IIETA International Information and Engineering Technology Association
Advancing the World of Information and Engineering

Mathematical Modelling of Engineering Problems
Vol. 12, No. 10, October, 2025, pp. 3348-3360
Journal homepage: http://iieta.org/journals/mmep

A New Multi-Agent Access Mechanism Based on the Fairness Principle

Ridha Mehalaine[1,2*], Mohammed El Habib Souidi[1,2], Abdeldjalil Ledmi[1,2], Sami Badis[1], Mouhamed Amine Tabti[1]

[1] Department of Computer Science, Abbes Laghrour University Khenchela, Khenchela 40000, Algeria
[2] ICOSI Laboratory, Department of Computer Science and Mathematics, Abbes Laghrour University, Khenchela 40004, Algeria

Corresponding Author Email: r_mahalaine@univ-khenchela.dz

https://doi.org/10.18280/mmep.121002

Received: 2 September 2025
Revised: 7 October 2025
Accepted: 15 October 2025
Available online: 31 October 2025

Keywords:
multi-agent systems, organization, coalition formation, pursuit-evasion, Markov Decision Process

ABSTRACT

Multi-agent coalition formation can be defined as a process allowing agents to temporarily form groups to accomplish complex tasks. Given this, a complex task requires the collaboration of a group of agents to be completed. The Pursuit-Evasion (PE) game is considered an NP-hard multi-agent system (MAS) problem in which the pursuers must block the movement of the evaders while taking into account time constraints. This problem is often addressed through the development of coalition formation algorithms, in which each pursuit group is assigned to block the movement of each detected evader. In this paper, we propose a pursuit coalition formation algorithm known as Fairness-Based Agent-Group-Role-Membership-Function (FBAGRMF). The main objective of this novel extension of the Agent-Group-Role-Membership-Function (AGRMF) model is to provide an efficient access mechanism for the pursuit groups that can overcome the limitations of the original model. In other words, the fairness principle is incorporated to achieve a degree of balance among the pursuit groups during their formation. To reflect the advantages of our proposed approach, we conduct a comparison study against the AGRMF model. During this study, we took into consideration the capture time and reward evolution of the pursuers. Moreover, we studied the PE game by examining the capture of static and mobile evaders. The obtained results demonstrate that FBAGRMF minimizes the capture time and improves the pursuers' performance during the pursuit process.

1. INTRODUCTION

Multi-Agent Systems (MASs) are computerized environments where several agents can interact with each other as well as elements in the environments, these agents may have different objectives that may or may not be conflicting with each other, but they can also have a common objective where they either cooperate or even compete to achieve said objective, this study focuses on cooperative agents with a common goal, where agents form groups in which the work together to complete the required tasks allowing the entire group to succeed in their goals as a team.

The forming of coalitions is a prolific topic when it comes to multi-agent systems in the current landscape as it is a cornerstone when it comes to the cooperation between agents, the coalitions are generally formed to solve problems that an agent cannot achieve by itself, and there many papers that seek to study different ways of forming coalitions in different scenarios to enhance the task completion through a better organization.

Pursuit-Evasion (PE) games are a distinctive area of distributed artificial intelligence dealing with the problem of cooperative decision-making in multi-agent systems. The pursuit problem requires organizing the pursuers in a manner that makes the capture easier for the coalition members finishing the task quicker.

In this work, a novel extension of the Agent Role Group Membership Function (AGRMF) organizational model we proposed for coalition formation research. Specifically, we will process the problem linked to the negative externality where a pursuer might be forced into a group where it performs below its potential due to the forming of the chase group of the first evader in the set with the pursuers that suit the most even if it contains pursuers better suited elsewhere, creating a bias towards evaders higher ranked in the set. The main objectives of this paper can be resumed as follows:

- The proposition of a new pursuit coalition formation algorithm based on an extension of AGRMF model.
- The introduction of an efficient pursuit group access mechanism to provide equilibrium during the groups' assignment and avoid the negative externalities.
- The proposition of a comparative study with AGRMF model by taking into account static and mobile evaders. Moreover, the comparison study focuses on the pursuit capturing time and also the pursuers' payoff acquisition.

we will be taking a general view on multi-agent systems and some of their concepts in-line with our topic, such as the organization-centered systems where the focus lies in the coalition rather than individual agents, we’ll also be taking a brief look at Markov Decision Process (MDP), as well as viewing some organizational models that have been aimed at the pursuit-evasion problem like the Agent-Group-Role (AGR) model, the Yet Another Multi-Agent Model (YAMAM), and mainly AGRMF. Will focus on the AGRMF model and its algorithm so that we can compare it to the extension we’re proposing, focusing on the similarities and especially the differences that are present between the two, as we try to improve upon it with a different group access mechanism optimizing pursuers’ performances. finally, a thorough comparison is presented between the previously mentioned algorithms in different scenarios with multiple metrics about the group performances by measuring their capture time and individual pursuers' Performance through their reward development and gains throughout the capturing process.

The paper is organized as follows: in Section 2, we discuss some related works in relation to our proposed approach. In Section 3, we introduce the different principles used in this paper, such as, Markov Decision Process, multi-agent organizational models and dedicated to the description of the PE game. Our proposed pursuit coalition formation algorithm is introduced in the Section 4. Section 5 reflects the main obtained results in comparison with AGRMF model. We conclude the paper by a conclusion in Section 6 to resume the main work effectuated in this paper and to provide brief perspectives of the future works.

2. RELATED WORK

Multi-agent coalition formation allows autonomous agents to dynamically form groups (coalitions) in order to collaborate on tasks that no single agent can efficiently perform alone. Recent research activities are based on scalable, more flexible, and online mechanisms. In relation to Internet of Things (IoT) [1], a dynamic coalition formation model among IoT service providers improves grouping based on changing demand and outperforms static schemes. In multi-robot task allocation [2], learning-based frameworks allows the robots to form coalitions through the use of local information and reorganize groups in real time. The generation of optimal coalition formation remains an attractive research activity. New hybrid algorithms that reduce search spaces make coalition formation tractable for large number of agents [3]. These advances improve the multi-agent coordination, flexibility and dynamism in cooperative multi-agent systems.

In the last decade, multi-agent organizational models were hugely used to provide pursuit coalition formation algorithms. In the study by Souidi et al. [4], the authors have based on the AGR organizational model [5] to introduce a pursuit coalition algorithm. Specifically, the authors were inspired by the concepts of AGR model to generate a pursuit model. On the one hand, the concept Role was used to define the roles pursuer. On the other hand, the concept group reflected the pursuit groups composed of pursuers to capture the evaders. In comparison to our proposed approach in this paper, Souidi et al. [4] did not take into consideration the dynamism of pursuit groups during the process. In other words, the authors have based on pursuit groups and not on pursuit coalition. The difference between a group and coalition in MAS is that the coalition can change during the task execution, however, the group remains static. In addition, and in comparison, with the work proposed in this paper, Souidi et al. [4] did not introduce pursuit groups access mechanism allowing the pursuers to join the groups in which there are more efficient.

In the same context, Souidi et al. [6] and Siam et al. [7] introduced a new extension of AGR model based on fuzzy logic principles (AGRMF). The authors had modeled the concept group as a fuzzy set equipped with a membership function that permits to define the membership degree of each agent in relation to each group. Knowing that in AGR model, there is no mechanism allowing the agent to access to the groups. Souidi et al. [6] applied AGRMF model to the PE game in comparison with AGR [4]. From the obtained results, AGRMF model improved the pursuit capturing time and pursuers’ development during the process. Also, AGRMF provided the dynamism of the groups by allowing their reorganization when a more efficient coalition is identified during the pursuit process. In comparison with the proposed Fairness-Based Agent-Group-Role-Membership-Function (FBAGRMF), the pursuit algorithm based on AGRMF model generates negative externalities between the pursuit groups. Precisely, for each pursuit group, AGRMF assigns the pursuers without taking into account the other groups. This fact makes the order of the groups’ assignment a very important criteria for pursuit processing. In FBAGRMF, we propose an pursuers’ assignment mechanism that avoids the negative externalities of the formed groups.

In the study by Souidi et al. [8], Overlapping AGRMF (OAGRMF) was proposed. In this work, the authors introduced a pursuit coalition method permitting a limited overlapping of the pursuit groups. in other words, in OAGRMF, a pursuer can pursue more than one evader at the same time. This solution eliminates the problem linked to the negative externalities of the pursuers and also allows the processing of the PE game in the case where the number of required pursuers is not sufficient to capture the detected evaders. In comparison with FBAGRMF, the capture of evader became very complex when the distance between the evaders is too long in OAGRMF case. This fact made the displacement of the overlapped pursuers between the concerned evaders a very complex task.

Regarding other multi-agent organizational models, YAMAM was also used to provide a pursuit coalition formation algorithm in the study by Souidi et al. [9]. The authors have based on the concept’s agent, role, task, and skill composing YAMAM to process the PE game. In their proposal, the concept role defines the roles pursuer and evader of the agents. The task represents the evaders that must be captured. The skill represents the position of the pursuers in the environment in relation to the evader and also their internal velocities. The role pursuer is attributed to the agents according to their skills. In relation to the proposed FBAGRMF, the coalition based on YAMAM did not allows the dynamic reorganization of the pursuit groups during the PE process.

Game theoretic methods are also often used to process the PE game. Souidi et al. [10] based on the Iterated Elimination of Weakly Dominated Strategy (IEWDS) to provide a pursuit coalition algorithm. in their proposition, a strategy is considered as a possible coalition of the agents. After the generation of the possible coalitions, the agents eliminate their dominated strategies in turn until obtaining the undertaken strategy. The main advantage of this method is that it allows a dynamic reorganization of the agents during the PE process. Moreover, it avoids the selection of the coalitions with negative externalities. However, in comparison with FBAGRMF, the calculation of the possible coalitions raises a serious problem when the number of agents increases.

Recently in the study by Ma et al. [11], the authors proposed a cooperative game-theoretic method for multi-agent PE game. The authors introduced a coalition formation algorithm that allows pursuers to form dynamic coalitions under communication and interaction limits. Through the definition of coalition values and stability conditions, the approach improved coordination and resource allocation among agents. Simulation results showed that the proposed method ameliorated the evaders' capture efficiency and reduces redundancy in comparison with non-cooperative strategies. This work provides a significant contribution to cooperative control and multi-agent coalition in complex PE game environments. In comparison to FBAGRMF, the two approaches are based on dynamic coalitions, however, regarding the communication between the pursuers used in FBAGRMF is completely indirect via the use of the PE game environment. The main advantage of using mobile agents is that we can only base on indirect communication (through the environment) between the agents.

3. PROBLEM DESCRIPTION

The notion of MAS is relatively new in the field of Distributed Artificial Intelligence (DAI), it has proven to be quite promising over the years it's been in existence, and this notion also become very popular in mainstream computer science.

3.1 Definition

MASs are computer-based environments comprised of multiple intelligent agents that interact with one another. MAS is defined by Stone and Veloso [12] as "a loosely coupled network of problem solving entities (agents) that work together to find answers to problems that are beyond the individual capabilities or knowledge of each entity (agent)" [13].

Several MAS research projects focus on the behavior, construction and study of a set of potentially pre-existing autonomous agents that interact with each other and with their environments [14].

3.2 Multi-agent learning

Due to the importance of multi-agent learning [15], aspects that encourage its separate study from traditional machine learning. First, multi-agent learning can involve multiple learners, each adapting and learning from the context of the others; this introduces unresolved game-theoretic issues into the learning process.

Second, since multi-agent learning deals with problem domains involving multiple agents, the relevant search space can be exceptionally large; these agents can frequently cause unpredictable changes in the emergent properties of the multi-agent group, following their small changes in learned behaviors.

3.3 Agents

"Interaction" [16] indicates that agents can be influenced by other agents, or even humans, in the pursuit of their goals and the execution of their tasks. Interaction [16] can occur directly through a shared language, or indirectly, through the environment in which they operate. To avoid situations undesirable by one or more agents, coordination is the most effective means. Competition and cooperation are two fundamental models of coordination. In the case of competition, agents attempt to accomplish conflicting goals as a team by working against each other. In the case of cooperation, agents attempt to accomplish common goals as a team by working together. Cooperating agents attempt to accomplish as a team what individuals cannot accomplish, and thus fail or succeed together.

3.4 Environment types

A task environment [17] is effectively fully observable if the sensors detect all aspects relevant to the action choice; relevance, on the other hand, depends on the performance measure. The environment is unobservable if the agent has no sensors. The agent to track the world, does not need to maintain an internal state to track the world so fully observable environments are practical. Due to noisy and imprecise sensors the environment may be partially observable.

3.5 MDPs

To decide on the optimal next action, MDPs consider the agent's actions, environmental information, and rewards. Depending on the set of available states, actions, and the frequency of decision-making, MDPs can be characterized as discrete or continuous, infinite or finite [18, 19].

It is a stochastic process $\{X_t,\ t \in T\}$ is a set of random variables indexed by a parameter t (discrete-time) and defined on the same probability space. The variable X_t represents the state of the process at time t. Moreover, the set of all possible values for this variable is called the state space of the process and will be denoted S.

A discrete-time Markov chain is a discrete-time stochastic process $\{X_t,\ t = 0,1,\ldots\}$, defined on a finite space of states S and satisfying the following Markov property:

$$\begin{gathered}\forall i \in, \forall t > 1[X_t = i|\ X_0,\ldots,X_{t-1}] = \\ P[X_t = i \mid X_t = i \mid X_{t-1}]\end{gathered} \tag{1}$$

In such a process, known as a memory less process, predicting the future from the present does not require knowledge of the past. Markov chains are used to model stochastic systems but do not allow an agent to intervene and act on the evolution of the system.

The controller can use several actions for a purpose. His goal is to build an optimal plan to maximize the reward he can get. The evolution of the system is considered a Markov process, i.e., the evolution of a temporal sequence of distinct states according to transition probabilities, based solely on the phenomena of the previous state [20].

3.6 Organizational models in multi-agent systems

An organizational model is a task coordination mechanism for multi-agent systems. This model allows agents to cooperate and share tasks to be able to solve complex problems. All this is done within the framework of a metamodel which makes it possible to define the relations between the agents as well as the role of each one [21].

3.6.1 Organizations in MASs

Jennings [22] stated that one of the ways to simplify the design of complex software systems is MAS. With higher-order abstractions, and managing the complexity of current software systems, MAS without real structure is not suitable, should be used and some way of structuring is necessary to more accurately model the problem addressed [23, 24].

Gasser [25] stated that we have virtually no concrete experience building truly heterogeneous, realistically coordinated, and co-operating multi-agent systems. Horling and Lesser [26] stated that our current real world demands new software engineering paradigms such as MAS.

To reduce or manage uncertainty, provide strength in numbers, limit the scope of interactions, and formalize high-level goals while no agent can be conscious [27, 28]. Organizations provide a framework for managing and structuring agent interactions to regulate the level of agent autonomy [29, 30]. Different types of organizational structures are adapted to particular environmental conditions [31, 32].

The approach proposed by Corkill and Lander [33] is more suitable for the engineering of complex large-scale adaptive multi-agent systems, especially with the evolution of the IoT [34]. Several application areas include smart grids [35], Cyber-Physical Systems (CPS) [36], ubiquitous computing [37], global Supervisor Control and Data Acquisition (SCADA) [38], ubiquitous computing [39], etc.

3.6.2 Organization types

In studies [40-42], several types of organizations are distinguished:

i. Group: several types of groups exist:

- Simple group: As soon as a group exists, we can have cooperative coordination to achieve a common and shared goal.
- Team: A collection of individuals who have been brought together to work together. In an organization, several teams are formed for communication reasons. In this definition, individuals belonging to a team must necessarily communicate with each other, which leads to the introduction of the notion of environment (framework in which agents exist and evolve).
- Interest group: Each member has the same interests; they share information and cooperate to achieve a common goal.
- A community of practice is formed when professionals come together and organize themselves to share information and experiences relating to their activities. The members of these communities can thus exchange and cooperate to jointly solve the problems with which they may be confronted, thus learning from each other and building common knowledge and practices together.

ii. Hierarchy where we distinguish:

- Simple hierarchy: Based on a master/slave relationship, this type of organization is no longer used.
- Multi-level hierarchy: Authority links form a tree. Control in this type of organization is very complex, for example, the problem of resource allocation or planning must be taken into account.

iii. Decentralized organization: It is a multi-division hierarchy where each top of a branch is an organization in its own right. The main difficulty in this type of organization is the integration of the different results from the different divisions.

iv. Market: This type of organization is based on the consumer/producer relationship. A special instance of the market is the Contract Net Protocol which is a protocol allowing the development and execution of a contract between a manager agent and a contracting agent. It involves agents interacting with each other during the development and execution of the contract employing performatives.

v. Coalitions: To enable agents to benefit from their respective skills, the use of a short-term organization based on specific and contextual commitments is necessary.

3.6.3 AGR

The AGR model is based on three concepts: Agent, Group, and Role.

A group is a set of agents that share a common characteristic [21, 43]. A group serves as a context for an activity model and allows organizations to be partitioned. If agents belong to the same group, they can communicate.

AGRMF [44] is a fully observable, multiagent, stochastic, episodic, static, continuous in a known environment flexible organizational model extension of the AALAADIN model that aim at the Pursuit problem, this model allows the interaction, organization, and re-organization of pursuers.

YAMAM [44] model is an alternative to other multi-agent models such as AGRMF [45] and AGR [4]. The YAMAM model is characterized by its scalability, modularity, and task planning tools. The four essential concepts of YAMAM are: agent, skill, role, and task. The agent is built on reactive properties, which implies that skills cannot be added dynamically. The relationships between agents are paramount for agents and their behavior.

The agent plays a role if he is able to perform the required tasks. Typically, this role requires the ability to perform multiple tasks. In other words, agents have no view of existing groups and are completely unaware of their membership information. In our proposal, the groups will be represented by the various coalition formations formed to capture detected escapees.

3.7 The pursuit-evasion problem

The pursuit-evasion has always been a popular topic in AI research, as many combinations of different scenarios and situations could be explored, it can be generally defined as the problem of capturing mobile target(s) with one or more pursuers, with the possibility of having obstacles in the environment using different strategies. There are four main types of problems that we'll be exploring below.

3.7.1 The single pursuer, single evader

This type of problem contains a single pursuer chasing after a single mobile target throughout the environment, here the pursuer and the evader can move at the same speed, like in Yamashita et al. [46] where the evader has unbound speed or Alonso et al. [47] Lion ad Man problem where the pursuer possesses a higher speed.

3.7.2 Multi-pursuers, single evader

Here there are multiple pursuers attempting to catch a single mobile evader, in his type, the pursuer can be competitive where the first pursuer to capture the target is the winner or in a cooperative environment like in Wan et al. [48] where the pursuers form flocks that coordinate between its members to corner and apprehend the evader.

3.7.3 The single pursuer, multi-evader

In this type, a single pursuer tries to catch multiple evaders, the evaders can move independently from one another, or they can cooperate such as in the case of Salmon et al. [49].

3.7.4 Multi-pursuer, multi-evader

This type has many different scenarios and might require one or more pursuers to capture each evader, where every evader requires a different number of pursuers to be captured resulting in a cooperative environment where pursuers form coalitions and coordinate to optimize the capture time which is the case in the AGRMF [44] model.

4. THE PROPOSED ORGANIZATION MECHANISM

In this paper, we'll be proposing a new group access mechanism for the AGRMF algorithm based on a fair approach to coalition formation. The main purpose of this approach is to form balanced coalitions where pursuers can chase after evaders that may be closer it rendering the capture time faster and more balanced, to do that we must eliminate the issue where the earliest evader in the set monopolizes the closest pursuers giving it fast capture time while the remaining evaders can take much longer to be captured due to their optimal pursuers joining the first evader's coalition.

The main goal is to solve the issue of negative externality that plagues the AGRMF model (Figure 1). A negative externality is where an agent joins a coalition where it's possible performance decreases instead of joining a coalition where it can reach or at least gets close to its full potential.

While the mechanism that will propose may decrease the performance of certain groups, it will allow the coalitions to be more balanced leading to all pursuits finishing at approximately the same time instead of having one pursuit take much longer than others.

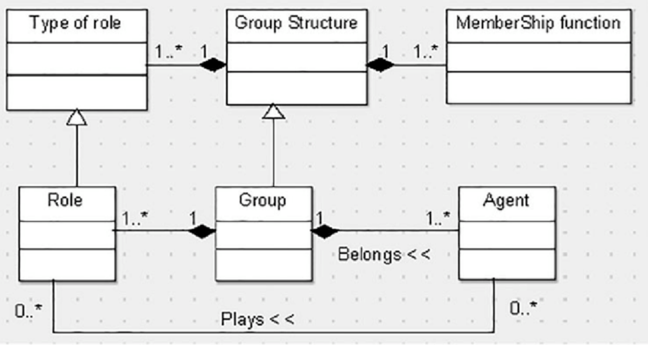


Figure 1. AGRMF meta-model

To achieve that task, we must first understand the pursuit-evasion scenario that we are facing, as well as a proper understanding of AGRMF and its components.

4.1 AGRMF

Souidi et al. [44] created an extension of the AGR model that changed the groups' vision, providing methodological guidance which enables their flexibility within a multiagent system.

4.2 The pursuit problem

In this type of problem [44], the agents are split into two different roles, the evaders, which are agents moving freely and completely independently around the environment, and the pursuers, which form coalitions to cooperate in catching the evaders, these two roles, as well as the method of forming coalitions will be explained more thoroughly later on.

4.2.1 The pursuer

The set of pursuers that is responsible for the capturing is denoted by $P = \{P1; 2; P3; \dots; Pn\}$ where each pursuer possesses its capacity parameters explained below.

***Self-confidence degree*:** A metric that represents the agent's competence, calculated using the number of tasks that the agent has successfully finished Cs, and Ct; the number of tasks in which the agent has participated, this parameter is denoted as follows:

$$\forall Conf \in (0.1;1): Conf = \max(0.1,(Cs/Ct)) \qquad (2)$$

***Credit*:** Each agent receives credit after tasks, the credit is affected negatively if the agent cannot complete a task, the credit is denoted bellow with Cb being the number of the abandoned tasks by the agent:

$$\forall Credit \in (0;1): Credit = \min(1,1-(Cb/(Ct-Cs))) \qquad (3)$$

The distance: Capture will be easier if the pursuer is closer. So distance is a very important criterion. The position Pos is computed as follows:

$$Pos = \min((;)) \qquad (4)$$

where, $Dist$ is the distance between the pursuer and the evader, Sp is the state (cell) of the pursuer, SE is the state (cell) of the evader.

$$(Sp;SE) = \sqrt{(\mathrm{Copi}-\mathrm{CoEi})2+(\mathrm{Copj}-\mathrm{CoEj})2} \qquad (5)$$

where, $(Copi)$ is the Cartesian coordinates of the pursuer, and $(CoEi)$ the Cartesian coordinates of the evader.

***The evader*:** The set of evaders which are mobile agents that move in random directions is denoted by $E = \{E1;\ E2;\ E\ 3;\ \dots;\ En\}$.

***The membership function*:** Using fuzzy logic, the degrees of membership are admitted to a given set. These values represent how well an agent fits in a certain role:

$$\mu_{Group}(Agent) = \frac{Coef_1 \times Pos + Coef_2 \times Conf^3 + Coef_3 \times Credit^3}{\sum_{i=1}^{3} Coef_i} \qquad (6)$$

4.3 AGRMF algorithm

The steps in Algorithm 1 are explained in the following manner: the organizer evaluates the parameters of each evader agent and creates n groups of the alliance (Figure 2).

Algorithm 1: AGRMF algorithm

```
1. Create-MemFu();
2. Broadcast ($Pos$);
   Waiting-Response;
3. SendResponse ($Pos, Conf, Credit$);
4. $\mu Group(Pi)$
5. $List1$: contains the membership degree of each agent pursuer.
   $List2$: empty list.
   Nbr: number of agent pursuers.
   $\lambda \leftarrow 0$;
   Repeat
        $If\ (Pj \notin List2)\ \&\&(\mu(Pj)) = Max(List1)\ then$
          $Add\ (Pj, 2(Ei))$;
          $Delete\ (Pj, 1)$;
            $\lambda \leftarrow \lambda + 1$;
        End if;
        $i \leftarrow i + 1$
   Until $\lambda = x$
        Else
             Waiting
        End if;
6. If $Pi \in List2$ then Join group;
7. Launch of the chase:
   $C_{life} = \begin{cases} e^{-klt^2} & t \leq T_{max} \\ 0 & t > T_{max} \end{cases}$
8. $C_{life} = 0$
   If $(capture\ ==\ true)$ then
      Go to step 9;
   Else
      Go to step 2; $t \leq T_{max}$
   $t > T_{max}$
   End if;
9. For each pursuer
     $r(St) \leftarrow r(Sn)$;
     $Ct \leftarrow Ct + 1$;
     $Cs \leftarrow Cs + 1$;
```

4.4 The proposed pursuit coalition formation algorithm

In our proposition, we are looking to further optimize the Organizer, to minimize the capture time, through a more even distribution of pursuers. The Organizer in the AGRMF model allows the first in the set of evaders to be pursued by the pursuers in the most optimal position (closest distance) completely disregarding the remaining evaders, as opposed to our proposed method of allowing all evaders in the set to add a single pursuer to their respective groups at a time resulting in a fairer pursuit.

The Organizer is the entity responsible for admitting pursuers into groups, in our proposition, the organizer goes through the evaders one by one, choosing the closest pursuer and admitting it to the chase group of the current evader, this process is repeated as many times as needed to fill the chase groups. The main difference between our proposition and AGRMF is that in AGRMF the evaders receive higher priority based on their placement in the set, allowing the first evader to take up the most optimal pursuers leaving the remaining evaders with scraps, in our proposition, however, each evader has a chance of having closer pursuers due to adding a single pursuer to each group at a time while repeating this until all groups are filled, resulting in an even distribution of pursuers.

The steps in Algorithm 2 are also shown in Figure 3, which are explained as follows: the organizer integrates the membership function into the alliances needed after evaluating the parameters of the evader agents. The organizer sends as many messages as there are groups to the pursuer agents containing the *Pos* of each evader, after that it waits for a certain amount of time for the response from the pursuers (02). The pursuers now must respond to the organizer with a message containing the *Pos*, *Conf*, and *Credit* (03). The organizer now uses the membership function to compute the degree of membership of each pursuer (04). The organizer goes through the set of evaders one by one, adding a single pursuer to the current evader's chase group, this pursuer has the highest membership degree of the available pursuers. When a pursuer gets added to a group chase, it is automatically removed from *List* which contains all the available pursuers. This process is repeated *Re* times, where *Re* is the number of pursuers that the evader needs to be captured (05). The organizer must keep track of the chase's progress from the start. This hunt will not continue indefinitely; the alliance has a *Clife* value (06). When *Clife* = 0 and the evader is captured, and the organizer ends the chase (07).

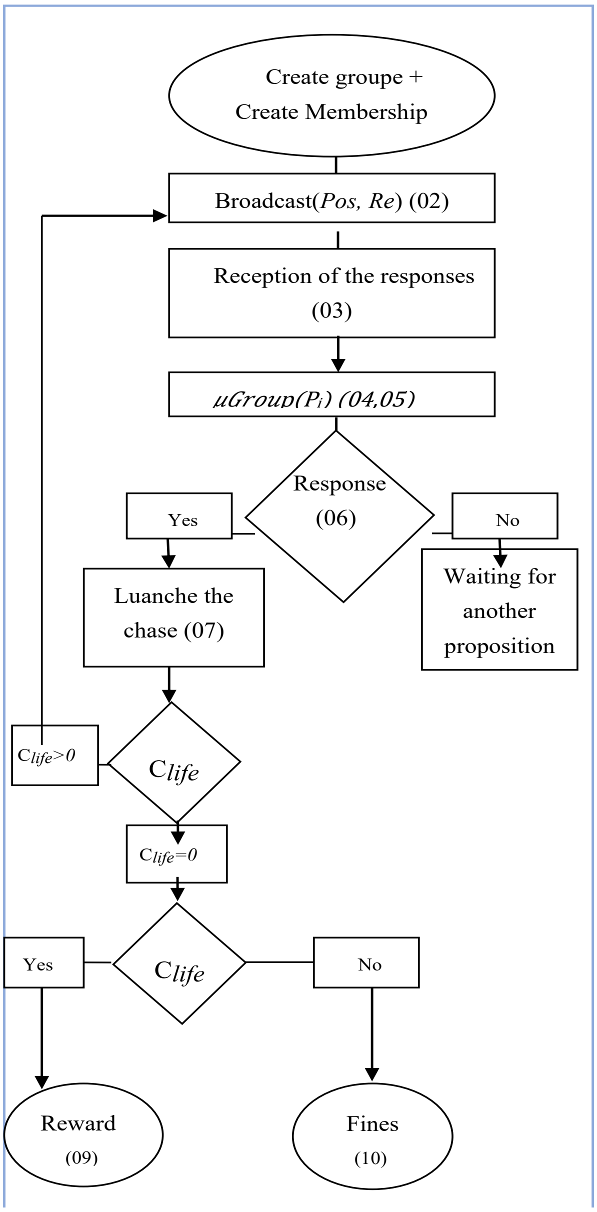


Figure 2. AGRMF flowchart

Algorithm 2: FBAGRMF algorithm

```
1. Create-MemFu();
2. Broadcast ($Pos$);
   Waiting-Response;
3. SendResponse ($Pos, Conf, Credit$);
4. ($Pi$)
5. $x = 0$;
Repeat
        For $i = 0$ to $n$ do
            Select ($Pj$) from List;
            Add $Pj$ to ($Ei$);
            Delete $Pj$ from List;
              $i \leftarrow i + 1$;
        End for
        $x \leftarrow x + 1$;
Until $x = Re$.
6. Launch of the chase:
$C_{life} = \begin{cases} e^{-klt^2} & t \leq T_{max} \\ 0 & t > T_{max} \end{cases}$
7. If ($capture == true$) then
        End;
      Else
        Go to step 2;
      End if;
```

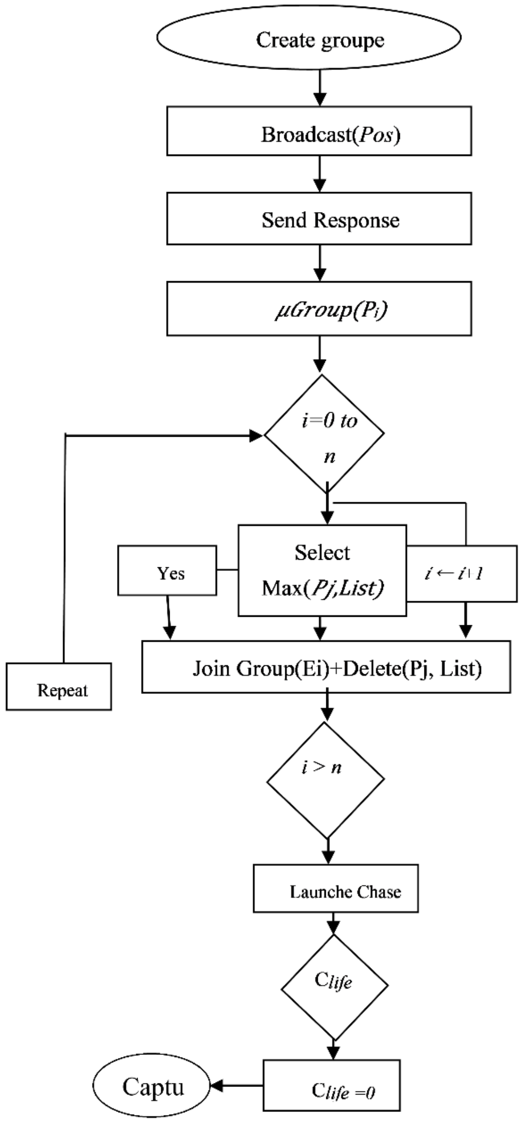


Figure 3. Fairness-based AGRMF flowchart

Table 1. Role attribution

Algorithm Group	P1	P2	P3	P4	P5	P6	P7	P8
Group 01	23%	50%	45%	80%	61%	20%	30%	55%
Group 02	28%	44%	54%	80%	62%	23%	17%	90%

The main difference between the two algorithms is how they distribute the pursuers; our proposition seeks to increase the efficiency by allowing the pursuers to join coalitions where they are better suited, rather than joining a coalition that only suits the first evaders in the set, this is demonstrated by better through the following example in Table 1.

In this example, the coalitions formed with AGRMF are Group 1: (P4, P5, P8, P2) = 61.5%, Group 2:(P3, P1, P6, P7) = 30.5%, while the fairness-based algorithm forms the coalitions like so Group 1: (P4, P5, P2, P7) = 55.3%, Group 2: (P8, P3, P1, P6) = 48.8%.

Even though the fair distribution caused Group 1's attribution to drop by 5%, it more than made up for it by increasing Group 2's attribution by over 18%.

According to this example, we can note that the pursuer P8 is a member of Group 1 in AGRMF case with a membership degree of 55%. However, we can constate that P8 can be more efficient in Group 2 with membership degree of 90%. According to this fact, we can conclude that P8 provoked a negative externality on Group 2 of AGRMF case. In formal way, negative externality can be explained as follows:

Let N = {1, 2, …, n} be the set of agents and v(S) the value (or utility) of a coalition S⊆N. A coalition structure CS = {S_1, S_2, …, S_k} is a partition (or an overlapping configuration) of N.

A negative externality occurs when the formation or composition of one coalition negatively affects the utility of another coalition. Formally, for two coalitions S_i, S_j⊆N: $v(S_i)>v'(S_i)$; where $v'(S_i)$ denotes the value of S_i after the change in S_j composition or formation. That is, the payoff or performance of S_i decreases due to the presence, behavior, or reorganization of agents in another coalition.

4.5 Pursuers path planning

In this section, we will explain how the pursuers are moving in the environment according to MDP principles where agents try to end up the maximum reward possible at the end of the chase through the quadruplet (S, A, T, R) where S is the space of states [3], A is the space of actions, T represents a transition function where $T(s, a, s')$ is the probability of reaching a state $s' \in S$ from an initial state $s \in S$ after executing an action $a \in A$. And the reward function R where $R(s, a) \in R$ is the reward of executing action $a \in A$ from a state $s \in S$. A policy is a function $\pi: S \rightarrow A$ that associates each state S with a corresponding action A as so:

$$\pi(S_t) = a_t \tag{7}$$

The value function of policy π links each state with the accumulated rewards gained from the state through the function $V^{\pi}: S \rightarrow R$.

$$V^{\pi}(s) = E\left[\sum_{k=0}^{\infty} \gamma^{k} \mathrm{R}(\mathrm{sk}, \pi \mathrm{sk}) s_0 = s\right] \tag{8}$$

The discount facture $\gamma \in [0,1]$ reduces the emphasis on distant rewards. A fundamental property of policy π is that it checks a recursive equation of Bellman:

$$V^{\pi}(s) = R\left(s, \pi(s), s^{'}\right) + \gamma \cdot \sum s^{'} T\left(s, \pi(s), s^{'}\right) \cdot V\left(s^{'}\right) \qquad (9)$$

The primary goal of MDP is to establish the best policy. We take notice of V^*, the optimal value function, which associates the best possible reward with each state.

$$V^{*}(s) = \max_{\pi} V^{\pi}(s) \qquad (10)$$

We can easily determine the optimal policy π by selecting the greedy policy among the values of V^* if the optimal value function is known:

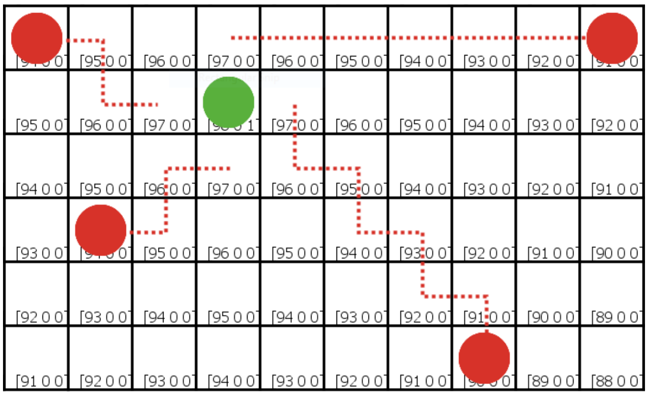
$$\pi^{*}(s) = \operatorname{argmax}_{a \epsilon A} R(s,a) + \gamma \cdot \sum s^{'} T\left(s, a, s^{'}\right).V^{*}\left(s^{'}\right) \qquad (11)$$

Our pursuers' movement strategy (Figure 4) is entirely centered on the best policy π. Given $V^*(s)$, an agent can maximize its reward by employing the "greedy" strategy of choosing the action with the highest reward. This technique is greedy because it uses $V^*(s)$ as a substitute for immediate reward before maximizing its immediate profit. It is ideal since the function V^* is a precise summary of future rewards.

In our simulation model, the possible states for each agent are {S_{up}, S_{down}, S_{left}, S_{right}}. Knowing that, at each iteration, the agent can move in these four directions. The action of each agent at each iteration allows it to change its state in the environment. For example, P (s' |s, a) is the transition probability to go from the state (s) to (s') through the application of the action a. The reward of each state (as shown in Figure 4) is calculated according to the distance separating it of the concerned evader:

$$R(s) = \frac{\alpha}{\text{distance}\left(s, s^{*}\right)} \qquad (12)$$

S: actual state, s^*: goal state, $\alpha > 0$: normalization factor, distance (s, s^*): Euclidian distance between s and s^*.



Note: Red agents: pursuer; Green agent: evader.

Figure 4. Pursuers' motion strategy

5. SIMULATIONS AND RESULTS

In this section, we'll introduce the concept of Agent-oriented Programming highlighting the differences between it and classical programming paradigms. we'll also be taking a look at the multi-agent platforms that use said paradigm which we chose for this study; NetLogo denoting its history and presenting both its components and its features.

We shall also be discussing the results of the comparison between the Fairness-based AGRMF (Figure 3) that was discussed in the previous section, and AGRMF as well as mentioning the testing methodologies for the different scenarios, the scenario with a dynamic evader where both pursuers and evader move at the same speed throughout the entire chase. In the second scenario, evaders can make a single move at the beginning of the chase and then remain static for the remaining time. The final scenario will have an evader that's dynamic and moves twice as fast as its pursuers increasing the difficulty of the chase.

The results will be displayed visually through charts that'll help us makes an objective comparison between the two approaches, the results will also be summarized in a table to highlight the differences that we'll be finding.

5.1 Agent-oriented programming

Agent-oriented programming was proposed by Wang et al. [50] as a new programming paradigm, which can be seen as a specialization of object-oriented programming. In this approach, agents are the central elements of the language, in the same way, that objects are central for object-oriented languages. The perspective on agents is cognitive: agents are characterized by mental notions such as their beliefs, decisions, and obligations. In addition, each agent is associated with a set of skills that represent what the agent can do. At the same time, agent-oriented programming assumes that we will develop programs in which several agents interact, which emphasizes the social dimension of agents. The programming language proposed by Shoham as a demonstration of this new paradigm is called AGENT0.

The main difference between such a language and a classical programming language that one could use to develop agents, comes from the fact that the mental notions which characterize the agents appear in the language itself, and that the semantics of the language is intimately linked to the semantics of these mental notions. Agent-oriented programming can be seen as a specialization of object-oriented programming because program modules are now agents, i.e., objects with a state that defines the associated mental notions, and the messages between objects are replaced by messages between agents. How do messages between agents differ from messages between objects? Firstly, these messages are modeled based on speech act theory, which is concerned with communicative actions such as informing, asking, offering, accepting, rejecting, and more. Second, since agents are autonomous and mentally capable, they have the freedom to decide whether or not to perform the action specified in the message. In contrast, an object receiving a message will always perform the action specified in the message.

5.1.1 The NetLogo platform

NetLogo is a simple and primitive. This platform is distributed in open-source in its current version. The design and development of NetLogo were carried out with the aim of modeling.

Both mobile and immobile agents can be modeled by NetLogo. It deals with multiple agents belonging to the same physical space and can also include thousands of agents in the same simulation. In addition, there are other capabilities including agent and network binding capabilities, two-dimensional and three-dimensional options regarding display modes, as well as easy switching between one-step execution and continuous simulations. The features distinguishing NetLogo could be summarized as follows:

In recent research activities, NetLogo has been used to study the impact of ant colony optimization on the traffic problem, the traffic control problem is modeled as a multiagent-multi-uses (MAMP).

During our work, we found the usefulness of using NetLogo to describe the pursuit escape environment as well as the process of pursuing agent coalitions until the capture of the detected escapees.

A NetLogo program can be composed of different types of agents:

- Turtles: They are our agents, they have characteristics, they move around the world, they are born and die; they can be of several types (Breeds, consumers, firms).
- Patches: They are spatial components of the modeled "world"; agents move on patches and these can store variables.
- Links: A particular type of agent which connects two agents and is represented as a line drawn between these agents. This link may or may not be oriented.
- World: It corresponds to the spatial representation (in 2D or 3D) of the modeled environment.

5.2 Running the simulations

The environment for our tests was done in a 100 × 100 cell grid, containing ten pursuer agents, and two evader agents, the agents can move by a distance of one cell per episode in one of the four cardinal directions; north, south, east, and west, each evader needs a group of four pursuer to be captured.

5.2.1 Dynamic simulations

The following chart depicts the result of running a 100 simulation or both the normal AGRMF and the fairness-based AGRMF.

In the superseding chart, we see that the fairness-based algorithm performs consistently better than the normal AGRMF, as it has a lower capture time aside from a few edge cases here and there, with an average of 83 for the fairness-based AGRMF, compared to the 107 of the normal AGRMF showing a 22.5% improvement overall, while the highest capture time for the fairness-based AGRMF is 140, it's still 17 episodes lower than the normal AGRMF which peaked at 157, and there's an even bigger difference of 21 episodes when it comes to their fastest times of 58 for the fairness-based AGRMF, and, 79 for the normal AGRMF.

In order to confirm the results reflected in Figure 5, an analysis was performed using an Independent Samples T-Test (a procedure equivalent to a one-way ANOVA when comparing only two groups) to evaluate the difference in the mean number of iterations required by the FBAGRMF and AGRMF methods. The null hypothesis, stating that there is no significant difference between the mean iteration counts of the two methods ($\mu_{FBAGRMF} = \mu_{AGRMF}$), was tested against the alternative hypothesis.

The T-test results indicated a statistically highly significant difference between the two methods ($t = -9.60$, $p < 0.0001$). Given this highly significant result, the null hypothesis is rejected. This finding was further supported by a non-parametric Mann-Whitney U test, which also yielded a significant result ($p < 0.0001$).

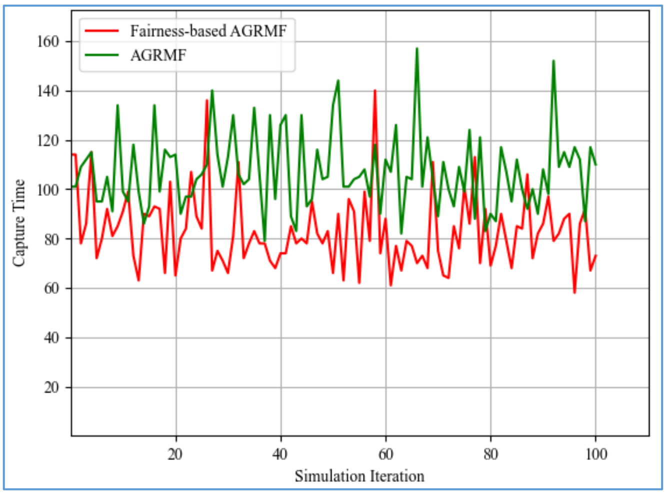


Figure 5. Capture time per run (Dynamic evader)

The mean number of iterations for FBAGRMF ($\bar{x} = 84.80$) was found to be significantly lower than that for AGRMF ($\bar{x} = 107.59$). The mean difference of 22.79 iterations confirms that the FBAGRMF method achieves convergence with substantially fewer steps. Therefore, based on the iteration count across 100 episodes, the FBAGRMF method demonstrates superior performance and significantly greater computational efficiency compared to the AGRMF method.

Figure 6 illustrates the evolution of the reward for an execution of a given duration relative to the average of each algorithm. We can see that the starting position for the fairness-based AGRMF has the advantage, which it maintains until around the 40th second, the moment when it intersects with the normal AGRMF, even occasionally falling below it before recovering just before the end of the 40 seconds, until it reaches the target, while the normal AGRMF remains constantly below until the end.

These results are extracted from the same run as the previous graph, the fairness-based AGRMF suffers a loss in reward in 05 instances, about 6% of the time, while the old implementation loses rewards about 8.4% of the time, or in 9 instances. From this we can deduce that pursuers are more effective when they are distributed in a balanced manner where they can be a lot more effective. The average reward gain is represented in Figure 7.

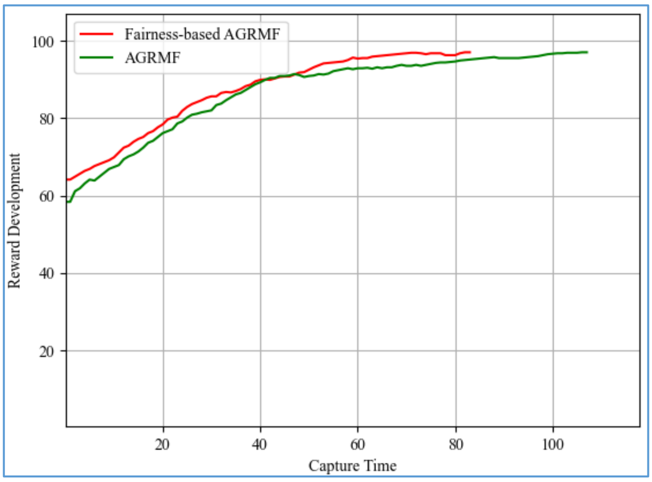


Figure 6. Average reward development (Dynamic)

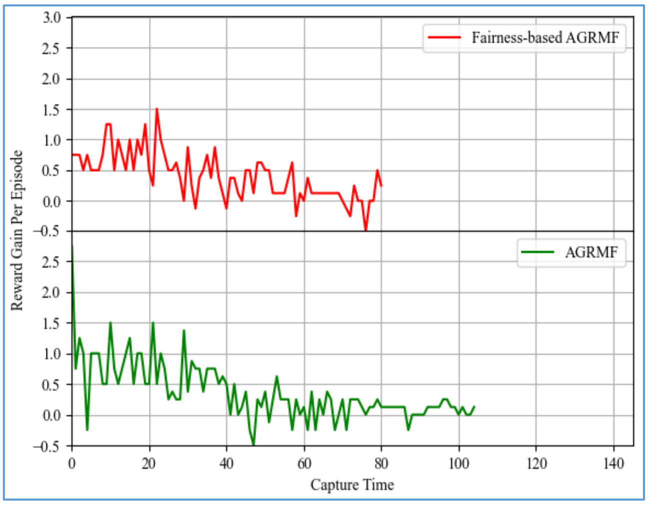


Figure 7. Average reward gain (Dynamic)

5.2.2 Static simulation

The static simulation included evaders that move by a single cell in a random direction at the beginning of the chase and remain static for the rest of it. The capture time per run of static evader is represented in Figure 8.

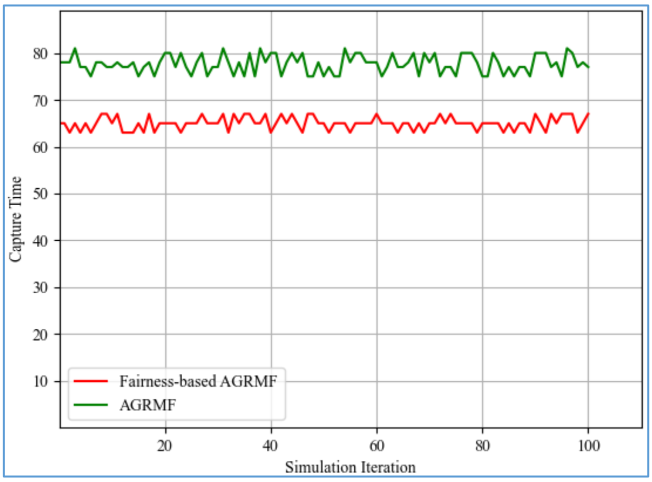


Figure 8. Capture time per run (Static evader)

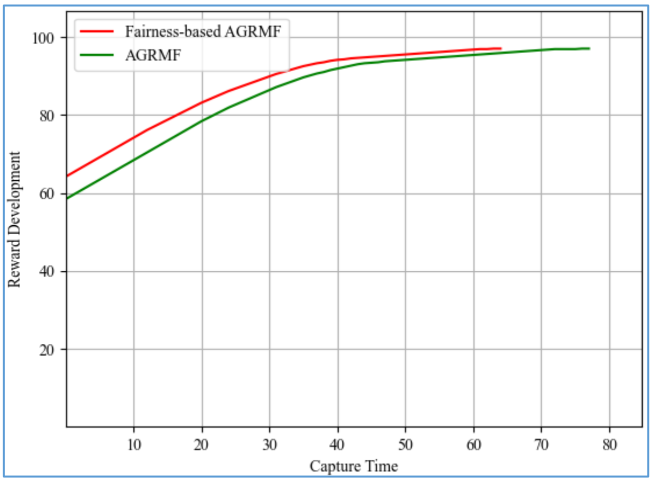


Figure 9. Average reward development (Static evader)

The advantage of fair distribution is much more evident, with an average of 65 episodes for the fairness-based AGRMF, compared to 77 episodes for the standard AGRMF, representing a 15.4% reduction in captures. While the decrease in average capture time is less than that achieved when the escapee is dynamic, it is much more consistent because the fairness-based algorithm is always faster.

Figure 9 tells the same story as its dynamic counterpart, with a consistent rise for both algorithms with a maintained advantage for the fairness-based AGRMF of the normal AGRMF from beginning to end.

Here we see that both instances maintain the same gain but for nearly twice the number of episodes for the normal AGRMF before a steep curve down in contrast to the much gentler one for the fairness-based AGRMF right before flattening around the later third for both of them until the end. The average reward gain with the static evader is represented in Figure 10.

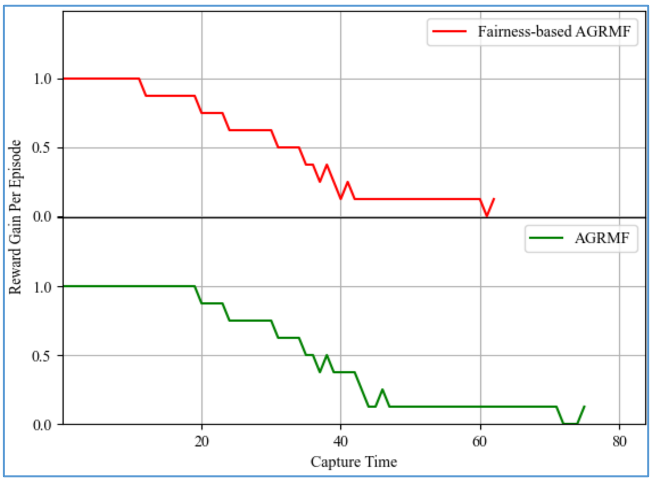


Figure 10. Average reward gain (Static evader)

5.2.3 Fast evader simulations

The following graph depicts the result of running a 100 simulation or both the normal AGRMF and the fairness-based AGRMF where the evader is allowed to move twice each episode making it faster than the pursuers. The capture time per run with the fast evader is represented in Figure 11.

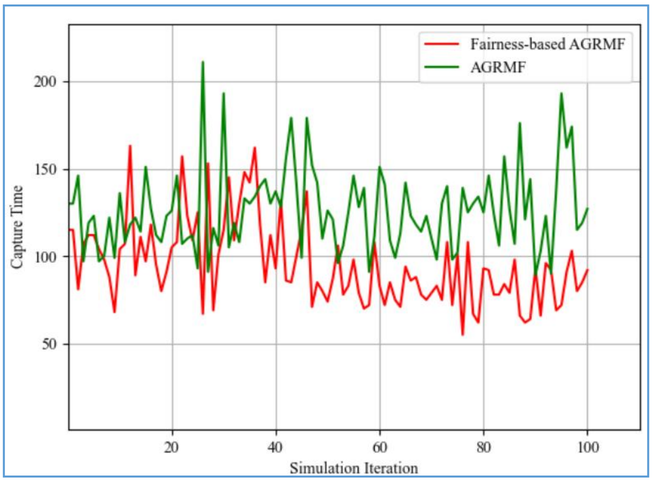


Figure 11. Capture time per run (Fast evader)

The previous chart shows the result that further solidifies our findings above, with a substantial difference of 31 episodes on average between our implementation of 95, and

AGRMF with an average of 126, an improvement of around 24.6%. The difference is more apparent in their fastest runs with our implementation and AGRMF having 55 and 89 respectively, the same narrative is repeated in their slowest runs with the fairness-based AGRMF scoring 163 as opposed to the 211 episodes that took AGRMF to complete its slowest run, further highlighting the improving our implementation provided. The average reward development with the fast evader is represented in Figure 12.

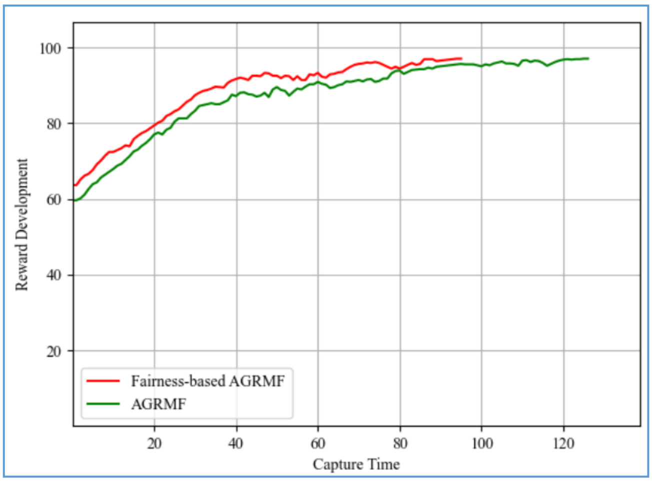


Figure 12. Average reward development (Fast evader)

This chart is also in line with previous results, with the fairness-based AGRMF permanently staying above AGRMF. Even though the charts here are a bit more erratic due to the inconsistency of speed between the pursuers and evader, it still defines a clear advantage for the coalitions that were formed in a fair manner, with the fairness-based AGRMF remaining on top for the its entire run time.

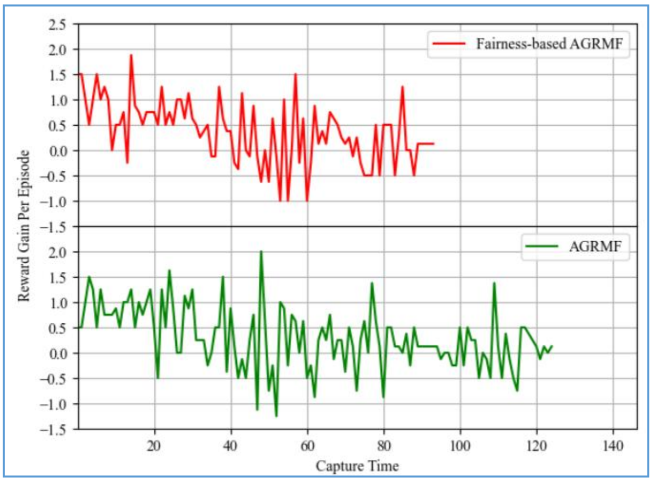


Figure 13. Average reward gain (Fast evader)

These charts are much more erratic compared to the previous ones (Figure 13), but we can still see that AGRMF has more reward losses with 18.25% of the instances or 28 times in the negative versus the 16.84% or 19 instances of the fairness-based implementation, giving a slight advantage to that implementation.

5.2.4 Result summary

Table 2 depicts the final results obtained from all the simulations, and it shows that the coalitions formed through the Fairness-based AGRMF brought an improvement of more than 20% proving that an even distribution of pursuers can dramatically increase the efficiency of the chase.

Table 2. Final results

Evader Type	Fairness-AGRMF	AGRMF
Dynamic evader	83 22.5%	107
Static evader	65 15.4%	77
Fast evader	95 24.6%	126

6. CONCLUSION

In this paper, we proposed a novel multi-agent coalition formation algorithm based on organizational principles. The proposed FBAGRMF coalition algorithm aims to provide equilibrium between the generated pursuit groups during the processing of the multi-pursuer muti-evader game. In comparison with AGRMF, FBAGRMF avoids the problem associated with negative externalities of the pursuers during the pursuit groups assignment. In this context, a negative externality concerns the case where a pursuer is belonging to a pursuit group but would be more efficient in another one. To showcase the improvement brought by the proposed coalition method, we have implemented a multi-pursuer multi-evader scenario based on ten agents that must form two pursuit groups of four evaders each one to capture two detected evaders. In order to generalize the studied cases, we have used two types of evaders, static and mobile. According to the obtained results, we concluded that FBAGRMF model improves the capturing time of the evaders as well as the development of the pursuers during the tasks processing. As mentioned in section 2 of this paper, it is very difficult to obtain optimal coalition formation during the processing of dynamic PE game. Thus, the main limitation of the proposed approach is that it does not take into account the case where the organizer of each pursuit group breaks down. Consequently, in future work, we will study the PE with considering the mentioned fault tolerance problem. Moreover, we will propose a novel multi-agent path planning based on deep-reinforcement learning [51], which is more adaptable with PE game with dynamic evaders.

REFERENCES

[1] Shakya, J., Chopin, M., Merghem-Boulahia, L. (2024). Dynamic coalition formation among IoT service providers: A systematic exploration of IoT dynamics using an agent-based model. Sensors, 24(11): 3471. https://doi.org/10.3390/s24113471

[2] Bezerra, L.C., Dos Santos, A.M., Park, S. (2025). Learning policies for dynamic coalition formation in multi-robot task allocation. IEEE Robotics and Automation Letters, 10(9): 9216-9223. https://doi.org/10.1109/LRA.2025.3592080

[3] Aripin, W.T., Haoxiang, X. (2025). Optimizing coalition formation in the global plastic waste trade using multi-agent reinforcement learning approach. Journal of Cleaner Production, 529: 146823. https://doi.org/10.1016/j.jclepro.2025.146823

[4] Souidi, M.E.H., Piao, S., Li, G., Chang, L. (2015). Coalition formation algorithm based on organization and Markov decision process for multi-player pursuit evasion. Multiagent and Grid Systems, 11(1): 1-13. https://doi.org/10.3233/MGS-150226

[5] Ferber, J., Gutknecht, O., Michel, F. (2003). From agents to organizations: An organizational view of multi-agent systems. In International Workshop on Agentoriented Software Engineering, Melbourne, Australia, pp. 214-230. https://doi.org/10.1007/978-3-540-24620-6_15

[6] Souidi, M.E.H., Songhao, P., Guo, L., Lin, C. (2016). Multi-agent cooperation pursuit based on an extension of AALAADIN organisational model. Journal of Experimental & Theoretical Artificial Intelligence, 28(6): 1075-1088. https://doi.org/10.1080/0952813X.2015.1056241

[7] Siam, A., Maamri, R., Sahnoun, Z. (2014). Fuzzy organization of self-adaptive agents based on software components. International Journal of Intelligent Information Technologies, 10(3): 36-56. https://doi.org/10.4018/ijiit.2014070103

[8] Souidi, M.E.H., Siam, A., Pei, Z. (2019). Multi-agent pursuit coalition formation based on a limited overlapping of the dynamic groups. Journal of Intelligent & Fuzzy Systems, 36(6): 5617-5629. https://doi.org/10.3233/JIFS-181471

[9] Souidi, M.E.H., Siam, A., Pei, Z., Piao, S. (2019). Multi-agent pursuit-evasion game based on organizational architecture. Journal of Computing and Information Technology, 27(1): 1-11. https://doi.org/10.20532/cit.2019.1004318

[10] Souidi, M.E.H., Maarouk, T.M., Ledmi, M., Ledmi, A., Rahab, H. (2023). Multi-pursuer multi-evader games based on dynamic elimination priorities of the dominated strategies. Journal of Computer and Systems Sciences International, 62(2): 398-411. https://doi.org/10.1134/S106423072302017X

[11] Ma, X., Lv, M., Duan, K., Xie, W., Yang, J., Zhang, W. (2025). Multi-player pursuit-evasion game with interaction constraints: A cooperative game theoretic approach based on coalition structure. IEEE Transactions on Automation Science and Engineering. https://doi.org/10.1109/TASE.2025.3581695

[12] Stone, P., Veloso, M. (2000). Multiagent systems: A survey from a machine learning perspective. Autonomous Robots, 8(3): 345-383. https://doi.org/10.1023/A:1008942012299

[13] Rocha, J., Boavida-Portugal, I., Gomes, E. (2017). Introductory chapter: Multi-agent systems. Multi-Agent Systems, 3-13. https://doi.org/10.5772/intechopen.70241

[14] Durfee, E.H., Lesser, V.R. (1989). Negotiating task decomposition and allocation using partial global planning. Distributed Artificial Intelligence, 229-243. https://doi.org/10.1016/B978-1-55860-092-8.50014-9

[15] Panait, L., Luke, S. (2005). Cooperative multi-agent learning: The state of the art. Autonomous Agents and Multi-agent Systems, 11(3): 387-434. https://doi.org/10.1007/s10458-005-2631-2

[16] Weiss, G. (1999). Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence. MIT Press.

[17] Rani, M., Nayak, R., Vyas, O.P. (2015). An ontology-based adaptive personalized e-learning system, assisted by software agents on cloud storage. Knowledge-Based Systems, 90: 33-48. https://doi.org/10.1016/j.knosys.2015.10.002

[18] Rizal, J., Gunawan, A.Y., Yosmar, S., Nuryaman, A. (2024). Seismicity pattern recognition in the Sumatra megathrust zone through mathematical modeling of the maximum earthquake magnitude using Gaussian mixture models. Mathematical Modelling of Engineering Problems, 11(5): 1179-1188. https://doi.org/10.18280/mmep.110506

[19] Kuppusamy, V., Gowrishankar, L. (2024). Performance evaluation of a M/G/1 queue model for patient flow in a health care system. Mathematical Modelling of Engineering Problems, 11(4): 863-871. https://doi.org/10.18280/mmep.110403

[20] Puterman, M.L. (1990). Markov decision processes. Handbooks in Operations Research and Management Science, 2: 331-434. https://doi.org/10.1016/S0927-0507(05)80172-0

[21] Feinberg, E.A., Shwartz, A. (2012). Handbook of Markov Decision Processes: Methods and Applications (Vol. 40). Springer Science & Business Media. https://doi.org/10.1007/978-1-4615-0805-2

[22] Jennings, N.R. (2000). On agent-based software engineering. Artificial Intelligence, 117(2): 277-296. https://doi.org/10.1016/S0004-3702(99)00107-1

[23] Odell, J.J., Dyke Parunak, H.V., Fleischer, M. (2002). The role of roles in designing effective agent organizations. In International Workshop on Software Engineering for Large-Scale Multi-agent Systems, pp. 27-38. https://doi.org/10.1007/3-540-35828-5_2

[24] Gutknecht, O., Ferber, J. (1998). A model for social structures in multi-agent systems. Technical Report, LIRMM/CNRS, Université Montpellier II.

[25] Gasser, L. (2001). Perspectives on organizations in multi-agent systems. ECCAI Advanced Course on Artificial Intelligence, 1-16. https://doi.org/10.1007/3-540-47745-4_1

[26] Horling, B., Lesser, V. (2004). A survey of multi-agent organizational paradigms. The Knowledge Engineering Review, 19(4): 281-316. https://doi.org/10.1017/S0269888905000317

[27] Barber, K.S., Martin, C.E. (2001). Dynamic reorganization of decision-making groups. In Proceedings of the Fifth International Conference on Autonomous Agents, Montreal Quebec, Canada, pp. 513-520. https://doi.org/10.1145/375735.376432

[28] Broek, E.L., Jonker, C.M., Sharpanskykh, A., Treur, J. (2005). Formal modeling and analysis of organizations. In International Conference on Autonomous Agents and Multiagent Systems, pp. 18-34. https://doi.org/10.1007/11775331_2

[29] Hübner, J.F., Vercouter, L., Boissier, O. (2008). Instrumenting multi-agent organisations with artifacts to support reputation processes. In International Workshop on Coordination, Organizations, Institutions, and Norms in Agent Systems, Chicago, USA, pp. 96-110. https://doi.org/10.1007/978-3-642-00443-8_7

[30] Burns, T., Stalker, G.M. (1969). The management of innovation. The Economic Journal, 79(314): 403-405. https://doi.org/10.2307/2230196

[31] Posey, Rollin B. (March). Modern organization theory edited by mason Haire. Administrative Science Quarterly 5(4): 609-611. https://doi.org/10.2307/2390625

[32] Hertz, D.B., Livingston, R.T. (1950). Contemporary

organizational theory: A review of current concepts and methods. Human Relations, 3(4): 373-394. https://doi.org/10.1177/001872675000300403

[33] Corkill, D.D., Lander, S.E. (1998). Diversity in agent organizations. Object Magazine, 8(4): 41-47.

[34] Mattern, F., Floerkemeier, C. (2010). From the Internet of computers to the Internet of Things. In From Active Data Management to Event-Based Systems and More, pp. 242-259. https://doi.org/10.1007/978-3-642-17226-7_15

[35] Rajkumar, R., Lee, I., Sha, L., Stankovic, J. (2010). Cyber-physical systems: The next computing revolution. In Design Automation Conference, Anaheim, California, pp. 731-736. https://doi.org/ 10.1145/1837274.1837461

[36] Momoh, J.A. (2012). Smart Grid: Fundamentals of Design and Analysis (Vol. 63). John Wiley & Sons. https://doi.org/10.1002/9781118156117

[37] Abbas, H.A. (2014). Future SCADA challenges and the promising solution: The agentbased SCADA. International Journal of Critical Infrastructures, 10(3-4): 307-333. https://doi.org/10.1504/IJCIS.2014.066354

[38] Saha, D., Mukherjee, A. (2003). Pervasive computing: A paradigm for the 21st century. Computer, 36(3): 25-31. https://doi.org/10.1109/MC.2003.1185214

[39] Friedewald, M., Raabe, O. (2011). Ubiquitous computing: An overview of technology impacts. Telematics and Informatics, 28(2): 55-65. https://doi.org/10.1016/j.tele.2010.09.001

[40] Mkadmi, A., Saleh, I. (2008). Bibliothèque Numérique et Recherche d'Informations. Hermès Science Publications: Lavoisier. https://doi.org/10.3917/docsi.473.0070c

[41] Akoka, J., Comyn-Wattiau, I. (2006) Encyclopédie de l'Informatique et des Systèmes d'Information. Vuibert, Paris.

[42] Rahwan, T., Jennings, N.R. (2007). An algorithm for distributing coalitional value calculations among cooperating agents. Artificial Intelligence, 171(8-9): 535-567. https://doi.org/10.1016/j.artint.2007.03.002

[43] Hannoun, M., Boissier, O., Sichman, J.S., Sayettat, C. (2000). MOISE: An organizational model for multi-agent systems. In Advances in Artificial Intelligence, SP, Brazil, pp. 156-165. https://doi.org/10.1007/3-540-44399-1_17

[44] Souidi, M.E.H., Songhao, P., Guo, L., Lin, C. (2016). Multi-agent cooperation pursuit based on an extension of AALAADIN organisational model. Journal of Experimental & Theoretical Artificial Intelligence, 28(6): 1075-1088. https://doi.org/10.1080/0952813X.2015.1056241

[45] Savall, M., Pécuchet, J., Chaignaud, N., Itmi, M. (2001). YAMAM–An organisation model for the multiagent systems. In Implementation in the Phoenix Platform, 3rd Francophone Conference of Modeling and Simulation "Conception, Analyze, Management of Industrial Systems" MOSIM.

[46] Yamashita, M., Umemoto, H., Suzuki, I., Kameda, T. (2001). Searching for mobile intruders in a polygonal region by a group of mobile searchers. Algorithmica, 31(2): 208-236. https://doi.org/10.1007/s00453-001-0045-3

[47] Alonso, L., Goldstein, A.S., Reingold, E.M. (1992). “Lion and man”: Upper and lower bounds. ORSA Journal on Computing, 4(4): 447-452. https://doi.org/10.1287/IJOC.4.4.447

[48] Wan, K., Wu, D., Zhai, Y., Li, B., Gao, X., Hu, Z. (2021). An improved approach towards multi-agent pursuit–evasion game decision-making using deep reinforcement learning. Entropy, 23(11): 1433. https://doi.org/10.3390/e23111433

[49] Salmon, J.L., Willey, L.C., Casbeer, D., Garcia, E., Moll, A.V. (2020). Single pursuer and two cooperative evaders in the border defense differential game. Journal of Aerospace Information Systems, 17(5): 229-239. https://doi.org/10.2514/1.I010766

[50] Wang, Y., Zhong, F., Xu, J., Wang, Y. (2021). Tom2c: Target-oriented multi-agent communication and cooperation with theory of mind. arXiv preprint arXiv:2111.09189. https://doi.org/10.48550/arXiv.2111.09189

[51] Wang, Y., Wang, Y., Tian, F., Ma, J., Jin, Q. (2025). Intelligent games meeting with multi-agent deep reinforcement learning: A comprehensive review. Artificial Intelligence Review, 58(6): 165. https://doi.org/10.1007/s10462-025-11166-1