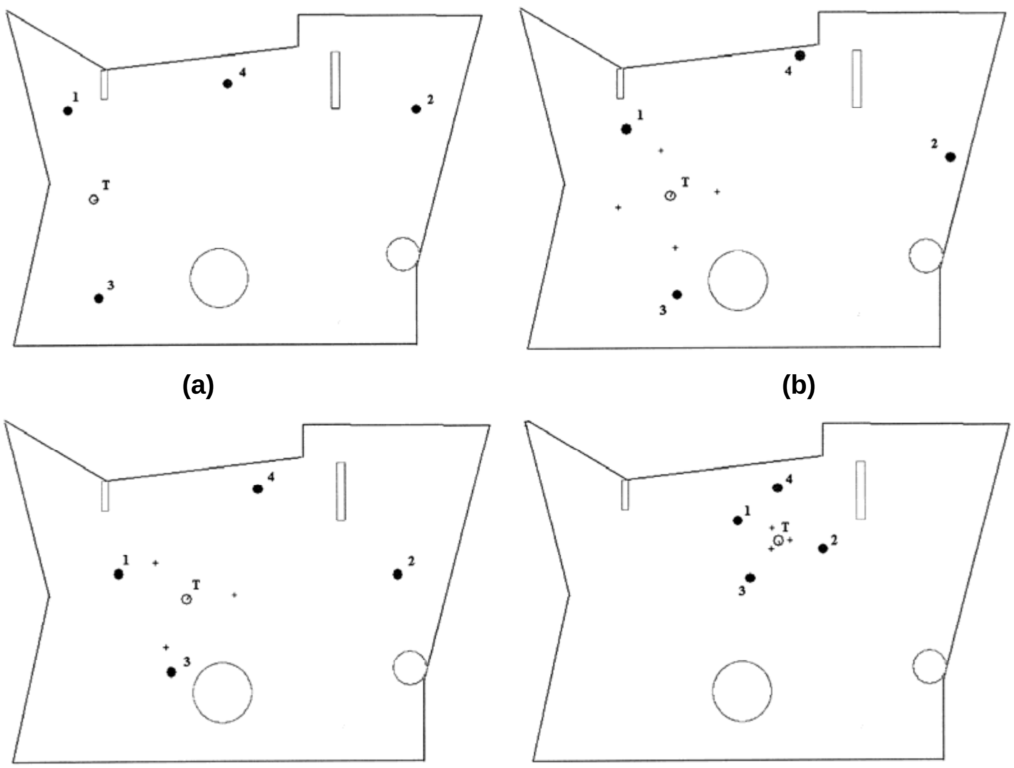
In the simulations, a team of robots of ID $1,2,\ldots$ hunts an evader T, which is regarded as one special robot. They have the same physical parameters: the radius, maximum step size and maximum sensing range are 0.2, 0.1 and 3.0, respectively. The parameters for proposed strategies are shown in Table I.

Simulation 1 is used to illustrate the robustness of the approach. Figure 5 shows several selected images. Four robots move from their initial positions as shown in Figure 5.a. When the evader T is seen (see Figure 5.b), the system begins to pursue the evader. After the robot of ID 4 becomes dysfunctional and stops moving (see Figure 5.c), other robots re-adjust themselves. When the robots are in positions exhibited in Figure 5.d, the besieging circle begins to shrink and finally the evader is captured.

To further confirm the performance of BCSLA, it is also compared with BCSLF (a BCS approach with a fixed leader), DSLA and DSLF. A series of simulations (simulation 2) were conducted with the distance d increasing in two environments without boundaries (see Figure 6). Figure 7 describes the variations of step numbers of the robots adopting different approaches according to different d and no label in some cases means the failure to complete the task. From the simulation results, the BCSLA approach is considered as a rational one.

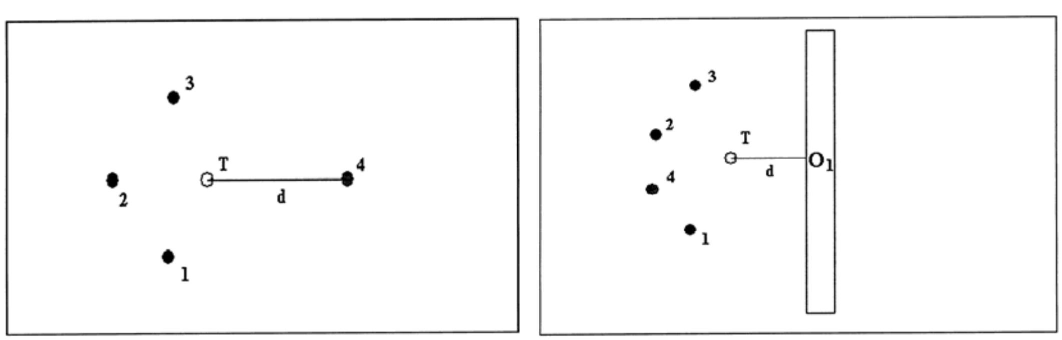
Table I. Parameters for proposed strategies.

Parameter	Strategy	Value	Parameter	Strategy	Value
τ	*dispersion-search*	$\frac{\pi}{18}$	$N_{prediction}$	*predict*	20
r_s	*leader-adjusting*	2.0	$\zeta_{r\max}$	*direction-optimization*	$\frac{\pi}{2}$
r_c		0.5	D_{safe}		0.3
k_d		0.75	Q	*safety-motion*	24
k_{as}		0.9	L_{safe}		0.39



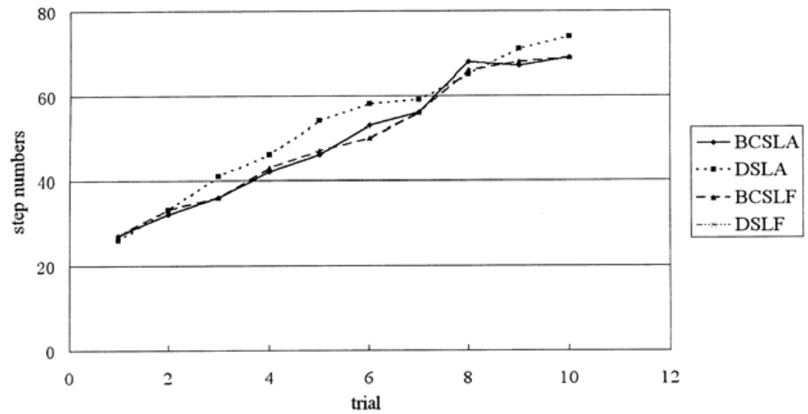
(a) (b) (c) (d)

Figure 5. Selected images for the cooperative hunting process in simulation 1.

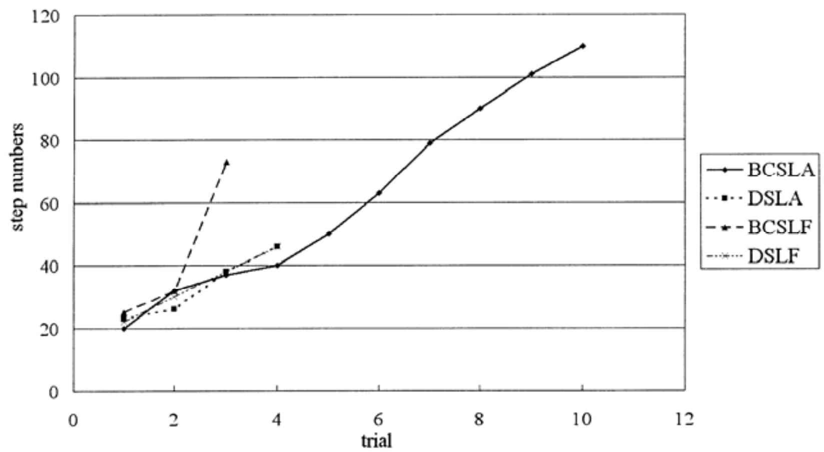


(a) Environment 1 **(b) Environment 2**

Figure 6. Two test environments of simulation 2.



(a) Environment 1



(b) Environment 2

Figure 7. The comparisons of different approaches for simulation 2.

7. CONCLUSION

This paper has mainly focused on the problem of cooperative hunting by multiple mobile robots in dynamic environments. An approach called BCSLA has been proposed. The approach may solve the *drag* problem and adjust dynamically to adapt to environmental changes. As the evader tries to escape by a *safety-motion* strategy, the difficulty of hunting is increased. Conducted simulations testify the rationality of the BCSLA approach.

ACKNOWLEDGEMENT

This work is funded in part by the National High Technology Research and Development Program of China (863 Program) under Grants 2006AA04Z258, 2006AA04Z219, and in part by the National Natural Science Foundation of China under Grants 50475179, 60635010.

REFERENCES

1. Y. U. Cao, A. S. Fukunaga, and A. B. Kahng, “Cooperative mobile robotics: antecedents and directions,” *Autonomous Robots*, 4(1), 7-27, 1997.
2. P. U. Lima and F. C.A. Groen, “Special issue on multi-robots in dynamic environments,” *Robotics and Autonomous Systems*, 50(2-3), 81-83, 2005.

3. T. Arai, E. Pagello, and L.E. Parker, "Guest editorial advances in multirobot systems," *IEEE Transactions on Robotics and Automation*, 18(5), 655-661, 2002.
4. H. Yamaguchi, "A cooperative hunting behavior by mobile-robot troops," *International Journal of Robotics Research*, 18(8), 931-940, 1999.
5. H. Yamaguchi and T. Arai, "Distributed and autonomous control method for generating shape of multiple mobile robot group," *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, 800-807, 1994.
6. P. Pirjanian and M. J. Mataric, "Multi-robot target acquisition using multiple objective behavior coordination," *Proceedings of the IEEE International Conference on Robotics and Automation*, 2696-2702, 2000.
7. P. Stone and M. Veloso, "Multiagent systems: a survey from a machine learning perspective," *Autonomous Robots*, 8(3), 345-383, 2000.
8. E. Osawa, "A metalevel coordination strategy for reactive cooperative planning," *Proceedings of the First International Conference on Multi-agent Systems*, 297-303, 1995.
9. J. Denzinger and M. Fuchs, "Experiments in learning prototypical situations for variants of the pursuit game," *Proceedings of the second International Conference on Multi-agent Systems*, 48-55, 1996.
10. R. Vidal, S. Rashid, C. Sharp, O. Shakernia, J. Kim, and S. Sastry, "Pursuit-evasion games with unmanned ground and aerial vehicles," *Proceedings of the IEEE International Conference on Robotics and Automation*, 2948-2955, 2001.
11. R. Vidal, O. Shakernia, H. J. Kim, D. H. Shim, and S. Sastry, "Probabilistic pursuit-evasion games: theory, implementation, and experimental evaluation," *IEEE Transactions on Robotics and Automation*, 18(5), 662-669, 2002.
12. N. Ono, K. Fukumoto, and O. Ikeda, "Collective behavior by modular reinforcement-learning animats," *Proceedings of the fourth International Conference on Simulation of Adaptive Behavior*, 618-624, 1996.

ABOUT THE AUTHORS

Z. Cao received the B.S. and M.S. degrees from Shandong University of Technology, Jinan, China, in 1996 and 1999, respectively. In 2002, he received the Ph.D. degree in control theory and control engineering from Institute of Automation, Chinese Academy of Sciences, Beijing, China.
He is currently an associate professor in the Laboratory of Complex Systems and Intelligence Science, Institute of Automation, Chinese Academy of Sciences. His research interests include multi-robot coordination, biomimetic robots, intelligent control.

N. Gu works at School of Engineering and IT Technology, Deakin University as Research Fellow. His research interests include robust control, modeling and optimization for manufacturing system, signal processing and wireless communication.

M. Tan received the B.S. degree in control engineering from Tsinghua University, Beijing, China, in 1986 and the Ph.D. degree in control theory and control engineering from Institute of Automation, Chinese Academy of Sciences, Beijing, China, in 1990.
He is currently a professor in the Laboratory of Complex Systems and Intelligence Science, Institute of Automation, Chinese Academy of Sciences. His research interests include advanced robot control, multi-robot system, biomimetic robots.

S. Nahavandi received BSc (Hons), MSc and a PhD in Automation and control from Durham University (UK). Professor Nahavandi holds the title of Alfred Deakin Professor, Chair of Engineering and is the leader for the Intelligent Systems research Lab. at Deakin University (Australia). He won the title of Young Engineer of the Year for his novel intelligent robotic end effector in 1996 and has published over 260 peer reviewed papers in various International Journals and Conference and is the recipient of six international awards in Engineering. His research interests include modeling of complex systems, simulation based optimization, robotics, haptics and augmented reality. Professor Nahavandi is the Associate Editor - IEEE Systems Journal, Editorial Consultant Board member - International Journal of Advanced Robotic Systems, Editor (South Pacific region) - International Journal of Intelligent Automation and Soft Computing, Editorial Board member - International Journal of Computational Intelligence. He is a Fellow of Engineers Australia (FIEAust) and IET (FIET) and Senior Member of IEEE.

X. Mao received B.S. degree from Peking University, Beijing, China, in 2001 and M.S. degree in control theory and control engineering from Peking University in 2004.
He is currently a Ph.D. candidate and graduate assistant in Mechanical Engineering Dept. of the Pennsylvania State University, PA, US. His research interests include nonlinear and robust control design for underwater vehicles, multiple robots.

Z. Guan is a Ph.D student at School of Engineering and IT Technology, Deakin University. Her research interests include multiple mobile robots and intelligent control.