

Multi-agent pursuit coalition formation based on a limited overlapping of the dynamic groups

Mohammed El Habib Souidi^{a,*}, Abderrahim Siam^a and Zhaoyi Pei^b

^a*Department of Computer Science, University of Khenchela, ICOSI Lab, Khenchela, Algeria*

^b*School of Computer Science and Technology, Harbin Institute of Technology, Harbin, China*

Abstract. Coalition formation algorithms based on organizational modeling frameworks can be considered as the earliest approaches applied to the Pursuit-Evasion problem. In this paper we have based on an extension of AGRMF (Agent-Group-Role-Membership-Function) organizational model with a limited overlapping degree of the pursuit groups to allow the coalition of the pursuers. The limited overlapping provides certain equilibrium between the pursuit groups through the elimination of negative externality problem. Regarding the path planning we have based on Markov Decision Process principles (MDP) resolved via the application of value iteration algorithm. To avoid the obstacles encountered, we have based on the recent Reward Bug Algorithm (RBA). This algorithm is based on the payoff returned by the application of MDP. The simulation results reflect the improvement provided by this new approach in relation to recent methods applied to this kind of problems.

Keywords: Pursuit-evasion, organizational modeling frameworks, overlapping coalition, Markov decision process

1. Introduction

Agent-Based Models (ABMs) seek to clone human behavior in a virtual reality, or artificial environment. In an artificial environment, the agents' decision-makers operate with each other with the aim of producing possible phenomena understandable by social scientists. This definition may be not sufficiently clear to those who are not experts in Distributed Artificial Intelligence (DAI), but it provides a clear signification if we take into consideration that all of us are acquainted with video games, which reflect a specific type of ABM. As a matter of fact, video games integrate players in a virtual world in which

some characters (the agents) interact: monsters, pursuers, evaders, astronauts, soldiers, and so on. Each of these agents undertakes a specific protocol of behavior (a behavior algorithm) in the case of interaction with the encountered agents. However, there exist so many possibilities that random encounters can readily transform the expected results of the game.

There exist two definition cases of multi-agent organizational structure. On one hand, it can be implicit and integrated in the structure of individual agents. On the other hand, it can be explicit and defined at the system level. Deciding among one and the other depends on the specifications of the application domain [1]. The formation of organizations can be effectuated in different ways. In organization theory, an organization could be spontaneous; emerging via interactions of the individual decision makers, or it could be a predefined structure in which the

*Corresponding author. Mohammed El Habib Souidi, Department of Computer Science, University of Khenchela, ICOSI Lab, 40000 Khenchela, Algeria. E-mail: souidi.mohammed@univ-khenchela.dz.

members should try to fit themselves. In multi-agent systems these two tendencies also exist. In certain cases, the organization's structure is depicted as a set of rules answering to the questions what, whom, and how to organize the communication. In other cases the interest focuses on an emergent formation of the organization, in which its objective is to have the organizational structure integrated into the agents.

The organizational modeling frameworks can realize the application of organizational constraints explicitly into a multi-agent system. There exist several frameworks in the literature for modeling organizations. Some examples of these frameworks are: AGR (Agent, Group, Role) [2], YAMAM (Yet Another Multi-Agent Model) [3], AGRMF (Agent-Group- Role- Membership- Function) [4, 9], GAIA [5], and MESSAGE [6]. These frameworks vary in different points like internal or external representation of the organization, existence of a distinct or distributed management, and normative angles of the organization [7].

In this paper, we propose an organizational architecture allowing dynamic and optimal overlapping coalitions of the multi-agent system. This new organizational structure known as OAGRMF (Overlapping-Agent-Group- Role- Membership- Function) can be considered as a new improvement of AGR organizational model via the implementation of new concepts. The proposed architecture is based on four different concepts: Agent, the Overlapped Groups, the Role, and the Membership function. In relation to the coalition based on AGR [8], the proposed work introduces the membership function concept representing the groups' access mechanism. Knowing that in AGR, there is no mechanism permitting the access to the groups. Moreover, this proposal is based on dynamic groups in relation to the compared work.

Comparing with the coalition based on AGRMF [9], this new approach allows the overlapping of the groups (each agent can belong to more than one group) to avoid the problem linked to the negative externalities. The negative externality is related to the agents already belonging to a specific group but their impact can be more important in other groups. Furthermore, the overlapping makes this approach applicable in the case where the number of the existing agents is less than the requirement.

Markov chains are used to model the stochastic systems but do not allow an agent to intervene and act on the evolution of the system. MDPs (Markov Decision Processes) are extended from Markov chains to model the dynamics of a system under the control of

an agent through actions and attributing rewards to the transitions between the states. In other words, MDP furnishes a formalism of model and solves planning and learning problems under uncertainty. Assume that we have on the one hand, a system changing over time as a probabilistic automaton and on the other hand, a kind of external controller examining the current state of the system. This controller can use a number of actions to build an optimal plan maximizing the reward that it can get. The evolution of the system is seen as a Markov process, i.e. the evolution of a temporal sequence of distinct States according to probabilities of transition, only based on the phenomena of the previous state. Puterman [10] describes these models located at the confluence of decision and probabilities theories. MDPs allow taking sequential decisions under uncertainty.

MDP is composed of two distinct elements: Markov process and a controller element. At each time step, the controller of the system observes the process and has the ability to act upon it, by making an action possibly nil. A reasoning based on MDP can lead to the following speech: "I know that I'm in such situation and if I make such action there are so many chances to find myself in this new situation by obtaining such reward". In relation to the work proposed in this paper, we introduce a multi-agent path planning allowing the motion of the agents as well as the obstacles avoidance based on MDP principles.

Pursuit-Evasion Game (PEG) is a popular multi-agent problem related to the optimization of the path and task planning of the agents. Based on uniform criteria between pursuers and evaders, the pursuit-evasion problem can be categorized into single and multi-object pursuits. The principal goal of Multi-Agent pursuit-evasion games is to allow the cooperation of a group of pursuers with each other with the aim of blocking the movement of another group of evaders. Furthermore, this problem has been taken into consideration in many interesting works and its application was extended from a various set of techniques and useful approaches such like: Graph theory [11], Polygonal environment [12, 13], Data mining [14], and Reinforcement learning [15]. In this type of problem, the participating agents are presented in different kinds. On the one hand, the agent pursuer reflects its pursuit potential (capacities). On the other hand, the agent evader returns the number and type of pursuers needed to capture the evader concerned. The value of an evader returns the expected rewards should be obtained by the pursuers implicated in the pursuit in the case of the evader capture.

OAGRMF organizational model can be considered as a type of task coordination mechanisms. As known, the task coordination mechanisms could be applied to several problems such as: Pursuit-evasion [16], Pick-up delivery [17], task allocation and scheduling [18], air traffic control [19] as well as supply chain management [20]. In this work, we have applied our mechanism to the pursuit-evasion problem due to the emergence of the organizational phenomena through this kind of game. Moreover, the concepts proposed in OAGRMF are very close to concepts of PEG. For example, the groups in OAGRMF are represented by the pursuit groups in PEG, the roles in OAGRMF are represented by the roles pursuers and evaders in PEG. In other words, in order to integrate a pursuit group, the concerned agent must be able to play the role proposed by the concerned group.

Regarding the PEG, the main objectives of this proposal is to improve the pursuers capturing time as well as the internal development of the pursuers during the pursuit achievement in relation to the previous approaches proposed. Also, we seek to decrease the number of pursuers used to perform the pursuits.

This paper is organized as follows: in section 2, we discuss the main works related to the proposition introduced in this paper. In section 3, we describe the environment and the different component of the pursuit. In section 4, we detail the pursuit algorithm based on the Overlapping AGRMF from the detection of the evaders to their capture. Moreover, we explain how MDP principles are used in order to provide a path planning to pursuers. In section 5, we present our simulation part, in which we expose a comparison study between the algorithm proposed in this paper and relevant related works. Finally in section 6, we focus on concluding remarks regarding the approach proposed.

2. Related work

In the recent research activities, two approaches based on Multi-agent organizational modeling frameworks are considered as the first works applied to the pursuit-evasion problem. In the first approach [8], the authors have proposed a pursuit coalition formation algorithm based on AGR organizational model in order to evaluate the impact of the agents grouping in non-overlapping groups on the pursuit processing.

In the second approach, they proposed a new organizational modeling framework known as AGRMF [9] extended from AGR through the application of

fuzzy logic principles. The groups in this model are equipped with a membership function calculating the membership degree of each agent in relation to each group. Moreover, they have extracted a dynamic pursuit coalition formation algorithm from this model, in which the membership function dynamically determines the access to the role pursuer proposed by the pursuit group concerned.

In [21], the authors introduced an interesting literature analysis regarding the different Multi-agent organizational modeling frameworks used. They evaluated the qualities of the Agent-Oriented software engineering methodologies, focalizing on their relations with industrial practices. Comparing with GAIA [5], OAGRMF is based on a meta-model allowing the description of the different concepts forming the model as well as the relations between them. Moreover, the new extension of AGR proposed in this paper is based on a support tool known as Madkit [2]. In relation to MESSAGE [6], the both are based on support tools and meta-models. The major difference is the concept Group proposed in OAGRMF meta-model which simplifies the tasks repartition between the formed groups of agents. Furthermore, the meta-model of OAGRMF allows the groups' overlapping to decrease the number of agents used to resolve the encountered problems. YAMAM organizational model [3] is also based on a meta-model describing the relations between four different concepts: Agent, Role, Task, and Skills, which can easily model the PE problem. In relation to OAGRMF, the concept Group and the overlapping between the set of groups make this model more adequate to allow the flexible grouping of the agents.

In [22], they have based on a game theoretic method (IEDS) to provide a pursuit task coordination mechanism. This coalition algorithm is based on the iterated elimination of the dominated pursuit groups causing 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.

Regarding the recent contributions in multi-agent path coordination [23], the authors have focused on the processing of the complex obstacles such as U and H encountered during the pursuit of targets. In other words, they proposed an obstacle avoidance process based on Bug-Algorithms and the rewards generated through the application of MDP. This new process is known as Reward Bug Algorithm (RBA). In relation to the precedent Bug-Algorithms, this method

increases the use of the environment's data returned via the sensors equipping the pursuers. Moreover, we demonstrate how RBA improves the goal orientation of the pursuers as well as their decision regarding the obstacles' leaving points. In the context of Pursuit-Evasion games, MDP is frequently introduced to schedule the path planning of the agents via the optimization of the payoff returned during the pursuit. In [24], a Partially Observable Markov Decision Process (POMDP) algorithm is implemented to localize the mobile target in a known graph. The major goal is to minimize the time regarding the capture of the targets through the clearing of the graph. In [25], the authors presented a new approach known as Continuous Time Markov decision process (CTMDP) to process the PEG. In relation to MDP, CTMDP takes into consideration the effect of the transition time between the states implying a strong robustness against the variations in transition probability. In relation to the path planning approaches proposed in [26], the motion strategy used in this paper is based on the value iteration algorithm detailed in the section 4.2. The point making the difference is the implantation of the new RBA [23] allowing the avoidance of the encountered obstacles. RBA is based on the values generated by the reward function. This fact makes RBA one of the most appropriate obstacle avoidance algorithm can be implemented in stochastic domains.

In order to tackle the lack of existing research on coalition formation, in which an agent can take part in several groups, the authors proposed [27] a system known as Dynamic Multi-agent Overlapping Coalition (DynaMOC). In this system, each agent can be defined as dynamic driven data application which receives and evaluates data from physical system in real time. The system applies a mechanism to reduce the computational and communication efforts based on dynamic overlapping coalitions of agents. Each agent is capable of organizing coalitions and calculating their own schedules of tasks according to the knowledge obtained within the coalitions concerned. In the same context [28], a particle swarm algorithm with dynamic weight value based on the similarity calculation is introduced to resolve overlapping coalition formation of serial tasks. This algorithm allows an Agent to take part in several different coalitions, and also reduces the waste of agent resource in certain extent. Lin e Hu [29] introduced the binary Particle Swarm Optimization (PSO) to process the coalition formation problem, taking into consideration the overlapping coalition and parallelizability simultaneously. In fact, they developed an algorithm

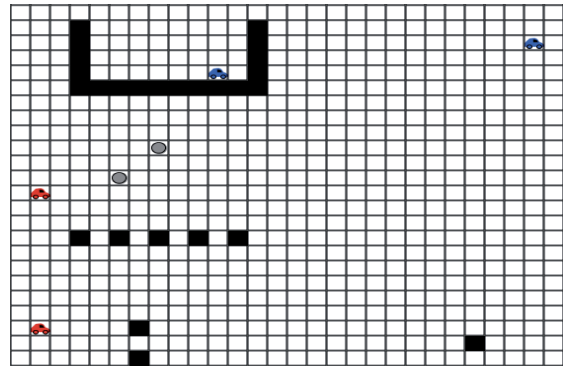


Fig. 1. A part of the PEG environment. Black cells: cells containing obstacles. White cells: free cells. Circle agent: Evaders. Car agent: Pursuers.

to parallelize the process of overlapping coalition formation, and also to assure the coordination of tasks at the same time. Kraus et al. [30] took into account the probability of incomplete information. Their work proposed protocols regarding the formation of coalitions based on the payoff sharing between agents in cases where groups of self-interested agents can only execute tasks cooperatively. Specifically, they consider situations where a complex task is composed of sub-tasks, and each sub-task must be executed by a different agent.

3. PEG description

In this paper, the PEG will be processed in an environment decomposed into a grid of rectangular cells. There exist two types of agents, the Evader E which moves randomly in the environment until its capture (that's mean all the possible next states or cells are occupied). The second type concerns the Pursuer P, which must be integrated into a group or groups according to a task planning method and equipped with a path planning method to effectuate the pursuit. The existing Evaders are differentiated by the number of Pursuers required to accomplish the chase of each one. Moreover, there exist some complex static obstacles characterized by different shapes and sizes situated randomly in the environment. Knowing that the environment cells have the same size as shown in Fig. 1, this representation seems to be perfect to clarify the implementation of MDP principles.

MDP is based on the following components, namely states, actions, transitions, rewards, policy, and value function:

- States (S): A state represents how the system currently exists. Ideally, it is an abstract and compact representation that only considers the necessary elements for the resolution of the problem, everything else being overlooked in modeling. For example, it may be the position of an agent in its environment. If we imagine all the possible values' combinations of the modeled elements, then we obtain a set of states called State space.
- Actions (A): An action is an available means that the effect can change the state of the process. A problem that we will discuss will be about the action that the Pursuer must choose in a particular State of the environment.

Rewards (R): MDPs aim to automate the decision making process, also, we must be able to have a measure of the usefulness for possible actions. This is why we specify the earned values corresponding to the choice of such action in such State. A reward, so-called immediate, is levied for each transition, but it can be nil if the state concerned contains an obstacle. Therefore, we can try to act so as to maximize the total reward in the short, medium or long term. In this paper the reward of each cell is inversely proportional to the distance between the cell concerned and the cell containing the target chased. The distance between the Pursuer and the Evader purchased in the environment implies a considerable impact on the pursuit processing. It is calculated via the use of the Cartesian coordinates of the agents concerned as follows:

$$\text{Dist}(s, s') = \sqrt{(Co_{s_x} - Co_{s'_x})^2 + (Co_{s_y} - Co_{s'_y})^2} \quad (1)$$

$$R(s) = \eta - \text{Dist}(s, s') \quad (2)$$

(Co_x, Co_y) : Cartesian coordinates of cell containing the agent in the environment.

η : the maximum reward or the reward of the cell containing the target.

s : the state containing the pursuer.

s' : the state containing the evader.

- Transitions (T): The transitions describe the evolution of system state over time and in particular under the effect of an action. MDPs are powerful tools because they take into account the fact that the effects of actions can be probabilistic and therefore they are not guaranteed.
- Policy: The agent's behaviour is guided by a policy π . A policy is defined as a function from the

states space to the actions space:

$$\forall s \in S \pi : S \rightarrow A \quad (3)$$

This function signifies that every policy associates to each state s an action a . Solving MDP is effectuated via the calculation of a policy indicating for each state of the system the best action to perform and maximizing a certain optimality criterion. This solution is known as the Optimal Policy π^* . Bellman [31] demonstrated that each MDP possesses at least a deterministic and stationary optimal policy. Therefore, the search regarding the policy can be achieved in the space of deterministic policies.

- Value function (V): In order to maximize the gains, the agent must choose among several policies according to the value of each one. The value function of policy π , associates to each state s the value of the performance criterion (considered), following π from s :

$$\forall \pi \in \Pi, V^\pi : S \rightarrow R \quad (4)$$

Taking into account the γ -weighted criterion, the value function of policy π is written as follows:

$$V^\pi(s) = E^\pi \left[\sum_{t=0}^{\infty} \gamma^t r_t \mid s_0 = s \right] \quad (5)$$

Where E^π denotes the mathematical expectation under politics π is followed.

A policy π^* is considered as Optimal when:

$$\forall \pi \in \Pi \forall s \in S V^{\pi^*}(s) > V^\pi(s) \quad (6)$$

4. The proposed solution

In this section, we detail how the pursuit of the different evaders detected is processed. On one hand, we clarify how the coalition of pursuers is effectuated according to the organizational model proposed OAGRMF. On the other hand, we showcase how the pursuers move to the evaders during the pursuit iterations according to MDP principles and the obstacle avoidance method RBA:

4.1. Pursuit coalition based on OAGRMF

Multi-agent coalitions are principally concerned with the evaluation of optimal grouping. The objective of an optimal coalition structure is to partition the teams into subgroups with or without intersection and equitably assign the tasks to those subgroups. Early

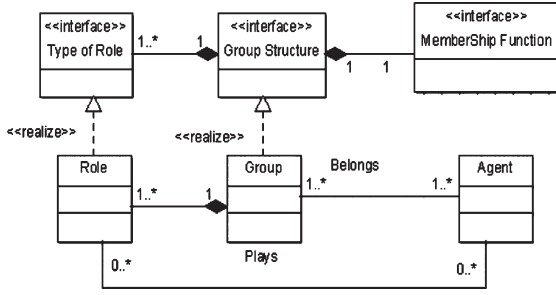


Fig. 2. OAGRMF meta-model.

research activities regarding the overlapping coalitions in multi-agent systems have been only recently introduced. Some of these algorithms use the formation of coalitions for joint achievement of tasks in the case where such tasks require different skills to be improved.

In this paper, we propose a new organizational modeling framework known as Overlapping Agent Group Role Membership Function (OAGRMF). The goal of this model is to optimize the multi-agent tasks execution as well as the agents' development during this process. In order to get a Role, the Agent must belong to the Group proposing the Role concerned with a specific Membership degree. In comparison with AGRMF, OAGRMF allows the overlapping of the existing groups:

$$\forall gr_1, gr_2 \in GR: \begin{cases} \text{if } gr_1 \cap gr_2 = \emptyset \text{ then AGRMF} \\ \text{else} & \text{OAGRMF} \end{cases} \quad (7)$$

gr_1, gr_2 : groups composed of agents.

GR: the set of the existing groups.

Figure 2 reflects the relationships between the different concepts forming the organizational proposed model. The concepts Group and Role implement the interfaces GroupStructure and Type of role which contain the main parameters allowing the structuration of the group and the role respectively. The structure of the group is composed of one membership function and can contain more than one type of role. If an agent wants to integrate a group, it must play one of the roles of the concerned group. In order to be authorized to play a role in a group, the agent must belong to the group with a specific membership degree. Moreover, each agent can play more than one role in the same group or in different groups which means that an agent can belong to more than one group allowing the overlapping of organizational architecture proposed.

In this section, we introduce our pursuit coalition formation based on a limited overlapping of the pursuit groups. This algorithm details the process of the pursuit-evasion game from the detection to the capture of the evaders:

Algorithm name: Pursuit coalition algorithm

Inputs: Agents position

Outputs: Evaders capture

$x \leftarrow$ number of evaders detected;

$y \leftarrow$ number of pursuers

Create x Groups;

Broadcast (offer);

For each-evader **do**

While ($\text{Group}_k \neq \text{Complete}$) **do**

repeat

$T_1[i] \leftarrow \mu(\text{Group}, A_i)$;

$T_2[i] \leftarrow A_i$;

$i \leftarrow i + 1$;

Until $I = y$

repeat

$\alpha \leftarrow \text{index}(\max(T_1))$;

Agent-Selected $\leftarrow T_2[\alpha]$;

If overlapping-degree () < 50% **then**

Role-attribution(Agent-Selected);

Delete ($\max(T_1)$);

Delete ($T_2[\alpha]$);

$j \leftarrow j + 1$;

nbr-role (Agent-Selected)++;

If nbr-role (Agent-Selected) > 1 **then**
overlapping-degree ()++;

endif;

Until $j = Re_k$

$k \leftarrow k + 1$;

endFor;

While (Time > 0) **do**

If obstacle() = true **then**

RBA();

Else

Move-to-target();

endif;

If (Capture() = true) **then**

Update (Rewards);

End while

If (Time = 0 and Capture() = false) **then**

Update (Fines);

End.

After the detection of the existing evaders in the environment, a pursuit group concerning each evader will be created. As explained in [9], each pursuit group is equipped with a membership function $\mu()$ determining the membership degree of each agent in relation to the group concerned according to the following equation:

$$\begin{aligned} \mu_{\text{group}}(P) &= \frac{(\text{Coef}_1 \times R + \text{Coef}_2 \times \text{Tying} + \text{Coef}_3 \times \text{Credit})}{\sum_{i=1}^3 \text{Coef}_i} \end{aligned} \quad (8)$$

This function is based on 3 parameters representing the pursuer skills (R, Tying, and Credit). R is calculated Equation (2) in relation to the distance between the pursuer and the evader concerned as detailed in Section 3. The coalition of the pursuers is considered as the first step of the pursuit. This grouping dissolves when the time spent on the pursuit is over. When the same group of pursuers is affected to an identical task in the future, it is necessary to repeat the process again and reconstitute a new coalition. Consequently, the factor of the task acquaintance (Tying) is used in order to decrease the communication rate as well as to avoid repeated information used in interactions. Tying, denoted by $Tying_{ij}$ is the degree of experience for a group of agents_i concerning a specific task T_j. This dynamic parameter will be updated when the task concerned is performed as follows:

$$Tying'_{ij} = \begin{cases} \min(Tying_{ij} + \lambda_1, 1), & Tying_{ij} \neq 0 \\ \lambda_2, & Tying_{ij} = 0 \end{cases} \quad (9)$$

λ_1, λ_2 : The coalition factors.

If a pursuer is not able to execute the task assigned in time then its credit will be decreased. The credit of the pursuer is computed as follows:

$$\forall \text{Credit} \in (0, 1) : \text{Credit} = \min \left(1, 1 - \frac{\gamma_b}{\gamma_t - \gamma_s} \right) \quad (10)$$

γ_s : the number of tasks that the agent has finished with success.

γ_t : the number of tasks in which the agent has participated.

γ_b : the number of the abandoned tasks by the agent.

In this paper we propose a new approach based on our previous work. In fact, we allow the overlapping of the pursuit groups with a specific degree. In other words, each agent can play the role Pursuer in more than one pursuit group. The different steps of the pursuit coalition algorithm are discussed as follows:

During the pursuit iterations, the membership degree of each agent in relation to each pursuit group is updated. The role Pursuer will be attributed to the agent after the verification of these two conditions:

1. If the agent belongs to the group concerned with highest_{Re} membership degree. Knowing that Re represents the number of pursuers required to capture the evader.
2. The second condition concerns the overlapping degree of the groups retuning how many pursuers belong to more than one pursuit group. In fact, if this degree exceeds 50% and the

agent concerned already belong to one or more other pursuit groups then the role will not be attributed.

After the attribution of the role Pursuer, the roles' number of this agent will be incremented if it belongs to more than one pursuit group.

After the coalition formation of the pursuit groups, each pursuer will undertake the stochastic path planning method based on MDP detailed in 4.2 with the aim of stopping the movement of the evader concerned.

If the evader is captured before the end of pursuit time then the pursuers concerned will be awarded. In the other case, the guilty pursuers will pay some fines to the pursuers belonging to the same group.

4.2. Path planning resolution based on value-iteration

In this section, we expose how the pursuers plan their paths in the grid of cells environment. Our path planning method is totally based on MDP principles. In fact, the pursuer can move in 4 different directions (Up, Down, Left, Right) according the rewards perceived in the environment. In order to make the pursuer able to choose the direction returning the best payoff we have based on the Value-Iteration algorithm.

Iterating over the values algorithm is the most commonly used dynamic programming algorithm because of its simplicity. It consists on gradually improve the value function arbitrary initialized. According to the optimality principles of Bellman, each algorithm's iteration is an update of the value function applied to the set of States. Knowing that Bellman equation is a contraction, it has been shown [30] that V_t asymptotically converges to the optimal value function V^* (regardless of the initialization of V_0):

$$V^\pi(s) = R(s, \pi(s), s') + \gamma \cdot \sum_{s'} T(s, (\pi(s), s')) \cdot V(s') \quad (11)$$

Different methods can be used to stop the algorithm. The most common is to stop the algorithm when the distance between two successive values falls below a certain threshold ϵ :

$$\|V_t - V_{t-1}\| < \epsilon \quad (12)$$

Therefore, a warranty on the remaining distance in relation to the optimal value function V^* is

established:

$$\|V_t - V^*\| < \frac{2\gamma}{1 - \gamma} \in \quad (13)$$

From this iterative calculation and the stopping criterion explained, we can deduce the value-iteration algorithm:

Algorithm name: Path planning algorithm

Inputs: MDP (S, A, P, R, T), $\gamma \in [0, 1[$, $\epsilon \geq 0$
Outputs: π^*

```

1 If obstacle = false then
2 For s  $\in$  S do
3  $V_0(s) \leftarrow 0$ ;
4 t  $\leftarrow 0$ ;
5 Repeat
6 For s  $\in$  S do
7  $V_{t+1}(s) \leftarrow \max_{a \in A} [R(s, a) + \gamma \times (T(s, a, s_u) \times V_t(s_u), T(s, a, s_d) \times V_t(s_d), T(s, a, s_l) \times V_t(s_l), T(s, a, s_r) \times V_t(s_r))]$ ;
8 t  $\leftarrow t + 1$ ;
9 Until  $\|V_{t+1} - V_t\| < \epsilon$ ;
10 For s  $\in$  S do
11  $\pi^*(s) \leftarrow \arg \max_{a \in A} [R(s, a) + \gamma \times (T(s, a, s_u) \times V_t(s_u), T(s, a, s_d) \times V_t(s_d), T(s, a, s_l) \times V_t(s_l), T(s, a, s_r) \times V_t(s_r))]$ ;
12 Else
13 Index  $\leftarrow$  false;
14 While (Index == false) do
15 Follow the obstacle edge;
16 For (i = 1; i  $\leq$  4; i++) do
17 R (Sensori, a);
18 If (Position (Sensori)  $\neq$  obstacle adjacent and R(Sensori, a) = max) then
19 index  $\leftarrow$  true;
20 End-if
21 End while
22 End-if

```

In order to converge to an optimal policy π^* , the algorithm requires the utilization of different resources. Thus, we focus on the study of the temporal and the spatial algorithm complexities. The temporal complexity of Value Iteration depends essentially on two factors: the complexity of the iteration and the necessary number of iterations to converge to an optimal solution:

- Complexity of an iteration: the operation with the highest consumption is the calculation of a transition between the states, where the complexity is $O(|S|^2|A|)$.
- Number of iterations: this number varies according to the size of the problem, but remains polynomial in relation to S, A, γ , and B, where B is the number of bits needed for the representation of the functions P and R.

The space complexity is linear in accordance to $|S| |A|$ (the memory space of states and actions as well as $V_t(s)$ during the phase of planning must be stored).

In the case of obstacle detection, the pursuers follow the principles of one of the earliest obstacle avoidance algorithm known as Reward Bug-Algorithm (RBA) [23]. This extension of Bug-Algorithms [32] is based on the following of the obstacle edge. During this operation, the sensors of the pursuer return the average rewards detected in the different possible directions. The obstacle leaving point is when the average reward is lowest in the direction adjacent to the obstacle.

5. Simulation results

In order to prove the feasibility of the approach proposed in this paper, we propose in this section a comparison study of OAGRMF algorithm with relevant technics treating the pursuit evasion problem. The pursuit-evasion problem is processed in a limited rectangular grid of 100×100 cells environment. Moreover, there exist obstacles of different size and shape implemented to complex the movement of the agents. These obstacles are avoided by the pursuers using the recent RBA as explained in section 4 of this paper. This case studied is based on 10 Pursuers and 2 Evaders situated randomly in the environment. Knowing that each Evader needs a coalition formed by 4 Pursuers to be captured.

During this work, we have seen the usefulness to use an open source agent-based modeling development platform known as NetLogo with the aim of describing the pursuit-evasion environment as well as the pursuit process from the agents' coalitions to the capture of the evaders detected. In fact, this platform offers the opportunity to create different types of agents which could be distinguished by several features such as the mobility degree defining the speed of the movement and the available paths to follow, the shape and the color defining the role attributed during the pursuit process. Moreover, the agents' environment proposed in NetLogo more known as (the Patches) corresponds perfectly to our grid of cells pursuit-evasion environment. In other words, each patch characterized by Cartesian coordinates is considered as a cell which can be occupied by an agent or an obstacle. The colors of the Patches are modifiable allowing us the definition of the free cells, the obstacles, as well as the environment edges.

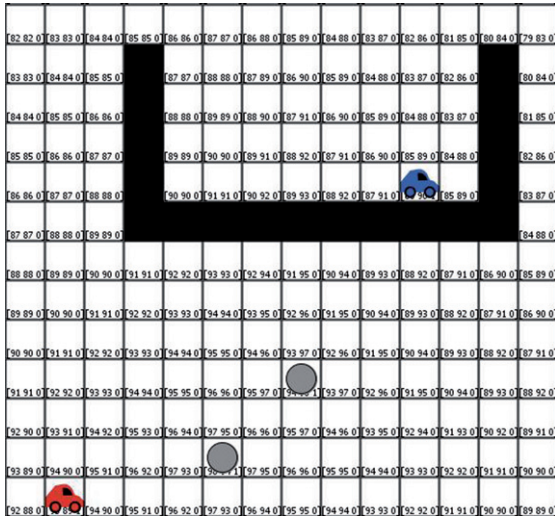


Fig. 3. Pursuit rewards distribution.

In addition to the part of the environment shown in Figs. 1, 3 details how the rewards are distributed in the environment in order to allow the pursuers path planning as explained in section 4.2. We can note that each cell of the environment contains a vector $[x, y, z]$:

- x: the reward could be obtained by a pursuer in the case of purchasing the first evader.
- y: the reward could be obtained by a pursuer in the case of purchasing the second evader.
- z: the obstacle index. This variable can receive two different values, 1 if the cell contains an obstacle and 0 in the other case.

Knowing that the resolution of PE problem via the application of MAS organizational modeling frameworks is a recent research field [8, 9], we have seen the usefulness to compare this work with a relevant approach [9] also extended from AGR [2].

As explained in section 2, there exist several types of MAS tasks coordination mechanisms which can be applied to the PE problem. In order to vary these types, we also compare our work to a PE algorithm based on an auction mechanism [16]. In relation to OAGRMF, this approach is also based on an efficient coalition access mechanism.

The three cases compared in this section can be detailed as follows:

- OAGRMF case: pursuit-evasion based on the approach proposed in this paper. Knowing that, OAGRMF algorithm is well detailed in section 4 of this paper.

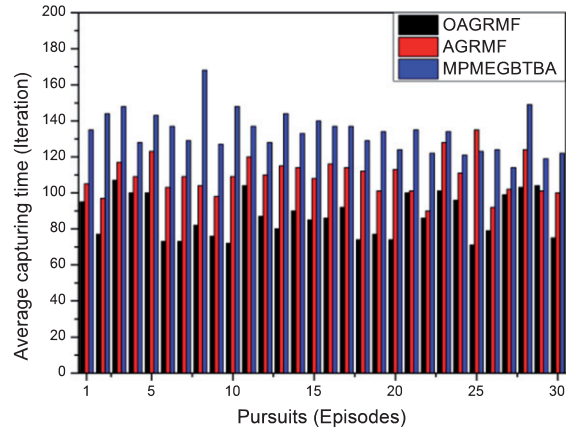


Fig. 4. Average capturing time obtained via the use of different number of pursuers.

- AGRMF case: Pursuit-evasion based on AGRMF organizational model proposed recently [9]. In comparison to OAGRMF, AGRMF is based on non-overlapping pursuit groups.
- MPMEGBTBA case: Pursuit-evasion based on an economical auction mechanism (MPMEGBTBA) [16]. In this approach, an advanced task negotiation process based on Task Bundle Auction was introduced with the aim of allocating the task dynamically via the dynamic grouping of multiple agents.

Figures 4 and 6 represent the average evaders capturing time obtained after 30 pursuits. The capturing time is calculated in relation to the average number of iterations effectuated by the pursuers from the beginning to the end of the pursuit. In other words, each time unit refers to one pursuit iteration according to the literature review [8, 9, 16, 22, 23]. The pursuit iteration can be defined as a pursuer state change or the motion of an agent from a cell to another adjacent cell (up, down, left, right). Knowing that, all the agents used have the same speed (one state change in each time unit).

The results shown in Fig. 4 are not based on the same number of pursuers used in all the cases compared. In fact, the number of pursuers used in OAGRMF decreases in relation to the increase of the limited overlapping degree determined. In our case study, the maximum overlapping degree is 50%. That's to say no more than two pursuers can belong to more than one pursuit group. The average overlapping degree obtained in the pursuits taken into

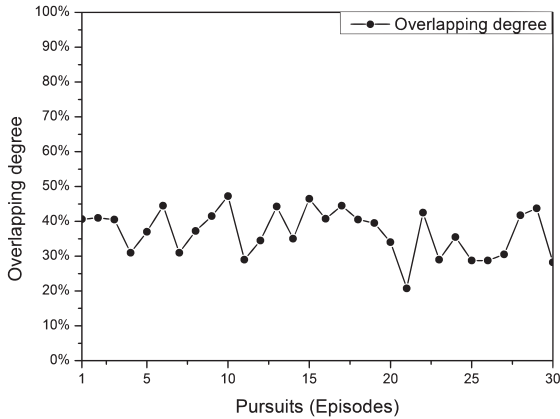


Fig. 5. Overlapping degree used during 30 pursuit episodes.

account in the results reflected is 36.87% as shown in Fig. 5. Then, we can conclude that the number of the pursuers used to effectuate the capture in OAGRMF case is 6.525 pursuers. However, the number of pursuers used in AGRMF and MPMEGBTBA is 8 pursuers.

After the application of the Friedman test on the results shown in Fig. 4, we obtained p -value = 6.6877191086071E-13. The result obtained is less than the significance level 0.05, which proves that there exists a significant difference between the 3 compared cases.

In addition, we note an interesting decrease in the evaders capturing time in OAGRMF case (87.419 pursuit iterations) in in comparison with AGRMF and MPMEGBTBA, in which the averages were 109.064 and 133.718 pursuit iterations respectively as shown in Fig. 4.

Also, we have studied the capturing time using the same number of pursuers in the three cases as shown in Fig. 6. In fact, we have decreased the number of pursuers in AGRMF and MPMEGBTBA in accordance to the number of pursuers used in OAGRMF case. The average number of pursuers used in the pursuit episodes is 6.525. The results show an important increase of 52.95% and 87.20% in AGRMF and MPMEGBTBA in relation to OAGRMF case. This improvement can be explained by the elimination of the negative externalities between the pursuit groups by allowing the multi attribution of the tasks to the individual agent as well as the dynamism of the pursuit group during the capture. The negative externality concerns the groups which are not integrated by the best pursuers because of their belonging to other groups. In other words, the negative externality

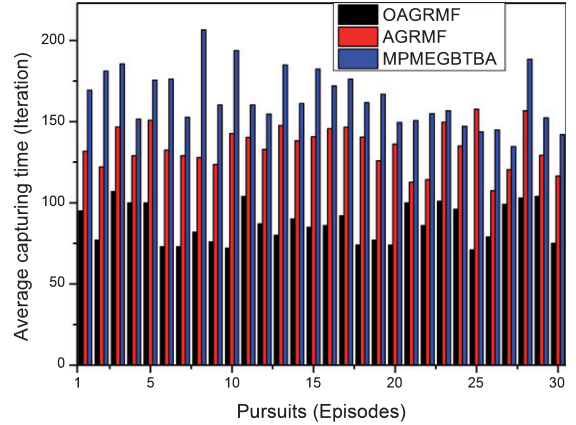


Fig. 6. Average capturing time obtained via the use of the same number of pursuers.

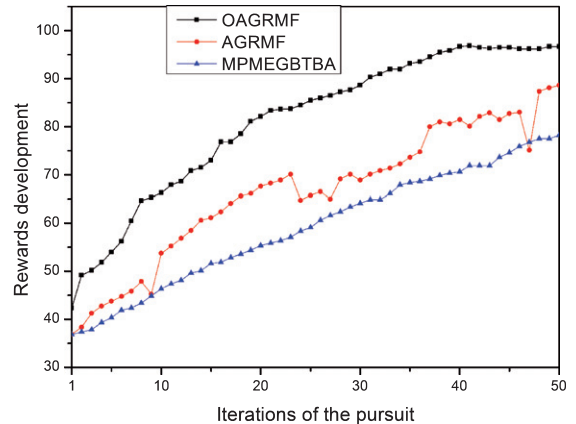


Fig. 7. Pursuers rewards development during a specific part of pursuit.

concerns the case when an agent is not belonging to the group in which this agent is the most efficient.

After the application of the Friedman test on the results shown in Fig. 6, we obtained p -value = 2.4602743683859E-13. The result obtained is less than the significance level 0.05, which proves that there exists a significant difference between the 3 compared cases.

Figure 7 represents the pursuers reward development in three cases studied during a specific part of the pursuit. We note an increase in the first iteration in favour to OAGRMF case due to the equilibrium provided by our algorithm regarding the construction of the pursuit groups. In order to explain this fact, we propose an example in which 8 pursuers are proposed to integrate 2 pursuit groups:

Table 1
Roles' attribution (AGRMF vs OAGRMF)

	P ₁	P ₂	P ₃	P ₄	P ₅	P ₆	P ₇	P ₈
GR ₁	23%	50%	45%	80%	61%	20%	30%	55%
GR ₂	28%	44%	54%	80%	62%	23%	17%	90%

Grouping based on AGRMF: GR₁: (P₄, P₅, P₈, P₂) = 61.5%. GR₂: (P₃, P₁, P₆, P₇) = 30.5%. Grouping based on OAGRMF: GR₁: (P₄, P₅, P₈, P₂) = 61.5%. GR₂: (P₈, P₄, P₃, P₁) = 63%.

The values shown in Table 1 are supposed to be the membership degree of each pursuer in relation to each group. This example perfectly exposes the problem of negative externality. In other words, GR₂ in AGRMF case is clearly weakened because the best pursuers P₈ and P₄ already belong to GR₁. This problem is resolved in OAGRMF knowing that this algorithm allows these two pursuers to belong to two groups in the same time. This fact improves the average membership degree of the agents in OAGRMF (62.25%) in comparison with AGRMF (46%).

Figure 8 showcases the average reward obtained by the pursuers during the pursuit iterations regarding the 3 cases studied. In MPPEGBTBA, the average reward obtained was 46.58%. This average increase

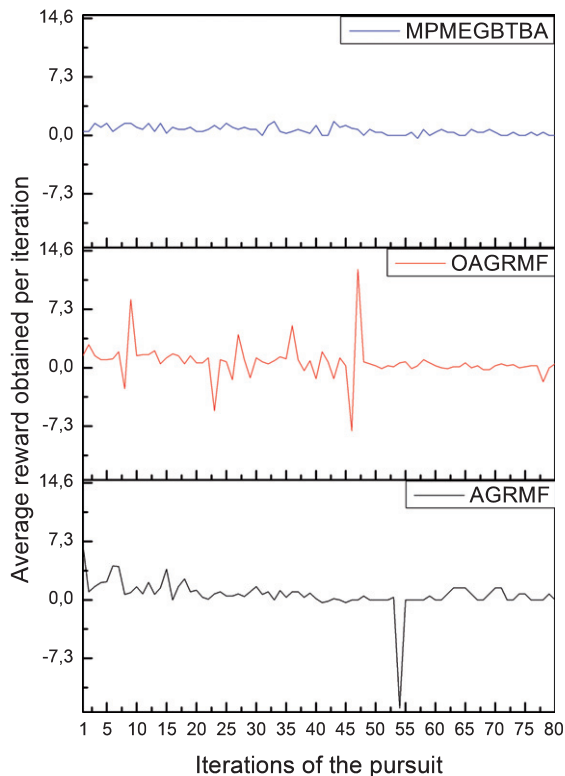


Fig. 8. Rewards obtained during the pursuit iterations.

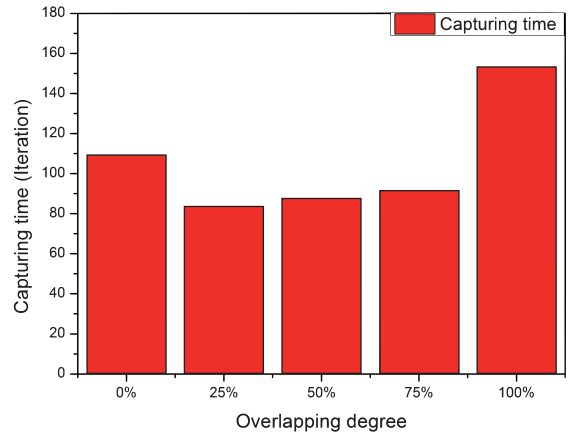


Fig. 9. Capturing time calculation in relation to the overlapping degree.

Table 2
Main results obtained

	Capturing time (with the same number of pursuers)	Capturing time (without the same number of pursuers)	Reward achieved per pursuit iteration
MPMEGBTBA	133.48	163.65	46.58%
AGRMF	109.06	133.71	57.81%
OAGRMF	87.41	87.41	64.31%

until 57.81% in AGRMF due to the application of the membership function optimizing the groups' access as well as the dynamic reconstitution of the pursuit groups during each pursuit iteration. However in OAGRMF, the average reward exceeds the precedent cases by the acquiring of 64.31% of the possible reward. This positive result reflects the efficiency returned through the implementation of a limited overlapping principle between the pursuit groups.

Finally, we have studied the capturing time in relation to the overlapping degree of the pursuit groups as demonstrated in Fig. 9. We note that the capturing time decreases when the overlapping degree is limited between 25% and 75% in comparison with the standard case (without overlapping). However, the time increases when the overlapping degree reaches the pick. This fact means that only 4 pursuers are used to capture 2 evaders in our case study.

Table 2 summarizes the exact results obtained during the simulations explained. These results showcase the improvement brought by OAGRMF concerning the evaders capturing time as well as the pursuers' development (reward achievement) during the execution of the pursuit.

6. Conclusion

In this paper, we have proposed a task planning method based on organizational modeling framework to allow an optimal and dynamic task sharing between the multi-agent groups as well as a path planning method based on stochastic principles to allow the displacement of the different agents. Concerning the task coordination, we have based on an extension of AGRMF organizational model by allowing a limited degree of the groups overlapping. Regarding the path planning, we have based on MDP principles resolved by the value-iteration algorithm as explained in section 4. Moreover, we have used a recent obstacle avoidance algorithm known as RBA to dodge the collision of the agents with the different obstacles encountered.

In comparison with AGRMF organizational model, this new approach is more adequate in the case where the number of existing agents is less than the requirement. Moreover, this model eliminates the problem related to the negative externalities emerging from the agents groupings via the overlapping principle.

In order to showcase the improvement brought by this approach, we applied it to the PEG in section 5. Also, we have seen the usefulness to compare this proposal with AGRMF and MPMEGBTBA, which are recent works in this field. The approach proposed in this paper improves the evaders capturing time as well as pursuers' developments during the pursuit. Furthermore, this approach allows the decrease of the pursuers used to effectuate the different pursuits. These facts are due to the limited overlapping of the pursuit coalitions which provides a certain groups balance.

References

- [1] V. Dignum, J. Vazquez-Salceda and F. Dignum, On the Engineering of Multi Agent Organizations, *Proceedings of the 12th International Workshop on Agent-Oriented Software Engineering*, 2011, pp. 53–65.
- [2] J. Ferber, O. Gutknecht and F. Michel, From agents to organizations, an organizational view of multi-agent systems, *Agent Oriented Software Engineering (AOSE) IV* **2935** (2004), 214–230.
- [3] M. Savall, J. Pécuchet, N. Chaignaud and M. Itmi, YAMAM - An organisation model for the multiagent systems, *Implementation in the Phoenix Platform 3rd Francophone Conference of Modeling and Simulation "Conception, Analyse, Management of Industrial Systems" MOSIM*, Troyes - France, 2001.
- [4] A. Siam, R. Maamri and Z. Sahnoun, Fuzzy organization of self-adaptive agents based on software components, *International Journal of Intelligent Information Technologies (IJIT)* **10**(3) (2014), 36–56.
- [5] F. Zambonelli, N.R. Jennings and M. Wooldridge, Multi-agent systems as computational organizations: The Gaia methodology, In *Agent-Oriented Methodologies*, 2005, pp. 136–171. IGI Global.
- [6] F.J. Garijo, J.J. Gomez-Sanz and P. Massonet, The MESSAGE methodology for agent-oriented analysis and design, In *Agent-Oriented Methodologies*, 2005, pp. 203–235. IGI Global.
- [7] D. Isern, D. Sanchez and A. Moreno, Organizational structures supported by agent-oriented methodologies, *Systems and Software* **84** (2011), 169–184.
- [8] M.E.H. Souidi, S. Piao, G. Li and L. Chang, Coalition formation algorithm based on organization and Markov decision process for multi-player pursuit evasion, *International Journal of Multiagent and Grid Systems (MGS)* **11** (2015), 1–13.
- [9] M.E.H. Souidi, S. Piao, G. Li and L. Chang, Multi-agent cooperation pursuit based on an extension of AALAADIN organisational model, *Journal of Experimental & Theoretical Artificial Intelligence (JETAI)* **28** (2016), 1075–1088.
- [10] M.L. Puterman and L. Martin, Markov decision processes: Discrete stochastic dynamic programming, 2014, John Wiley & Sons.
- [11] B. Goode, A. Kurdila and M. Roan, A Graph Theoretical Approach Toward a Switched Feedback Controller for Pursuit-Evasion Scenarios, *Proceedings of the American Control Conference*, 2011, pp. 4804–4809.
- [12] V. Isler, S. Kannan and S. Khannan, Randomized pursuit-evasion in a polygonal environment, *IEEE Transactions on Robotics* **21** (2005), 875–884.
- [13] J. Thunberg, P. Ögren and X. Hu, A Boolean Control Network Approach to Pursuit Evasion Problems in Polygonal Environments, *IEEE International Conference on Robotics and Automation (ICRA)*, 2011, pp. 4506–4511.
- [14] J. Li, Q. Pan and B. Hong, A new approach of multi-robot cooperative pursuit based on association rule data mining, *International Journal of Advanced Robotic Systems* **7** (2010), 165–172.
- [15] J. Liu, S. Liu, H. Wu and Y. Zhang, A Pursuit-Evasion Algorithm Based on Hierarchical Reinforcement Learning, *International Conference on Measuring Technology and Mechatronics Automation (ICMTMA '09)*, vol 2, 2009, pp. 482–486.
- [16] Z.-S. Cai, L.-N. Sun, H.-B. Gao, P.-C. Zhou, S.-H. Piao and Q.-C. Huang, Multi-robot cooperative pursuit based on task bundle auctions, In: *International Conference on Intelligent Robotics and Applications*, Springer, Berlin, Heidelberg, 2008. pp. 235–244.
- [17] T. Guerram, A multi agent based organizational architecture for dynamic pickup and delivery problem, *Journal of Computing and Information Technology* **25**(4) (2018), 259–277.
- [18] A. Maoudj, B. Bouzouia, A. Hentout, et al., Multi-agent approach for task allocation and scheduling in cooperative heterogeneous multi-robot team: Simulation results, *IEEE 13th International Conference on Industrial Informatics (INDIN)*, 2015, pp. 179–184.
- [19] A.K. Agogino and K. Tumer, A multiagent approach to managing air traffic flow, *Autonomous Agents and Multi-Agent Systems* **24**(1) (2012), 1–25.
- [20] E.L. Yasmine, A.S. Larbi, et al., Supply chain management using multi-agent systems in the agri-food industry, *Service*

Orientation in Holonic and Multi-Agent Manufacturing and Robotics, 2014, pp. 145–155.

- [21] J.J. Gomez-Sanz and R. Fuentes-Fernández, Understanding agent-oriented software engineering methodologies, *The Knowledge Engineering Review* **30**(4) (2015), 375–393.
- [22] M.E.H. 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).
- [23] M.E.H. Souidi, S. Piao and G. Li, Mobile agents path planning based on an extension of Bug-Algorithms and applied to the Pursuit-Evasion game, *Web Intelligence* **15** (2017), 325–334.
- [24] E. Ehsan and F. Kunwar, Probabilistic search and pursuit evasion on a graph, *Transactions on Machine Learning and Artificial Intelligence* **3** (2015), 57–65.
- [25] S. Jia, X. Wang and L. Shen, A continuous-time markov decision process-based method with application in a pursuit evasion example, *IEEE Transactions on Systems, Man, and Cybernetics: Systems* **46**(9) (2015), 1215–1225.
- [26] L.P. Kaelbling, M.L. Littman and A.R. Cassandra, Planning and acting in partially observable stochastic domains, *Artificial Intelligence* **101**(1-2) (1998), 99–134.
- [27] V.A. dos Santos, G.C. Barraso, et al., DynaMOC: A Multiagent System for Coordination of Mobile Teams Based on Dynamic Overlapping Coalitions, *Intelligent Distributed Computing V Springer Berlin Heidelberg*, 2011, 303–308.
- [28] J. Xu, and W. Li, Solution of overlapping coalition formation based on discrete particle swarm optimization, *4th International Conference on Wireless Communications, Networking and Mobile Computing*, 2008, pp. 1–4.
- [29] C.F. Lin and S.L. Hu, Multi-task overlapping coalition parallel formation algorithm, *Proceedings of the 6th International Joint Conference on Autonomous Agents and Multiagent Systems*, 2007, pp. 211:1–211:3.
- [30] S. Kraus, O. Shehory and G. Taase, The advantages of compromising in coalition formation with incomplete information, *Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems*, vol. 2, 2004, pp. 588–595.
- [31] R. Bellman, *Dynamic programming*, Courier Corporation, 2013.
- [32] V. Sezer and M. Gokasan, A novel obstacle avoidance algorithm: Follow the gap method, *Robot Auton Syst* **60**(9) (2012), 1123–1134.