CIT. Journal of Computing and Information Technology, Vol. 27, No. 1, March 2019, 1–12
doi: 10.20532/cit.2019.1004318

Multi-Agent Pursuit-Evasion Game Based on Organizational Architecture

Mohammed El Habib Souidi[1], Abderrahim Siam[1], Zhaoyi Pei[2] and Songhao Piao[2]

[1]Department of Computer Science, University of Khenchela, Khenchela, Algeria
[2]Computer Science and Technology, Harbin Institute of Technology, Harbin, China

Multi-agent coordination mechanisms are frequently used in pursuit-evasion games with the aim of enabling the coalitions of the pursuers and unifying their individual skills to deal with the complex tasks encountered. In this paper, we propose a coalition formation algorithm based on organizational principles and applied to the pursuit-evasion problem. In order to allow the alliances of the pursuers in different pursuit groups, we have used the concepts forming an organizational modeling framework known as YAMAM (Yet Another Multi Agent Model). Specifically, we have used the concepts Agent, Role, Task, and Skill, proposed in this model to develop a coalition formation algorithm to allow the optimal task sharing. To control the pursuers' path planning in the environment as well as their internal development during the pursuit, we have used a Reinforcement learning method (Q-learning). Computer simulations reflect the impact of the proposed techniques.

ACM CCS (2012) Classification: Computing methodologies → Artificial intelligence → Distributed artificial intelligence → Multi-agent systems

Theory of computation → Theory and algorithms for application domains → Algorithmic game theory and mechanism design → Convergence and learning in games

Keywords: Pursuit-Evasion games, organization, coalition formation; Q-learning

1. Introduction

Pursuit-Evasion (PE) games are considered as one of distinctive distributed artificial intelligence areas treating the cooperative decision problem in Multi-Agent Systems (MAS). It mainly focuses on the cooperative approaches used to connect many autonomous agents playing the role of pursuers to capture the mobile evaders. In fact, the pursuit problem requires coordination of the pursuers' actions. It is widely used in relation to the different types of coordination mechanisms and coalition formation algorithms since it came into being.

With the aim of describing this problem, several kinds of environment have been used in the recent research activities. For example, in [1] the authors used an m-dimensional Euclidean space where the evaders and pursuers can move at the same speed. Noting that, for each evader there exists a finite set of pursuers trying to capture it. They have proved through a simple constructive method that a k-capture is always achievable when the pursuer lies inside the pursuers' k-hull (The k-hull is defined as a set of all points p such that any line passing through p divides the given points into two sets of k points each at least). In [2], the problem was studied under the wireless sensor and actor networks (planar environment) where a novel multi-step cooperative pursuit based on Daisy-Chain Formation algorithm and a sliding mode-based method were used to control the pursuit process. Moreover, the game was introduced in two cases. In the first one, the evader is always assumed to be static in the environment. In the other one, the evader will move once it detects the pursuers. In [3], the pursuit game was presented in an environment totally based on graph theory, in which the main objective was to localize the node containing the evader. For this purpose, the Iterative Greedy Node Search (IGNS) algorithm was undertaken to enable offline guaranteed search. In this paper, we are focused on two-dimensional grid of cells environment [4], [5], in which the evader is considered as captured when the expected locations (cells) are occupied by the pursuers or by static obstacles.

Coordination is a part of multi-agent systems, in which the interdependencies between activities of the agents might rise from the need to have shared resources or to unify efforts to solve a more complex problem. Division of a task into sub-tasks, processing with overlapping subtasks, or even impossibility of dividing a task to appropriate sub-tasks [6], [7] need having access to efficient process to select which agent, when, and how to assign tasks to. Durfee [8] determines coordination of agent as "an agent's fundamental capability to take the decision regarding its own actions in the context of activities of other agents around it". The coordination allows the agents to intentionally join their efforts and resources to resolve global goals. In this paper we have focused on organizational structuring framework, which is considered as coordination mechanism's type. Indeed, we have used the different concepts proposed in YAMAM organizational model [9] as well as the relations between them in order to propose a pursuit coalition formation algorithm enabling optimal grouping of the pursuers to achieve the capture of different evaders detected.

Regarding the pursuers' path planning, we have applied the reinforcement learning (RL) method known as Q-learning to optimize the movement of the agents during the pursuit. RL algorithm is regarded as a simple framing of the learning problem based on the environmental interactions in order to reach a specific objective [10]. In RL, each agent characterized by a level of skills enabling the interactions with the environment is called the decision-maker. Information generated in the environment is communicated to the agent through interactions between each other. In fact, the agent selects an action to execute in the environment from the received information. The environment changes provoked in different ways by the actions are transmitted to the agent via a scalar reinforcement signal. The actions of the agent are performed to maximize the rewards generated by the environment as special numerical values. The temporal difference learning methods, the dynamic programming, and Monte-Carlo represent the three main methods used in RL. Indeed, temporal difference learning methods are based on different concepts extracted from both dynamic programming methods and Monte Carlo techniques [11].

The paper is organized as follows: The problem formulation and the description of the environmental elements are defined in Section 3. Section 4 contains the principles of YAMAM organizational model as well as the relations between their different concepts and information how the coalition formation algorithm is extracted from this organizational model. In the same section, we highlight the motion strategy of the agent based on RL method. Section 5 reveals the positive impact imposed by this proposition in relation to some relevant work.

2. Related work

There are several benefits regarding the application of multi agent systems. One of the most important benefits is the parallel resolution of the problems [7]. This fact aims to model classical artificial intelligence to simultaneously execute several tasks. We can note this point in the PE problem when the pursuit of the detected pursuers is effectuated in parallel way. The second benefit is the distributed resolution of the problems [10]. This principle is reflected by the agents' autonomy. We can note that in PE problem through the communication of the pursuers with each other and with the environment during the pursuit. Multi-agent based simulation can also be considered as a great benefit, it easily allows the implementation of different social phenomena on micro and macro levels (flexible number of used agents). We can deduce this fact in PE problem through the flexibility of the used pursuers and evaders.

The Organization concept is often used in MAS in different ways such as in Holonic multi-agent manufacturing systems [12], in which a two-dimensional self-organization mechanism was conceived taking into account structural and behavioural vectors to achieve truly evolutionary and flexible systems. Moreover, other works use an underlying organization to lead the coalition formation of the agents [13], [14].

Application of the pursuit coalition formation process based on MAS organizational models is a recent research activity regarding the distributed artificial intelligence. In [15], the authors focused on Agent Group Role (AGR) organizational model [16] in order to extract a coalition formation algorithm allowing the alliances of the pursuers in different pursuit groups. Also, they demonstrated effects of the pursuit groups' stability on the capturing time as well on the pursuers' development during execution of the tasks. In the same context, they proposed a flexible organizational model extended from AGR through the application of fuzzy logic principles in order to equip each pursuit group with a membership function used to determine the membership degree of each pursuer. This model is known as Agent Group Role Membership Function (AGRMF) [17], [18], [19]. In addition, they showcased the impact of this fuzzy access mechanism on the roles' attribution and on the dynamism of coalition formation.

In relation to game theory, a pursuit coalition formation algorithm based on the iterated elimination of dominated strategy (IEDS) model was proposed to provide a pursuit task coordination mechanism [20]. This coalition formation algorithm is based on the iterated elimination of the dominated pursuit groups, causing a certain equilibrium between the pursuit groups selected and excluding any problem related to the negative externalities. Furthermore, this approach showcases an interesting decentralized calculation of the possible coalition formations.

Application of coalition formation in PE games is very common, such as in [21] where the authors proposed a pursuers' coalition formation algorithm to improve the real-time computation of the optimal coalition. Knowing that, this computation exponentially increases in relation to the number of the pretending pursuers. Specifically, this algorithm is based on greedy optimal gains, allowing assignment of the pursuers in different coalitions.

Also, there are other types of coordination mechanisms such as Contract Net Protocol (CNP) applied to the pursuit-evasion games where a cooperation mechanism of multiple agents used to form the pursuit alliance to capture different kinds of evaders is introduced [22]. Specifically, they have extended CNP through the improvement of the alliance decision making as well as its dynamism to develop a cooperative pursuit algorithm based on the dynamism of the formed alliances.

RL is usually used in multi-agent path planning. Q-learning can be used through several approaches, not requesting the agent to have access to the information on the functioning of the environment. Q-learning evaluates state-action values (Q values), which are numerical evaluators of quality for a given action to transit between two different states [23]. Also, there are several interesting works based on RL and treating PE games in grid of cells environment [24], [25] with the aim of finding the optimal path of the pursuers during the pursuit process. In [26], the authors introduced a Stochastic Shortest path-based Q-learning (SSPQL) method improving the convergence speed in comparison with Q-learning. Moreover, SSPQL works very well when the state transitions are stochastic. However, it needs considerable memory consumption in order to store the state-transition probabilities as well as stochastic shortest paths. In [27], they presented a Reinforcement Learning algorithm regarding the problems in which a poissonian stochastic time delay is encountered in the agents' reinforcement signal. This algorithm can design an adapted control policy for the agents' environment despite the presence of environment noise.

3. Pursuit-evasion Problem Description

In this section, we depict the pursuit evasion setting considered. The environment is represented by a limitary rectangular grid of cells of the same size and shape [28]. Knowing that, this approach is also applicable to any other type of environment. There are three types of cells in the environment: free cells, cells occupied by agents, and cells occupied by obstacles. The agents are equipped with environmental sensors permitting avoidance of the different obstacles as well as detection of each other's location in the environment. Moreover, we note that the environment is assumed to be fixed with static obstacles. Pursuer and evader reflect the two types of agent used. Both agents can only move horizontally and vertically to the adjacent cells. The pursuer and the evader act at discrete time steps and execute pursuit and evasion iterations respectively at each pursuit-evasion step. The evaders differ according to the number of the pursuers required to accomplish their capture. When the evader's adjacent cells are occupied, we deduce that the capture is performed. The pursuers are characterized by different dynamic skills allowing their differentiation during formation of the pursuit coalitions:

3.1. Task Processing Degree (Ω)

In pursuit-evasion game, the pursuers are equipped with the skills enabling performance of the different tasks required. The task processing degree determines the achievement's probability of the pursuers' set in relation to the evader concerned. Denoting the number of pursuers required to perform the pursuit of one evader (E) with Ψ, and $Gr = \{P_1, P_2, \ldots, P_\Psi\}$, this factor is computed as follows:

$$\Omega\left(Gr,E\right)=\sum_{i=1}^{\Psi}\frac{\max\left(0.1,\dfrac{t_S^i}{t_a^i}\right)}{\Psi},\quad \forall\Omega\in\left[0.1,1\right] \tag{1}$$

where t_S is the number of tasks achieved by the pursuer and t_a is the number of tasks affected to the pursuer.

3.2. The Average Goal Distance

The distance from the evader to the pursuers in the environment involves a significant impact on the pursuit processing. More precisely, it enables calculation of the minimum average distance between the set of pursuers and the purchased evader. The position **Pos** is computed as follows:

$$Dist\left(Gr,E\right)=$$

$$=\sum_{i=1}^{\Psi}\frac{\sqrt{\left(Cor_{p_x^i}-Cor_{E_x}\right)^2+\left(Cor_{p_y^i}-Cor_{E_y}\right)^2}}{\Psi} \tag{2}$$

where (Cor_{P_x}, Cor_{P_y}) are the coordinates of a specific pursuer and (Cor_{E_x}, Cor_{E_y}) are the coordinates of the evader concerned.

3.3. The Task Acquaintance (Tying)

The pursuit coalition is formed at the beginning of the task execution, and it dissolves when the task is performed. Next time when the same group of agents is assigned to the same task, it is obligatory to repeat the interactions and reconstitute a new coalition. The factor of the task acquaintance is introduced here with the aim of reducing the communication rate as well as avoiding the repeated information in interactions. Tying, denoted by $Tying_{ij}$ is the level of experience for a group of agents$_i$ in relation to the task T_j. The initial value of this dynamic parameter is zero, and when the task is performed, it will be updated as follows:

$$Tying_{ij}=\begin{cases}\min\left(Tying_{ij}+\Upsilon_1,1\right), Tying_{ij}\neq 0\\ \Upsilon_2, \qquad\qquad\qquad Tying_{ij}=0\end{cases} \tag{3}$$

Figure 1 describes the environment proposed. It determines how the agents move between the cells, and how the pursuers intercept the evaders to perform the capture. Different colors of the pursuers mean that they belong to different pursuit groups.

4. The Proposed Solution

In this section we will explain the different concepts forming YAMAM organization model

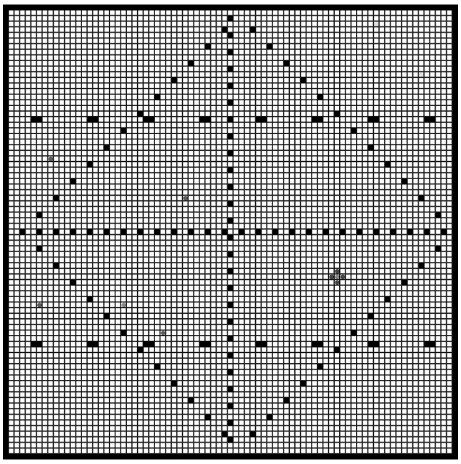

Figure 1. Grid of cells simulation environment.

and also describe how they are used in order to allow the coalition of pursuers in different pursuit groups to chase the detected evaders in the environment (Task planning). In addition, we will explain the proposed stochastic Q-Learning method allowing the pursuers to move in the direction of the evader concerned after the groups' formation (Path planning).

4.1. YAMAM Organizational Modeling Framework:

The YAMAM organizational model [9] is an alternative to other models based on agents such as AGR [16], and AGRMF [17]. The main advantages of this model can be summarized on its good modularity, and possible scalability. Also, it can be interfaced with task planning tools, as it will be used in this paper. YAMAM is based on four different concepts: Agent, Skill, Role and Task. In this model, the organization is described by its inherent structure. Therefore, relations between the agents are primordial in relation to the agents and their behavior.

4.1.1. Agent

In this organizational structure, an agent is defined as an autonomous and communicating entity situated in an explicit or implicit environment. In fact, the agent is built on reactive proprieties, thus implying that the skills cannot be added dynamically. However, it is possible to instantiate a cognitive agent able to use reactive and cognitive skills so as to evaluate over time.

4.1.2. Role

The role reflects a service or an agent identification form. The agent can manage with one or several roles regarding a specific environment. Moreover, the role is based on a set of tasks to perform in different ways. It is supposed that an agent can play a role only if it is able to execute the tasks involved, and thus if it possesses the skills required. Usually, the role requires execution capacity of several tasks. Consequently, an agent must be characterized by the skills related the tasks concerned. In this paper, this concept is used to attribute the role "Pursuer" which differs from one pursuit group to another. Moreover, we note that each agent cannot play the role "Pursuer" in more than one pursuit group.

4.1.3. Skill

The skill is a necessary unit of knowledge in the processing of a given task. There are two types of skills: the cognitive skill, which could be received only by the cognitive agents, and the reactive skill, which could be also received by the reactive agents. An agent can aggregate several skills in order to execute the set of expected tasks required to play a specific role. In this paper, the pursuer's skills are represented by the task processing degree (*Ω*), the distance separating the pursuer from the target (*Dist*), and the task acquaintance (*Tying*) as detailed in Section 3.

4.1.4. Task

In order to be achieved the task can be regarded as the operation of a skill or as an action requiring one or several skills. In pursuit-evasion game, the different pursuits represent the tasks to execute. Also, each evader requires a specific number and type of pursuers to be captured (skills).

Figure 2 details the different relations between the concepts constituting YAMAM organizational model. However, unlike AGR organizational model, the concept group is not physically implemented in this model. In other words, the agents have no vision regarding the existing groups and totally ignore their membership information. In our proposal, the groups will be represented by the different coalition formations formed to capture the detected evaders.

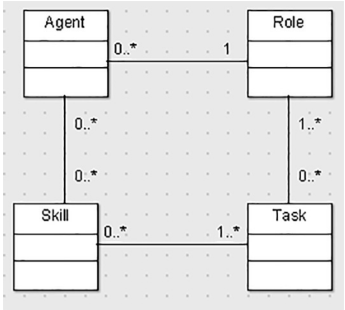

Figure 2. YAMAM meta-model.

4.2. Pursuers' Coalition Formations Based on YAMAM Organizational Model

The coalition formation problem attracts the efforts of recent researches in multi-agent systems [27], [29]. The problem implicates exchange of information amongst collaborating agents to form mutually profitable coalitions. The main objective of this section is to showcase how the coalition formations of the pursuers are performed through the different concepts proposed in YAMAM organizational model. In other words, we explain how the pursuers integrate the pursuit groups (playing the role) according to their skills' degree in order to execute the existing tasks (capture of the evaders). The pseudo-code of the algorithm's steps can be summarized as follows:

Algorithm 1. Pursuit coalition based on YAMAM.

```
Input: Evaders detected
Output: Optimal_coalition
Evaders_Detection ();
$Nbr_{cl} \leftarrow N_p! / (N_p - X)!$;
while coalition_list ≠ end do
   Coalition-fragmentation (i);
   Roles-supposition (i);
   TSD (i) ← Average (Ω, Dist, Tying);
end while;
while coalition_list ≠ end do
      if TSD (i) = max then
Optimal_coalition ← coalition (i);
end if;
end while;
while pursuers-list ≠ end do
      if p (i) ∈ Optimal_coalition then
Role_confirmation (p (i));
      end if;
while life > 0 do
Pursuit_iteration;
     Update (Ω);
end while;
Update (tying);
end.
```

Different steps shown in the algorithm above are explained in the following manner: after the detection of the different evaders in the environment, the possible coalitions to perform the pursuits at the same time will be determined. Noting that each coalition enrolls the supposed number of pursuers required to capture all the evaders detected. The number of possible coalitions is computed as follows:

$$Nbr_{cl} = \frac{y!}{(y-\psi_1)!\psi_1!} \times \frac{y-\psi_1!}{\left(y-(\psi_1+\psi_2)!\psi_2!\right)} \times \ldots \times \frac{y-(\psi_1+\ldots+\psi_{m-1})!}{\left(y-(\psi_1+\psi_2+\ldots+\psi_m)\right)!\psi_m!} = \Pi_{j=1}^{m} \frac{\left(y-\sum_{k=0}^{k=j-1}\psi_k\right)!}{\left(y-\sum_{k=0}^{k=j}\psi_k\right)!\psi_j!} \tag{4}$$

$$X = \psi_1 + \psi_2 + \ldots + \psi_{\mathrm{m}},\ \psi_0 = 0,$$

where m is the number of evaders detected and y is the number of pursuers existing in the environment.

Secondly, the generated coalitions will be fragmented into several pursuit groups in which each one of them represents the alliance to capture only one evader. Specifically, in each coalition the pursuer can only belong to one pursuit group, that is to say, it plays only one role:

$$cl = \{Gr_1, Gr_2, \ldots, Gr_x\}$$
$$Gr = \{P_1, P_2, \ldots, P_\psi\}$$
$$\forall P_x \rightarrow \exists Gr_x \in cl : P_x \in Gr_x.$$

Where Gr is the pursuit group concerning the capture of one evader, while cl is the set of pursuit groups.

After the specification of the coalitions, the task's skill degree of each one will be computed. This calculation is totally based on the pursuers' ability factors (pursuers' skills) explained in Section 3:

$$TSD = \frac{\eta_1 \times \Omega + \eta_2 \times Dist + \eta_3 \times Tying}{\eta_1 + \eta_2 + \eta_3} \tag{5}$$

where η is the pursuer skill's factor.

According to the obtained degrees, the maximum value will be selected in order to choose the optimal coalition. Also, the supposed roles of the agents belonging to the optimal coalition will be confirmed.

After the implementation of the coordination mechanism, the pursuits will start simultaneously and will last for a specific time. Their duration is unified and determined by a temporal variable "life".

The task processing degrees of the pursuers are updated after each pursuit step (iteration) is completed, as defined by the agents' displacement in the environment which follows the motion strategy explained in the next subsection. Otherwise, the task acquaintance will be updated in case of successful capture at the end of the pursuit life cycle. Figure 3 shows how the concepts forming YAMAM are implemented to model the pursuit processing (group access mechanism, roles attribution, and the task definition). In other words, the concept Role is reflected by the two roles Pursuer and Evader. The concept Task is reflected by the agents playing the role Pursuer (should be captured during the pursuit). The concept Skill is reflected by the task processing degree (*Ω*), the average goal distance, and the task acquaintance (*Tying*) explained in Section 3. These parameters are used in order to determine if an agent is able to integrate a pursuit group.

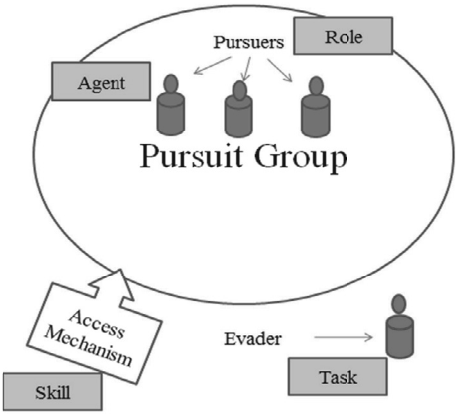

Figure 3. YAMAM concepts applied to the pursuit-evasion game.

4.3. Motion Planning Control Based on Q-learning

The goal of this section is to highlight how the motion strategy of the pursuers is implemented after their coalition according to a reinforcement learning method. Multi-Agent Q-learning is an extension of QL to multi-agent environments. It concerns the problems in relation with the learning of optimal behaviour from the point of view of an agent acting in a multi-agent environment. At the beginning, the environmental dynamics and the algorithms undertaken by the other players are unknown to the given agent. All agents select their actions independently and concurrently, execute them in parallel, and observe the same reward related to the joint action. In this single-agent case, the RL can usually be described by Markov decision process MDP, in which an agent must select the sequence of actions that maximizes some reward-based optimization criterion. MDP is defined as follows:

$$MDP = \langle S, A, T, R \rangle$$

S determines the finite set of the environment states $S = \{s_1, s_2, ..., s_n\}$, *A* determines the finite set of the actions $A = \{a_1, a_2, ..., a_\mathrm{n}\}$, T: $S \times A \times S \rightarrow [0, 1]$ defines the transition probabilities between the states *s* and *s*' through the execution of the action *a*, R: $S \times A \times S \rightarrow \mathrm{R}$ defines the reward function returning the immediate payoff obtained via each possible transition.

In order to implement the greedy strategy, each pursuer must select its actions with the aim of maximizing the value function defined as follows:

$$V(s) = E\left[\sum_{f=0}^{\infty} \lambda^f r_f\right] \qquad (6)$$

where r_f determines the payoff obtained during the transition from s to *s*', and the scalar ($\lambda \in [0,1]$) represents the discount factor. The resolution of *MDP* consists of finding a mapping from state to action. This mapping is known as Policy π: $S \times A \rightarrow [0, 1]$. $\pi(s, a)$ represents the probability of selecting the action a in state s. In the case of finite *MDP*, there exists at least one optimal policy π* which is consistent with the following rule:

$$V^{\pi*}(s) \geq V^{\pi}(s) \qquad (7)$$

Q-learning enables values to be approximated from empirical samples occurred during the actual experiences. The agent launches with arbitrary Q-values and updates them as follows:

$$Q_{i+1}(s,a) = (1-\alpha_1)Q_i(s,a) + \alpha_i\left[r_{i+1} + \max_{\alpha'} Q_i(s',a')\right] \quad (8)$$

α_i is Step-size sequence, $Q_i(s, a)$ is the i-th estimation of $Q^*(s, a)$.Under the standard RL hypotheses, the sequence Q_i probably converges to Q^* , and the optimal policy is simply selecting the action to maximize $Q_{i+1}(s, a)$ at any state s.

5. Computer Simulations

To test the performance of the proposed approach, some simulation results are presented in this section. We conduct our simulation experiments using an open source agent-based modeling development platform, NetLogo [30]. The pursuit-evasion game is illustrated in 80 × 80 grid of cells provided with some static obstacle in which the pursuers are clustered in several groups to capture the different evaders detected. Noting that, both pursuers and evaders have the same speed and information regarding each other provided by the environment. We use the same initial conditions (environment and agents' position) to compare our algorithm explained in subsection 4.2 with a recent research activity [15] treating the pursuit-evasion problem through the use of an organizational coordination mechanism. Indeed, the concepts Agent, Group and Role presented in AGR organizational model were simultaneously used to allow the coalition of the pursuers as explained in Section 2 of this paper. Knowing that, the pursuers' path planning in AGR case is totally based on MDP principles. Specifically, our case study is based on ten agents able to play the role "Pursuer" allowing them to purchase two mobile evaders.

Figure 4 reflects the average evaders' capturing time achieved after different pursuit episodes. In relation to the pursuit based on AGR, the results obtained via the application of our coalition formation algorithm (YAMAM) decrease until 33.88% in relation to the other case. This fact is due to the optimal coalitions formed in the beginning of the pursuit through the application of the access mechanism.

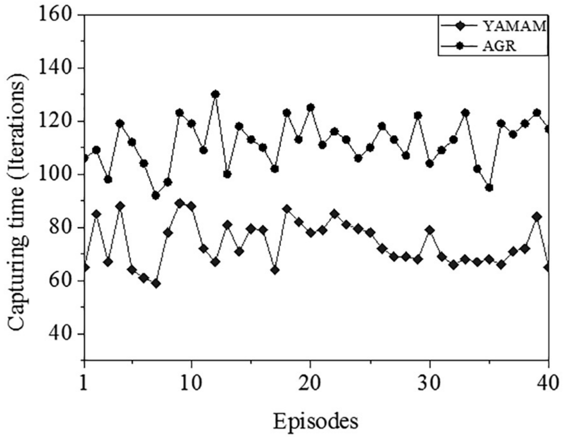

Figure 4. The average capturing time achieved during forty (40) pursuit episodes.

To highlight the internal development of the pursuers during tasks' executions, we have calculated the pursuers' tasks processing degree (Ω) during each pursuit step performance as shown in Figure 5. This skill is calculated in accordance to the Equation 1 as detailed in subsection 3.1. The results obtained through the proposed algorithm in this paper reveal an estimated increase of 15.38% in comparison to the results achieved through AGR model.

To test the effectiveness of the motion planning proposed, we have also calculated the pursuers' rewards development as returned in Figure 6. The immediate application of the access mech-

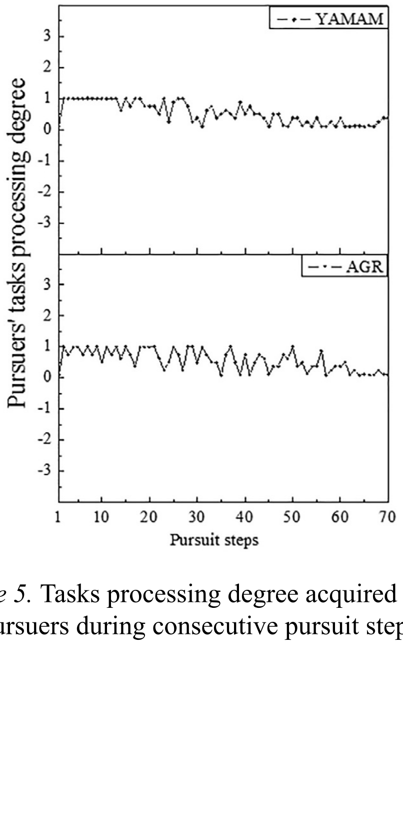

Figure 5. Tasks processing degree acquired by the pursuers during consecutive pursuit steps.

anism provokes an interesting increase in comparison to AGR at the first pursuit step. Moreover, we note certain positivity in the rewards' development in both cases during the pursuit processing, which is mainly caused by the stability of the coalitions formed as well as the greedy strategy provided by Q-Learning in YAMAM case and MDP principles in AGR case. Knowing that, the pursuers' rewards are totally based on the distance (Dist) separating the pursuers from the evader concerned. This distance (Skill) is calculated in accordance to the Equation 2 given in Section 3 of this paper.

Otherwise, we have seen the usefulness to showcase the evaders' utility decrease until the capture is performed in both cases. The results shown in Figure 7 confirm that the access mechanism provided by our algorithm allows optimal execution of the tasks with aim to swiftly reach the requirements through the increase of the pursuers' goal orientation regarding the evaders' capture, when compared with the case based on AGR.

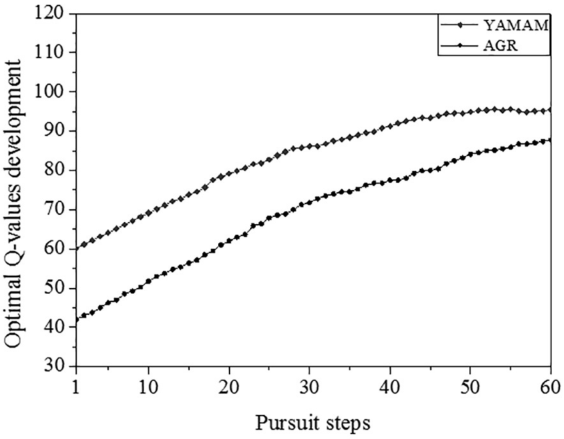

Figure 6. The average learning development of the pursuers during a specific pursuit part.

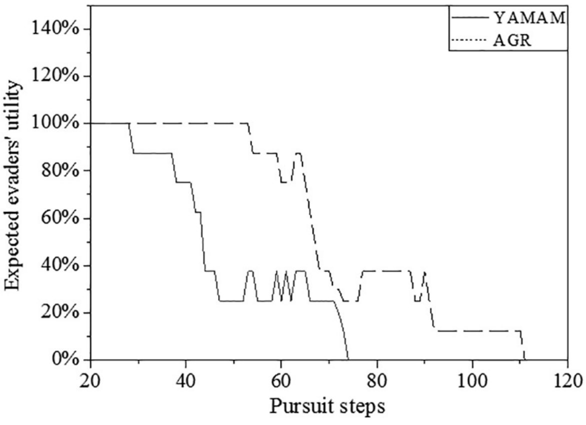

Figure 7. Evaders' utility decrease during a complete pursuit.

During these experiments, we have noted that the path planning methods used in the compared approaches provide approximately the same results regarding the pursuers' trajectories in this case study. By this fact, we can deduce that the main result's improvement is due to the pursuit group access mechanism based on YAMAM.

Table 1 summarizes the main results acquired during the simulation experiments concerning the average evaders' capturing time, the pursuers' internal development as well as the achievement of the evaders' requirements. We can deduce that the method proposed in this paper improves the pursuit processing in comparison to the AGR approach.

Table 1. The main results achieved.

	Average Capturing time (Iterations)	**Average Task processing degree**	**Average expected evaders' utility**
AGR	111.925	0.44	67.09%
YAMAM	74	0.52	65.81%

6. Conclusion

In this paper, we presented a new variant of the classical pursuit-evasion game in a two-dimensional grid of cells environment, which requires multiple pursuers to simultaneously block the movement of the evaders for capture. In order to enable coalition of the pursuers, we proposed a coordination mechanism based on YAMAM organizational model. Different concepts composing this model allow the definition of the groups' access mechanism providing certain optimality during the pursuers' roles attribution. Knowing that in AGR there is no predefined mechanism allowing the agents' access to the groups. Also, we have implemented a Reinforcement Learning method (Q-learning) to control the action selection of the pursuers as well as to make their behaviour "greedy" during the rewards acquisition. Furthermore, we have compared the proposed method with a recent research based on organizational principles and treating the same problem. The results achieved reflect the positive impact imposed through the application of the access mechanism implemented on the pursuers' internal development as well as the evaders' capturing time. To improve these parameters, our future work will focus on the dynamism of the pursuit groups formed. Also, to prove the flexibility of our research activities, the simulation environment will be totally based on graph theory where the agents can move from one node to another.

References

[1] S. D. Bopardikar and S. Suri, "K-Capture in Multi Agent Pursuit Evasion Or the Lion and the Hyenas", *Theoretical Computer Science*, vol 522, pp. 13–23, 2014.
http://dx.doi.org/10.1016/j.tcs.2013.12.001

[2] J. Yan *et al*., "A Cooperative Pursuit-evasion Game in Wireless Sensor and Actor Networks", *Journal of Parallel and Distributed Computing*, vol. 73, pp. 1267–1276, 2013.
http://dx.doi.org/10.1016/j.jpdc.2013.05.009

[3] A. Kehagias *et al*., "A Graph Search Algorithm for Indoor Pursuit-evasion", *Mathematical and Computer Modelling*, vol 50, pp. 1305–1317, 2009.
http://dx.doi.org/10.1016/j.mcm.2009.06.011

[4] F. Amigoni and N. Basilico, "A Game Theoretical Approach to Finding Optimal Strategies for Pursuit Evasion in Grid Environments", *IEEE International Conference on Robotics and Automation (ICRA)*, 2012, River Centre, Saint Paul, Minnesota, USA, pp. 2155–2162.
http://dx.doi.org/10.1109/ICRA.2012.6224924

[5] J. Li *et al*., "A New Approach of Multi-robot Cooperative Pursuit Based on Association Rule Data Mining", *International Journal of Advanced Robotic Systems*, vol. 7, no. 3, pp. 165–172, 2010.
http://dx.doi.org/10.1109/FSKD.2009.403

[6] V. Lesser and L. Erman, "Distributed Interpretation. A Model and an Experiment", *IEEE Transactions on Computers, Special Issue on Distributed Processing*, vol. 29, pp. 1144–1163, 1980.
http://dx.doi.org/10.1109/TC.1980.1675519

[7] V. Lesser and D. Corkill, "Functionally Accurate, Cooperative Distributed Systems", *IEEE Transactions on Systems, Man and Cybernetics*, vol. 11, pp. 81–96, 1981.
http://dx.doi.org/10.1109/TSMC.1981.4308581

[8] E. Durfee, "Scaling Up Agent Coordination Strategies", *Computer*, vol. 34, pp. 39–46, 2001.
http://dx.doi.org/10.1109/2.933502

[9] M. Savall *et al*., "YAMAM – An Organisation Model for the Multiagent Systems", Implementation in the Phoenix platform, *3rd Francophone Conference of Modeling and Simulation* "*Conception, Analyze, Management of Industrial Systems*" *MOSIM*, 2001, Troyes, France.

[10] R. S. Sutton and A. G. Barto, "Reinforcement Learning: An Introduction", The MIT Press, Cambridge, Massachusetts, USA/London, England (1998).
http://dx.doi.org/10.1109/TNN.1998.712192

[11] B. Abdulhai and L. Kattan, "Reinforcement Learning: Introduction to Theory and Potential for Transport Applications", *Can. J. Civ. Eng*., vol. 30, pp. 981–991, 2003.
http://dx.doi.org/10.1139/l03-014

[12] J. Barbosa *et al*., Dynamic Self-organization in Holonic Multi-agent Manufacturing Systems: The ADACOR Evolution", *Computers in Industry*, vol. 66, pp. 99–111, 2015.
http://dx.doi.org/10.1016/j.compind.2014.10.011

[13] V. K. Singh *et al*., "Self-Organizing Agent Coalitions in Distributed Multi-agent Systems", *IEEE International Conference of Computational Intelligence and Communication Networks (CICN)*, pp. 650–655, 2010.
http://dx.doi.org/10.1109/CICN.2010.128

[14] S. Abdallah and V. Lesser, "Organization-based Cooperative Coalition Formation", *IEEE/WIC/ACM International Conference of Intelligent Agent Technology (IAT)*, pp. 162–168, 2004.
http://dx.doi.org/10.1109/IAT.2004.1342939

[15] M. Souidi *et al*., "Coalition Formation Algorithm Based on Organization and Markov Decision Process for Multi-player Pursuit Evasion", *International journal of Multiagent and Grid Systems (MGS)*, vol 11, pp. 1–13, 2015.
http://dx.doi.org/10.3233/MGS-150226

[16] J. Ferber *et al*., "From Agents to Organizations, an Organizational View of Multi-Agent Systems", *Agent Oriented Software Engineering (AOSE) IV, P*., vol. 2935, pp. 214–230, 2004.
http://dx.doi.org/10.1007/978-3-540-24620-6_15

[17] M. Souidi *et al*., "Multi-agent Cooperation Pursuit Based on an Extension of AALAADIN Organisational Model, *Journal of Experimental & Theoretical Artificial Intelligence (JETAI)*, vol. 28, pp. 1075–1088, 2016.
http://dx.doi.org/10.1080/0952813X.2015.1056241

[18] A. Siam *et al*., "Fuzzy Organization of Self-adaptive Agents Based on Software Components", *In-*

ternational Journal of Intelligent Information Technologies (IJIIT), vol. 10, no. 3, pp. 36–56, 2014. http://dx.doi.org/10.4018/ijiit.2014070103

[19] A. Siam *et al.*, "An Approach Based on Software Components and Mobile Agents for Developing Distributed Applications with Verification of Validity Criterion", in *Complex, Intelligent and Software Intensive Systems (CISIS)*, *6th International Conference on IEEE*, 2012, pp. 407–413. http://dx.doi.org/10.1109/CISIS.2012.97

[20] M. Souidi and S. Piao, "A New Decentralized Approach of Multi-agent Cooperative Pursuit Based on the Iterated Elimination of Dominated Strategies Model", *Mathematical Problems in Engineering*, 2016. http://dx.doi.org/10.1155/2016/5192423

[21] W. Hao *et al.*, "Pursuers-coalition Construction Algorithm in Multi-robot Pursuit-evasion Game", *ROBOT*, pp. 142–150, 2013. http://dx.doi.org/10.3724/SP.J.1218.2013.00142

[22] Y. Chen *et al.*, "Mas-based Pursuit-evasion Algorithm Under Unknown Environment", *Proceedings of the 4th International Conference on Machine Learning and Cybernetics*, vol. 1, pp. 265–269, 2005. http://dx.doi.org/10.1109/ICMLC.2005.1526956

[23] A. L. C. Bazzan *et al.*, "Learning in Groups of Traffic Signals", *Eng. Appl. Artif. Intell.*, vol. 23, pp. 560–568, 2010. http://dx.doi.org/10.1016/j.engappai.2009.11.009

[24] T. Watanabe, "A Study on Multi-agent Reinforcement Learning Problem Based on Hierarchical Modular Fuzzy Model", *IEEE international conference of fuzzy systems*, pp. 2041–2045, 2009. http://dx.doi.org/10.1109/FUZZY.2009.5277268

[25] T. Wada *et al.*, "A Study on Hierarchical Modular Reinforcement Learning for Multi-agent Pursuit Problem Based on Relative Coordinate States", *IEEE International Symposium of Computational Intelligence in Robotics and Automation (CIRA)*, pp. 302–308, 2009. http://dx.doi.org/10.1109/CIRA.2009.5423188

[26] W. Kwon *et al.*, "Fast Reinforcement Learning Using Stochastic Shortest Paths for a Mobile Robot", *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 82–87, 2007. http://dx.doi.org/10.1109/IROS.2007.4399040

[27] J. S. Campbell *et al.*, "Multiple-model Q-learning for Stochastic Reinforcement Delays", *IEEE International Conference on Systems, Man and Cybernetics (SMC)*, pp. 1611–1617, 2014. http://dx.doi.org/10.1109/SMC.2014.6974146

[28] M. Souidi *et al.*, "Mobile Agents Path Planning Based on an Extension of Bug-algorithms and Applied to the Pursuit-evasion Game", *Web Intelligence*, vol. 15, 2017. http://dx.doi.org/10.3233/WEB-170369

[29] J. Jian-Guo *et al.*, "Multi-task Coalition Parallel Formation Strategy Based on Reinforcement Learning", *Acta Automatica Sinica*, vol. 33, pp. 349–352, 2008. http://dx.doi.org/10.3724/SP.J.1004.2008.00349

[30] U. Wilensky, "NetLogo", Center for connected learning and computer-based modeling, Northwestern University, Evanston, IL, 1999. http://ccl. northwestern. edu/netlogo

Received: July 2018
Revised: December 2018
Accepted: January 2019

Contact addresses:

Mohammed El Habib Souidi
Department of Computer Science
University of Khenchela
ICOSI Lab
40004 Khenchela
Algeria
e-mail: souidi.mohammed@univ-khenchela.dz

Abderrahim Siam
Department of Computer Science
University of Khenchela
ICOSI Lab
40004 Khenchela
Algeria
e-mail: siamabderrahim@univ-khenchela.dz

Zhaoyi Pei
School of Business Administration in Karvina
Computer Science and Technology
Harbin Institute of Technology
Harbin 150001
China
e-mail: peizhaoyi@stu.hit.edu.cn

Songhao Piao
School of Business Administration in Karvina
Computer Science and Technology
Harbin Institute of Technology
Harbin 150001
China
e-mail: piaosh@stu.hit.edu.cn

Mohammed El Habib Souidi received his BSc degree in Computer Science from the University of Khenchela (Algeria) in 2011. He also received his Master degree in Computer Science from the same university in 2013, and his PhD degree in Computer Science from the Harbin Institute of Technology (China) 2017. He is working as a Lecturer in the Department of Mathematics and Computer Science at the University of Khenchela (Algeria). His research interests include multi-agent task coordination, reinforcement learning, game theory, and path planning.

Abderrahim Siam received his BSc degree in Computer Science from University of Batna (Algeria) in 2002, and MSc degree in Computer Science from the University of Oum El Bouaghi (Algeria) in 2005 and his PhD degree in Computer Science from the University of Constantine (Algeria). He is working as a Lecturer in the Department of Mathematics and Computer Science at the University of Khenchela (Algeria).

Zhaoyi Pei received his BSc degree in Computer Science from Harbin Institute of Technology, Harbin (China), in 2015. He is currently a PhD student in the Computer Science and Technology School at the same institution. His primary research interests include multi-agent systems, mechine learning and robotics.

Piao Songhao is currently a professor and doctor supervisor in the School of Computer Science and Technology at Harbin Institute of Technology. He received his PhD degree from the Harbin Institute of Technology (HIT) in 2004. From 2006 to 2009, he worked as a postdoctoral researcher in national key technology in robot technology and system at the same institution. His main research interests are robot intelligence control, pattern recognition, motion planning, and robot vision.