Int. J. Nonlinear Anal. Appl.
Volume 12, Special Issue, Winter and Spring 2021, 1987–2018
ISSN: 2008-6822 (electronic)
http://dx.doi.org/10.22075/ijnaa.2021.24783.2825

A bankruptcy based approach to solving multi-agent credit assignment problem

Hossein Yarahmadi[a], Mohammad Ebrahim Shiri[b,*], Hamidreza Navidi[c], Arash Sharifi[d]

[a]*Department of Computer Engineering, Science and Research Branch, Islamic Azad University, Tehran, Iran*
[b]*Department of Mathematics and Computer Science, Amirkabir University of Technology, Tehran, Iran*
[c]*Department of Mathematics and Computer Science, Shahed University, Tehran, Iran*
[d]*Department of Computer Engineering, Science and Research Branch, Islamic Azad University, Tehran, Iran*

(Communicated by Madjid Eshaghi Gordji)

Abstract

Multi-agent systems (MAS) are one of the prominent symbols of artificial intelligence (AI) that, in spite of having smaller entities as agents, have many applications in software development, complex system modeling, intelligent traffic control, etc. Learning of MAS, which is commonly based on Reinforcement Learning (RL), is one of the problems that play an essential role in the performance of such systems in an unknown environment. A major challenge in Multi-Agent Reinforcement Learning (MARL) is the problem of credit assignment in them. In this paper, in order to solve Multi-agent Credit Assignment (MCA) problem, we present a bottom-up method based on the bankruptcy concept for the effective distribution of the credits received from the environment in a MAS so that its performance is increased. In this work, considering the Task Start Threshold (TST) of the agents as a new constraint and a multi-score environment, as well as giving priority to agents of lower TST, three methods PTST, T-MAS and T-KAg are presented, which are based on the bankruptcy concept as a sub branch of game theory. In order to evaluate these methods, seven criteria were used among which density was a new one. The simulation results of the proposed methods indicated that the performance of the proposed methods was enhanced in comparison with those of the existing methods in six parameters while it proved a weaker performance in only one parameter.

Keywords: Multi-agent Systems, Credit assignment problem, Bankruptcy, Reinforcement Learning, Game Theory, Global Reward Game,Machine Learning.

*Corresponding author
Email addresses: hs.yarahmadi@srbiau.ac.ir (Hossein Yarahmadi), shiri@aut.ac.ir (Mohammad Ebrahim Shiri), navidi@shahed.ac.ir (Hamidreza Navidi), a.sharifi@srbiau.ac.ir (Arash Sharifi)

Received: August 2021 *Accepted:* December 2021

1. Introduction

In recent years, attention has shifted from centralized to distributed systems [1]. One branch of distributed systems is Distributed Artificial Intelligence (DAI) [2]. DAI has been manifested in three areas, which are: parallel artificial intelligence, distributed problem solving, and finally MAS [3]. MAS deal with the behaviors of computing entities called agents, which collaboratively interact to solve a problem [4]. These problems span a wide range of applications such as traffic control [5], cancer modeling [6], complex systems and networks [7], cyber-physical systems [8], medical images fusion [9], Intelligent transportation systems [10], etc. One challenging problem with MAS is their learning [11]. Since MAS are able to operate independently and automatically, they are used in unsupervised environments. Therefore, RL may be suitable for MAS in unknown environments [12]. In MARL, agents collaborate to solve a problem and receive rewards or punishments from the environment for taking the correct or incorrect action, respectively. Ultimately, such credits and punishments result in the learning of MAS in these environments [13]. Now this question is raised that how the credit or punishment that the environment grants the MAS as a vector is to be distributed among the agents to enhance the performance of the agents. This problem is known as the MCA in the literature [14].

Since the agents in MAS collaborate to solve the problem, the issue may be considered from the perspective of cooperative games [15]. Cooperative games comprise an important research area in the MARL since many real-life problems can be modeled as such games. Instances of this can be seen in the collaboration of autonomous vehicles [16], energy efficiency in LTE networks [17], and search-and-rescue robots [18]. Global-reward games are a subclass of cooperative games in which agents aim to increase the global reward [19]. In such games, credit assignment is an important problem whose purpose is to find a way to distribute the global rewards. There are two approaches to solving this problem, namely the shared reward approach [20] and the local reward approach [21]. The shared reward approach directly attributes a global reward among all agents. This is while the local reward approach assigns a fraction of the global reward to each agent based on its contribution.

The purpose of this study was to find an appropriate way of distributing reward among agents so that the performance of the distributed system and accordingly the reward gained from the environment were increased. In a plenty of studies conducted in this field, the environment is considered both uniform and homogeneous so that by solving part of the problem a reward or punishment equal to solving the other part is received. Cases like this can be found in [14][22][23]. This is while there are a lot of problems, which are not essentially uniform, so that solving one part brings a different reward or punishment than solving the other part. Examples of these problems are abundant in everyday life, including the grades that students receive for each course exam and their grade point average [24], economic, social and political problems of a country and their priorities [25], proper distribution of resources among the individuals of a society in terms of their performance [26], investing in the stock market according to their technical and fundamental parameters [27]. In all of the mentioned problems, the environment in question is a non-uniform environment, which has not been addressed a lot in the relevant studies. These problems, in which solving one part provides a different reward or punishment than for the other part, are called Multi-Score Problems (MSP). Although heterogeneous MAS have received much attention [28][29], problems in which the environment is not uniform have received less attention.

In this paper, our focus is on such problems. Furthermore, this study used a constraint called the Task Start Threshold (TST). This means that any agent shall start to work only when it receives a reward greater than or equal to this threshold, while in a majority of previous studies in this field, agents started to work upon receiving any reward. This constraint allows us to get closer to real world problems. There are many problems in real life in which people (agents) start to work only if

they are granted a certain amount of reward; otherwise they will not start to work. Instances can be found in problems such as corporate tenders and auctions [30], problem of workers [31], etc. A company starts to work if it receives a suitable fee or a worker starts to work if they receive certain salaries. Since many agents of good performance only start to work if they receive the appropriate reward, and since such a reward may not be available in the beginning, in this study, as a bottom-up innovation, priority was given to agents that might have lower performance, but they could get reward faster.

One method for distributing resources between the agents when they are limited and claimants are many is the bankruptcy method, which is a subclass of the game theory first introduced by O'Neill [32]. In the problem of MARL, when agents are interacting with the environment and receive reward or punishment from that environment, from the local reward approach point of view, we are faced with the problem that how to distribute this limited reward/punishment among agents so that the performance of the MAS is increased. We thus appear to be dealing with a bankruptcy problem. Therefore, in this paper, as the next innovation, we used the prioritized bankruptcy method to solve the problem so that the priority is given to agents with lower TST. These agents are rewarded faster and therefore will launch agents that perform better but have a higher TST. They, in turn, will launch more agents in the same manner. Examples of the real-world situation in which the same process occurs are wildfires that begin with a small fire that ignites shrubs and then larger trees are involved. Another example is the beginning of revolutions in different societies, as in the Arab Spring, which started in Tunisia with the self-immolation of a salesman, then the disobedience spread to the people of the city, then to the whole Tunisia, and finally to other Arab countries. In studies related to this problem, parameters such as group learning rate, confidence, expertness, certainty, efficiency, and correctness are employed to evaluate the proposed methods. As the last innovation in this research, a new criterion, i.e. density, was considered in addition to the above mentioned parameters, which is an indication of the number of agents that collaborate in problem solving. An example of this can be found in the team work of a group of students to solve a math problem that the teacher assigns to involve as many students as possible.

2. Related Works

RL can be considered the most important method of innate learning among living beings. This type of learning, which occur based on rewards or punishments, is found in all living things [33]. The RL method that occurs in intelligent systems is based on this fact.

One of the most important features of RL is its ability to learn in unknown environments. This feature makes it as a suitable method to autonomous systems learning. MASs also use this feature, as an autonomous system, for learning [34].

Rahaei and Beigi [23] generally divided the MCA into the following four categories:

1. Temporal Credit Assignment
2. Structural Credit Assignment
3. Social Credit Assignment
4. Multi-agent Credit Assignment

In temporal credit assignment [35][36], which is related to a single agent system, the agent is considered as an entity that does not immediately receive the result of its action, which may be a reward or a punishment, after interacting with the environment, so the agent is not able to recognize what action did the reward/punishment belong to? This method is looking for answer this question.

In structural credit assignment [37], the reason of the reward/punishment is the agent's knowledge, and therefore it must be determined which part of the agent's knowledge caused the reward/punishment. In this method, the relevant system is a single agent system.

In the social credit assignment, which is introduced by Mao [38], attempt is to determine the cause of the agent's behavior based on internal or external factors.

The last type of credit assignment in this categorize is the MCA, which is part of the MARL process. The MCA was first introduced by Skinner [39]. It was stated that the success of a system depends on the cooperation of its components. MCA solving often occurs in two ways. In the first case, which is the simplest and meanwhile the most unfair and inefficient method, the received reward is divided equally between the agents. This method is called shared reward approach [20]. The next method, which is called the local reward approach method [21], rewards of each agent based on its contribution to the success of the MAS in achieving its goal. This method is fairer and more efficient than the first method, but it is difficult to determine the participation of each agent.

From another point of view, the MCA problem may be considered implicitly or explicitly:

1. Explicit Credit Assignment
2. Implicit Credit Assignment

Explicit credit assignment introduces strategies for assigning credits to the agents, which are at least locally optimized [40]. COMA [41] is an example of this approach, which uses a centralized critic to estimate the advantages and disadvantages of an agent's action. However, in complex collaborative behaviors it loses its effectiveness. SQDDPG [42] is another instance, which works based on a theoretical framework for credit assignment according to the approximate contribution of agents, which is sequentially added to a group of agents. This framework, while theoretically justified, assumes an initial timetable for agents with a public supervision, which is often unachievable in practice. In contrast, the implicit methods do not purposefully assess the agent actions based on a specific baseline, but they use former methods to assign credits in such a way that the agents' learning from the distribution of the global credits among them occurs based on their individual functions. One of the earliest tasks in this field is VDN [43], in which the value decomposition is linear and the state-related information is ignored during the training phase. QTRAN [44] tried to prove the limitations with a general factorization, but there are computational limitations that could lead to a poor experimental performance.

One of the first works to solve the MCA, based on the agent's knowledge, was done by Harati [14]. This work was later extended by Rahaei and Beigy [23][45]. To solve the MCA, they proposed two methods, that were history based method and ranking method [23].

In the history-based method [23], prior knowledge of the agents is modeled as an undirected graph. In this graph a set of variables with unknown values is introduced to model the environment. Then, once the values of those variables are specified, the environment model is complete. After the environment modeling, the critic can decide how to assign the global reward between agents.

The next method to solve the MCA problem is the ranking method [23]. In the ranking method, the knowledge of the agents is first extracted based on the criteria introduced in [14] and then ranked. In this method, the critic distributes the global reward among the agents based on this ranking.

The last method that we use in this paper to compare with the proposed methods is called the dynamic method [45]. In the dynamic method, to solve the MCA problem, the global reward is decomposed into sum of weighted rewards among the agents.

3. Preliminaries

3.1. Markov Decision Process

In the one-agent RL problem, the problem is usually modeled as the Markov Decision Process (MDP) [46]. Markov decision process is formally defined as the multiple (S, A, P, R, γ) in which,

S is the space state;

A is the action space;

$$P: \ S \times A \to \delta(S)$$

.

is the probability that transition from $s \in S$ to $s' \in S$ happens through the action $a \in A$.

$$R : S \times A \times A \longrightarrow \mathbb{R}$$

is the reward function, which returns the reward received by an agent due to transition from the pair (s, a) to the state s'.

$$\gamma \in \ [0, 1]$$

is a discount factor and a parameter used to compensate for the instant effect in the learning process of the agent.

3.2. Markov Game

When more than one agent is involved in RL, because the behavior of the agents strongly affects the performance of other agents, the MDP is not appropriate for describing and modeling the environment and the problem. A generalization of MDP is called Markov games [47], which is also called stochastic games [48].

A Markov game is expressed as a multiple $\left(N, S, \{A_i\}_{i \in N}, P, \{R_i\}_{i \in N}, \gamma\right)$ in which,

$N = 1, 2, 3 \ldots, N$ is the set of $N > 1$ agents;

S is the observable space for all agents;

A_i : is the action space for the i^{th} agent and

$$A = A_1 \times A_2 \times \ldots \times A_N \tag{3.1}$$

is called joint action space.

$$P: \ S \times A \to \delta(S) \tag{3.2}$$

is the probability of transition to any state $s' \in S$ starting from $s \in S$ and taking the common action $a \in A$.

$$R_i : S \times A \times A \longrightarrow \mathbb{R} \tag{3.3}$$

is the reward function of the i^{th} , which indicates the instant reward resulting from transition from (s, a) to state s'.

$$\gamma \in \ [0, 1]$$

is the discount factor. In this paper, we represent all agents with Ag and agent i with Ag_i. Here, we are dealing with a number of agents that, together, they form MAS and collaborate to achieve a certain goal. If we suppose that this MAS consists of N agents, then,

$$MAS = \{\mathrm{Ag}_1, \mathrm{Ag}_2, \ldots, \mathrm{Ag}_N\}$$

In this paper, it was assumed that every agent did not start to work at the beginning, unless the received reward was higher than its TST. The reward received by agent i at time t is displayed with r_i^t and its TST is displayed with TST_i:

$$\forall\ \mathrm{Ag}_i \in MAS\ :\ \text{if } \mathrm{r}_i^t \geq \mathrm{TST}_i \text{ then } \mathrm{Ag}_i^{t+1}\ is\ Active \tag{3.4}$$

Eq. (3.4) shows that each agent may be in an active or inactive state. We denote each active agent at time t with Ag_i^t. The set of active agents is the set of agents interacting with the environment at time t. The number of active agents is n so that $n \leq N$. This set of active agents is displayed with MAS^t:

$$\mathrm{MAS}^t = \left\{\mathrm{Ag}_1^t, \mathrm{Ag}_2^t, \ldots \mathrm{Ag}_n^t\right\}\ ,\ where\ \mathrm{MAS}^t \subseteq MAS$$

Every active agent Ag_i^t performs an action in the environment at time t, which is denoted by a_i^t. The actions set of the agents in the environment at time t is displayed with A^t,

$$A^t = \left\{a_1^t, a_2^t, \ldots, a_n^t\right\},\ where\ A^t \subset A$$

Since the environment regards the MAS as a single entity, it returns the resultant of the rewards/punishments for each agent in the form of a global reward to the MAS and delivers it to the critic. This global reward is displayed with R. Because this global reward is not the same at all times, we display it at time t with R^t. The critic has the task of distributing the received global reward among the agents. In other words, the critic must produce the vector $(r_1^t, r_2^t, \ldots, r_n^t)$ in such a way that,

$$\sum_{i=1}^{n} r_i^t = R^{\mathrm{t}} \tag{3.5}$$

This is illustrated in Figure 1.

How to distribute this reward among the agents is the basis of the present work. For arranging the agents according to their TSTs, we use the symbol "$\ll$" as follows,

$$\mathrm{Ag}_k \ll \mathrm{Ag}_j \tag{3.6}$$

This expression means that the TST of agent k is smaller than the TST of agent j.

In this work, we introduced the TST as a constraint to get closer to the real situation. The existence of this constraint will cause each agent to start to work only if the received reward is greater than TST.

Therefore, the function $SAg(.)$ indicates that the agent starts to work on following conditions:

$$\mathrm{SAg}\,(r) = \begin{cases} False & ,\ r < TST \\ True & ,\ r \geq TST \end{cases} \tag{3.7}$$

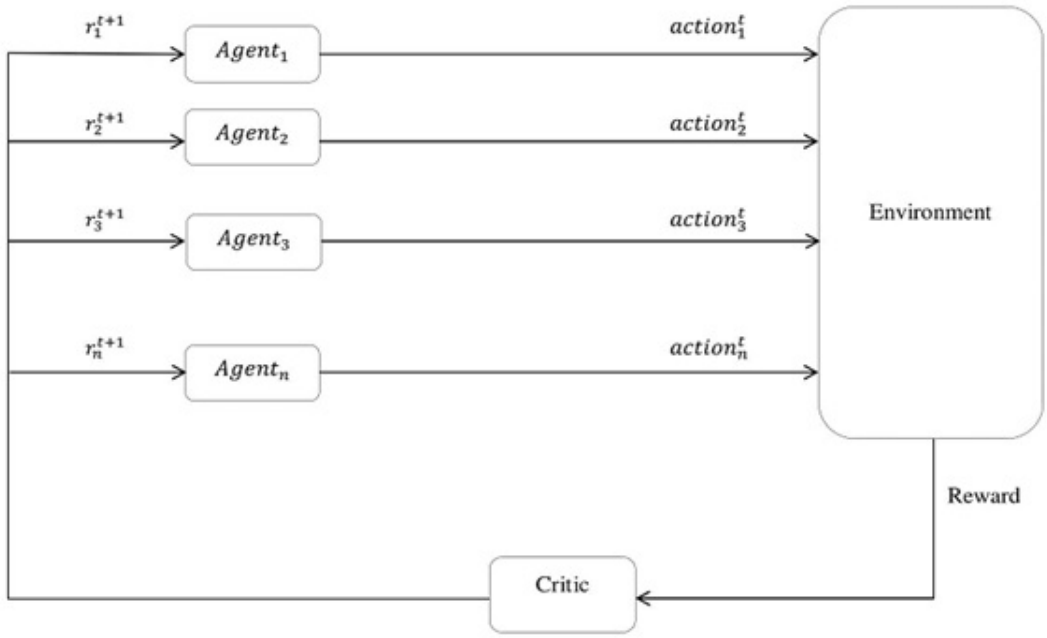

Figure 1: Interactions of the agents of a multi-agent system with the environment and receiving global reward from the environment by the critic.

Eq. (3.7) states that if the received reward, r, is greater than the TST value, then the agent will start to work and the function will return a *True* value; otherwise, it shall not start to work and the function will return a *False* value.

In this work, the Q learning method according to Eq. 8, AKA the Bellman equation, was used for the RL of the agent.

$$Q(s,a) = Q(s,a) + \alpha \left[r(s,a) + \gamma \max \left(Q\left(s', a'\right) \right) - Q(s,a) \right] \tag{3.8}$$

In Eq. (3.8), s is the current state of the agent and a is the action being taken in this state. s' is the state to which the agent is transitioned after performing action a in state s. r is the amount of the reward received by the agent due to a transition from the pair (s,a) to the state s', Q is the learning table of the agent and γ is the discount factor. In this paper, the operating environment was the Scrabble game [49]. Each agent was responsible for placing one or more letters in the appropriate place. Upon solving part of the problem by the agent i at time $t+1$, the amount of the reward that the environment provides increases by c_i with respect to time t. If we display the set of successful agents with Sc,

$$Sc = \{i : \text{ Ag}_i \text{ is successful at time t}\}$$

Then,

$$r_i^{t+1} = r_i^t + c_i \tag{3.9}$$

$$R^{t+1} = R^t + C \ , \ where \ C = \sum_{i \in Sc} c_i \tag{3.10}$$

3.3. Bankruptcy Problem

In this paper, the bankruptcy concept was used to distribute rewards among agents. The bankruptcy problem is a branch of game theory that deals with how a debtor's assets, which are less than the total claims of the creditors, are distributed among them, so that the amount assigned to each creditor is non-negative and not greater than the claimed amount. If we denote the sum of the creditors' claims by D, each creditor's claim by d_i, debtor's assets by E, the fraction of the debtor's assets allocated to each creditor by x_i, and the number of the debtors by N, then we have,

$$E = \sum_{i=1}^{N} x_i \tag{3.11}$$

$$D = \sum_{i=1}^{N} d_i \tag{3.12}$$

$$E \leq D \tag{3.13}$$

$$x_i \leq d_i \tag{3.14}$$

The problem of bankruptcy can be summarized as follows [50]: If we consider the pair (E, D) as a bankruptcy problem, then one solution to the bankruptcy problem (E, D) is an n-fold assignment in which $E = \sum_{j \in N} x_{\mathrm{j}}$. In order to select the mode that results in the highest efficiency, an allocation function is used. In a cooperative game with n players during the formation of the allocation function, the pair (N, c) is defined in such a way that $N = \{1, 2, \ldots, n\}$ is a finite set of players, $c : 2^n \to R$ is the allocation function, 2^n represents the number of subsets of N and $c\,(\varnothing)$ is assumed be 0. In fact, in these problems, we refer to the S subsets of N as allocations, and the amount of $c(S)$ is introduced as the value (asset) of S. In allocation, each player's asset is interpreted as the player's maximum profit or cost. Now, consider a fixed set of players by a game (N, c) where c is an allocation function. The bankruptcy game is then defined according to the bankruptcy problem (E, D) by Eq. (3.15),

$$c_{E,d}\,(S) = max\{(E - \sum_{j \in N \setminus S} d_j)\ , 0\} \tag{3.15}$$

As stated before, in Eq. (3.15), E is the amount of the debtor's assets and S is the subset of N that the debtors' assets should be assigned to the members of the S, which are the creditors. j is any member of N that can be a member of the subset S, and therefore d_i is the amount of the asset allocated from the debtor to the creditor j. An optimal value of zero would indicate bankruptcy. If the amount of a player's asset is less than or equal to the sum of the creditors' claims on that player, the amount obtained in the phrase,

$$E - \sum_{j \in N \setminus S} d_j$$

is negative and all assets have been paid to creditors as receivables; this indicates that the player is bankrupt, and a non-zero value indicates non-bankruptcy. This means that if the amount obtained in the phrase

$$E - \sum_{j \in N \setminus S} d_j$$

is positive, it is interpreted as non-bankruptcy of the player in question. There are various ways for solving the bank bankruptcy problem, among which the following can be mentioned.

3.3.1. Proportional Rule (P rule)

The ratio or proportionality method is the simplest and most famous method of the bankruptcy theory [50][51]. In this method, the allocation coefficient is obtained through dividing the inventory by the amount claimed by the claimants according to Eq. (3.16). Therefore, the share of every claimant is calculated using Eq. (3.17) and with an equal coefficient of their needs.

$$\beta = E/D \tag{3.16}$$

$$x_i = \beta d_i \tag{3.17}$$

3.3.2. Adjusted Proportional Bankruptcy Rule (AP rule)

According to this method [50], and based on Eq. (3.18), first other claimants assign an initial value of v to claimant i:

$$v_i = \text{Max}\{0, E - \sum_{j \neq i} d_j\} \tag{3.18}$$

$$x_j = v_i + (d_j - v_i) \left(\sum_{j \in N} (d_j - v_i) \right)^{-1} \left(E - \sum_{j \in N} v_j \right) \tag{3.19}$$

In the AP method (Eq. (3.18)), as an initial allocation to person i, the needs of all claimants but person i are satisfied first. Then the remainder is allocated to claimant i, and if nothing is left or a negative value is calculated as the remainder, a zero value is allocated to that claimant. In Eq. (3.18) v_i indicates the share of claimant i. Once v_i is calculated share of each claimant can be calculated based on the Eq. (3.19) where N is the number of claimants. Because in this method, first other representatives are considered and prioritized to determine the initial allocation of claimant i, the initial allocation is called the minimum claimant i.

3.3.3. Constrained Equal Award (CEA)

The basic idea behind the CEA is to meet the levels of the needs in an equal way so as to the amount allocated to each individual does not surpass the level of the need [50][52]. The following steps are taken for the calculation of the CEA:

In the first step, the lowest claims are considered as an initial allocated value for all creditors. After the fulfillment of this request, through the elimination of the claimant with the minimum allocation, the process continues with other claimants. Eq. (3.20) shows how the allocation is worked out in this method:

$$x_i = CEA\,(d_i, \lambda) = Min\,(d_i, \lambda) \tag{3.20}$$

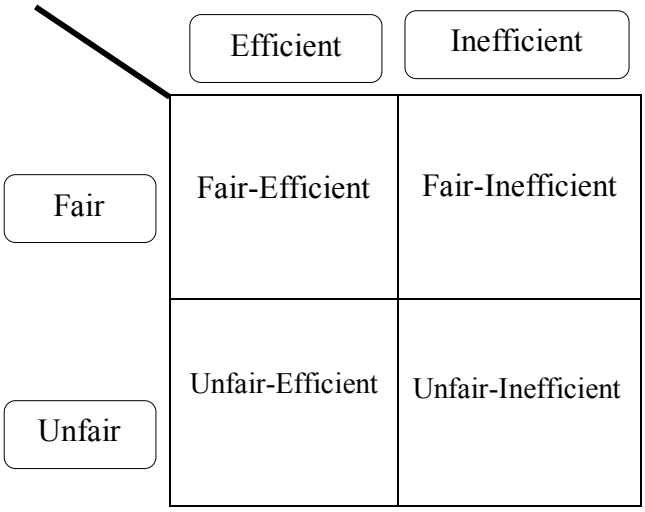

Figure 2: Classification of MCA problems based on fairness and effectiveness.

The value of λ is chosen in a way that,

$$\sum_{i \in N} Min(d_i, \lambda) = E$$

3.3.4. Constrained Equal Loss (CEL)

In the CEL method, it is attempted to distribute the value of the existing deficit evenly among all claimants [50], [53]. Based on Eq. (3.21), the difference between the number of claims and the source inventory is computed and divided by the number of claimants. The computed value, which is indeed considered as an equal loss, is deducted from the claims by all claimants and considered as the amount allocated to each claimant.

$$x_i = CEL\,(d_i, E) = \text{Max}\,(0, d_i - \lambda) \tag{3.21}$$

The value of λ is selected in a way that,

$$\sum_{i \in N} \text{Max}\,(0, d_i - \lambda) = E$$

3.3.5. Problem Definition

Based on fairness and effectiveness, the MCA problem can be considered as illustrated in Figure 2.

The ideal state can be considered as one in which, in addition to a fair distribution, efficiency increases as well. In the studies presented in this field, it has been attempted to move in this direction, but often fair distribution has been under the spotlight. There are many problems in which fair distribution not only does not increase the efficiency but also results in efficiency reduction. Consequently, two more general cases can be found in the MCA:

- Fair and inefficient
- Unfair and efficient

In this work, we tried to increase the efficiency of MAS so that all agents are benefited from this increase in efficiency. The problem becomes more challenging when the reward that the agents receive from the environment through critic is less than their TST values. In that case, several agents will not start to work. Otherwise stated, the amount of reward received from the environment through the critic is less than sum of TSTs of the agents, and therefore we face the problem of bankruptcy. As illustrated in Figure 1, the agents interact with the environment in MAS. Each agent is assigned a task to do and attempts to do it. This task may be done correctly or incorrectly. Upon this interaction, the environment returns a resultant vector of the agents' actions to the MAS, which in turn is delivered to the critic. Now the critic is faced with the challenge that how to distribute the global reward, which is less than sum of TSTs among the agents (which in spite of being possibly unfair) it increases the efficiency of the MAS. In this paper, we look for the answer to this problem and providing ways to solve it.

4. Methodologhy

This section presents a novel way of solving the MCA problem based on the bankruptcy concept. The proposed method was modified based on the adjusted proportional rule (AP), but it works in a reverse manner, which we call it the Reverse AP. Besides, the proposed methods used the concept of ranking so that the agents were ranked and prioritized based on their demands.

On the basis of the AP method, first the needs of all claimants except claimant i are satisfied to allocate to that claimant; then, if anything left from the assets of the debtor, the needs of claimant i will be satisfied. If the remaining reward is zero or negative, nothing is allocated to claimant i. In this method, claimant i is the one with the lowest request. In the proposed method, the process was reversed. That is to say that first the request of claimant i (the one with the lowest claim) was satisfied. Then requests of other claimants were addressed; this is why we called it Reverse AP. On the other hand, in order to prioritize the agents, we ranked them according the amounts of their requests (their TST values), so that the request of the agent with the lowest claim (lowest TST value) was satisfied first. In other words, since the claimants were prioritized based on the amount of their requests, first the needs of claimants with the least priority were satisfied. Based on what mentioned above, here we used the TST values of the agents as a constraint to prioritize. Thus the agents were sorted in ascending order (Eq. (4.1)) based on their TST values according to the definition 3.6; then rewards were allocated.

$$\mathrm{Ag}_k \ll \mathrm{Ag}_s \ll \ldots \ll \mathrm{Ag}_p, \qquad where \;\; \forall \mathrm{Ag}_i \in MAS \;,\; i = \{1, 2, \ldots N\} \tag{4.1}$$

To allocate rewards to agents, we used the Reverse AP method, which means first that reward was allocated to the agent with the lowest TST value in the ascending order of agents based on their TST values. The minimum reward received by an agent must be equal to its TST in order to start to work. Thus, the critic must, if possible, allocate the least possible amount to agents according to the described method. When the agent with the lowest amount of TST was rewarded according to the mentioned method, the next agent, which was after the first agent in the ascending order of TST values, had to be rewarded. The process continued until the remaining reward was zero or not available anymore. Eq. (4.2) shows how rewards were allocated to each agent and Eq. (4.3) shows how the rewards available were updated:

$$r_i^{t+1} = \begin{cases} \mathrm{TST}_i, & R^t - \mathrm{TST}_i \geq 0 \\ 0\;, & o.w \end{cases} \tag{4.2}$$

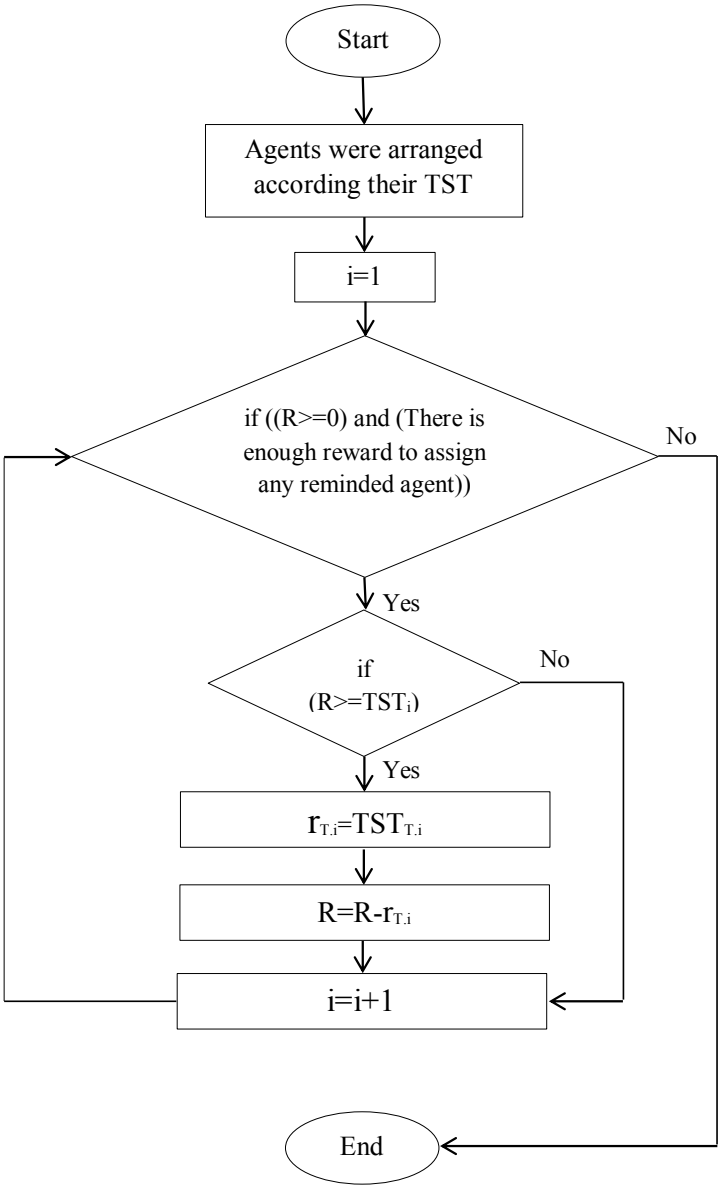

Figure 3: The Flowchart for reward allocation to the agents based on the TST values.

$$R^t = R^t - \mathrm{TST}_i \tag{4.3}$$

In Eq.(4.2), R^t is the global reward, TST_i is the TST for agent i and r_i^{t+1} is the reward allocated to agent i at time $t+1$.

if T.i represents the order of the agents among the agents based on their TST, so that $\mathrm{Ag}_{T.1}$ has the lowest and $\mathrm{Ag}_{\mathrm{T.N}}$ has the highest TST values, then the flowchart in Figure 3 represents the mentioned procedure.

If the residual reward is positive, after this process is completed and the rewards are allocated, but not allocable to the agents according to Eq. (4.2) (i.e. not enough to be capable of launching an agent), the residual reward must be distributed among the agents to which it has already been allocated. Now the question is raised that how this residual value should be distributed among the agents. In order to answer this question in the present paper, three methods will be presented as

follows.

- Pure TST (PTST)

In this method, the global reward is first distributed among the agents based on the flowchart in Figure 3. If the remainder amount of the reward is positive but not attributable to another agent (the residual value is less than the TST of the next agent based on the TST order), this reward will be neglected. In other words, this method simply seeks to get agents to start to work with the lowest amounts of rewards.

- TST satisfaction and distribution among MAS Based on their TST (T-MAS)

This method is based on the PTST method. The two methods are similar except that in the in the latter, the residual reward, which we denote by R_{rem}, will not be neglected if it is positive after the allocation. In this method, the allocation of this residual value takes place on the basis of the needs of each active agent as in Eq. (4.4),

$$r_i^t = r_i^t + \left(\frac{(r_i^t)^2}{\sum_{i=1}^{n} (r_i^t)^2} \right) \times R_{\text{rem}} \tag{4.4}$$

- TST satisfaction and allocation of the remainder to the most Knowledgeable Agent (T-KAg)

This method works like the T-MAS method with the exception that the residual reward distributed in a different manner. In this method, the remaining reward is assigned to the most knowledgeable agent. If we denote this agent with $\text{Ag}_{\text{knowledeable}}$, then the allocation of this residual value will be as Eq. (4.5),

$$r_i^t = \begin{cases} r_i^t + R_{\text{rem}}, & i = knowledeable \\ r_i^{\text{t}} \ , & o.w \end{cases} \tag{4.5}$$

4.1. Training phase

In the training phase, since the agents have no knowledge, they start to work in the beginning by randomly choosing a letter(s) to be placed in the right place. Then they select cells to place these letters. Such selections may be true or false. After all the letters are in placed in the cells (some in the correct and some in wrong places), the environment returns a consequent vector based on the way the letters are placed in the cells, which is the result of rewards and punishments. The critic is now faced with the fundamental question we sought to answer in this work i.e. how the critic should distribute the global reward among the agents. Owing to the fact that the critic has no information regarding the agents' knowledge, in this phase, it distributes the global reward (R^t) equally among the agents. In other words, each agent's share from the global reward is obtained based on Eq. (4.6),

$$r_i^{t+1} = \frac{R^t}{\text{N}} \tag{4.6}$$

In Eq. (4.6), N is the number of the agents and R^t is the global reward that the environment grants to the critic and the critic must distribute it among the agents. In the training phase, the learning process of an agent takes place based on RL, and each agent, after receiving a reward from the critic according to the Eq. (3.8) takes action to updates its Q-table. In the end of the training phase, the critic sorts the agents based on their knowledge and according to the criteria given in [14][23]. This training phase is illustrated in Figure 4.

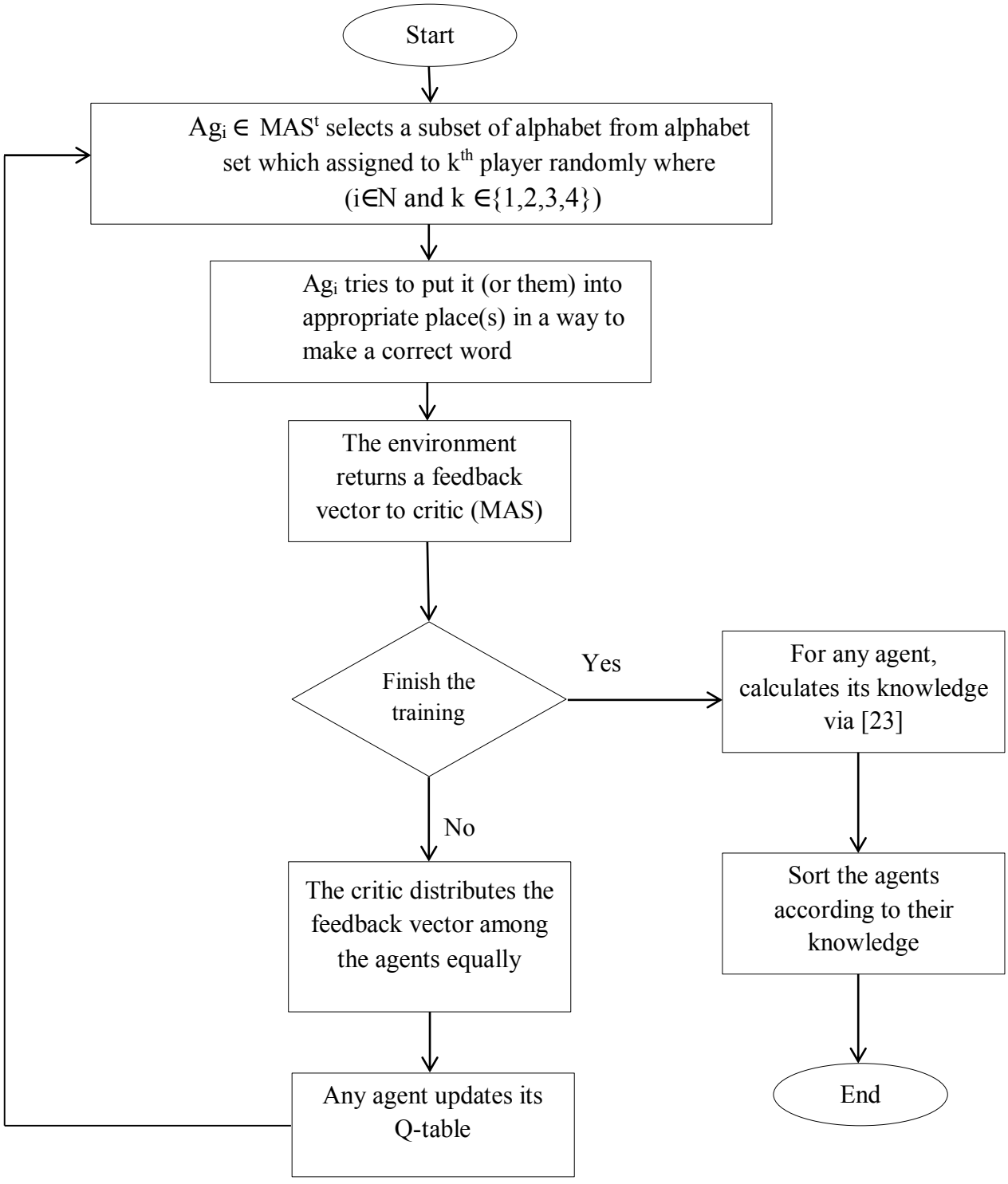

Figure 4: Training phase.

4.2. Test Phase

After the training phase, the test phase commences. In this work, we used the TST constraint. This constraint secured that the agents would start to work if the reward received was higher than their TST values. Therefore, in case of improper distribution of the rewards, it is possible that agents that require high TST do not start to work and consequently, the MAS will never start to work or will suffer from a low performance. This faces us with the basic problem of this study, which is how distribution should be in order to enhance the performance of MAS.

In order to implement the proposed method, the agents needed be prioritized. If priority was given to agents with high TST values, the system might never start; therefore, in this paper, priority was given to agents with lower TST values. On the other hand, according to the Reverse AP that we used in this work, the credits assignment to the agents had to be done. That is, first the reward was given to the agent with the highest priority –or the lowest amount of TST, and then the reward was distributed among other agents in the order of their TST values.

Once every agent started to work, if it did it properly, the environment returned a reward to the MAS, which was received by the critic. This would increase the rewards received by the MAS according to Eqs. (3.9) and (3.10) even more. This increase in the reward of the MAS at time t in comparison with the previous time would cause one or more agents (including more knowledgeable agents) start to work upon the distribution of the new reward based on the presented methods. These agents would in turn launch other agents by earning higher scores. In other words,

$$\exists\ c_k, \mathrm{Ag}_k \in MAS\ : \left(r_k^{t+1} = r_k^t + c_k\right) \geq \mathrm{TST}_k \tag{4.7}$$

Eq. (4.7) asserts that if the agent(s) Ag_i do its (their) task(s) correctly at time t, the reward received by the MAS will increase by C according to Eq. (3.10). In this case, it is possible for the critic to distribute more rewards among agents at time $t+1$. This causes agent(s) such as Ag_k that did not start previously due to not receiving the appropriate reward, to start to work at this time due to receiving a reward equal to or greater than it's(their) TST values (an increase equal to c_k compared to the previous time) and participate in the problem solving process. This process continues until the problem is solved. Figure 5 illustrates the test phase.

In this flowchart, after the interaction of the agents with the environment, in other words, after the agents try to solve the problem, the environment returns a reward/punishment resultant vector to the critic. In the next step, the critic distributes this global reward among the agents based on the presented methods. After that, each agent that receives a reward updates its Q-table based on the Q learning method. Since each agent is responsible for solving part of the problem, in the next step, the agent may have completed its tasks, in which case it is excluded from the set of active agents, MAS^t, according to Eq. (4.8),

$$\mathrm{MAS}^{t+1} = \mathrm{MAS}^t - \left\{\mathrm{Ag}_j : \mathrm{Ag}_j \text{ Completes its task}\right\} \tag{4.8}$$

Otherwise, that is, if the agent does not complete the task depending on the received reward, it will be faced with two scenarios. First, if the amount of the received reward is less than the TST value, $\mathrm{SAg}(r_i^t)$ will be equal to *False* and the agent must wait for a new distribution at a later time by the critic. In the second scenario, the received reward is higher than or equal to the TST value; in this case, $\mathrm{SAg}(r_i^t)$ will be equal to *True*, and the agent will be among the agents of the active MAS to participate in solving the problem at a later time. However, it is possible to think of conditions in which no agent is removed from the MAS set. In this scenario, it may prevent the participation of other agents in the problem solving process. After the MAS is updated according to Eq. (4.8), it should be checked if the problem is solved. When the problem solving is completed, the test phase

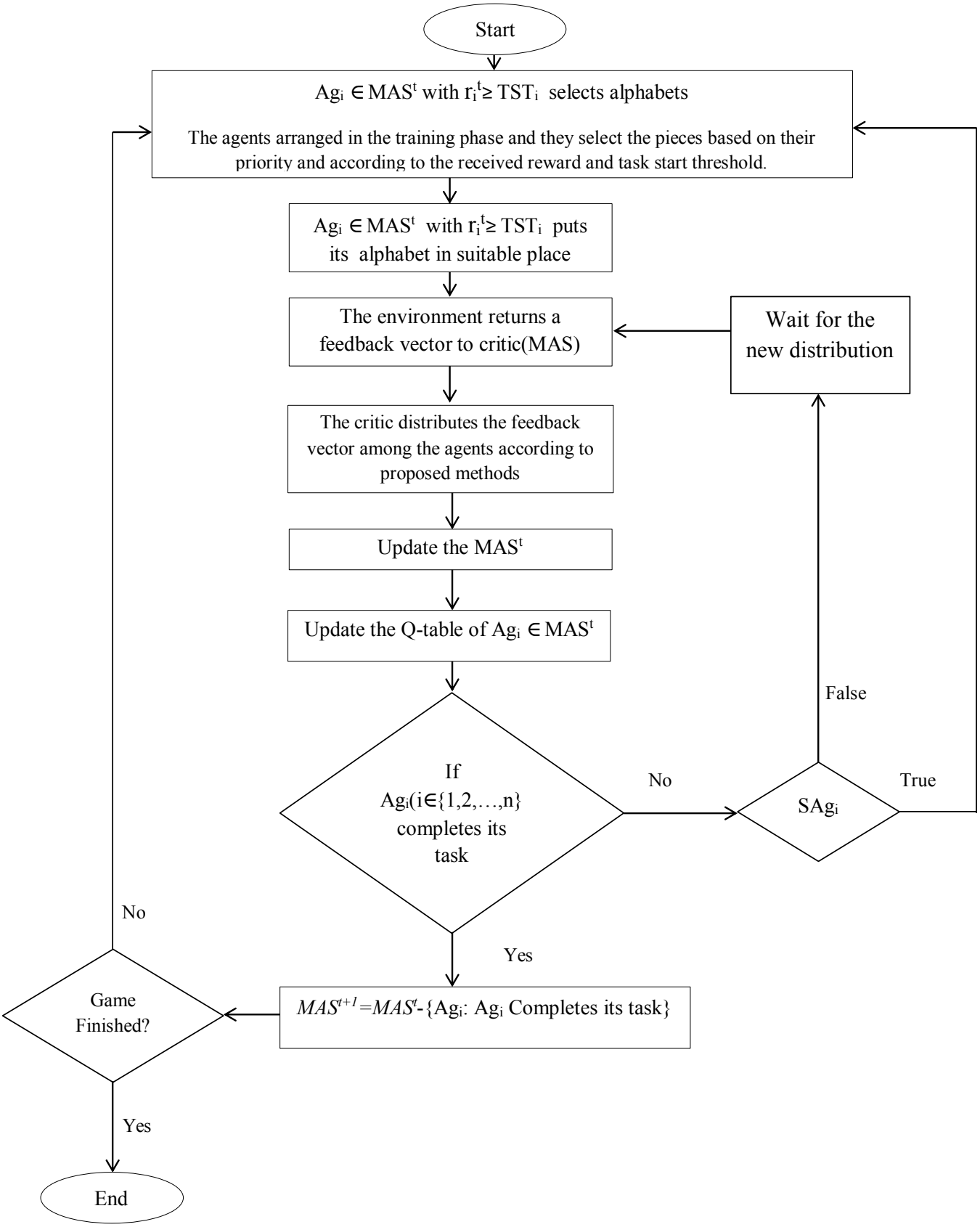

Figure 5: Test phase.

ends. Otherwise, the problem solving process continues with the new MAS. This process goes on until the problem is solved.

5. Results

The Scrabble game and its related rules were employed to evaluate the proposed methods. We utilized a MAS, which consisted of seven agents. According to the conditions introduced in this work, the agents with higher TST values were more knowledgeable agents and therefore chose the letters with higher scores. In contrast, the remaining letters with lower scores would be selected by agents of lower TST values and less knowledge. In this evaluation, the agents with the numbers 1,2,...,7 were considered. Agents 6 and 7 were the most knowledgeable while agents 1 and 2 had the lowest knowledge. The agents started to choose letters. Agents 6 and 7 chose the letters with the highest score as they learnt more and more. Furthermore, they chose more letters. Agents 1 and 2 selected fewer letters with lower scores due to lower learning rates. Therefore, their only advantage was their lower TST values. Agents were divided into three groups i.e. of high knowledge and high TST values including agents 6 and 7; of medium knowledge and medium TST values including agents 3, 4 and 5; and of low knowledge and low TST including agents 1 and 2. All agents started to work in the first round, but would work in subsequent rounds only if the reward they received from the critic was higher than their TST values. Consequently, owing to the fact that more knowledgeable agents had a higher TST values, they might take the whole reward but would not start to work due to not receiving the reward relevant to the TST value. Therefore, for the MAS to start, first agents with low TST values were launched first. These agents started to work due to their low TST and were rewarded. Then, this reward made it possible for the reward allocated by the critic to increase even more and launch other agents, including the more knowledgeable ones, which had reached their TST values owing to this increase in reward. Upon the entry of more knowledgeable agents into the set of active agents, i.e. MAS^t and construction of more valuable words, the reward received by the MAS increased, and consequently, the reward received by the critic increased as well. This process continued until other agents reached their TST and participated in the problem solving process. In order to evaluate the proposed methods, we used the criteria presented in [14][23], which are as follows.

- Group learning rate
- Confidence
- Expertness
- Certainty
- Efficiency
- Correctness

In addition, one more criterion, namely, *density* was introduced here and used to compare the proposed methods with other methods. Density refers to the number of agents that contribute to a problem solving process per time unit.

A noteworthy point, in the methods presented so far such as [23], all agents are rewarded with a global reward. Given the fact that in the proposed methods this reward might always be less than the TST value, it was possible that in these methods, the agents never started to work and the

problem remained unsolved. Based on the following criteria, our proposed methods were compared with three methods i.e. ranking method, history-based method and dynamic method. The results are presented in the following subsections.

5.1. Group learning rate

The average agent learning rate of the agents is computed based on Eq. (5.1).

$$\mathrm{LR}^t = \left(\frac{1}{N}\right) \sum_{i=1}^{N} \mathrm{LR}_i^t \tag{5.1}$$

Eq. (5.3) shows the individual learning rate of an agent.

$$\mathrm{LR}_i^t = \frac{|Learnt(S_i^t)|}{|S|} \tag{5.2}$$

The individual learning rate of an agent is defined based on Eq. (5.3) as the number of learning states:

$$\text{Learnt}\left(S_i^t\right) = \left\{\forall a_i^t : feasible\left(a_i^t, s_i^t\right) \rightarrow f_{a_i}^{\text{suggested}} = f_{a_i}^{\text{real}}\right\} \tag{5.3}$$

In Eq. (5.2), $|Learnt(S_i^t)|$ denotes the number of states that the agent learns. As learning is regarded as the highest value of Q in all states, the correct action is chosen using the greedy method. This means that if in the Q-table we consider the rows as states (cells) and the columns as actions (letters), then the word of the highest score will have the highest value in that row in the learning mode. $|S|$ denotes the number of states available for each agent. Figure 6 shows the results of comparing the proposed methods, i.e. PTST, T-MAS, T-KAg with existing methods based on the criterion of group learning rate.

The common denominator of the proposed methods was that in view of the low reward received from the environment, priority was assigned to agent(s) with low TST values so that they were launched and according to the reward that the environment returned to the MAS and increased according to Eqs. (3.9) and (3.10), other agents would be launched (Eq. (4.7)). The difference between the proposed methods becomes apparent when, after this assignment, if the amount of the residual reward is not sufficient to launch another agent, it must be distributed among the active agents. In other words, the difference is in the assignment of the residual reward after the initial assignment. The important point here is that due to the fact that the priority was assigned to agents with low TST values and those agents were also less knowledgeable with low learning rates, the learning rate of the MAS was low in the early episodes. In other words, in the early episodes, there were agents in MAS^t with low learning rates. Consequently, the learning rate of the MAS was low too. After a while, more knowledgeable agents were added to MAS^t according to the mentioned process, and consequently, they increased the learning rate of the MAS For this reason, the learning rate of the MAS in the proposed methods was low up to the middle episodes, but from the middle episodes onwards, it increased upon the entry of more knowledgeable agents and their addition to MAS^t. Given the inclusion of the TST constraint in other methods that seek fair distribution, it was possible that the agents never reach a starting point and do not start to work, which was not the case we are interested in. The next case was when in other methods, the distribution of rewards would launch a number of agents. In other methods, the way of distribution would cause more knowledgeable agents to be placed in MAS^t or not to start to work at all. As a result, the group learning rate would be slow and quite late. In view of the above, and the graph in Figure 6,

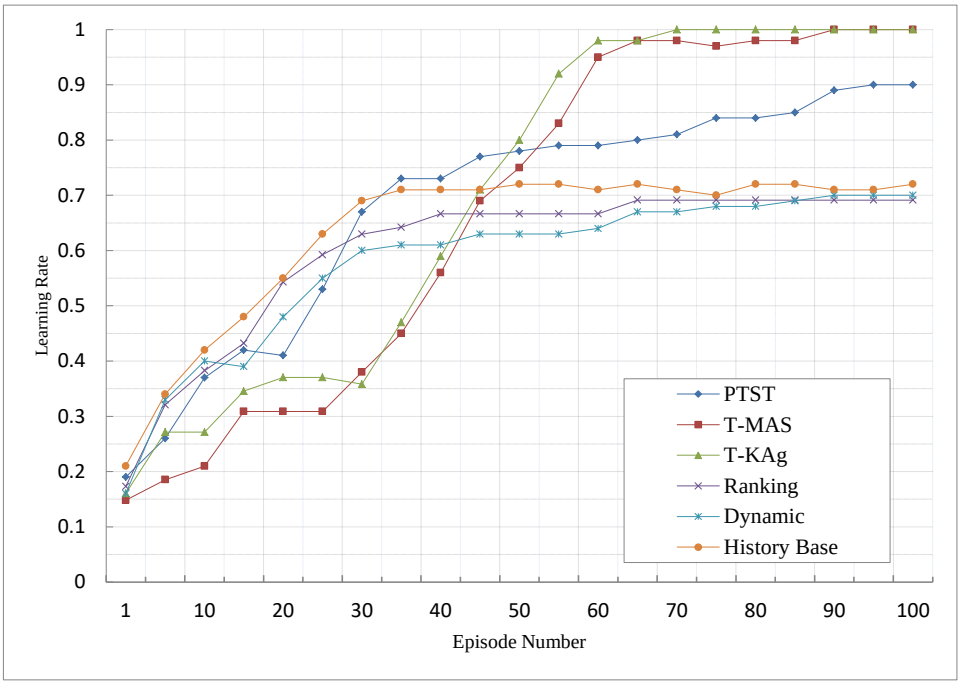

Figure 6: Comparison of six MCA methods based on the criterion of group learning rate.

the intermediate episodes of the proposed methods performed more poorly than the other methods. From the middle episodes onwards, the proposed methods which worked based on the TST criterion exhibited a much better performance than other methods which was due to the entry of agents with higher learning rates into MAS^t. Among the proposed methods, PTST exhibited poorer performance due to the fact that it neglected the residual reward. This was while the other two methods, i.e. T-MAS and T-KAg, caused the entry of the agents with higher learning rates and increased the group learning rate as they distributed the remaining reward among the agents. Nevertheless, from intermediate episodes onwards, all three proposed methods performed better than the history-based method, which had the best performance among the existing methods.

5.2. Confidence

The next criterion according to which the proposed methods were compared to the existing methods was *confidence*. Extracted when completion of the Q-table is in progress, this parameter is obtained by subtracting the second largest value of the Q-table from its largest value. The greater this difference, the more inclined the agent is to choose the appropriate action. If $[\,\mathrm{q}\,(1),\ q\,(2),\ \mathrm{q}\,(3),\ldots\ ,q\,(|A|-1),\ q\,(|A|)]$ are the values in the Q-table, which are in an ascending order, then the confidence of each agent is obtained based on Eq. (5.4):

$$\mathrm{Cnf}\left(S_i^t\right) = q\,(|A|) - q\,(|A|-1) \tag{5.4}$$

Figure 7 illustrates a comparison of the proposed methods with existing methods based on this parameter.

In the proposed methods, if simultaneous launching of agents with low knowledge and agents with high knowledge was possible, as mentioned in the Methodology Section, priority was given to more knowledgeable agents. However, these types of agents also had higher TST values, which meant receiving a higher reward for starting to work. If it was possible to launch these types of agents, they

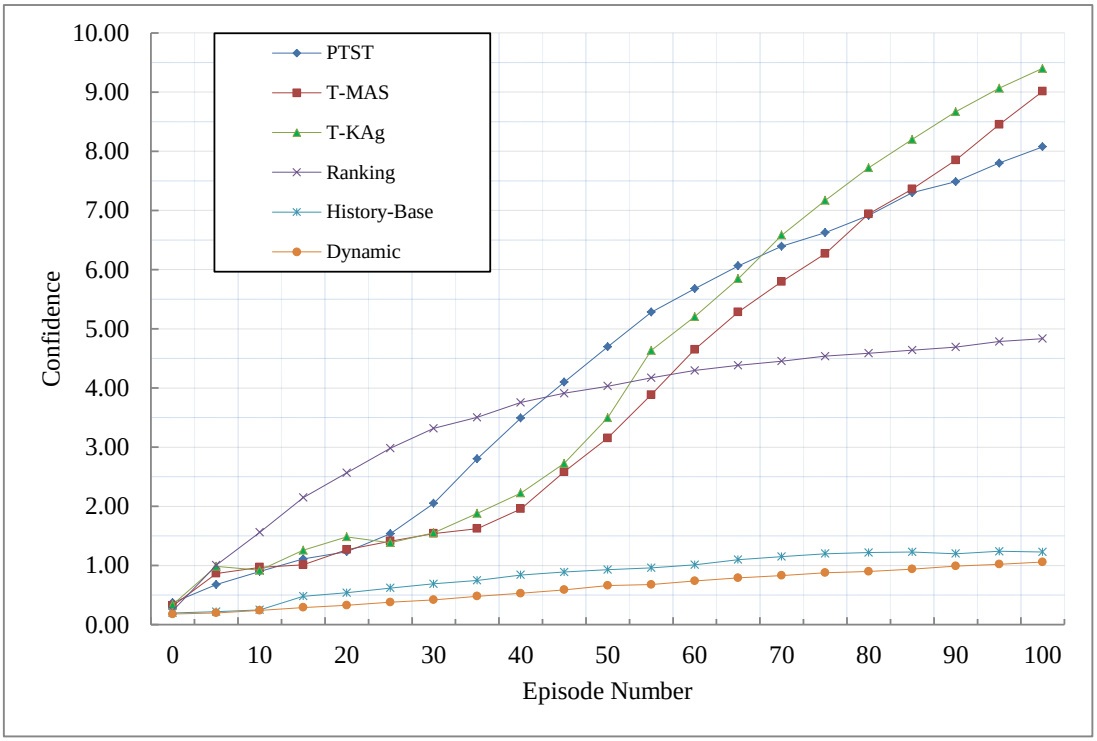

Figure 7: Comparison of six MCA methods based on the confidence criterion.

would obviously receive more rewards and therefore the values in their Q-table would increase, which meant that their confidence would increase. However, in the early episodes, this was not possible due to the low reward received from the environment. Therefore, to get the MAS start to work and to receive more rewards in a gradual manner, priority was given to agents with lower TST values. This caused agents with lower TST values and knowledge to start to work and complete their Q-table. As a result, in the early episodes, when agents with low TST complete their table, the confidence value of MAS was low. Once such agents started to work to do their tasks, they increased the global reward received from the environment and the result was the entry of more knowledgeable agents. More knowledgeable agents allocated higher rewards to themselves because they had higher TSTs, so their Q-table values started to increase at a higher rate. Therefore, after the arrival of agents with knowledge higher than the episode ˜40 onwards, we observed the improvement of the MAS process over this parameter.

Other existing methods had almost uniform linear rates as more knowledgeable agents were less likely to enter the active MAS. Among the existing methods, the ranking method had a better performance up to the intermediate episodes as it is based on the knowledge of the agents; however, due to the fact that from the intermediate episodes the existing agents reached an almost steady state and knowing the way the reward was allocated, no agent with more knowledge would enter MAS^t, this criterion would have had a steady growth rate. Because the proposed methods let more knowledgeable agents enter MAS^t over time and such agents had higher values in their Q-table, they performed better than other methods from intermediate episodes onwards. Among the proposed methods, the PTST method neglected the residual reward. Therefore, the reward for more knowledgeable agents was less than those of T-MAS and T-KAg methods. For this reason, from episode ˜70 onwards, it had a poorer performance than the other two methods.

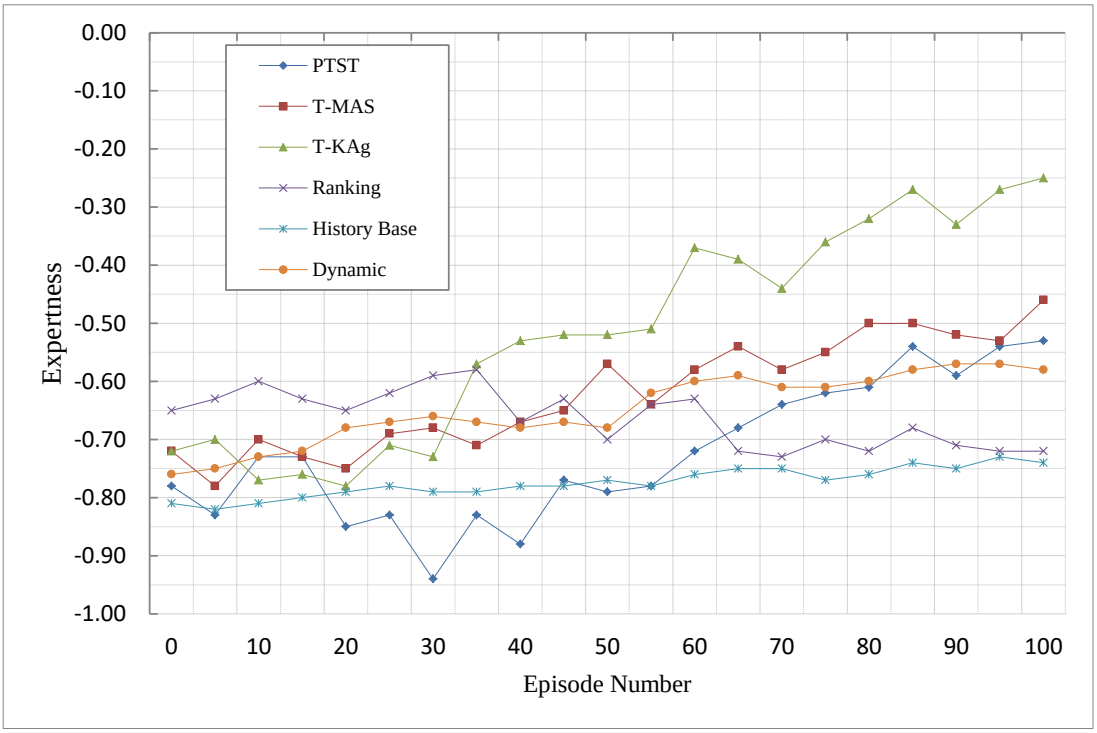

Figure 8: Comparison of six MCA methods based on the expertness criterion.

5.3. Expertness

The next parameter utilized to evaluate the proposed methods was *expertness*, which indicates the difference between the number of times an agent receives a reward (i.e. takes the right action), and the number of times it receives a punishment (i.e. takes the wrong action). Eq. (5.5) expresses this criterion,

$$Expertness = N_r - N_p \tag{5.5}$$

N_r is the number of times the agent receives a reward.

N_p is the number of times the agent receives a punishment.

Figure 8 shows a comparison of the proposed methods with the existing methods based on this criterion.

As mentioned before, in this paper, the TST constraint was used, according to which agents would start to work when their reward was higher than their TST value. Since in most cases this was not the case, the agents did not receive a reward. For this reason, and in view of Eq. (5.5), this graph was often lies below the x-axis. Ranking method, history-based and dynamic methods, allocated a reward to agents if possible and did not differentiate between them. For this reason, the rewards distributed among agents were often not large enough to get the agents start to work; in other words, the rewards granted to them were often less than their TST values, and consequently they received a punishment according to this criterion. Since there was no change in the assignment process in these methods, the diagram process of these methods as illustrated in the Figure 8 was almost steady and linear.

In the proposed methods, due to the fact that less knowledgeable agents were initially less likely to be considered due to their low TST values, priority was given to them. Therefore, if they received

a low reward at least equal to their TST value, they would start to work. These types of agents, given their less knowledge (especially in comparison with more knowledgeable agents) had a poor performance in choosing letters with higher scores. Consequently, in competition with other agents, they would get low-score letters. Because of this, in the early episodes, the reward received was low since most of those agents were in MAS^t even though the reward received by the critic from the environment at time t was lower than at time $t-1$ according to Eq. (3.10). However, the reward received by the critic from the environment might not be enough so that a large number of agents were rewarded according to their TST values. As a result, few agents were rewarded and this slowed down the reward receiving process. This process continued until more knowledgeable agents were placed in MAS^t. At this time, the number of more knowledgeable agents that chose letters of higher values increased, and the reward received from the environment by the critic increased as well. As a result of this increase, more agents were rewarded in subsequent episodes, and the expertness of the MAS increased. This can be observed in the diagram in Figure 8. The proposed methods exhibited an increasing trend up to the middle of the problem solving process but at very low rates. From the middle episodes onwards, these methods took an upward trend with steeper slopes, which indicated the entry of more knowledgeable agents and choosing letters with higher scores, thus increasing the collective reward and increasing the expertness of the MAS. Among the proposed methods, the T-KAg method performed better. This improvement performance was due to the fact that in this method, after the entry of more knowledgeable agents into the set of active agents, the remained reward was assigned to them, so that these agents solved more valuable parts of the problem thus causing more rewards from the environment are received by the critic. This reward increased the assignment of rewards to more agents and therefore increased the level of expertness. Figure 8, which compares the proposed methods with existing methods, confirms this.

5.4. Certainty

The fourth criterion for the evaluation of the proposed methods was *certainty*. This criterion is computed based on Eq. (5.6) and compares the value of Q in the action a and the state s with other values of the state s,

$$\text{Cert}\left(s_i^t, a_i^t\right) = \frac{\exp\left(\frac{Q(s_i^t, a_i^t)}{T}\right)}{\sum_{a_i^t \in A} \exp\left(\frac{Q(s_i^t, a_i^t)}{T}\right)} \tag{5.6}$$

Eq. (5.7) specifies the value of T for each episode,

$$T = Max\left\{\frac{T_0}{1+\log\left(\text{episode}\right)}, T_{\min}\right\} \tag{5.7}$$

Based on [14][23], we set T_0 to 10 and $T_{\min}$ to 1 for our experiments. Figure 9 illustrates a comparison of the proposed methods with other methods based on this criterion.

This criterion is computed based on the values in the Q-table. In the early episodes, the values of this parameter are low, since most of the values in the Q-table are zero –as can be seen in the diagram. In the ranking method (which had the best performance among other methods based on this criterion), due to the TST constraint, the agents will reach the threshold value with a delay and start to work in a multi-agent environment. Therefore, the number of zero values in the Q-table was high and the number of non-zero values was less than those of the proposed methods. For this reason, the performance of other methods was worse than those of the proposed methods according to this criterion. Among our proposed methods, the two methods of T-KAg and T-MAS had higher

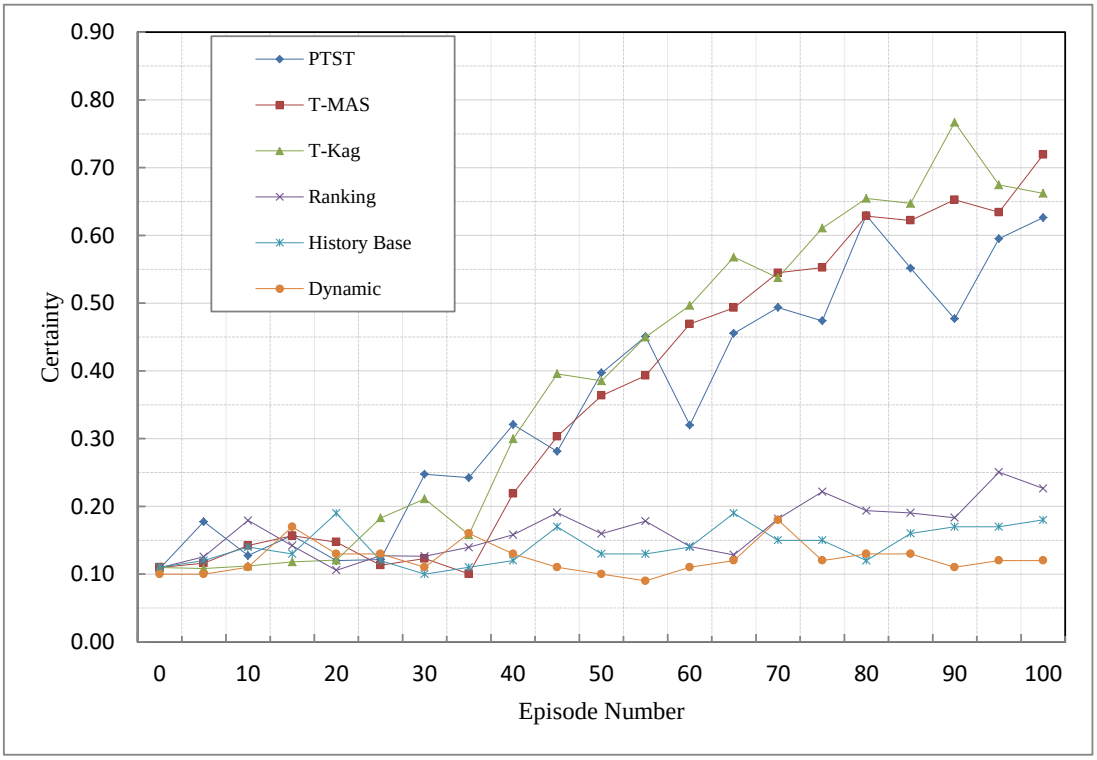

Figure 9: Comparison of six MCA methods based on the certainty criterion.

values in the Q-table because they distributed the reward first based on the TST values and then redistributed it if there was any global reward remained. Therefore they performed better than the PTST method, which neglected the remaining of the global reward.

Furthermore, in the proposed methods, later more knowledgeable agents entered the set of active MAS; thus the values of the Q-tables for the existing agents, which were often less knowledgeable agents, were low and almost close to each other. This continued until more knowledgeable agents entered the middle episodes and increased the correct values in the Q-table due to the higher rewards they received. Therefore, in Figure 9 it is evident that the proposed methods grew in the intermediate episodes much faster than in the early episodes.

5.5. Efficiency

Defined by Eq. (5.8), *efficiency* was the fifth criterion we utilized for the evaluation of the proposed methods. This criterion is an indication of how many times a non-zero reward is allocated to an agent. The process of allocating rewards to agents by the critic has to be conducted conservatively to lead the agent to the goal. This means that if the agent goes astray, it will lose the goal or spend more time to reach it. Therefore, any non-zero assignment to the agent by the critic means that the critic takes such a risk and judges about the agent's choice of action.

$$\text{Eff} = \sum_{i=1}^{F} I\left(r_i^t \neq 0\right) \tag{5.8}$$

$$I\left(x\right) = \begin{cases} 1, & x : \text{True} \\ 0, & x : \text{False} \end{cases}$$

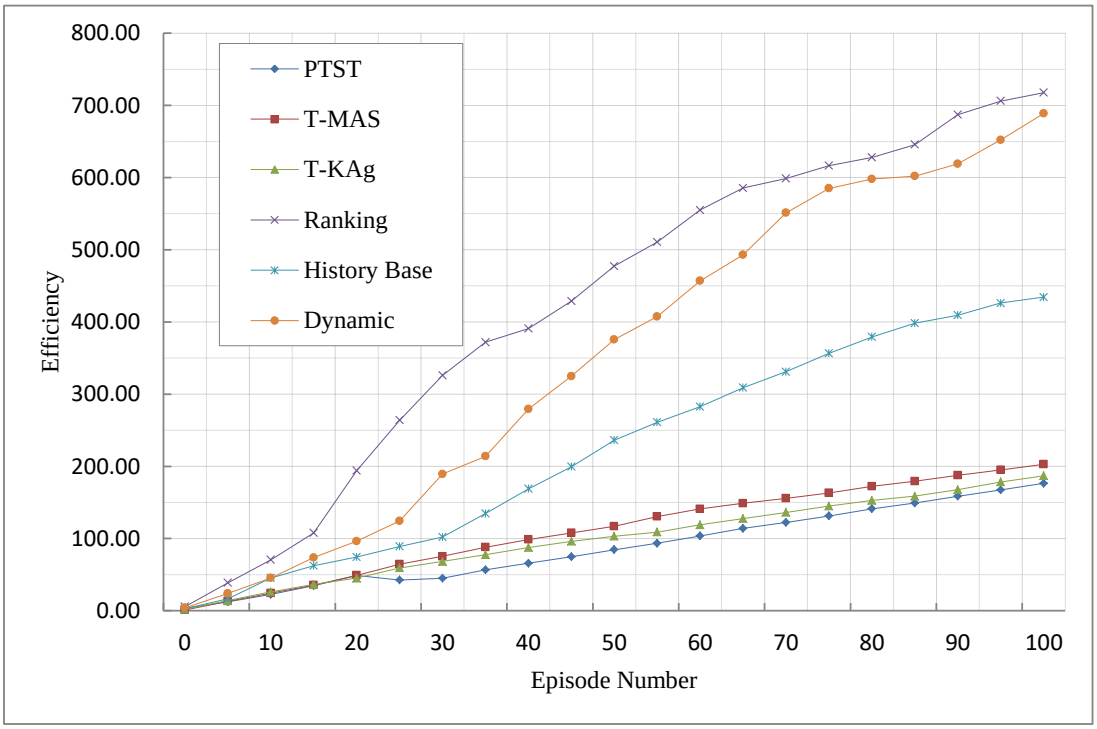

Figure 10: Comparison of six MCA methods based on the efficiency criterion.

F : the number of feedbacks

The results of comparing the methods based on this criterion are illustrated in Figure 10.

The ranking method had a very good performance in comparison with all other methods. Because in the ranking method and the other two methods (history-based and dynamic methods) a partial reward is given which is the result of the distribution of a global non-zero reward, unlike the proposed methods in which a large number of agents are likely not to be rewarded, they performed better than the proposed methods. In other words, in the proposed methods, which are based on the TST constraint, a reward is assigned to an agent if this reward is greater than its TST value. Otherwise, no reward will be allocated. Therefore, the probability of receiving a non-zero reward is decreased. This causes the ranking method (as the best method in the existing methods based on this criterion) to have a much better performance than the proposed methods based on the TST value. Since the proposed methods, as mentioned before, are based on TST, they are less likely to receive non-zero rewards and work very similarly. Moreover, because the frequency of receiving non-zero rewards, and not their amount, is important in this criterion, they will not be much different.

5.6. Correctness

The sixth criterion used for the comparison of the proposed and the existing methods, was *correctness*, which may be expressed based on a variety of criteria. The most flexible definition of correctness is based on the threshold value. If the difference between the assigned reward and the real reward is less than the threshold, this assignment is regarded as a correct assignment, otherwise it is incorrect. This is expressed in Eq. (5.9).

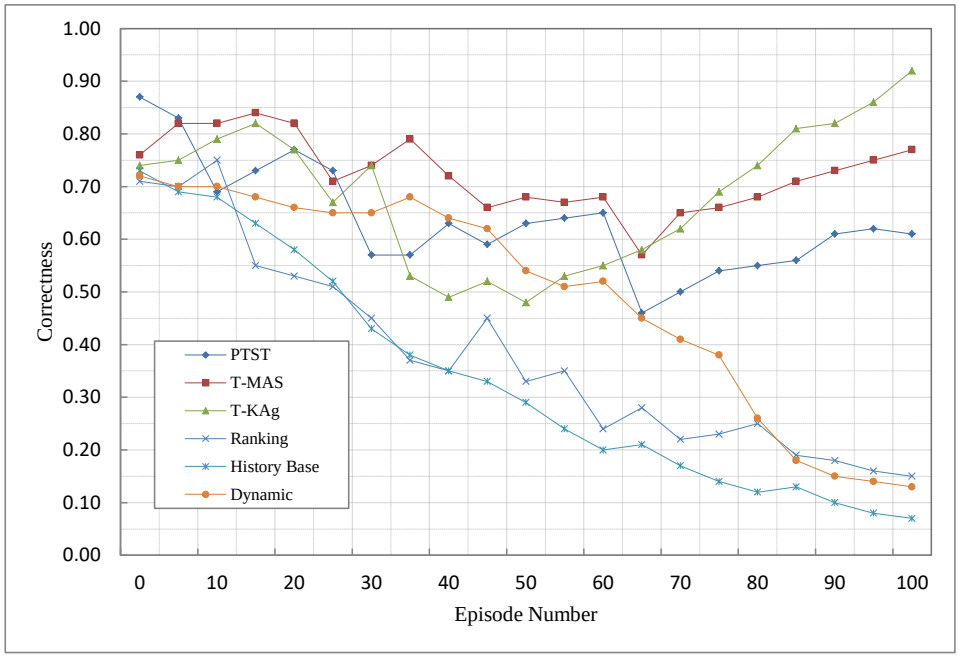

Figure 11: Comparison of six MCA methods based on the correctness criterion.

$$\text{Corr} = \sum_{i=1}^{F} I(\left|r_i^t - r_{i\ \text{cor}}^t\right| < T) \tag{5.9}$$

$$I(x) = \begin{cases} 1, & x : \text{True} \\ 0, & x : \text{False} \end{cases}$$

r_i^t : The done assignment

$r_{i\ \text{cor}}^t$: The actual assignment

F : The Number of Feedbacks

$T : Threshold$

Figure 11 illustrates a comparison of the proposed methods with the existing methods based on this criterion.

In the proposed methods, priority was initially given to agents with lower TST values which were also less knowledgeable. Subsequently, upon the entry of more knowledgeable agents, priority was given to these agents. It should be noted, however, that this assignment was made if the assigned reward was higher than their TST values. Consequently, even if less knowledgeable agents took a correct action, the resulting reward would go to more knowledgeable agents so that they are launched. The important point in Figure 11 is that for the proposed methods in the episodes that happened in the first one-third of time, agents with low knowledge and TST values were working and more knowledgeable agents and with higher TST values had less presence; therefore, in this time period,

i.e. up to episode ˜30, the assignment was made almost correctly and the correctness trend was almost linear. In the middle episodes, i.e. approximately from episodes 30 to 60, owing to the entry of more knowledgeable agents with higher TST values, more assignment was made to them, even though less knowledgeable agents had performed correctly. For this reason, it can be seen that in the middle episodes, the critic's correctness has diminished and the chart is declining. This downward trend continues until more knowledgeable agents start to work on more valuable letters to receive more global reward from the critic. Therefore, from the episode ˜70 onwards, most of the work was done by such agents, and the right action was taken (by the critic) if the critic assigned rewards to them. Therefore, in this graph, it can be seen that from the episode ˜70 onwards, the precision of assignment by the critic increases and the graph takes an upward trend for the proposed methods.

Here, in view of the assignment of rewards to agents and considering that in the existing methods, it was possible to assign rewards to most agents, they acted very similarly to the proposed methods in the early episodes. Subsequently, in the higher episodes, due to the fact that the probability of assigning rewards to some agents decreased and the critic assigned rewards to certain agents, the proposed methods performed better than other methods. Among the available methods, the T-KAg method assigned more rewards to more knowledgeable agent(s). On the other hand, in higher episodes, more knowledgeable agents tried to solve more valuable pieces; as a result, the critic received more global reward from the environment. One could thus say that the most knowledgeable agent deserved the highest reward, which was the case with the T-KAg method. Therefore, in the higher episodes, the T-KAg method performed better than the other proposed methods.

5.7. Density

The next parameter based on which the proposed methods and existing methods were compared was *density*. This parameter is one of the criteria addressed in this paper and was not used in previous works. Because we used MAS to solve the problem, the number of agents that participated in solving the problem was important. If we consider the MAS from a cooperative game perspective, the number of agents involved in solving the problem will be important. In solving the problem by a MAS, we are faced with two sets of agents, i.e. the whole set of agents (MAS) and the set of active agents (MAS^t) so that the set of active agents is a subset of the whole set of agents, that is

$$\mathrm{MAS}^t \subseteq \mathrm{MAS}$$

Since the number of active agents is n and the total number of agents is N, at any time, t, density can be defined as in Eq. (5.10),

$$Dens = \frac{n}{N} \tag{5.10}$$

The higher is the density, the greater is the participation of the agents in problem solving. The figure 12 shows a comparison of the existing methods with the proposed methods based on this criterion.

In the proposed methods, it was attempted to enter and launch other agents, including ones with high knowledge and high TST values by launching agents with low TST values. For this reason, initially priority was given to agents with low TST values. All methods started to work with one agent and then according to the allocation mode by the critic and taking into account the TST values of other agents, the entry of agents into the set MAS^t takes place. In the proposed methods, the focus was on the entry of other agents; therefore, in these methods, agents were gradually added to the set MAS^t according to the assignment method, while in other methods, due to the fact that all

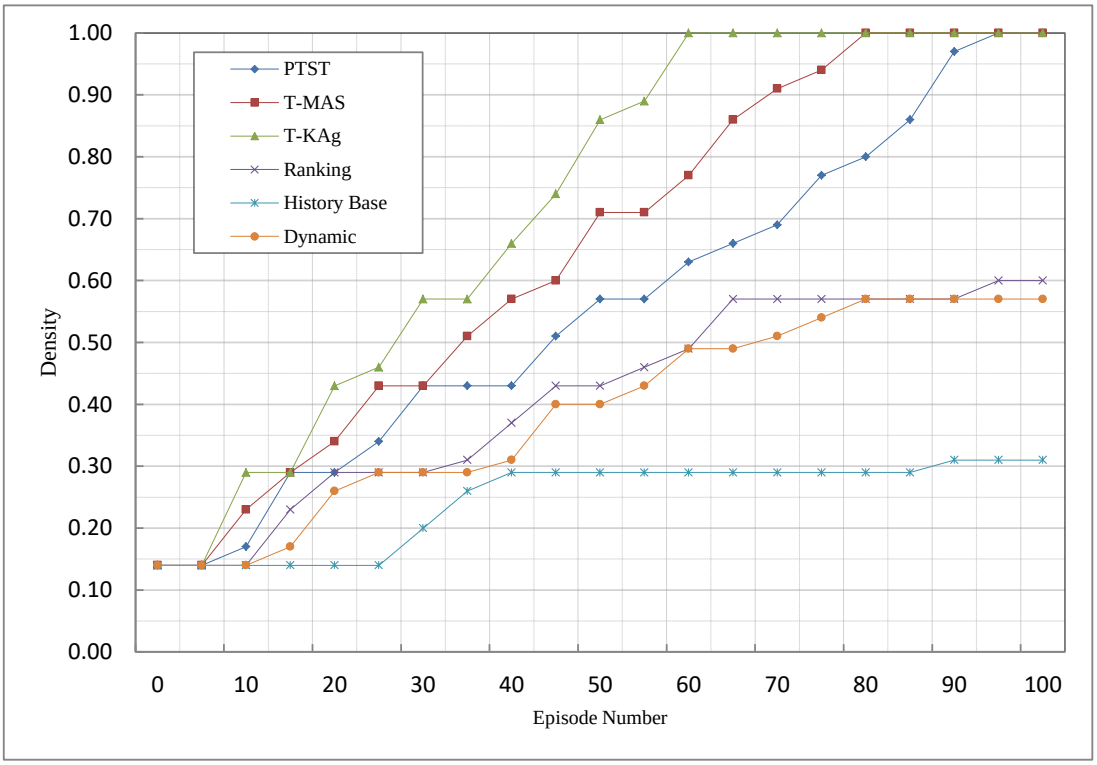

Figure 12: Comparison of six MCA methods based on the density criterion.

agents are rewarded, it is possible that many agents that have not reached their TST and therefore do not fall into the set MAS^t. Therefore, the density of these methods was lower than those of the proposed methods. Among the proposed methods, T-MAS and T-KAg worked better than PTST. This is because in the PTST method, if the residual reward is neglected after the assignment, less reward is assigned to the active agents than in the other two methods, and therefore the probability of launching more knowledgeable agents and consequently gaining higher scores is decreased. However, in the other two methods, the remaining reward is distributed among the agents based on their own formulas.In the T-KAg method, due to the assignment of the whole reward to more knowledgeable agents, the more valuable letters of the problem were solved so that more scores were gained by the critic and distributed among the agents; this additional reward (compared to the previous time) caused other agents to start and fall into MAS^t thus increasing the density. Therefore, this method worked a little better than the T-MAS method, which distributed the remaining reward according to the amount of TST values of the agents which are directly related to their knowledge.

6. Discussion

One of the tools for solving problems as bottom-up is MAS. MASs are used to solve many problems such as traffic control, complex system modeling, resource allocation in pandemics such as Covid-19, cyber physical systems and cyber security, etc.

One of the strengths of these systems is their ability to be located in unknown environments and learn in these environments. This learning often takes the form of RL that is based on receiving rewards and punishments. These rewards / punishments are the result of agents interacting with the environment and performing actions correctly or incorrectly.

In a single-agent system, this reward / punishment is assigned entirely to one agent, but in a

MAS, the outcome of the individual agent's reward / punishment is returned to the MAS in the form of a single vector called global reward that should be distributed among the agents.

How to distribute this global reward among agents is a challenging problem. Two general approaches to solve this problem can be considered, which are the fair approach and the efficiency increase approach. Most of the works done in this field were looking for a fair approach and the approach of increasing efficiency has received less attention. Therefore, from this perspective, there seems to be a gap between the MCA and increasing the efficiency of the MAS.

In this paper, in order to fill this gap and solve the MCA in a way that increases the efficiency of the MAS, three methods were presented. These methods were based on the concept of bankruptcy. Bankruptcy occurs when the amount of resources available is less than the total amount requested. In this paper, a new constraint called TST was introduced. This constraint means that each agent will start to work, if the received reward is more than a certain amount, otherwise it will not start to work. The concept based on which this paper was formed given the above mentioned points is that agents of high knowledge may not start to work because of having a high TST. Therefore, in this paper, priority was given to agents with low TST values in spite of having knowledge. This will help MAS start to work by launching such agents, even though with a limited number of them, so that they launch other agents, including more knowledgeable ones and get them enter the set MAS^t by receiving more rewards than ever even to a little extent.

7. Conclusion and Future works

7.1. Conclusion

In order to assess the proposed methods and compare them to the existing methods, i.e. the ranking method, history-based method and dynamic method, seven criteria were used, which included a new criterion of *density* indicating the ratio of the number of agents in the set of active agents MAS^t to the total agents. Other parameters according to which the methods proposed in this paper were evaluated were group learning rate, confidence, expertness, certainty, efficiency, and correctness.

In the comparison of the proposed methods with the existing ones based on the parameter of group learning rate, the proposed methods had poorer performances than the others because the bankruptcy concept was used and less knowledgeable agents with less TST values were given priority up to intermediate episodes; however, from intermediate episodes onwards, due to launching more knowledgeable agents with less TST values a much better performance, approximately 70% compared to other methods, was observed. Among the proposed methods, the T-KAg method performed better than other proposed methods due to the allocation of the residual reward to more knowledgeable agents. In terms of confidence, the proposed methods had lower performances than other ones. This could be attributed to the use of the bankruptcy concept and giving priority to agents with less TST values to launch other agents up to the middle episodes. From the middle episodes onwards, with the entry of more knowledgeable agents to the problem solving process, the performance of the proposed methods and particularly the T-KAg method improved in comparison with that of the ranking method as the best method among others in this criterion. This improvement was about twice as much as the ranking method. Another criterion according to which the proposed methods were compared with the existing ones was expertness. From the perspective of this criterion, the proposed methods performed quite close to each other, but better than the existing ones. However, this improvement was more evident from the intermediate episodes. Among the existing methods, the dynamic method performed better than the others, with a performance quite close to that of the PTST method; however, in the lower episodes, the PTST method exhibited a better performance.

Among the proposed methods, the best performance was observed for the T-KAg with an enhancement of about twice in comparison with the dynamic method as the best available method. The next criterion according to which our proposed bankruptcy-based methods were compared with the existing ones was certainty; the performances of the proposed methods were much better than those of the other methods (about 3.5 times), even though in the early episodes, up to episode ˜30, all methods performed very closely.

The only criterion based on which the proposed methods performed worse than the existing methods was efficiency; this was due to the fact that the criterion used the number of times the agents were rewarded, which was in contradiction with the very nature of the methods that worked on the basis of prioritizing agents. In the sixth criterion, i.e. correctness, when we divided the episodes into three parts of early, middle and final, we could contend that in the early episodes, the proposed methods had almost downward trends but with low slopes due to the use of less knowledgeable agents with low TST values. Subsequently, with the entry of more knowledgeable agents of higher TST values, the distribution of rewards by the critic using the existing methods was done in a wrong way from the perspective of this criterion; therefore, in the middle episodes, the downward trend took a steeper slope. In the final episodes, due to the fact that most of the work was done by more knowledgeable agents, the graph improved and an upward trend was observed. In all these episodes, the proposed methods performed better than the others, so that in the last episode, the performance of the T-KAg method as the best proposed method was about 9 times better than that of the dynamic method among other methods.

The last criterion studied was density, which we introduced in this paper. In this criterion, the number of active agents was considered. In the proposed methods, first agents with low TST values were launched and according to the way the rewards were allocated in these methods, other inactive agents were gradually added to the set of active agents. Consequently, it was observed that in these methods, the number of active agents gradually increased, so that by episodes ˜60-70, all agents were active. This is while in other existing methods, the reward allocation process is a fixed process and therefore from the middle episodes onwards, no new agent is added to the set of active agents.

7.2. Future works

In this paper, the concept of bankruptcy was used to solve the MCA problem, and based on that, we presented the reverse adjusted proportional bankruptcy method to solve this problem. Since in this research the TST was used to get closer to the real situation, other bankruptcy methods along with considering the TST constraint may be used to solve this problem. This method can be used in many areas and the problem solving method presented in this paper can be used to solve them in future tasks. For instance, one can map an organization or company to MAS and use this method as a way of allocating salaries to the personnel according to their performances. As a future proposal for solving the MCA problem, the critic can be trained using RL methods.

References

[1] van Steen, Maarten, and Andrew S. Tanenbaum. *A brief introduction to distributed systems,* Computing 98(10)(2016) 967-1009.

[2] Yadav, Satya Prakash, Dharmendra Prasad Mahato, and Nguyen Thi Dieu Linh, eds. *Distributed Artificial Intelligence: A Modern Approach,* CRC Press, 2020.

[3] Vlassis, Nikos. *A concise introduction to multiagent systems and distributed artificial intelligence,* Synthesis Lectures on Artificial Intelligence and Machine Learning 1(1)(2007) 1-71.

[4] Qadir, Muhammad Zuhair, Songhao Piao, Haiyang Jiang, and Mohammed El Habib Souidi. *A novel approach for multi-agent cooperative pursuit to capture grouped evaders,* The Journal of Supercomputing 76(5)(2020) 3416-3426.

[5] Habibi, M., Broumandnia, A., Harounabadi, A. *An Intelligent Traffic Light Scheduling Algorithm by using fuzzy logic and gravitational search algorithm and considering emergency vehicles,* International Journal of Nonlinear Analysis and Applications, 2020; 11(Special Issue) 475-482. doi: 10.22075/ijnaa.2020.4706

[6] Kazemi, A., Shiri, M., Sheikhahmadi, A., Khodamoradi, M. *A new parallel deep learning algorithm for breast cancer classification,* International Journal of Nonlinear Analysis and Applications, 2021; 12(Special Issue) 1269-1282. doi: 10.22075/ijnaa.2021.24247.2702

[7] Li, Xueyan, and Hankun Zhang. *A multi-agent complex network algorithm for multi-objective optimization,* Applied Intelligence (2020) 1-28.

[8] Challenger, Moharram, and Hans Vangheluwe. *Towards employing ABM and MAS integrated with MBSE for the lifecycle of sCPSoS,* In Proceedings of the 23rd ACM/IEEE International Conference on Model Driven Engineering Languages and Systems: Companion Proceedings, pp. 1-7. 2020.

[9] Amiri, Ehsan, Mina Rahmanian, Saeed Amiri, and Hadi Yazdani Praee. *Medical images fusion using two-stage combined model DWT and DCT,* International Advanced Researches and Engineering Journal 5(3)(Under Construction) (2021) 344-351.

[10] Asadi, Mehrdad, Mahmood Fathy, Hamidreza Mahini, and Amir Masoud Rahmani. *An Evolutionary Game Approach to Safety-Aware Speed Recommendation in Fog/Cloud-Based Intelligent Transportation Systems,* IEEE Transactions on Intelligent Transportation Systems (2021).

[11] Panait, Liviu, and Sean Luke. *Cooperative multi-agent learning: The state of the art,* Autonomous agents and multi-agent systems 11(3)(2005) 387-434.

[12] Wang, Ruyan, Xue Jiang, Yujie Zhou, Zhidu Li, Dapeng Wu, Tong Tang, Alexander Fedotov, and Vladimir Badenko. *Multi-agent reinforcement learning for edge information sharing in vehicular networks,* Digital Communications and Networks (2021).

[13] Al-Dayaa, H. S., and D. B. Megherbi. *Reinforcement learning technique using agent state occurrence frequency with analysis of knowledge sharing on the agent's learning process in multiagent environments,* The Journal of Supercomputing 59(1)(2012) 526-547.

[14] Harati, Ahad, Majid Nili Ahmadabadi, and Babak Nadjar Araabi. *Knowledge-based multiagent credit assignment: A study on task type and critic information,* IEEE systems journal 1(1)(2007) 55-67.

[15] Airiau, Stéphane. *Cooperative games and multiagent systems,* The Knowledge Engineering Review 28(4)(2013) 381-424.

[16] Jing, Shoucai, Fei Hui, Xiangmo Zhao, Jackeline Rios-Torres, and Asad J. Khattak. *Cooperative game approach to optimal merging sequence and on-ramp merging control of connected and automated vehicles,* IEEE Transactions on Intelligent Transportation Systems 20(11)(2019) 4234-4244.

[17] Wang, Zeng, Bo Hu, Xin Wang, and Shanzhi Chen. *Cooperative game-theoretic power control with a balancing factor in large-scale LTE networks: an energy efficiency perspective,* The Journal of Supercomputing 71(9)(2015) 3288-3300.

[18] Meng, Yan. *Multi-robot searching using game-theory based approach,* International Journal of Advanced Robotic Systems 5(4)(2008) 44.

[19] Chang, Yu-Han, Tracey Ho, and Leslie P. Kaelbling. *All learning is local: Multi-agent learning in global reward games,* (2004).

[20] Guisi, Douglas M., Richardson Ribeiro, Marcelo Teixeira, Andre P. Borges, and Fabricio Enembreck. *Reinforcement learning with multiple shared rewards,* Procedia Computer Science 80 (2016) 855-864.

[21] Bagnell, Drew, and Andrew Ng. *On local rewards and scaling distributed reinforcement learning,* Advances in Neural Information Processing Systems 18 (2005) 91-98.

[22] Rădulescu, Roxana, Manon Legrand, Kyriakos Efthymiadis, Diederik M. Roijers, and Ann Nowé. *Deep multi-agent reinforcement learning in a homogeneous open population,* In Benelux Conference on Artificial Intelligence, pp. 90-105. Springer, Cham, 2018.

[23] Rahaie, Zahra, and Hamid Beigy. *Critic learning in multi agent credit assignment problem,* Journal of Intelligent and Fuzzy Systems 30(6)(2016) 3465-3480.

[24] George, Marcus L. *Effective teaching and examination strategies for undergraduate learning during COVID-19 school restrictions,* Journal of Educational Technology Systems 49(1)(2020) 23-48.

[25] Tisdell, Clement A. *Economic, social and political issues raised by the COVID-19 pandemic,* Economic analysis and policy 68 (2020) 17-28.

[26] Arias, Michael, Rodrigo Saavedra, Maira R. Marques, Jorge Munoz-Gama, and Marcos Sepúlveda. *Human resource allocation in business process management and process mining: A systematic mapping study,* Management Decision (2018).

[27] Boonpeng, Sabaithip, and Piyasak Jeatrakul. *Decision support system for investing in stock market by using OAA-*

neural network, In 2016 Eighth International Conference on Advanced Computational Intelligence (ICACI), pp. 1-6. IEEE, 2016.

[28] Wakilpoor, Ceyer, Patrick J. Martin, Carrie Rebhuhn, and Amanda Vu. *Heterogeneous Multi-Agent Reinforcement Learning for Unknown Environment Mapping,* arXiv preprint arXiv:2010.02663 (2020).

[29] Calvo, Jeancarlo Arguello, and Ivana Dusparic. *Heterogeneous Multi-Agent Deep Reinforcement Learning for Traffic Lights Control,* In AICS, pp. 2-13. 2018.

[30] Cripps, Martin, and Norman Ireland. *The design of auctions and tenders with quality thresholds: the symmetric case,* The Economic Journal 104(423)(1994) 316-326.

[31] Peiró, José M., Sonia Agut, and Rosa Grau. *The relationship between overeducation and job satisfaction among young Spanish workers: The role of salary, contract of employment, and work experience,* Journal of applied social psychology 40(3)(2010) 666-689.

[32] O'Neill, Barry. *A problem of rights arbitration from the Talmud,* Mathematical social sciences 2(4)(1982) 345-371.

[33] Castro, Leyre, and Edward A. Wasserman. *Animal learning,* Wiley Interdisciplinary Reviews: Cognitive Science 1(1)(2010) 89-98.

[34] Busoniu, Lucian, Robert Babuska, and Bart De Schutter. *A comprehensive survey of multiagent reinforcement learning,* IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews) 38(2)(2008) 156-172.

[35] Jiang, Yuqian, Sudarshanan Bharadwaj, Bo Wu, Rishi Shah, Ufuk Topcu, and Peter Stone. *Temporal-Logic-Based Reward Shaping for Continuing Learning Tasks,* arXiv preprint arXiv:2007.01498 (2020).

[36] Sutton, Richard Stuart. *Temporal credit assignment in reinforcement learning,* PhD diss., University of Massachusetts Amherst, 1984.

[37] Yu, Zhong, Gu Guochang, and Zhang Rubo. *A new approach for structural credit assignment in distributed reinforcement learning systems,* In 2003 IEEE International Conference on Robotics and Automation (Cat. No. 03CH37422), vol. 1, pp. 1215-1220. IEEE, 2003.

[38] Mao, Wenji, and Jonathan Gratch. *The social credit assignment problem,* In International Workshop on Intelligent Virtual Agents, pp. 39-47. Springer, Berlin, Heidelberg, 2003.

[39] Skinner, Burrhus Frederic. *The behavior of organisms: An experimental analysis,* BF Skinner Foundation, 2019.

[40] Foerster, Jakob, Gregory Farquhar, Triantafyllos Afouras, Nantas Nardelli, and Shimon Whiteson. *Counterfactual multi-agent policy gradients,* In Proceedings of the AAAI Conference on Artificial Intelligence, vol. 32(1. 2018.

[41] Foerster, Jakob, Gregory Farquhar, Triantafyllos Afouras, Nantas Nardelli, and Shimon Whiteson. *Counterfactual multi-agent policy gradients,* In Proceedings of the AAAI Conference on Artificial Intelligence, vol. 32(1. 2018.

[42] Wang, Jianhong, Yuan Zhang, Tae-Kyun Kim, and Yunjie Gu. Shapley Q-value: a local reward approach to solve global reward games. In Proceedings of the AAAI Conference on Artificial Intelligence, vol. 34(05, pp. 7285-7292. 2020.

[43] Sunehag, Peter, Guy Lever, Audrunas Gruslys, Wojciech Marian Czarnecki, Vinicius Zambaldi, Max Jaderberg, Marc Lanctot et al. *Value-decomposition networks for cooperative multi-agent learning,* arXiv preprint arXiv:1706.05296 (2017).

[44] Son, Kyunghwan, Daewoo Kim, Wan Ju Kang, David Earl Hostallero, and Yung Yi. Qtran: Learning to factorize with transformation for cooperative multi-agent reinforcement learning. In International Conference on Machine Learning, pp. 5887-5896. PMLR, 2019.

[45] Rahaie, Zahra, and Hamid Beigy. *Expertness framework in multi-agent systems and its application in credit assignment problem,* Intelligent Data Analysis 18(3)(2014) 511-528.

[46] Even-Dar, Eyal, Sham M. Kakade, and Yishay Mansour. *Experts in a Markov decision process,* Advances in neural information processing systems 17 (2005) 401-408.

[47] Ma, Chris YT, David KY Yau, Xin Lou, and Nageswara SV Rao. *Markov game analysis for attack-defense of power networks under possible misinformation,* IEEE Transactions on Power Systems 28(2)(2012) 1676-1686.

[48] Levy, Yehuda John, and Eilon Solan. *Stochastic games,* Complex Social and Behavioral Systems: Game Theory and Agent-Based Models (2020) 229-250.

[49] Mhatre, Manasi, Sakshi Nagaonkar, Sminil Shirsat, and Pournima Kamble. *Scrabble Game Using Java,* International Journal of Progressive Research in Science and Engineering 2(5)(2021) 114-116.

[50] Curiel, Imma J., Michael Maschler, and Stef H. Tijs. *Bankruptcy games,* Zeitschrift für operations research 31(5)(1987) A143-A159.

[51] Bergantiños, Gustavo, Leticia Lorenzo, and Silvia Lorenzo-Freire. *A characterization of the proportional rule in multi-issue allocation situations,* Operations Research Letters 38(1)(2010) 17-19.

[52] Hagiwara, Makoto, and Shunsuke Hanato. *A strategic justification of the constrained equal awards rule through a procedurally fair multilateral bargaining game,* Theory and Decision 90(2)(2021) 233-243.

[53] Lorenzo, Leticia. *The constrained equal loss rule in problems with constraints and claims,* Optimization 59(5)(2010) 643-660.