A New Approach of Multi-robot Cooperative Pursuit Based on Association Rule Data Mining

Jun Li, Qishu PAN, and Bingrong HONG
School of Computer Science and Technology, Harbin Institute of Technology, P. O. Box 405, Harbin 150001, People Republic of China
Corresponding author E-mail: xlmeng@hit.edu.cn

Abstract: An approach of cooperative hunting for multiple mobile targets by multi-robot is presented, which divides the pursuit process into forming the pursuit teams and capturing the targets. The data sets of attribute relationship is built by consulting all of factors about capturing evaders, then the interesting rules can be found by data mining from the data sets to build the pursuit teams. Through doping out the positions of targets, the pursuit game can be transformed into multi-robot path planning. Reinforcement learning is used to find the best path. The simulation results show that the mobile evaders can be captured effectively and efficiently, and prove the feasibility and validity of the given algorithm under a dynamic environment.

Keywords: multiagent system, multi-robot, pursuit problem, association rule, Q-learning

1. Introduction

In multiagent systems it is quite an important and interesting endeavor to study the development of cooperative behavior among agents. Multi-robot cooperative pursuit is a popular multiagent problem. Based on the same conditions between predators and evaders, the pursuit game can be sorted into single object pursuit and multi-object pursuit. For the first part, this game has also been considered in many references (Yamaguchi, H., 1991; Swastik, K. & Chinya, V., 2005; Kok, J. K. & Vlassis, N., 2004; Yuku, I.; Sato, T. & Kakazu, Y. 2003; Vidal, R.; Shakerinia, O. ; Kim, H. J. ; Shim, D. H. & Sastry, S. 2002); for the other part, Grinton, C. (1996) made use of Commitments & Conventions cooperative mechanism to explore the algorithm including the consistent commitments among predators to capture many static evaders. Schenato, L. et al. (2005) proposed a general framework for the design of a hierarchical control architecture that exploits the advantages of a sensor network by combining both centralized and decentralized real-time control algorithms to complete multi-object pursuit. Based on dynamic role assignment, a method for forming coordination multi-robot teams in the multiple mobile targets capturing has been proposed (Li, S. Q.; Wang, H.; Li, W. & Yang, J. Y. 2006). Cai et al. (2008) proposed a kind of multi-robot cooperative pursuit algorithm that allows fault injection dynamic alliance based task bundle auction. Based on these researches, in this paper we propose an approach of cooperative hunting for multiple mobile targets by multi-robot. In the present approach, there are two parts: one in which the sample data set by the attribute relationship between predators and evaders is created, association rule data mining technology is used to find out the interesting rules, then to build the pursuit team to every evader according to these rules; the other in which every pursuit team forecasts the position of its object in order to make sure of the predators’ aim positions, consequently, we can transform the pursuit problem into the path planning problem and then use multi-robot reinforcement learning to choose the pursuit teams optimal actions strategies to capture evaders in the shortest time.

2. Description of pursuit problems

In this section, we concentrate on a collaborative problem in which $n$ predators in a limitary toroidal grid environment $X$ have to capture $m$ different kinds of evaders. The expression of $P=\{p_1,...,p_n\}$ and $E=\{e_1,...,e_m\}$ stands for the collection of $n$ predators and $m$ evaders respectively. Predators and evaders are called for agents. Evaders have their styles $R_{e_i}$, $i \in m$, $R_{e_i}\in[ I, II, III, IV]$ shows how many predators evaders could be captured by. Here we think that predators can judge evaders’ styles. There are the fixed obstacles whose shape and size are random in $X$. The position could be destined for the mapping $m:X\rightarrow[0,1]$, such as $\forall x\in X, \ m(x)=1$, well then $x$ is an obstacle. If two grids have a same site, they are bordered upon. We mark the neighbor grid of $x$ as $A(x)$ and we can see $|A(x)|\leq 4$ in the rectangle plane. We denote the positions of predators and evaders by $X_t(\bar{i})=(x_1(t),...,x_n(t))$ and $X_t(\bar{e})=(x_1(t),...,x_m(t))$ respectively at time $t$, $t \in T=[1,2,...]$. Any two agents can’t stay in the same cell and each agent can only take one action which includes to stay the origin place or move to
its neighbor grid that is not taken up at time \( t \), that is \( X_i(t+1)=A(X_i(t))\cup\{X_i(t)\} \), \( k\in P_i \). In this paper we formally assume that all agents know the environment map, however they don’t know the objects’ positions and styles at \( t=0 \) and thus all predators begin to search the pursuit area at that time. Here we adopt the circular searching, that is, the predators do their best to get to grids that haven’t been scouted. With increasing the time, there is more and more probability of coming back to search. Hence we give every grid of the pursuit area a value, called searching expectation. For any grid \( x \in X \) at time \( t \), its searching expectation is \( v(x,t) \). For predators when they are searching, \( v(x,t) \) of those grids is zero, which are occupied by obstacles or in the perceptive area of predators, but for other grids, their \( v(x,t) \) is updated by

\[
v(x,t)=v(x,t)+\frac{\Delta N_{R_i}(t)}{N_X}
\]

(1)

where \( \Delta N_{R_i}(t) \) denotes the difference of the number of grids that the predator can see at the neighbor time spot and \( N_x \) is the sum of grids of the searching area. We assume that \( P_i \) is at the grid \( x \), the sum of \( v(x,t) \) for all grids in the perceptive area of \( P_i \) is denoted by \( G(y,t) \) when it moves to the target grid \( y \). In this paper we adopt local-max search strategy, that is, \( P_i \) chooses the grid as the next step position whose \( G(y,t) \) is the maximal value. So, let \( N_{A(x)}(t) \) denote the accessible grids set at time \( t \), obviously we can conclude that \( N_{A(x)}(t)\cup\{x\}\subseteq A(x) \) and \( m(x)=0 \) and let \( N_{R_i}(y) \) be the set of all grids that can be apperceived by \( P_i \) when it is at the grid \( y \). Well then, when \( P_i \) moves to \( y \) from \( x \), its \( G(y,t) \) is updated as the following:

\[
G(y,t)=\sum_{z\in N_{R_i}(y)}v(z,t)
\]

(2)

\( P_i \) can choose the grid from \( N_{A(x)}(t) \) as its expectation target position at time \( t+1 \) whose \( G(y,t) \) is the maximal value. We discuss two final successful capturing configurations: one in which the total number of predators in neighbor grids of the evader is more than the value of its style, the other in which evaders are surrounded by predators or blocked up in the border of \( X \) or the gap of obstacles. We show the initial and final circs of the second one as that seven predators hunt two evaders in Fig.1. The cooperative team can gain the reward \( S_{E_j} \) after they capture the evader \( E_j \).

3. Leaguing pursuit teams with association rule data mining technology

There are many predators and evaders in the pursuit area. When some predator finds the evader, it will inform other predators. In this section, we use association rule data mining technology to distribute these predators to hunt different evaders effectively, which creates many pursuit teams. Then the members of each team work together to hunt their goals.

Fig. 1. Sketch map of multi-robot successfully capturing the evaders
(a) is an initial state before capture
(b) is goal state where evaders were captured

3.1. Definitions of pursuit rule data mining with pursuit problems

Association rule was originally proposed by Agrawal and Srikant in 1993. They generalized it again (Agrawal, R. & Srikant, R. 1995; 1996). The aim of association rule is to find out the contact with different commodity (set) in the trade data base, that is, to look for the interesting contact in the appointed data collection. Through describing the potential relation rules between data sets in the data base, we can discover the dependence relations between many domains which are satisfied with the preconcerted threshold value of support and confidence. For discussing problems conveniently, we give several definitions of association rule with pursuit problems.

Definition 1. Let \( D_n \), \( D=[P_1,P_2,...,P_n,P_e] \) be association rule mining data collection denoting that predators \( P(i\leq n) \) hunt evaders \( E(j\leq m) \), where \( P_i \) is a pursuit robot. As an affair \( P=[k_1,k_2,...,k_q](i=1,2,...,n) \), the element in \( P_i \) is called by an Item which is the description of the correlative attribute relation between \( P_i \) and \( E_j \).

Definition 2. We assume that \( I=[k_1,k_2,...,k_q] \) \( (p\leq q) \) is composed of all items in \( D_i \). Any subset \( X \) of \( I \) is called item set. Association rule is the implication formula such as \( X\Rightarrow Y \), where \( X\subset I \), \( Y\subset I \), also \( X\cap Y=\emptyset \). X is the forward item and Y is the back item; Association rule can reflect the rule when the item of \( X \) appears, the item of \( Y \) must be. In pursuit problems, the interesting association rule can provide the reliable reason for some predator to hunt its goal.
Definition 3. The number of affairs including item $X$ in $D_i$ is modeled as the support number of item $X$, namely $\alpha_i$. The support rate of item $X$ is figured as support($X$),

$$support(X) = \frac{\sigma_x}{|D_i|} \times 100\%$$ (3)

where $|D_i|$ denotes the affair number of $D_i$. If support($X$) is not smaller than the appointed smallest support, called minsupport, we can reckon $X$ as a big item, conversely a small item. The support rate of $X \Rightarrow Y$ is the support rate of association rule $X \Rightarrow Y$, namely support($X \Rightarrow Y$)=support($X \cup Y$). And we denote the confidence of association rule $X \Rightarrow Y$ by confidence($X \Rightarrow Y$):

$$confidence(X \Rightarrow Y) = \frac{support(X \cup Y)}{support(X)} \times 100\%$$ (4)

The smallest confidence which is designated by the user could be figured as minconfidence.

3.2. Process of creating pursuit teams

The process of creating pursuit teams is categorized into the followings: (1) finding out all of big items; (2) coming into being association rules through big items; (3) building the pursuit teams.

(1) Finding out all of big items: According to definitions, the frequency that these big items appear is at least as time to depict its value, it is defined by Eq.(8)

$$Pr_{a_{ij}} = \begin{cases} \frac{T - t_0}{C_a} C_a & t - t_0 \leq T \\ \frac{C_s}{C_a} & t - t_0 > T \end{cases}$$ (8)

where $C_a$ is the times of $P_i$ capturing $E_i$, $C_t$ is the times of failure, $C_r$ is the times of being in for hunting $E_i$, $e_1$ and $e_2$ are updating thresholds. $k_i$ is $\text{Pr}_{a_{ij}}$ which shows the probability of $P_i$ capturing $E_i$ here we choose the frequency and actual effect of $P_i$ capturing $E_i$ in the latest time to depict its value, it is defined by Eq.(8)

$$k_i = \text{Cre}_{ij} = \begin{cases} \frac{C_t}{C_t} + e_1 & \text{capturing the goal} \\ 1 - \frac{C_f - e_2}{C_t} & \text{not capturing the goal} \end{cases}$$ (7)

where $C_t$ is the times of $P_i$ capturing $E_i$, $C_r$ is the times of failure, $C_t$ is the times of being in for hunting $E_i$, $e_1$ and $e_2$ are updating thresholds. $k_i$ is $\text{Pr}_{a_{ij}}$ which shows the probability of $P_i$ capturing $E_i$ here we choose the frequency and actual effect of $P_i$ capturing $E_i$ in the latest time to depict its value, it is defined by Eq.(8)

$$k_i = \text{Cre}_{ij} = \begin{cases} \frac{C_t}{C_t} + e_1 & \text{capturing the goal} \\ 1 - \frac{C_f - e_2}{C_t} & \text{not capturing the goal} \end{cases}$$ (7)

where $C_t$ is the times of $P_i$ capturing $E_i$, $C_r$ is the times of failure, $C_t$ is the times of being in for hunting $E_i$, $e_1$ and $e_2$ are updating thresholds. $k_i$ is $\text{Pr}_{a_{ij}}$ which shows the probability of $P_i$ capturing $E_i$ here we choose the frequency and actual effect of $P_i$ capturing $E_i$ in the latest time to depict its value, it is defined by Eq.(8)

$$Pr_{a_{ij}} = \begin{cases} \frac{T - t_0}{C_a} C_a & t - t_0 \leq T \\ \frac{C_s}{C_a} & t - t_0 > T \end{cases}$$ (8)

where $t$ is the current time, $t_0$ is the latest time which $E_i$ is captured , $T$ is the longest actual effect after capturing $E_i$ and $C_a$ is the times of $P_i$ hunting different evaders. $k_i$ is $\text{Rev}_{ij}$ that is the retained profits of $P_i$ after capturing $E_i$ which denotes the margin of the reward ($\text{Rev}_s$) and the cost ($\text{Cos}_s$), that is, $\text{Rev}_{ij}=\text{Rev}_s-\text{Cos}_s$, where $\text{Cos}_s$ is the sum of communication cost, resource consumption, punishment about hunting failure or exiting to pursuit teams and so on. $k_i$ is $\text{El}_{ij}$ which denotes the estimate whether $P_i$ have the ability to capture $E_i$ here $\text{El}_{ij}=\alpha \text{Cab}_s + \beta$, where $\alpha \text{Cab}_s$ is the ability value of $P_i$ hunting $E_i$, $\alpha$ and $\beta$ is the corresponding coefficient. $k_i$ is the current load state $\text{Pur}_{s_i}$, if $P_i$ doesn't have any task, the value of $\text{Pur}_{s_i}$ is spare, then $\beta<0$; if $P_i$ is hunting one goal, the value of $\text{Pur}_{s_i}$ is busy, then $\beta>0$. $P_i$ stands for predator $i$. $V_{r_i}$ is the corresponding value of $P_i$ and $k_i$. We can conclude from the definition of $k_i$ that there are two types in $V_{r_i}$, continuous and discrete value. In this paper, we use Apriori algorithm to mine the Boolean association rules so that the value of classificatory and quantitative attribute should be transformed to Boolean values. We will introduce the transformation means as following:

1) Dispersing the value of quantitative attribute. “$k_i$-$k_s$” are quantitative attributes, for the consistency of record
amounts in every group, we disperse the five quantitative attributes based on the same length or same distance method: $k_1$ is categorized into the followings: $a_1 \{V_1 \in [0, \infty]\}, a_2 \{V_2 \in (-\infty, 0]\}$

$k_2$ is categorized into the followings: $b_1 \{V_3 \in (0.9, 1]\}, b_2 \{V_3 \in [0.5, 0.9]\}, b_3 \{V_3 \in [0.1, 0.5]\}$

$k_3$ is categorized into the followings: $c_1 \{V_4 \in (0, 0.3]\}, c_2 \{V_4 \in [0.3, 0.6]\}, c_3 \{V_4 \in [0.6, 1]\}$

$k_4$ is categorized into the followings: $d_1 \{V_5 \in (-\infty, 0]\}, d_2 \{V_5 \in (0, 200]\}, d_3 \{V_5 \in (200, 1000]\}$

$k_5$ is categorized into the followings: $e_1 \{V_6 \in [0, 0.3]\}, e_2 \{V_6 \in [0.5, 0.9]\}$

2) Transforming the value of classificatory attributes. $Pur_{si}$ is a classificatory attribute, which should be transformed to the Boolean type, then $Pur_{si}$ is categorized into the followings: $f_1 \{V_7 = \text{"spare"}\}, f_2 \{V_7 = \text{"busy"}\}$.

For $p_i$, if $V_8 > 0, V_9 > 0.63, V_9 > 0.35, V_{10} = 500, V_{11} > 0.61, V_{12} = \text{"spare"}$, then Table 1 could be transformed to Table 2 after above disposal.

<table>
<thead>
<tr>
<th>TID</th>
<th>List of Transaction Item</th>
</tr>
</thead>
<tbody>
<tr>
<td>$p_i$</td>
<td>$a_1, b_2, c_2, d_3, e_1, f_1$</td>
</tr>
</tbody>
</table>

Table 2. Affair data transformed from Table 1

We can obtain the affair data base of all predator states through the homologous process.

2) Coming into being association rules through big items: Here we employ conventional Apriori algorithm, and we note minsupport by the type of evader in order that the pursuit team can not capture its goal because its scale is too small, that is, if the type of some evader $RDE_j$ is IV, it could be captured by four predators at least. Note that our affair data base has twelve sample records, minsupport=4/12=33%. To make sure of the confident degree of the gained rules, minconfidence should be larger than the maximum of $C_0$. And $C_0$ is updated constantly, if $C_0$ is too low, we can gain so many uninteresting rules, which leads to be hard to build the pursuit teams. So the lowest threshold of minconfidence is destined for 0.5, that is, minconfidence=max(0.5, max$C_0$).

We make the assumption any frequency item $l$ and its nonempty subset $s$, if there is the following formula:

$$\text{support}(l) \geq \text{minconfidence}, \text{ then we can input the rule }$$

$$s \Rightarrow \{l-s\}$$

3) Building the pursuit teams: We can judge whether $P_i$ choose $E_i$ as its goal or not from the gained association rules. In this paper, we therefore assume that $k_1, k_2, k_3$ are the former attribute, $k_4, k_5$ are the back attribute for which should be in $\{c_2, c_3, d_2, d_3\}$ theoretically, and $k_6$ is the reference attribute. If this results in interesting rules such as $A: a_1b_2e_1 \Rightarrow c_3d_2 
\{35\%\,72\%\}$, we can explain it: there are 35% of predators whose force pulled by $E_i$ is more than pushed by obstacles, who have the abilities to hunt the goal, whose dependability degree is good, if it will hunt $E_i$, the probability of success more than 0.6 and the value of its reward over 0-200 is 72%. We should scan the origin sample data item $D_t$ in the light of these association rules, then pick up 35% of predators who is satisfied with $a_1, b_2, e_1$. For the above case, $12 \times 0.35=4$, then the cooperative team can be made of these four eligible predators to hunt the same goal. However, if the amount of appropriate predators is more than $R_{DE_j}$, it is preconditions of choosing partners that we should keep the stability of system. Some predator has priority whose $Pur_{si}$ is spare. If their $Pur_{si}$ is same, we should think about $P_D$ and $R_{DE}$. Assuming $V_i=\mathbf{a}_1 \mathbf{b}_2 \mathbf{e}_1 \Rightarrow \mathbf{c}_1 \mathbf{d}_2 \{40\%\,72\%\}$, we also scan the data item $D_t$ and find out those predators which are fit for $B$, then ignore them. Furthermore, we make use of the same support and confidence to mine interesting rules which is satisfied with hypothesis condition from the leaving predators $(12-12 \times 0.4=8)$. If we find $C: a_1b_2e_1 \Rightarrow c_1d_2 \{37\%\,75\%\}$, we choose the predators which is satisfied with $C$ to join the pursuit team. We can conclude that there are about $3(8 \times 0.4 \approx 3)$ predators. If $R_{DE}=IV$, we let $U_i=\mathbf{a}_1 \mathbf{b}_2 \mathbf{c}_3 \mathbf{d}_3 \mathbf{e}_0$, whose $a_1$ and $a_2$ and $a_3$ are scale coefficients respectively. Well then, we choose the leaving predators in $D_t$ whose $U_0$ is the highest value to join the pursuit team. And circulating the process, we can build the cooperative team to $E_i$ called $CG(E_i)$.

4. Multi-robot cooperative pursuit algorithm based on reinforcement learning

4.1. Forecasting the mobile positions of pursuit objects

In this part we propose a multi-robot cooperative pursuit algorithm. When the dynamic evaders appear, we use the association rule mining data technology to build cooperative teams to different evaders. The members of $CG(E_i)$ coordinate their actions to hunt $E_i$. We assume that evaders can move randomly without pursuit threat, contrarily evade intelligently. To capture the intelligent evaders, predators must forecast the positions of goals. It is the aim of $P_i \in CG(E_i)$ that $P_i$ gets to the neighbor grid of the goal, which makes the distance $d(x(t), x_i(t))$ from itself to the goal shortest. $E_i$ does its best to evade the predators. According to the pursuit definition, the contribution probability of every predator in $A(x(t))$ is $1/R_{Ei}$. We assume that the distance, obstruction and surrounding contribution of pursuit members are $OC=1/N_0(x(t))$, $EC=\sum \theta_{\omega}(l) \pi$ respectively in the alert range of $E_i$, where $N_0(x(t))$ is the number of neighbor grids that $E_i$ can get to and $\theta_{\omega}(l)$ is
the anticlockwise angle between neighbor predators and $E_j$ at time $t$. So, we can compute the probability of capturing $E_j$ as follows:

$$\Pr(P_{\text{cap}}(E_j)) = \begin{cases} 0 & \forall \delta_i(t_i), \delta_j(t_j) > L_E_j \\ (\mu_1 \cdot DC + \mu_2 \cdot OC + \mu_3 \cdot EC) / L_E_j \\ P_i(t_i), \delta_j(t_j) \leq L_E_j, k_1 + k_2 + k_3 = 1 \end{cases}$$ (9)

$P_i$ anticipates that the intelligent $E_j$ will choose the cell as its position from $A(X_i(t)) \cup X_j(t)$, which makes $\Pr(P_{\text{cap}}(E_j))$ smallest at time $t+1$. So, $P_i$ moves to the goal position from $A(X_i(t)) \cup X_j(t)$ whose $d(x_i(t+1), x_j(t+1))$ is the shortest.

4.2. Cooperative teams hunting objects

To avoid the conflict with predators, if different predators who are in the same team choose the same grid as its target position, the predator who is near to the evader has the priority to hold its target position next $t$. If members who belong to the different teams choose the same cell as its target position, the predator who is in the evader’s alert range has the priority. We assume that $P = \Delta C_k + A_k \cdot C_k + A_k \cdot E_k$ is the predator's each action contribution to capture the evader. If any two predators who are in the different teams are all in the alert range of their objects, the predator whose $P$ is higher has the choice priority. On the contrary, the predator that is near to its object has it. After confirming the aim position, we can transform the pursuit problem into the multi-robot path planning problem based on the known environment. One of the key problems for pursuit teams is the problem of coordination: how to ensure that the individual decisions of the predators result in jointly optimal decisions for the team.

Reinforcement learning (Sutton, R. S. & Barto, A. G. 1998; Kaelbling, L. P.; Littman, M. L. & Moore, A. W. 1996) is the problem faced by an agent that must learn behavior through trial-and-error interactions with dynamic environments, and it has been applied successfully in many single agent systems for learning the policy of an agent. Learning from the environment is robust because agents are directly affected by the dynamics of the environment. In principle, it is possible to treat a multiagent system as a ‘big’ single agent and learn the optimal joint policy using standard single-agent reinforcement learning techniques and therefore in this paper we are interested in fully cooperative pursuit systems in which the predators have to learn to optimize a global performance measure. Q-learning (Watkins, C. J. & Dayan, P. 1992) is one of important reinforcement learning algorithms, which is one kind of model-free reinforcement learning algorithm that is based on dynamic programming. In essence Q-learning is a temporal-difference learning method. In this paper we are interested in using Q-learning to learn the coordinated actions of many groups of cooperative predators to find out the best path of capturing objects without collision between agents. So, we should think the pursuit team as a big agent. We can use Q-learning to directly compute the approximation of an optimal action-value function, called Q-value, by using the following update rule:

$$Q_{t+1}(s_i, a_i) = (1 - \alpha_r) Q(s_i, a_i) + \alpha_r \cdot \max_b Q_{t+1}(s_{t+1}, b)$$ (10)

where $Q$ is an estimate of the predators’ state-action pairs at time $t$, $s_i$, $a_i$, $r$, $n$ are state, action, the learning rate, discount factor and reward respectively; $\max_b Q_{t+1}(s_{t+1}, b)$ is gained by comparing the value of all kinds of actions. Here, let the relative positions between predators and evaders, the sensor distance information of predators as the input signal of the pursuit reinforcement learning system. Predators can choose the appropriate actions set $[a_1'...a_n']$ after learning, where $a_i'$ denotes the action of $P_i$ at time $t$, in this paper we set $a = 0.3$, $\gamma = 0.9$. Here the instantaneous reward is defined as the following rules. When all pursuit members are out of their objects’ alert ranges, we make use of distance to give rewards, let $\Delta D = \sum d(x_i(t), x_j(t)) \cdot d(x_i(t-1), x_j(t+1))) / N_{CG(\theta_P)}$ where $N_{CG(\theta_P)}$ denotes the member number of CG($E_j$). If $\Delta D > 0$, then to the positive rewarded and vice versa; however, we should set down the reward rules in the light of the degree of obstruction and encircling besides distance if there are predators in the alert range of $E_i$. It is the perfect condition that pursuit members distribute in the neighbor grids of evaders and $\theta_{\text{tr}}(t) = \pi/2$. So, let $\Delta N = N_{\text{alert}}(\theta_{\text{tr}}(t)) - N_{\text{CG}(\theta_{\text{tr}}(t))}$ if $\Delta N > 0$, then to the positive rewarded and vice versa; when $0 < \theta_{\text{tr}}(t) < \pi/2$, the reward is positive if $\theta_{\text{tr}}(t) > \pi/2$, and when $\pi/2 < \theta_{\text{tr}}(t) \leq \pi$, the reward is positive if $\theta_{\text{tr}}(t) < \pi/2$. The collision actions could be published. In this paper the reinforcement signal $r$ is: $r = r_{\text{tr}} + r_{\text{ob}} + r_{\text{enc}}$, and when $d(x_i(t), x_j(t)) > L_i$, $\delta = 1$, $\delta = 0$, $\delta = 0$, otherwise $\delta$, $\delta$, $\delta = (0,1), \delta + \delta + \delta = 1$, where $r_1$ $r_2$, $r_3$ are given by the following equations:

$$r_1 = \begin{cases} 1 & \text{if } \Delta D \leq 0 \\ -1 & \text{if } \Delta D > 0 \end{cases}$$

$$r_2 = \begin{cases} 0.5 & \text{if } N_{\text{alert}}(\theta_{\text{tr}}(t)) \leq 0 \\ -0.5 & \text{if } N_{\text{alert}}(\theta_{\text{tr}}(t)) > 0 \end{cases}$$

$$r_3 = \sum_{i \in \text{all agents}} \lambda \left[ 1 \cdot (e^{[\theta_{\text{tr}}(t) - \theta_{\text{tr}}(t-1)]} - 1) + \lambda \cdot (e^{[\theta_{\text{tr}}(t) - \theta_{\text{tr}}(t-1)]} - 1) \right] / \left[ 1 + e^{[\theta_{\text{tr}}(t) - \theta_{\text{tr}}(t-1)]} \right]$$ (11)

where $\lambda = 1$, $\lambda = 0$ if $0 < \theta_{\text{tr}}(t) \leq \pi/2$ and $\lambda = 0$, $\lambda = 1$ if $\pi/2 < \theta_{\text{tr}}(t) \leq \pi$. 

333
5. Experimental setting and simulation results

To evaluate the approach proposed in this paper, we perform simulation experiments on an instance of the pursuit problem in a 360×250 rectangle area which is divided to 900 toroidal grids of 10×10. The obstacles are constant solid areas. The predators and evaders are all 10×10 cycloidal bodies. The alert radiuses of all of agents are same to be 3. We mark every agent with the ID number. If the evader is captured, then the other evader with the same style is relocated at new random positions. They have perfect communication system and same speed (move one cell distance each time). After capturing its object successfully, the cooperative team is rewarded by the third power of the evader’s style, which is distributed to every member averagely, then the cooperative team is dismissed.

We use the same initial condition to compare our algorithm (after convergence) which we call it A with the algorithm proposed by Zhou, P. C.; Hong B. R. and Wang Y. H. (2005) which we call it B under above environment (ten predators hunt four evaders). A sample configuration is shown in Fig. 2(a). We take the capturing time of pursuit teams in each experiment as the evaluation index. From these results of A and B, we can find that the capturing time is little different for one algorithm of every experiment, however, the average time is much different in them. And Fig. 2(b) shows the reward comparison between them. We take one sample every ten experiments for the clear results. We can find that, there is a holistic performance increase of 18 percent of our algorithm. These can be explained as following: 1) we choose the pursuit members after contrasting the current state with the historical factors in A, which is relatively general, but case-based reasoning is used, which could lead to its adaptability worse in term of historical cases only under dynamic multi-robot environment. 2) It is assumed that evaders move randomly and predators have not learning ability in B, while in A we think about two conditions, moving randomly and evading intellectively, and reinforcement learning is used to choose the best actions which is more satisfied with the practice. So, A is better than B.

To test the effect of different decision mechanisms of building pursuit teams to capturing results, we compare the stochastic selection strategy(1) with the selection strategy based on case reasoning and assistant decision matrix(2) and the selection strategy based association rule(3) proposed in this paper. Under the same experiment environment, we increase the number of evaders incessantly from four to ten. The results for three algorithms are displayed at Fig.3. It is clear that the result based on association rule is the best which takes the least time, and the result of the stochastic selection strategy is the worst. This is because there are many factors considered such as ability, credibility, force and cost of predators, and so on, then to build reasonable cooperative teams to the special goal, which can improve the efficiency while avoiding to waste resources.

At last, we choose the same pursuit team to test two pursuit algorithms, one is based on reinforcement learning, and the other is the greedy actions strategy. We take the average pursuit benefit as the evaluation index, that is, how much the predator gain when it moves one grid every time. Table 3 shows the results. We see that the pursuit algorithm based reinforcement learning is better than the other. It is because that the predators are selfish.
Table 3. Pursuit benefits of different learning mechanisms

<table>
<thead>
<tr>
<th>Pursuit strategy</th>
<th>based on reinforcement learning</th>
<th>greedy actions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Average reward of pursuit teams (R)</td>
<td>1563.7</td>
<td>1396.2</td>
</tr>
<tr>
<td>Average number of predators covering grids (N)</td>
<td>199</td>
<td>216</td>
</tr>
<tr>
<td>Pursuit benefit (R/N)</td>
<td>7.86</td>
<td>6.42</td>
</tr>
</tbody>
</table>

In this paper, a multi-robot cooperative pursuit algorithm is proposed. Firstly, we use association rule data mining technology to choose the goals for every predator. Then the cooperative team for the appointed goal is composed of the predators whose goal is same. Secondly, the members of the cooperative team coordinate to hunt their goal. Through forecasting the position of evader, we transform the pursuit problem to the path planning problem and use multi-robot reinforcement learning to choose the best action strategies. Experimental results showed that the proposed algorithm could be used to achieve the optimal solution. However, confidence will affect the result of building teams. If it is higher, we could not find the suitable interesting rules to build pursuit teams. On the contrary, we could find the improper teams. Future work should examine more appropriate confidence. Furthermore, we use reinforcement learning to solve out the cooperative problems in the course of hunting evaders. However, both the state and action space scale exponentially with the number of predators and evaders, rendering this approach infeasible for overfull robots, which can affect the hunting efficiency. Consequently we should find out a new method to reduce the number of state and action pairs and then improve the learning speed. In addition, we will expand our algorithm ulteriorly to apply it to the unknown pursuit environment. To resolve the conundrums is a challenge.

7. References

