Research on Pursuit-evasion Games with Multiple Heterogeneous Pursuers and A High Speed Evader

Hongpeng Wang[1,2,3], Qiang Yue[1,2,3], Jingtai Liu[1,3]

1. Institute of Robotics & Automation Information System, Nankai University, Tianjin, 300071, China
E-mail: sunshineyq@mail.nankai.edu.cn
2. State Key Laboratory of Robotics, Shenyang, 110000 ,China
E-mail: hpwang@nankai.edu.cn
3. Tianjin Key Laboratory of Intelligent Robotics, Tianjin, 300071, China
E-mail: liujt@nankai.edu.cn

Abstract: We deal with Pursuit-evasion games with a high speed evader which has superiority in velocity over a group of heterogeneous pursuers in this paper. Heterogeneity in the group of pursuers is expressed as heterogeneity in the individual maximum speeds. We introduce Apollonius circles generated by every pursuer and the evader to analyse the criteria for a successful capture. Collective robots pursuit problem under unknown environment is investigated from the view of behavior-based control method, and Motor Schema-based reactive control architecture is adopted. Three basic behaviors using for pursuers are designed, namely Move_to_goal, Avoid_obatacle and Hunting, wherein hunting behavior is realized through a kind of reinforcement learning algorithm called Q-Learning. Pursuit of the evader is based on synthesized behavior, generated through summarizing the outputs of all behaviors weighed. Results of simulation experiments validate the effectiveness of our method.

Key Words: Pursuit-evasion games, Heterogeneity, Apollonius circle, Q-Learning, Motor Schema

1 INTRODUCTION

During the last few decades, research on multi-robot system has drawn extensive attention in robotics field. Various applications range from rescue tasks [1], security patrol to military reconnaissance [2]. The utilities of multi-robot system have indicated much more superiorities over the single one, such as higher efficiency, better adaptability and stability. As a result, strategies and control method are specifically pivotal for multi-robot system to accomplish coordinating tasks [3]. And we deal with Pursuit-evasion games with a high speed evader which outruns a group of heterogeneous pursuers in this paper.

As a typical multi-robot problem, Pursuit-evasion game has become a hot research area due to its applications in various problems such as air combat, search and security tasks, and autonomous navigation in an adversarial environment. In Pursuit-evasion games, a group of pursuers aim to capture evaders in minimum possible time, and evaders try to escape from pursuers for maximum possible time. In [4] the task of security was studied and a novel algorithm for adaptive formation control was raised. In [5], PE problem was studied in the view of evader-centric, and the goal for pursuer is to capture the slowest one among the evaders. Reinforcement learning was used in the control of pursuers [6], and the result is quite encouraging. In [7] PSO method was introduced to optimize both pursuers' and evaders' strategies. However, the proposed method is computational intensive and results are sub-optimal in some cases. The decentralized control approach was implemented through a novel modularized hybrid system architecture and experiments with physical robots were demonstrated in [8]. The majority of related scholars suppose that evaders run slower than anyone of the pursuers or set the evader stationary. In real world applications, however, it's common sense that evaders should be as fast as the pursuers at least. The slower evader can always be caught by a group of pursuers without additional strategies or approaches. And in this paper, Pursuit-evasion games with a fast evader which outruns pursuers are discussed.

Moreover, few people studied the scenario when members of the pursuit group are heterogeneous, namely, pursuers have different abilities during the process of hunting. The difference may include speed, communication radius or the sensor ranges. In [9], pursuit and evasion in the plane with a group of heterogeneous evaders and a single pursuer was studied. In this paper, we deal with pursuers of different speed, and the corresponding criteria for a success capture are analyzed by applying Apollonius circles.

The rest of this paper is structured as follows: In Section 2, mathematical concepts relating to hunting strategies are presented. Then, hunting behavior based on Q-Learning algorithm is introduced in detail in Section 3. The whole control architecture is described in Section 4. Simulation results are presented at Section 5 and conclusions and the future work are summarized in Section 6.

2 PRELIMINARIES

Let $P(t), E(t), V_P, V_E$ denote positions at time t and the maximum velocity of pursuer robots and evader robot

This work is supported by National Natural Science Foundation of China under Grant 61105096, 61375087 and Opening Project of State Key Laboratory of Robotics (No.2013-O02).

Hongpeng Wang is the corresponding author.
Email: hpwang@nankai.edu.cn.

978-1-4799-7016-2/15/$31.00 ©2015 IEEE

respectively. $m = V_P/V_E$, since $V_P < V_E$, then we have $m < 1$ .According to [10], m should satisfy:

$$m = \mathrm{V}_p/V_e \geq \sin(\pi/n) \tag{1}$$

n denotes the number of pursuers and in this paper we set the slowest pursuer follow this constraint condition. The Cartesian coordinate system is established with the origin on the evader. In Fig.1, $\mathrm{A}(x_A, y_A)$ is a random point. If $AP/AE = m$, then the trajectory of point A forms an Apollonius circle on which the points can be reached simultaneously by robot P and E. Through the related plane geometry knowledge, we can figure out the coordinate of the circle center as below:

$$O\left(\frac{x_P - m^2 x_E}{1-m^2}, \frac{y_P - m^2 y_E}{1-m^2}\right) \tag{2}$$

and the radius of the circle as Eq.3:

$$R = \mathrm{m}\sqrt{(x_P - x_E)^2 + (y_P - y_E)^2}/(1-m^2) \tag{3}$$

Also we can get the max value of θ according to the sine theorem $AP/\sin(\pi/2) = AE/\sin\beta$ .And the maximum value of θ can be achieved under $\beta \to \pi/2, V_P/V_E \to 1^-$ as Eq.4, thus the minimum number of pursuer is $n_{min} = 3$.

$$\theta_{max} = \pi|_{\beta \to \frac{\pi}{2}, \frac{V_P}{V_E} \to 1^-} \tag{4}$$

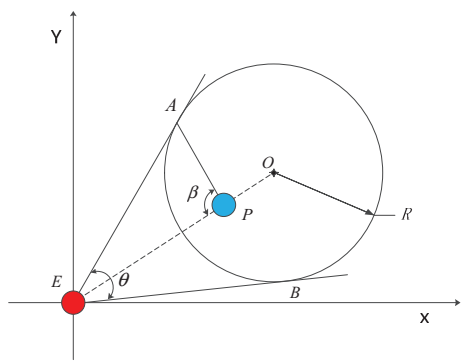

Fig.1 Apollonius circle *O* formed by a faster evader *E* and a pursuer *P*, wherein line *AE* and *BE* are tangent to it.

The adjacent Apollonius circles of different pursuers must be either tangent to a point or intersected with each other in order to form a closed Apollonius circles area [10]. Suppose that at time t, two robots of P_i and P_{i+1} are adjacent to each other as shown in Fig.2:

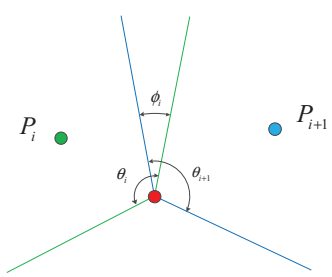

Fig.2 The red circle denotes the evader and angle $\phi_i$denotes overlapping angle of robots P_i and P_{i+1} in green and blue respectively.

As illustrated in Fig.2, overlapping angle of θ_i and θ_{i+1} is ϕ_i, and we define its value equals 0 when θ_i and θ_{i+1} don't overlap. So our goal is to achieve Eq.5 :

$$\Phi = \sum_{i=1}^{n} \theta_i - \sum_{i=1}^{n} \phi_i = 2\pi \tag{5}$$

If the Eq.5 is satisfied, pursuers could completely manage to capture the evader. Otherwise, as Fig.3 shows, the related robots should take actions to change the value of ϕ_i who equals zero at present time.

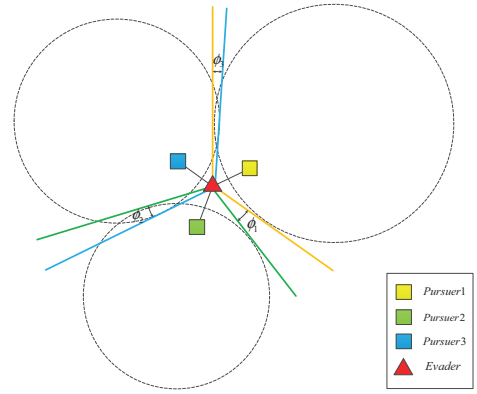

Fig.3 The evader E hasn't been surrounded by the Apollonius circle area formed by pursuers, so P_1, P_2 should run towards ϕ_1 to make it positive.

3 Hunting Behavior Based On Q-Learning

Over the last decades, a novel machine learning algorithm called reinforcement learning (RL) has become an important methodology for learning in a lot of research fields, including robotics [11]. It contains a class of approaches in which the robot try to fulfill tasks while learning based on punishment and reward it receives from the interaction with the environment. Conception of RL dates back to the early time of cybernetics and work in neuroscience, psychology, statistics, and computer science. In the last few years, RL has attracted swiftly increasing attention in the machine learning (ML) domain and artificial intelligence (AI) communities [12]. It controls the robot to learn behaviors through continuous interactions with the unknown and complex surroundings and getting punishment and reward from it. So robot doesn't need to be specified how the task is to be achieved, for it can choose the right action according to its state right now.

3.1 Overview of Q-Learning

In robotics domain, Q-Learning is widely known as a kind of reinforcement learning approach [13]. It has the advantages of simple arithmetic. And by selecting the action of the highest Q value in every step, it's not difficult to get the optimal policy for robot control. We can review the fundamentals of Q-learning briefly ahead of dealing with the details.

Assuming that the distinct states set of the surroundings, set $\boldsymbol{S}$, can be distinguished by the robot and it's able to carry out actions of set $\boldsymbol{A}$ upon the surroundings. Next, the surrounding is taken as a Markov process, which makes random transformations between different states based on the present state as well as actions the robot takes. We define $P(s, a, s')$ as the probability of transforming from the present state s to the next state s′ when taking action a. And the corresponding reward $\mathrm{r(s, a)}$ is defined for (s, a) , which is called the state-action pair [14].

The purpose of robot is to obtain the optimal policy through which it can receive maximum reward over time. A policy **f** maps from $\boldsymbol{S}$ to $\boldsymbol{A}$. The return is defined as the gross value of reward. Value of the return as seen from below:

$$\sum_{n=0}^{\infty} \gamma^n r_{t+n} \tag{6}$$

where γ is defined as discounting factor. It indicates the extent of influence over a policy's total value by getting rewards afterwards and $0 < \gamma < 1$. When the agent started in state **s** and followed policy **f**, the reward received at time t is defined as r_t.

Define $Q^*(\mathrm{s},\mathrm{a})$ as the value function after executing the action **a** and continuously follow the optimal policy in state **s**. See as defined below.

$$Q^*(s,\mathrm{a}) = r(s,\mathrm{a}) + \gamma \sum_{s' \epsilon S} P(s,\mathrm{a},s') \max_{a' \epsilon A} Q^*(s',\mathrm{a}') \tag{7}$$

The value of r and P are unknown initially, so making estimation of the values is quite necessary. Starting with $Q(\mathrm{s},\mathrm{a})$ at any value (generally zero), update the Q value after taking an action. It can be seen as follows:

$$Q(s,\mathrm{a}) = (1-\alpha)Q(s,\mathrm{a}) + \alpha[r(s,\mathrm{a}) + \gamma \max_{a' \in A} Q(s',\mathrm{a}')] \tag{8}$$

where α is called leaning rate (range between 0 and 1), and after taking action **a**, the transition from current state **s** to $\mathbf{s}'$ results in a reward called r. The Q-learning method we used in this paper is a simple and practical one-step edition shown as follows.

Initialize: Q ← randomly select initial value for the Q value function (e.g., zero).

Start looping:

1. Perceive the present state s .
2. Take action a , and let r be the reward received, and s' be the next state.
3. Update state action value function through Eq.8.
4. Return to step 1.

3.2 State Space Reduction

Inspired by [15], in order to reduce state space, we choose the following clustered condition predicates:

- near-evader ?
  return 1 if distance to evader satisfy $d_e < d_{eth}$; otherwise, return 0;
- near-pursuer ?
  return 1 if distance to a partner satisfy $d_p < d_{pth}$; otherwise, return 0;
- nearly-surround ?
  return 1 if $\Phi > \Phi_{th}$,which means the pursuit group is about to form the prospective surrounding formation; otherwise, return 0;

In this way, for a system of n robots, the three conditions above generate a n bits binary number, which contains 2^n states at all.

3.3 Action Set

Actions are taken according to subset of the state space determined by predicates on sensor reading. Accordingly, action set of Hunting behavior is consisted of the following actions:

- *Wander.*
  –keeps the robot moving about randomly without colliding with any objects.
- *Move_to_goal.*
  –let the robot run directly towards the evader.
- *Intercept_evader.*
  – makes prediction of evader's position and move to it ahead of the evader.
- *Move_to_peers.*
  –let the robot run in the direction of partner robot which is farthest form self.
- *Wait_Stationary*
  –makes the robot stay at the present location immovably.

3.4 Action Evaluation

As mentioned in Section 2, the objective is to increase the sum value of Φ.So the variation of Φ is introduced to evaluate the action. We have:

$$\Delta\Phi = \Phi(t) - \Phi(t-1) \tag{9}$$

Actions generating a positive ΔΦ are rewarded, whereas actions causing a negative ΔΦ result in negative reinforcement. And we introduce the following evaluation function:

$$r = 2\left(\frac{1}{2} - \frac{1}{1+e^{\Delta\Phi}}\right) \tag{10}$$

4 Motor Schema-based Control Method

4.1 Control Architecture

Motor Schema-based reactive control architecture is proposed by Arkin [16] in 1989, and it is widely used in mobile robot navigation [17]. As a kind of behavior-based decomposition mode, it breaks down the movement of robot into a series of basic behaviors, and a direct relation between sensors and actuators is established. As a result, robot's response to environment is accelerated. Another advantage is that this kind of architecture doesn't need to do environment modeling, and robustness of the system is strengthened [18][19].

As to pursuers in this paper, three basic behaviors are set up: Avoid_obstacle, Move_to_goal and Hunting. See the following Fig.4:

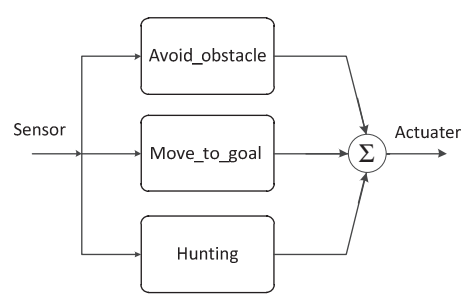

Fig.4 Motor Schema-based reactive control architecture

4.2 Behavioral Assemblage

Every basic behavior outputs a motion vector (magnitude and direction) representing the reaction to current stimulus

which is imposed by surroundings and other robots. In order to show the different importance, each behavior multiplies with a weight number. The final integrated behavior is calculated according to the equation below.

$$F = w_1F_1 + w_2F_2 + w_3F_3 \tag{11}$$

where F denotes motion vector and w denotes weight number. Apart from Hunting behavior, Avoid_obstacle behavior and Move_to_goal behavior are realized using Artificial Potential Field (APF), a common local path planning method [20], which has the advantage of rapid respondence, small amount of calculation and real-time decision.

Computation process is similar with [21], and result of behavioral assemblage is as follows:

$$\mathrm{v} = \sqrt{\left(\sum_i w_i v_i \cos\theta_i\right)^2 + \left(\sum_i w_i v_i \sin\theta_i\right)^2} \tag{12}$$

and,

$$\theta = \begin{cases} \tan^{-1}\left(\dfrac{\sum_i w_i v_i \sin\theta_i}{\sum_i w_i v_i \cos\theta_i}\right), & \sum_i w_i v_i \sin\theta_i \geq 0 \\ \tan^{-1}\left(\dfrac{\sum_i w_i v_i \sin\theta_i}{\sum_i w_i v_i \cos\theta_i}\right) + \pi, & \sum_i w_i v_i \sin\theta_i < 0 \end{cases} \tag{13}$$

where w_i is the weight number of behavior i and v_i is the output velocity of behavior i, and θ_i is the output angular of behavior i.

5 SIMULATION

In order to verify the algorithm we proposed above, simulation experiment is carried out in the environment of Matlab R2014a. In the virtual scenario, three heterogeneous pursuers and a fast evader conduct pursuit-evasion in a field with static obstacles. Every robot is equipped with eight virtual ultrasonic sensors and an omni-vision camera in order to run safely and distinguish teammates.

Fig.5 shows the initial deployment of simulation system, in which circles in color denote robots and the black objects denote static obstacles.

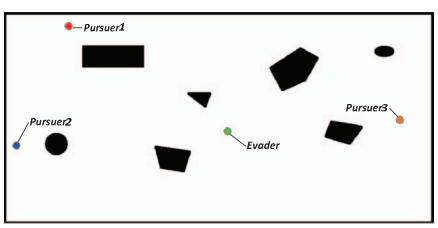

Fig.5 First frame: initial deployment of simulation system

Rules and parameters set in the experiment are as follows:

- Game will be over when step limit is up or the evader is captured.
- Step time of the simulation is 100 ms, and maximum step limit is set to 500.
- The size of virtual map is 1000×480 with some static obstacles.
- The maximum velocity of the evader's is set to 4.0, and the three pursuers' are set different from each other respectively, and they all should satisfy the condition as follows:

$$\frac{V_P}{V_E} > \sin\frac{\pi}{3} = 0.866 \tag{14}$$

- Diameter for each robot is 8, and pursuers are assumed to have global knowledge of other robots, while sensor range of the evader is set to 128.
- Evader's strategy is run randomly without colliding with other objects when opponent is sensed.
- Learning rate in Q-Learning is set to 1, and discount factor is 0.9.

When the three pursuers' maximum velocity is set to {3.5,3.7,4.0}, simulation result is as Fig.6 shows. The three pursuers move towards the evader while learning from the environment and avoiding obstacles. And finally the experiment results in a successful pursuit.

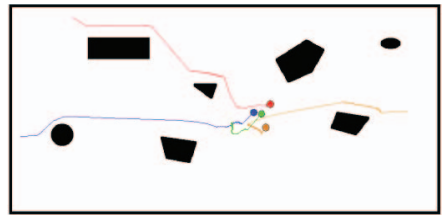

Fig.6 Final frame when game is over and tracks of each robot

Fig.7 shows the three pursuers' relative distance to the evader changing along with time step. As seen in Fig.7, distances between each pursuer and the evader decrease until the evader is captured.

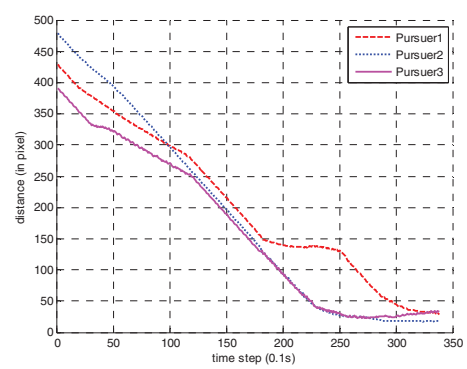

Fig.7 Relative distance of each pursuer to the evader

The three pursuers run to the evader as well as coordinate to surround it in a closed Apollonius circles area. Fig.8 illustrates the Apollonius circle to each pursuer, and it can be seen that the evader is encircled within the range of circles area from the magnified view.

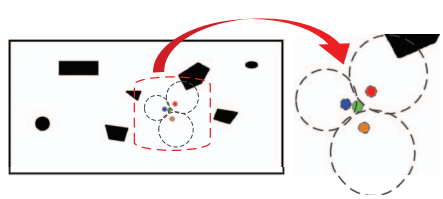

Fig.8 The evader in green has been encircled within the closed Apollonius circles area and is captured by the pursuer in blue.

Simulation results under different ratio value m of V_P/V_E are shown in Table 1. Experiment under each condition is

conducted for 20 times and it can be seen from this table that capture times increases with the ratio m increasing to 1, meanwhile, mean time decreases accordingly.

Table 1: Simulation results for different ratio m

Ratio m	Capture Times	Mean Time (s)
$\{0.87,0.88,0.9\}$	11	40.8
$\{0.87,0.93,1.0\}$	16	33.5
$\{0.95,0.98,1.0\}$	20	26.4

6 CONCLUSIONS

In this paper, we introduce the concept of Apollonius circle into Pursuit-evasion game, when the multiple pursuers are heterogeneous and the single evader has superiority in velocity. Pervious work usually aims at making pursuers evenly distributed around the evader, while this won't apply when, for example, pursuers have different speed with each other. And through the algorithm we realized using Q-Learning, this problem can be solved. What's more, Motor Schema-based reactive control architecture is adopted in this paper, and pursuit of the evader is based on the synthesized behavior of three basic behaviors in an unknown environment with static obstacles. Simulation experiment is carried out in virtual scenario and the result shows that multiple pursuers coordinate their movements to capture the evader, so the effectiveness of our solution is validated. In the following research, multi-evader will be added into the research content and study of evasion strategy under the research of this paper shall be interesting.

REFERENCES

[1] J.S. Jennings, G. Whelan, and W. F. Evans, Cooperative search and rescue with a team of mobile robots, in Proceedings of the International Conference on Advanced Robotics, Canada, 1997:193-200.

[2] W. Burgard, M. Moors, C. Stachniss, and F.E. Schneider, Coordinated multi-Robot exploration, IEEE Transaction on Robotics, vol.21, no.3, 2005: 376-386.

[3] Vidal R, Shakernia O, Kim H J, et al. Probabilistic pursuit-evasion games: theory, implementation, and experimental evaluation[J]. IEEE Transactions on Robotics and Automation, 2002, 18(5): 662-669.

[4] H. Yamaguchi. Adaptive Formation Control for Distributed Autonomous Mobile Robot Groups, in Proceedings of the 1997 IEEE International Conference on Robotics and Automation, New Mexico, 1997: 2300~2305

[5] Kumar A, Ojha A. An Evader-Centric Strategy against Fast Pursuer in an Unknown Environment with Static Obstacles. Control, in 2013 International Conference on Automation, Robotics and Embedded Systems (CARE). IEEE, 2013: 1-6.

[6] Ono N, Fukumoto K. Multi-agent reinforcement learning: A modular approach[C], in Proceedings of the Second International Conference on Multi-Agent Systems. 1996: 252-258.

[7] A. K. Sun and H. H. T. Liu, Multi-pursuer evasion, in AIAA Guidance, Navigation and Control Conference and Exhibit, August,18-21 2008: 795–798.

[8] Huang F, Wang L, Wang Q, Wu M, Jia Y. Coordinated control of multiple mobile robots in pursuit-evasion games[C], in American Control Conference(ACC'09), 2009. IEEE, 2009: 2861-2866.

[9] Scott W, Leonard N E. Dynamics of Pursuit and Evasion in a Heterogeneous Herd [J], in Proceedings of the IEEE Conference on Decision and Control, Los Angeles, California, 2014.

[10] Jin S, Qu Z. Pursuit-evasion games with multi-pursuer vs. one fast evader[C], in 2010 8th World Congress on Intelligent Control and Automation (WCICA). IEEE, 2010: 3184-3189.

[11] Kaelbling L P, Littman M L, Moore A W. Reinforcement learning: A survey [J]. arXiv preprint cs/9605103, 1996.

[12] Kamei K, Ishikawa M. Determination of the optimal values of parameters in reinforcement learning for mobile robot navigation by a genetic algorithm[C]. International Congress Series. Elsevier, 2004, 1269: 193-196.

[13] Asada M, Uchibe E, Noda S. Coordination of multiple behaviors acquired by a vision-based reinforcement learning[C]. Intelligent Robots and Systems' 94.'Advanced Robotic Systems and the Real World', IROS'94. Proceedings of the IEEE/RSJ/GI International Conference on. IEEE, 1994, 2: 917-924.

[14] Gaskett C. Q-Learning for robot control [D]. Doctoral dissertation, Australian National University, 2002.

[15] Matarić M J. Reinforcement learning in the multi-robot domain [M].Robot colonies. Springer US, 1997: 73-83.

[16] Balch, Tucker, and Ronald C. Arkin. Behavior-based formation control for multi-robot teams. IEEE Transactions on Robotics and Automation, 14.6 (1998): 926-939.

[17] Arkin R C. Motor schema—based mobile robot navigation [J]. The International journal of robotics research, 1989, 8(4): 92-112.

[18] Na Y K, Oh S Y. Hybrid control for autonomous mobile robot navigation using neural network based behavior modules and environment classification [J]. Autonomous Robots, 2003, 15(2): 193-206.

[19] Zhou P, Han Y, Xue M, Hong B. Multiple Robots Cooperative Pursuit Based on Motor Schema [J]. Computer Engineering, 2008, 34(7): 32-34.

[20] Huang B, Cao G. The path planning research for mobile robot based on the artificial potential field [J]. Computer Engineering and Applications, 2006, 27: 008.

[21] Su Z, Lu J, Tong L. Strategy of Cooperative Hunting by Multiple Mobile Robots [J]. Journal of Beijing Institute of Technology, 2004, 5: 008.