

NEW GAME-THEORETIC CONVOLUTIONAL NEURAL NETWORK APPLIED FOR THE MULTI-PURSUER MULTI-EVADER GAME

Nabila SID, Meriem DJEZZAR, Mohammed El Habib SOUIDI,
Mounir HEMAM

ICOSI Laboratory

University of Khenchela, Algeria

*e-mail: {sid.nabila, meriem.djezzar, souidi.mohammed,
hemam.mounir}@univ-khenchela.dz*

Abstract. Pursuit-Evasion Game (PEG) can be defined as a set of agents known as pursuers, which cooperate with the aim forming dynamic coalitions to capture dynamic evader agents, while the evaders try to avoid this capture by moving in the environment according to specific velocities. The factor of capturing time was treated by various studies before, but remain the powerful tools used to satisfy this factor object of research. To improve the capturing time factor we proposed in this work a novel online decentralized coalition formation algorithm equipped with Convolutional Neural Network (CNN) and based on the Iterated Elimination of Dominated Strategies (IEDS). The coalition is formed such that the pursuer should learn at each iteration the approximator formation achieving the capture in the shortest time. The pursuer's learning process depends on the features extracted by CNN at each iteration. The proposed supervised technique is compared through simulation, with the IEDS algorithm, AGR algorithm. Simulation results show that the proposed learning technique outperform the IEDS algorithm and the AGR algorithm with respect to the learning time which represents an important factor in a chasing game.

Keywords: Multi-agent system, Pursuit-Evasion Game (PEG), Convolutional Neural network (CNN), coalition formation

1 INTRODUCTION

In multi-agent systems (MAS), connected autonomous agents act in a limited environment to achieve objectives or to maximize rewards. MAS is a distributed system based on a set of agents that interact in most cases. The interaction is effectuated according to the mode of coordination used. This coordination can be represented through competition, cooperation, or either a negotiation between the agents. MAS have been mostly processed through the use of machine learning principles. In this paper, we propose a Convolutional Neural Network (CNN) to predict the approximate coalition according to the changes of the elimination priority. The CNN is effective in our case. CNN used to minimize error and maximize the probabilities to extract the indexes corresponding to the optimal coalition which make a capture in considerable time. The CNN reflects its simplicity and usefulness to model such problems. CNN consists of a multilayer stack of neurons, mathematical functions with several adjustable parameters, which preprocess the coalition's features. CNN categorized this features obtained from IEDS technique to generate the data set of input layer. A selection between several activation functions in our case was the Maxout function activation. This function activation is based on the fact that if k scalar products are provided for a node, effectively this node can learn a local nonlinear activation function by approximating it with a piecewise linear function consisting of k intervals. several researchs treated this field. In [1], the authors proposed to calculate the decentralized optimal strategy under uncertain environment in MAS. The learning in this case maintains five neural networks for each agent. Other authors [2], depend on principle of complete perception of environment for each agent, when he choose to use a recurrent network to extract the feautres needed to make a partial perception of environment.

MAS has multiple uses in artificial intelligence, pursuit evasion and game theory in particular [3, 4, 5]. In the pursuit-evasion problem, there are many pursuit groups, each one contains a certain number of agents known as pursuers. Each pursuit group attempts to capture a specific evader in the shortest possible time. The multi-agent pursuit problem was processed by using several methods of coordination such as cooperation [6], MAS organizational models [7]. PEG was developed in the extensive research [8, 9, 10, 11, 12]. During the pursuit, the pursuers have to cooperate to form coalitions or pursuit groups. Coalitions are formed to capture evaders and dissolve after the task processing.

Game theory studies the options of autonomous agents as well as their consequences during the interactions. The most recent game-theoretic principle used to form pursuit coalition formation was the Iterated Elimination of Dominated Strategies model (IEDS). During each pursuit iteration in the IEDS algorithm [13], the pursuers follow a specific grouping strategy known as a coalition. Each pursuer aims to be a part of the coalition returning the maximum value extracted by the IEDS algorithm knowing that the coalitions of the pursuers are generated using static elimination priority.

IEDS was used in combination with Markov decision process (MDP) that allows the displacement of the agents. MDP is a mathematical framework used for modeling decision-making problems, where the outcomes are partially random and partially controllable [14].

In this paper, we propose a machine learning method, which used to predict the best coalition according to the changes of the elimination priority. In this case, the most appropriate machine learning method to utilize is Convolutional Neural Network (CNN). This latter, is a universal approximator for continuous functions within a bounded domain which is considered as a sub-class of neural networks, used to minimize error and maximize the probabilities in chasing game. The CNN reflects its simplicity and usefulness to model such problems. CNN consists of a multilayer stack of neurons, mathematical functions with several adjustable parameters, which preprocess small amounts of information. These CNN can categorize information extracted from IEDS algorithm to generate the data set of input layer. A selection between several activation functions in this kind of neural network is the Maxout function activation. This function activation is based on the fact that if k scalar products are provided for a node, effectively this node can learn a local non linear activation function by approximating it with a piecewise linear function consisting of k intervals.

A nother work proves the effectiveness of multiple maxout activation function variants on 18 datasets using Convolutional Neural Networks [15]. In [16], Cai and Liu combined maxout neurons with convolutional neural network (CNN) and the long short-term memory (LSTM) recurrent neural network (RNN).

In this work, we used the maxout to predict the maximum value used to select the optimal coalition formation in the shortest time where the selection of agents is effectue randomly, unlike IEDS algorithm wich given a previously priorities to agents to make coalitions, each coalition corresponds to a particular strategy. An elimination of dominated strategies is applied to extract the optimal coalition corresponding to the maximum value.

The structure of this paper is as follows: We discuss related research in Section 2. Section 3 describes the multi-agent pursuit problem and clarifies the principal characteristics of pursuers and evaders. In Section 4, we describe our proposed framework for pursuit MAS game equiped with CNN. The different steps of the proposed coalition formation algorithm are detailed in Section 5. Section 6 is dedicated to the presentation of experimental results. Finally, Section 7 concludes and provides directions for future work.

2 RELATED WORK

Lately, the pursuit-evasion problem has attracted the attention of many research activities in the area of multi-agent systems. Someof them depends on a new form of probability density function (PDF) to solve the optimal pursuit-evasion strategies and a neural network to estimates an optimal control, and approximates the optimal

cost function [17]. In another work, they designed an algorithm based on relaxation problem to estimate future states of the game by introducing a polynomial-time algorithm to control action selection in visibility [18].

Tian et al. [19], established a hierarchical evasion strategy for the Air-breathing Hypersonic Vehicles (AHVs). Specifically, the authors presented multiple cooperative pursuers in the case where they come from different directions concurrently. To achieve successful evasion, successive pursuers come from the same direction and flight with proper spacing. Others preferred to investigate the interaction between two antagonistic agents in an environment without obstacles [20]. Another interesting work [21] is based on the selection of an intelligent evader to be hunted by a group of pursuers by retreating horizon control policies.

Among the most processed problems in a multi-agent system, we can cite the Coalition Formation. Recently, in [22] Guo et al. have proposed a genetic algorithm with heuristic initialization and repair strategy (GAHIR) to solve coalition formation problem, they treated the problem as a single-task single-coalition formation, a multi-task single-coalition formation, as well as a multi-task multi-coalition formation. In the same issue, we find another research activity [23] that focuses on organizational hierarchies of coalition formation structures.

Furthermore, in [24] Estrada et al. addressed the problem of task allocation in Mobile Crowd Sensing (MCS) by forming tasks publisher coalition taking into consideration workers' route preferences. On the other hand, Babu and Chitnis [25] introduced the utility function to achieve the optimal coalition among nodes.

In comparison with other research activities, processing the same problem, Souidi et al. [26] introduced a coalition formation algorithm based on Agent-Group-Role organizational membership function model (AGRMF), which is an extension of agent-group-role (AGR) model. In this model, a group is considered as a fuzzy set where the goal always remains to optimize a coalition of agents in order to capture an evader in the shortest time. In another work [27], Cruz et al. have used reinforcement learning principles and smoothing techniques to modify the path planning of the agents in an unknown environment. These modifications have for objective to predict the greedy actions of pursuers.

In AGR organizational model [28], each agent can play one or more roles simultaneously. Each agent can be a member of one or more groups in the same time. A class of agents is determined by its task that is contributed to each entity of agents. The agents can effectuate a set of operations in the same group such as communication, cooperation, and negotiation. Benoudina et al. [29] used a multi-agent platform to build a simulator based on reactive agents capable to transform this complex system into a data processing program that can represent its structure, its communication, its behavior, its control loops and verify the integrity and its proper functioning.

Boudjidj et al. [30] have introduced a new proposal that consists on a transformation of Agent-Group-Role (AGR) organizational model in a categorical way, which permits the analysis, the verification, and the validation of the organization at a high level of abstraction.

Qadir et al. [31] affirmed, that in AGR model, there is no mechanism defining the access conditions regarding the groups. Consequently, in AGRMF model, they used a binary variable instead of logic fuzzy set called degree of membership. This degree of membership function controls whether the agents will take the role. A membership function interpreted within some parameters the options of an agent to undertake a role.

In [32], the authors used concept of AGR model to represent a liquefied natural gas treatment process (decarbonation) at a gas plant in Algeria. Moreover in [30], the authors proposed a transformation of Agent-Group-Role (AGR) organizational model in a categorical way in order to obtain a formal semantics model describing the MAS organization, which allows a high level of abstraction.

MDP is used in several domains, such in [33], the author devises vehicle trajectories by coupling a locally-optimal motion planner with a Markov decision process (MDP) model that can capture network-level information.

Recently, many research activities have taken intense interests of the features in neural networks.

Other researches, such as [34], have taken into consideration the performance of hyperparameter optimization of Deep CNNs by adapting Q-learning and defining learning agents per layer to split the design space into independent smaller design sub-spaces. Consequently, each agent can fine-tune the hyperparameters of the assigned layer concerning a global reward. A combination of graphic convolutional neural network with deep Q-network has been used in [35] to form an innovative graphic convolution Q network, that serves as the information fusion module and decision processor in multi-agent cooperative control of connected autonomous vehicles.

3 PROBLEM STATEMENT

According to Schenato et al. [35], the pursuit-evasion game is defined as “*a mathematical abstraction arising from numerous situations, which address the problem of controlling a swarm of autonomous agents in the pursuit of one or more evaders*”.

Much research in the pursuit evasion field focuses on the capture of evaders neglecting the factor of time. In our work, achieving the minimum possible capture time was our interest.

For this purpose, we depended on a supervised learning which benefits from exploiting the coalitions categorized by the IEDS technique with a convolutional neural network so that the MAS be able to self learn its forming coalition by adjusting its parameters (i.e. the weights of the neurons), so as to reduce the difference between the capture time obtained and the expected capture time.

The margin of error is thus reduced over the training processs, with the aim of being able to generalize one’s learning to new cases. The weak point of this method is that it does not give all its predictive capacity when the input data are small. In other words, this technique gives good results as long as we have hyper data sets.

The point distinguishing this approach from the IEDS, is the implementation of an online prediction mechanism based on CNN.

Our work will depend on the performance provided by CNN using supervised learning. We depend on supervised learning or associative learning in which the network is trained by providing it with input and matching output patterns. Unlike the IEDS technique which selected the agents with priority, in our case the agents are selected randomly without priority to form the groups. We proposed two different types of evader, dynamic and static evader.

4 THE PROPOSED APPROACH

Forming efficient coalitions is one of the major research challenges in the area of multi-agent systems. In coalition formation, coherent sets of distinct, autonomous agents, interact to accomplish their individual or collective goals. A pursuit coalition begins with a task and dismisses when it is accomplished. A set of coalition extracted by IEDS technique is denoted by $S = \{c_1, c_2, \dots, c_m\}$.

In order to decrease the communication cost and avoid repeated information in interactions, we based ourselves on the performance of CNN. The latter, uses the process of backpropagation to adjust the weights of neuron. A CNN has to be configured such that it can approximate at each iteration the optimal coalition which verify the condition: $(c_i, index_i) > (c_j, index_j)$. In other words, only the coalition having the maximum index, extracted by CNN, can be combined to the lowest number of iterations, which means the shortest capturing time.

The agents playing the role *Pursuers* are denoted by $P = \{p_1, p_2, \dots, p_n\}$. In addition, the *Evaders* are denoted by $E = \{e_1, e_2, \dots, e_m\}$. The main goal of the pursuit-evasion game is to perform the capture of each evader by a group of pursuers in a finite time. Each evader is characterized by a degree of difficulty denoted D , $D = \{d_1, d_2, \dots, d_n\}$, which represents the number of pursuers needed in a given pursuit. An evader e is defined by a type Re , with $Re \in \{I, II, III, IV\}$. This latter allows specifying how many pursuers are required to achieve the capture of the evader e .

The Pursuers achieving the capture of an evader e will get a reward equal to Re . The reward is provided to pursuers achieving the performance of the capture. The stability degree represents the roles' reattribution in a given coalition. It is assumed that all players have the same motion velocity (i.e. one cell per iteration), a stable communication system, and a partial vision of the pursuit region. In a bounded environment containing obstacles that pursuers must avoid, we consider the existence of eight (08) pursuers and two (02) evaders.

We perform our pursuit-evasion game in a rectangular two-dimensional grid with 100×100 cells. The agents and obstacles are located randomly in the grid of cells. Each cell corresponds to a specific state.

MDP application aims to compute the possibilities of the next state according to actual ones. This computation is effectuated in order to allow the pursuers' movements according to the detected reward.

Each neuron has its weight. To speed up the generation weights at the nodes, we used a random number generator. These numbers form a sequence of independent and uniformly distributed random variables on $[0, 1]$. This sequence is interpreted as the realization of a random variable which follows the law of uniform density on $[0, 1]$.

$$\mu_{i+1} = \mathcal{L}(\mu_1, \mu_{i-1}, \dots, \mu_{i-k}), \quad k < i, \quad i = 1, 2, \dots, n, \tag{1}$$

where $\mu_{i+1} = \mathcal{L}(\mu_i)$, \mathcal{L} has value $\in [0, 1]$.

The multi-agent system is used to learn new behaviors such that the natural systems exploit its performance in approximation method [34]. We intend to train our system to learn how to predict the optimal coalition making the best shortest time for a multi-agent pursuit game. This inspired us to believe that extracting for each coalition, the maximum average of reward and the stability degree as a input data can optimize the training process.

We assemble our training Dataset D_t through IEDS technique which resulting Reward (x_i) and stability degree (y_i) in each given coalition. Due to of large coalitions generated at each iteration, we are going to exploit the convolution layer in CNN. This later will contain samples of these coalitions in a vector to extract the maximum from its maxout units.

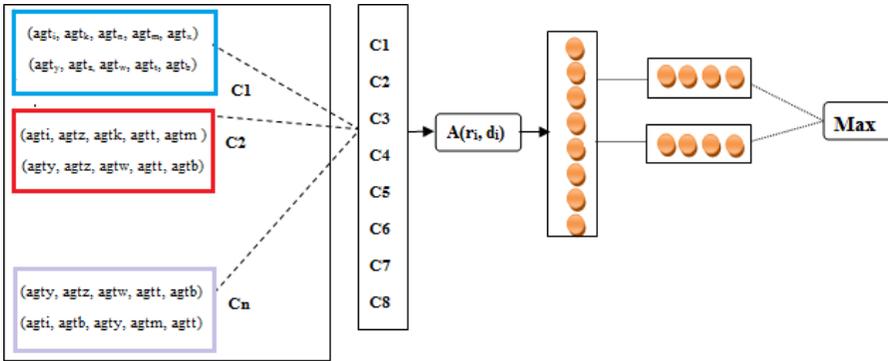


Figure 1. The CNN with tow unit maxout. A part of coalitions formation generated by IEDS technique are ranged in a vector. An average is extracted for each coalition. The tow Maxout unit calculated the maximum for each given coalition.

We use $(x_i, y_i) \in D, \forall i \in [1, n]$.

$$D_t = \begin{pmatrix} x_1 & y_1 \\ x_2 & y_2 \\ \dots & \dots \\ x_n & y_n \end{pmatrix}, \tag{2}$$

where $x_i \in R^d$ is a d -dimensional sample, with each dimension corresponding to a particular value of the reward, and $y_i \in R^1$ is the stability degree.

4.1 Training of CNN

The challenge is to train the CNN by feeding it teaching patterns and letting it change its weights according to output to achieve a desired capture time. First, the calculation of the weighted product of the inputs (h_i) is effectuated according to the following expression:

$$\forall i \in [1, n] : h_i = (w_i \times z_i). \quad (3)$$

To normalize this product and avoid a drastically different range of values, we use what is called an activation function. An activation function transforms these values into values between $[0, 1]$ or $[-1, 1]$ to make the whole process statistically balanced. From this value, a transfer function calculates the value of the state of the neuron. This value will be transmitted to downstream neurons. There are many possible forms concerning the transfer function. Most of transfer functions are continuous, offering an infinity of possible values included in the interval $[0, +1]$ (or $[-1, +1]$).

4.2 Supervised Convolutional Neural Network Model (IEDSNN)

A novel model of supervised learning based Convolutional Neural Network is established. A supervised training is used to update weights of the network until the obtainment of minimum error. The error function will serve in constructing efficient supervised training algorithms to accelerate the learning process.

4.3 Supervised Learning

In supervised learning, the data entering the process are already categorized according to IEDS algorithm, which the proposed algorithm must use to predict an approximated outcome. Our algorithm will learn the input to output mapping function:

$$H = f(Z). \quad (4)$$

The goal is to understand the mapping function. In other words, when new input data (z_i) are introduced, we can predict the output variables (h) for that data. In supervised learning, the robustness of the algorithm will depend on the precision of its training. A supervised content learning algorithm produces an internal map that allows its reuse to classify new amounts of data.

Data preprocessing. This step allows calculating the input values, which are ranged in a matrix of two dimensions. Each dimension reflects a result extracted by IEDS algorithm. The average is calculated for all lines in the matrix. Each average will form a new value as input values of our input layer.

An average Z_i of two dimensions has been placed to represent the Input of our CNN:

$$\forall i \in [1, n] : Z_i = \frac{(x_i + y_i)}{2}. \quad (5)$$

Pre-training. Training the CNN requires specifying an initial value for the weights.

A well-chosen initialization method will make learning easier. A distribution of random values is used to potential weights, we assign a constant number to all the weights. The constants numbers are in the range of $[-1, 1]$. The purpose of the random weight initialization is to break the symmetry. However, since the weights are no longer symmetrical, we can safely initialize all bias values with the same value. A well-chosen initialization can accelerate the convergence of the gradient descent, increasethe chance of gradient descent converging to lower training (and generalization) error.

Consequently, the parameters to initialize in our convolutional neural network are:

- Weight matrices

$$(W[1], W[2], W[3], \dots, W[L-1], W[L])(Z[1], Z[2], Z[3], \dots, Z[L-1], Z[L]).$$

- The bias vectors

$$(b[1], b[2], b[3]).$$

4.4 Function Activation Maxout

An activation function is a mathematical function used on a signal. It will replicate the activation potential found in the field of human brain biology. Moreover, it will allow the passage of information or not if the stimulation threshold is reached. Concretely, its role will be to predict whether or not to activate a neuron response. This prediction function that the CNN must learn is highly nonlinear. A neuron will only perform the following function:

$$z_i = \sum_{i=1}^n (input_i \times weight_i) + bias_i. \quad (6)$$

The Maxout activation function is chosen between severals activation function to capture the underlying nonlinearity. The Maxout activation function is a generalization of the Relu function [16].

The Maxout activation function is the most appropriate in our case. It is a piecewise linear function that returns the maximum of the inputs. Maxout activation function provides better optimization performance despite Castaneda et al. [21] are seen that in theory, a large number of extra parameters introduced by the k linear functions of each hidden Maxout unit result in large RAM storage memory cost and

considerable increase in training time, which affect the training efficient of very deep CNNs.

In general, a Maxout activation function is defined as follows:

$$h_i(x) = \max_{j \in [1,k]}(x_{ij}), \tag{7}$$

where

$$x_{ij} = x^T \times w_{ij} + b_{ij}. \tag{8}$$

$w \in R^{d \times m \times k}$ and $b_{ij} \in R^{m \times k}$ are the learned parameters, where m is the number of hidden units, d is the size of input vector and k represents the number of linear models. This nonlinearity can also be viewed as a feature selection process [24].

In our works, at each iteration by IEDS technique, a set of coalitions was generated resulting at the end of iteration a reward and degree of the maxout unit implements the following function:

$$H(x) = \max(\text{unit}_1, \text{unit}_2), \tag{9}$$

$$H(x) = \max(\max(W_1 \times z_1 + b_1, \dots, W_n \times z_n + b_1), \tag{10}$$

$$\max(Z_1 \times z_1 + b_2, \dots, Z_n \times z_n + b_2)). \tag{11}$$

The maxout-node applies n different scalar products to k offsets (b_1, b_2) and finally takes the maximum of these n values. Such model will estimate the optimum coalition formation as follows:

$$h_i = CNN((W_i, w_i), (Z_i, z_i)). \tag{12}$$

4.5 Error Function

Error function is used to determine the performance of a neural network during learning. The derivative of the error function is used by iterative learning algorithms. We have the squared error such that:

$$(I_{p,k})^2 = |d_{p,k} - o_{p,k}|^2, \tag{13}$$

where p is the p^{th} form, d is the desired value, o is the obtained value.

We then seek W such that W has to be minimized. Descending gradient method (generalized delta rule) has been used. So, the weight W must change in the same direction as $(-\partial E / \partial W)$.

4.6 Training Process

Learning consists of training the convolutional neural network (CNN) to predict the approximate the shortest time needed to make a shortest time in capture. Tow groups of pursuers chasing tow evaders. During this chase, a set of parameters was

extracted by IEDDS technique. This pramettters calculated for each given coalition are the number of changing role, wich represent the degree of stability of agents, number of iteration, reward.

For each coalition an average Zi was calculated for all iterations IEDS technique and ranged in a list. This list represent the input data for our CNN. The convolutionel layer will take at each time t a sample of this list and divided it into two lists: list1 [Z1 Z2 Z3 Z4] and list2 [Z5 Z6 Z7 Z8].

The Maxout unit will contain each given list. At this stage, an initialization step means to intialize the weights randomly, bias, threshold, learning rate. For each unit Maxout we extract the maximum value wich present the output: $h = \text{Max}(\text{Maxout unit1}, \text{Maxout unit2})$.

Unlike IEDS technique which is based on static percentages in the calculation of maximum value. The output value will serve to compare the coalition having the maximum couple of reward and degree of stability with the actual maximum wich used in capturing process.

We attempt to modify those weights according to the desired value with back-propagation algorithm. This algorithm is of the *online* type, when the weights are updated for each learning sample introduced to the neural network. Initially, the training process will propagate forward the inputs until obtaining an input calculated with the CNN. The second step consists of similitude between desired and calculated outcome. We adjust the weights such that in the next iteration the error must be minimized.

We propagate the signal forward in the layers of the CNN: $x_k^{(n-1)} \rightarrow x_j^{(n)}$

$$x_j^{(n)} = g^{(n)} \times v_j^{(n)} = g^{(n)} \times \sum_k w_{jk}^{(n)} x_k^{(n)}, \tag{14}$$

where g is the activation function Maxout and v_j is the agregation function.

Once the propagation is done we result our output value y . We can calculate the error between the y given by our CNN and the desired value t_i .

$$e_i^{output} = g(v_i^{output}[t_i - y_i]). \tag{15}$$

The weights updated as follow:

$$\Delta W_{ij}^{(n)} = e_j^{(n)} z_j^{(n-1)} \alpha, \tag{16}$$

where α is the learning rate, ($0 < \alpha \leq 1$).

$$W_{ij} = W_{ij} + \Delta W_{ij}. \tag{17}$$

5 IEDSNN ALGORITHM

The decentralized coalition formation algorithm, that we proposed, is an extension of the Iterated Elimination of Dominated Strategies (IEDS) and equipped with

Convolutional Neural Network (CNN) to form dynamic pursuit groups. A potential coalitions are formed depending on performances of CNN. The process begins with set of agents making the role of pursuers selected randomly. For each coalition we extract the stability degree and reward. A prediction of maximum indicators of optimal coalition formation have been developed by CNN. In order to generate the coalition making the capture of evaders in the shortest time (lowest number of iteration), we implement the pseudo-code Algorithm 1.

Algorithm 1 IEDSNN

Input:

– t : the vector of features extracted from coalitions generated by IEDS technique

Output:

– Max predicted value leading to detect optimal coalition making a shortest chase

Begin

Lanch_Chase();

Calculate_Desired_Max();

while ($C_{life} > 0$) and not obstacle) **do**

 Pursuit_Iteration ();

 extraction_Features ();

 Initializing indicators of CNN();

for $i \leftarrow 1$ to m **do**

$Inputnode_i \leftarrow T[i]$;

end for

 Calculate_Error();

while (Max calculated by CNN \leq Max desired) **do**

 Calculate_Max();

 Update_Weights();

 Propagate_Signal_Forward();

 Propagate_Error_Forward();

end while;

 index \leftarrow Max;

 Extraction(coalition, index);

end while

A description for IEDSNN algorithm is summarized, in Figure 2, by a flowchart, when it started by localization of agents in a pursuit closed environment with 100×100 cells, 08 pursuers, and 2 evaders needing each one 4 pursuers to be blocked. By the fact that we extended our proposal of IEDS approach, we have depended on stability degree and reward value extracted from each coalition as a training data for CNN. After the training step using IEDS algorithm which is considered as a preliminary outcome, the IEDSNN starts training the pursuers to make a fast capture of evaders in the shortest time.

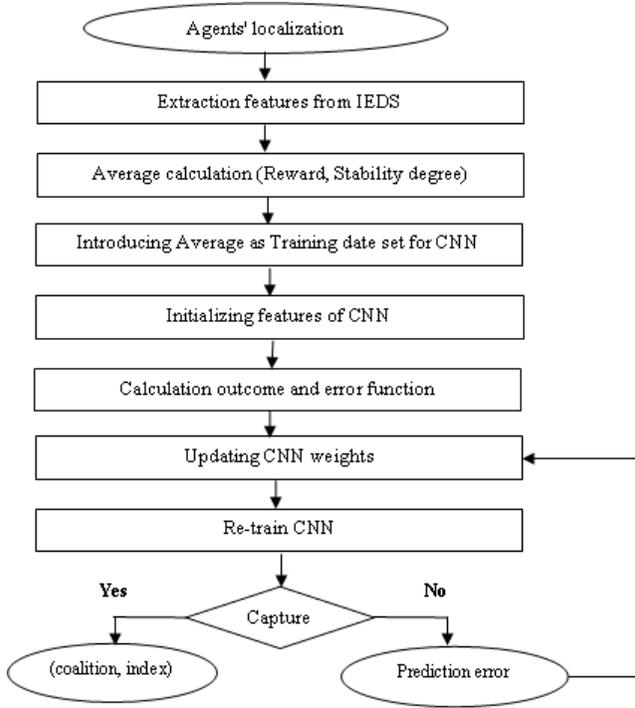


Figure 2. Flow chart of IEDSNN Algorithm

The process begins with the manual identification of the max value of the input vector; the vector containing the computation of the averages. This training iteratively decrease the error function which forces the algorithm to re-train the IEDSNN to finally extract the coalitions that have the highest contribution values to MAS efficiency.

First, the calculated averages are entered as input data. The phase of initialization of the CNN indicators took place. The training of the agents to form an appropriate coalition will be repeated as long as the least error function value is not yet achieved, implying that the expected coalition has not formed yet. Among all the coalitions carried out, the CNN extract the maximum value from each single unit which corresponds to any coalition. The training leads to perform an update of the weights always using the value of the error function obtained in each iteration.

Finally, the training will stop when the predicted value is achieved, extracting in return the optimal coalition making the best chase in the shortest time. A several episode of chasing have been developed by this process, at each episode the number of coalition making the quick capture decrease as mentioned in Table 1. This decrease of number of coalition is definitive proof of the success of our proposal in anticipation of optimal coalition formation of agents in a given chase.

6 EXPERIMENTAL RESULTS

The operating flow of our CNN algorithm is summarized from the moment of recovering the computations values of the algorithm IEDS until the capture in a considerable time. Here we investigate the performance of the proposed algorithm for eight (08) pursuers and two (02) evaders of type $R_e = IV$.

IEDSNN algorithm exploits the performance of CNN specifically the activation function Maxout which reduces extracting the minimum time in a pursuit game.

Table 1 indicates the average capturing time as well as the average obtained payoff per pursuit iteration regarding the three compared approaches, AGR [26], IEDS [13], and the proposed IEDSNN.

	AGR	IEDS	IEDSNN
Average capturing time (iteration)	100.33	78.5	64.16
Average pursuers' rewards obtained by iteration	0.34	0.36	0.45

Table 1. Pursuit result

The new coordination mechanism applied imposes an equitable sharing of the tasks between the different pursuers. The showcased results reveal the crucial difference between them through the distinctive decrease in average capturing time of the IEDSNN approach which only makes 64.16 iterations in comparison with AGR (100.33 iterations) and IEDS (78.5 iterations). Our proposal, according to Table 1, shows the fast prediction of optimal coalition to realize a speed pursuit capture in comparison with the other approaches.

Figure 3 represents the pursuit capturing times obtained in 30 pursuit episodes regarding the three compared cases. A learning of IEDSNN approach during the pursuit of agents is shown by the average capturing time achieved.

Figure 3 reflects the ability of pursuers to get learned how to form optimal coalition formation that leads to capturing the evaders in considered time. The number of coalitions established by the pursuers at the first time is increased knowing that the system at the beginning is not learned. The first phase consists on depending on IEDS algorithm computations. Extracting the first chase with the values needed in IEDSNN algorithm as a pre-training phase in the process of training the whole IEDSNN. Then the pursuers started learning the flow of IEDSNN. The pursuers learn quickly in forming optimal coalitions which, in turn, leads to effectuate a quick pursuit capture. This is why the curve decreases from 99 iterations to 52 iterations and remained on average in the range of 60 to 52 iterations. This fact represents the main contribution of the proposed IEDSNN.

The main results, indicated in Figure 4, reflect the efficiency of IEDSNN algorithm regarding the average reward development and obtained per iteration as plotted in cube Figure 4 and Figure 5 in the three compared cases. These results prove the efficiency and robustness of our approach concerning the progress in reaching the maximum value of reward. This increase of reward development explains

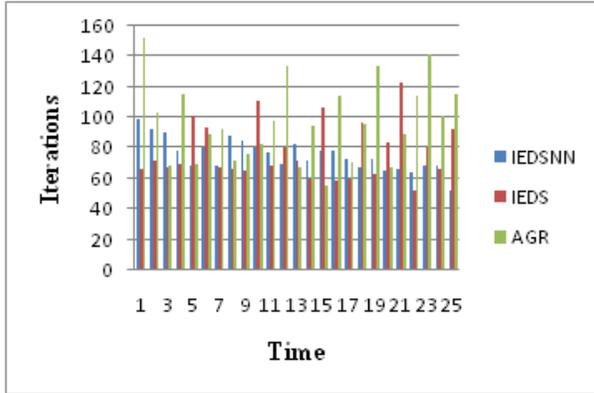


Figure 3. Average capturing time after 30 pursuits

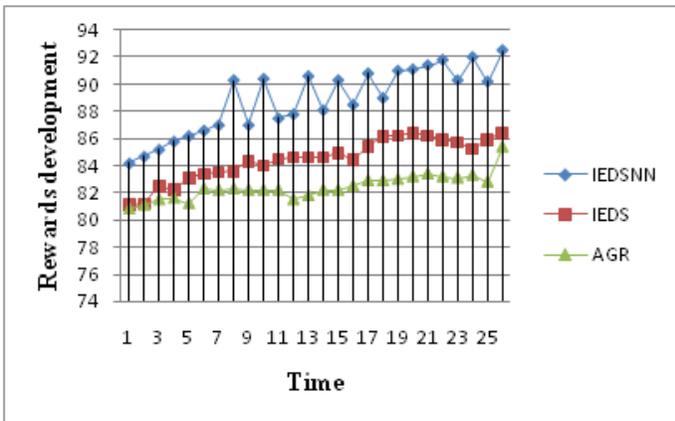


Figure 4. The pursuers' rewards development

clearly the training procedure of the pursuers in forming the appropriate coalitions attempt to make the shortest approximate time in capturing time.

The stability degree of the changing role of pursuers is shown in Figure 6, when we denote that the pursuers stop changing roles for the first 30 iterations and this is a well-proof of well learning process in our CNN.

To enrich our study, we use a static evader to understand the impact of the type of evader on the dynamics of the model. In this case, the evader only moves during the first iteration and stills static among the whole pursuit game process.

Figure 7 reflects the changes regarding the average capturing time using static evaders in comparison with the dynamic evaders used in the previous experiments. The significant advantage of the proposed method is the fast learning concern-

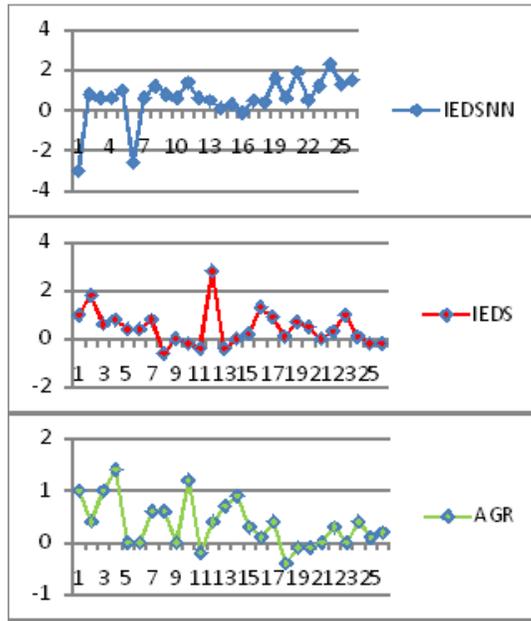


Figure 5. Average pursuers' reward per iteration

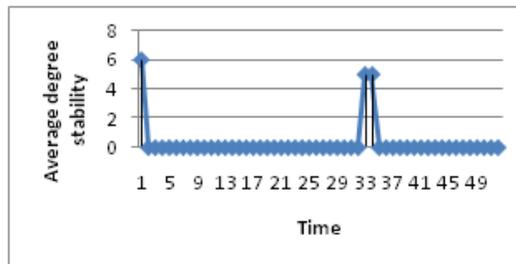


Figure 6. Average degree stability

ing the formation of an optimal coalition making a speed capture than depending on dynamic evader. The results prove that the type of evader affects the overall learning behavior of the simulation. More broadly, these results indicate that IEDSNN is a promising approach for well prediction that makes a speed capture.

From Figures 8 and 9, it can be seen that there is a proportional relationship between the number of iterations performed and the reward obtained for each iteration. When the system starts to well learn the process of chasing, it makes

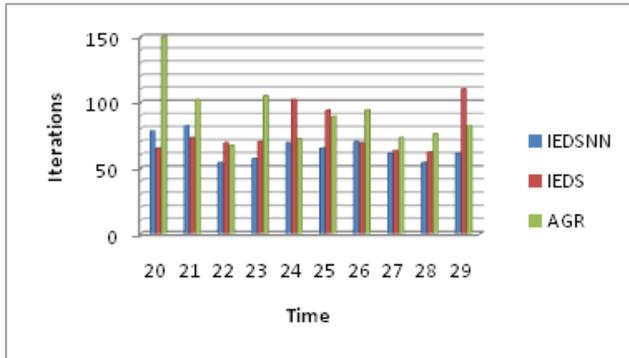


Figure 7. Average capturing time after 30 pursuits

a decrease of 17 in average capturing time which leads to the increase in average reward obtained in a given iteration.

Figure 9 shows the adequate dynamism degree of the roles' changes provided by the new proposal. This reduction, in comparison with the previous model using dynamic evaders, reflects the efficiency of CNN with static evaders. These results confirm the influence of the type of agent in PEG.

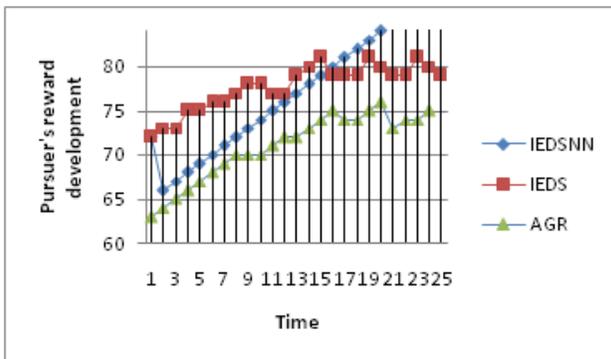


Figure 8. The pursuers' rewards development

7 CONCLUSION

In this paper, we have proposed a new pursuit coalition formation algorithm based on IEDS techniques as well as convolutional neural networks to allow the dynamic grouping of the implied pursuers. The used principles aim to provide appropriate coalition in accordance with the temporal constraints by accelerating the capture

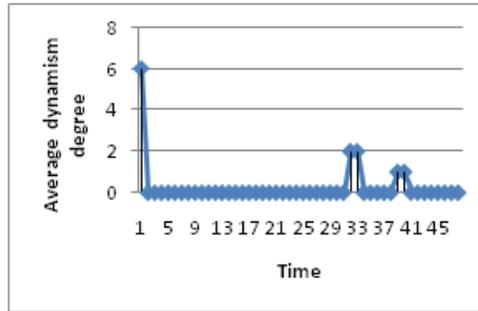


Figure 9. The average dynamism degree

process the detected evaders. From the used learning process, we can easily constate that the pursuers discover an approximated coalition in which they are able to be adapted to its changes after several experiments characterized by different pursuers' behaviors. This fact proves that IEDSNN exploits the principles of both convolutional neural networks and IEDS technique in the pursuit MAS game. To emphasize usefulness of our approach with, we effectuated a comparison study with a decentralized strategy of coalition based on AGR organizational model as well as IEDS algorithm. From the experimental results, we can deduce that this approach improves the pursuit capturing time, the coalition stability, as well as the payoff acquiring performed by the pursuers during the pursuit in comparison with the recent proposed approaches.

A solid foundation in this research is laid which can extend interesting other views in this kind of pursuit-evasion multi-agent game. For future work, we plan to use the learned representations (i.e. average capturing time, average rewards obtained, degree of stability), through our best CNN prediction system, to study the provided performance in different pursuit cases such as agent speeds and environment type. We will propose to use this information at the time of learning in order to learn all the possible coalition formations allowing a quick capture. We will also study the influence of the type of agent and the environment kind on the quality of performance prediction systems.

REFERENCES

- [1] ZHOU, Z.—XU, H.: Mean Field Game and Decentralized Intelligent Adaptive Pursuit Evasion Strategy for Massive Multi-Agent System Under Uncertain Environment. 2020 American Control Conference (ACC), IEEE, 2020, pp. 5382–5387, doi: 10.23919/ACC45564.2020.9147659.
- [2] ZHANG, L.—LI, J.—ZHU, Y.—SHI, H.—HWANG, K. S.: Multi-Agent Reinforcement Learning by the Actor-Critic Model with an Attention Interface. *Neurocomputing*, Vol. 471, 2022, pp. 275–284, doi: 10.1016/j.neucom.2021.06.049.

- [3] ZHOU, Z.—XU, H.: Decentralized Optimal Large Scale Multi-Player Pursuit-Evasion Strategies: A Mean Field Game Approach with Reinforcement Learning. *Neurocomputing*, Vol. 484, 2022, pp. 46–58, doi: 10.1016/j.neucom.2021.01.141.
- [4] LI, Z. Y.—ZHU, H.—LUO, Y. Z.: An Escape Strategy in Orbital Pursuit-Evasion Games with Incomplete Information. *Science China Technological Sciences*, Vol. 64, 2021, No. 3, pp. 559–570, doi: 10.1007/s11431-020-1662-0.
- [5] WANG, Y. F.: A Pursuit Evasion Game Approach to Obstacle Avoidance. Master Thesis. University of Waterloo, 2021.
- [6] TAN, X.—ZHOU, L.—WANG, H.—SUN, Y.—ZHAO, H.—SEET, B. C.—WEI, J.—LEUNG, V. C. M.: Cooperative Multi-Agent Reinforcement-Learning-Based Distributed Dynamic Spectrum Access in Cognitive Radio Networks. *IEEE Internet of Things Journal*, Vol. 9, 2022, No. 19, pp. 19477–19488, doi: 10.1109/JIOT.2022.3168296.
- [7] XU, Y.—YANG, H.—JIANG, B.—POLYCARPOU, M. M.: Multiplayer Pursuit-Evasion Differential Games with Malicious Pursuers. *IEEE Transactions on Automatic Control*, Vol. 67, 2022, No. 9, pp. 4939–4946, doi: 10.1109/TAC.2022.3168430.
- [8] CHIPADE, V. S.: Collaborative Task Allocation and Motion Planning for Multi-Agent Systems in the Presence of Adversaries. Ph.D. Thesis. University of Michigan, 2022.
- [9] LI, Z.—ZHU, H.—YANG, Z.—LUO, Y.: A Dimension-Reduction Solution of Free-Time Differential Games for Spacecraft Pursuit-Evasion. *Acta Astronautica*, Vol. 163, Part B, 2019, pp. 201–210, doi: 10.1016/j.actaastro.2019.01.011.
- [10] WANG, W.—LI, P.: Planning and Formulations in Pursuit-Evasion: Keep-Away Games and Their Strategies. *CoRR*, 2022, doi: 10.48550/arXiv.2206.08318.
- [11] BRAVO, L.—RUIZ, U.—MURRIETA-CID, R.: A Pursuit-Evasion Game Between Two Identical Differential Drive Robots. *Journal of the Franklin Institute*, Vol. 357, 2020, No. 10, pp. 5773–5808, doi: 10.1016/j.jfranklin.2020.03.009.
- [12] LIU, M.—ZHANG, J.—SHANG, M.: Real-Time Cooperative Kinematic Control for Multiple Robots in Distributed Scenarios with Dynamic Neural Networks. *Neurocomputing*, Vol. 491, 2022, pp. 621–632, doi: 10.1016/j.neucom.2021.12.038.
- [13] SOUIDI, M. E. H.—PIAO, S.: A New Decentralized Approach of Multiagent Cooperative Pursuit Based on the Iterated Elimination of Dominated Strategies Model. *Mathematical Problems in Engineering*, Vol. 2016, 2016, Art.No. 5192423, doi: 10.1155/2016/5192423.
- [14] KHOSRAVIFAR, B.—BOUCHET, F.—FEYZI-BEHNAGH, R.—AZEVEDO, R.—HARLEY, J. M.: Using Intelligent Multi-Agent Systems to Model and Foster Self-Regulated Learning: A Theoretically-Based Approach Using Markov Decision Process. 2013 IEEE 27th International Conference on Advanced Information Networking and Applications (AINA), 2013, pp. 413–420, doi: 10.1109/AINA.2013.70.
- [15] CASTANEDA, G.—MORRIS, P.—KHOSHGOFTAAR, T. M.: Evaluation of Maxout Activations in Deep Learning Across Several Big Data Domains. *Journal of Big Data*, Vol. 6, 2019, Art. No. 72, doi: 10.1186/s40537-019-0233-0.
- [16] CAI, M.—LIU, J.: Maxout Neurons for Deep Convolutional and LSTM Neural Networks in Speech Recognition. *Speech Communication*, Vol. 77, 2016, pp. 53–64, doi: 10.1016/j.specom.2015.12.003.

- [17] SRIVASTAVA, K.—SURANA, A.: Multi Agent AI for Tactical Maneuvering. In: Pham, T., Solomon, L. (Eds.): Artificial Intelligence and Machine Learning for Multi-Domain Operations Applications IV. Proceedings of the SPIE, Vol. 12113, 2022, pp. 166–176, doi: 10.1117/12.2617157.
- [18] RABOIN, E.: Model-Predictive Strategy Generation for Multi-Agent Pursuit-Evasion Games. Ph.D. Thesis. University of Maryland, College Park, 2015.
- [19] YAN, T.—CAI, Y.—XU, B.: Evasion Guidance Algorithms for Air-Breathing Hypersonic Vehicles in Three-Player Pursuit-Evasion Games. Chinese Journal of Aeronautics, Vol. 33, 2020, No. 12, pp. 3423–3436, doi: 10.1016/j.cja.2020.03.026.
- [20] RUIZ LÓPEZ, U.: Pursuit-Evasion Problems with a Differential Drive Robot and an Omnidirectional Agent. Ph.D. Thesis. CIMAT, Guanajuato, Mexico, 2020.
- [21] HESPANHA, J. P.—PRANDINI, M.—SASTRY, S.: Probabilistic Pursuit-Evasion Games: A One-Step Nash Approach. Proceedings of the 39th IEEE Conference on Decision and Control, Vol. 3, 2000, pp. 2272–2277, doi: 10.1109/CDC.2000.914136.
- [22] GUO, M.—XIN, B.—CHEN, J.—WANG, Y.: Multi-Agent Coalition Formation by an Efficient Genetic Algorithm with Heuristic Initialization and Repair Strategy. Swarm and Evolutionary Computation, Vol. 55, 2020, Art.No. 100686, doi: 10.1016/j.swevo.2020.100686.
- [23] KRAUSBURG, T.: Hierarchical Coalition Formation in Multi-Agent Systems. In: Rodríguez González, S., González-Briones, A., Gola, A., Katranas, G., Ricca, M., Loukanova, R., Prieto, J. (Eds.): Distributed Computing and Artificial Intelligence, Special Sessions, 17th International Conference (DCAI 2020). Springer, Cham, Advances in Intelligent Systems and Computing, Vol. 1242, 2021, pp. 210–214, doi: 10.1007/978-3-030-53829-3_23.
- [24] ESTRADA, R.—MIZOUNI, R.—OTROK, H.—MOURAD, A.: Task Coalition Formation for Mobile CrowdSensing Based on Workers' Routes Preferences. Vehicular Communications, Vol. 31, 2021, Art.No. 100376, doi: 10.1016/j.vehcom.2021.100376.
- [25] BABU, T. S. K.—CHITNIS, S.: Coalition Formation Based Cooperation Strategy for Routing in Delay Tolerant Networks. Materials Today: Proceedings, Vol. 45, Part 9, 2021, pp. 8182–8187, doi: 10.1016/j.matpr.2021.02.694.
- [26] SOUIDI, M.—PIAO, S.—LI, G.—CHANG, L.: Coalition Formation Algorithm Based on Organization and Markov Decision Process for Multi-Player Pursuit Evasion. Multiagent and Grid Systems, Vol. 11, 2015, No. 1, pp. 1–13, doi: 10.3233/MGS-150226.
- [27] CRUZ, D. L.—YU, W.: Path Planning of Multi-Agent Systems in Unknown Environment with Neural Kernel Smoothing and Reinforcement Learning. Neurocomputing, Vol. 233, 2017, pp. 34–42, doi: 10.1016/j.neucom.2016.08.108.
- [28] SOUIDI, M. E. H.—SONGHAO, P.—GUO, L.—LIN, C.: Multi-Agent Cooperation Pursuit Based on an Extension of AALAADIN Organisational Model. Journal of Experimental and Theoretical Artificial Intelligence, Vol. 28, 2016, No. 6, pp. 1075–1088, doi: 10.1080/0952813X.2015.1056241.
- [29] BENOUDINA, L.—REDJIMI, M.: Multi Agent System Based Approach for Industrial Process Simulation. Journal Européen Des Systèmes Automatisés (JESA), Vol. 54, 2021, pp. 209–217, doi: 10.18280/jesa.540202.
- [30] BOUDJIDJ, A.—MERAH, E.—SOUIDI, M. E. H.: Towards a Formal Multi-Agent

- Organizational Modeling Framework Based on Category Theory. *Informatica*, Vol. 45, 2021, No. 2, doi: 10.31449/inf.v45i2.2967.
- [31] QADIR, M. Z.—PIAO, S.—JIANG, H.—SOUIDI, M. E. H.: A Novel Approach for Multi-Agent Cooperative Pursuit to Capture Grouped Evaders. *The Journal of Supercomputing*, Vol. 76, 2020, No. 5, pp. 3416–3426, doi: 10.1007/s11227-018-2591-3.
- [32] REDJIMI, K.—REDJIMI, M.: A Multi-Agent System for Industrial Simulators Design. In: Troiano, L., Vaccaro, A., Tagliaferri, R., Kesswani, N., Díaz Rodriguez, I., Brigui, I., Parente, D. (Eds.): *Advances in Deep Learning, Artificial Intelligence and Robotics (icdlair 2020)*. Springer, Cham, *Lecture Notes in Networks and Systems*, Vol. 249, 2022, pp. 129–140, doi: 10.1007/978-3-030-85365-5_13.
- [33] LIU, X.—MASOUD, N.—ZHU, Q.—KHOJANDI, A.: A Markov Decision Process Framework to Incorporate Network-Level Data in Motion Planning for Connected and Automated Vehicles. *Transportation Research Part C: Emerging Technologies*, Vol. 136, 2022, Art.No. 103550, doi: 10.1016/j.trc.2021.103550.
- [34] ROCHOLL, N.: *Isolating Wildfires Using a Convolutional Neural Network Based Multi-Agent System*. Master Thesis. University of Groningen, 2021 (Bachelor Thesis).
- [35] IRANFAR, A.—ZAPATER, M.—ATIENZA, D.: Multiagent Reinforcement Learning for Hyperparameter Optimization of Convolutional Neural Networks. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, Vol. 41, 2022, No. 4, pp. 1034–1047, doi: 10.1109/TCAD.2021.3077193.



Nabila SID is a Ph.D. student in the Department of Mathematics and Computer Science, at Khenchela University, Algeria, a member of ICOSI Laboratory. Her research interests include multi-agent system, game theory and convolutional neural network.



Meriem DJEZZAR is an Associate Professor in computer science at the University of Khenchela, Algeria and a member of the ICOSI Laboratory. She completed her Ph.D. in computer sciences from University of Constantine 2 in 2016. Her research interests are in machine learning, cyber-physical systems, knowledge representation and reasoning.



Mohammed El Habib SOUIDI received his B.Sc. degree in computer science from the University of Khenchela, Algeria in 2011. He also received his Master's degree in computer science from the same university in 2013, and his Ph.D. in computer science from the Harbin Institute of Technology, China, in 2017. He is working as a Lecturer in the Department of Mathematics and Computer Science in University of Khenchela, Algeria. Moreover, he is affiliated as a researcher in ICOSI Lab, University of Khenchela. His research interests include multi-agent task coordination, reinforcement learning, game theory, and path planning.



Mounir HEMAM is an Associate Professor at the University of Khenchela, Algeria and the Head of the GECCO Research Group at the ICOSI Laboratory. He has completed his Ph.D. degree in computer sciences from the Mentouri University in January 2012. He received his habilitation qualifications (Accreditation to supervise research HDR) in 2017. His research interests include knowledge representation and reasoning; formal knowledge representation, machine learning and semantic IoT interoperability in the context of Industry 4.0.