IACSIT International Journal of Engineering and Technology, Vol.3, No.1, February 2011
ISSN: 1793-8236

Implementing a Decision Support Module in Distributed Multi-Agent System for Task Allocation Using Granular Rough Model

Sally M. El-Ghamrawy, Ali I. El-Desouky and MostafaSaleh

***Abstract*—A Multi-Agent System (MAS) is a branch of distributed artificial intelligence, composed of a number of distributed and autonomous agents. In MAS, an effective coordination is essential for autonomous agents to reach their goals. Any decision based on a foundation of knowledge and reasoning can lead agents into successful cooperation, so to achieve the necessary degree of flexibility in coordination, an agent requires making decisions about when to coordinate and which coordination mechanism to use. The performance of any MAS depends directly with the right decisions that the agents made. Therefore the agents must have the ability of making right decisions. In this paper, we propose a decision support module in a distributed multi-agent system, which enables any agent to make decisions needed for Task allocation problem; we propose an algorithm for Task Allocation Decision Maker (TADM) based on Granular Rough Model (GRM). Furthermore, a number of experiments were performed to validate the effectiveness of the proposed algorithm (TADM)); we compare the efficiency of our algorithms with recent frameworks. The preliminary results demonstrate the efficiency of our algorithms**

***Index Terms*—Decision Making, Task allocation, Coordination Mechanism, Multi-Agent System (MAS), Rough Set, Granular Computing**

I. INTRODUCTION

The notion of distributed intelligent systems (DIS) [1] has been a subject of interest for number of years. A Multi-Agent system (MAS) is one of the main areas in the DIS. Any Multi-agent system consists of several agents capable of mutual interaction, with heterogeneous capabilities, that cooperate with each others to pursue some set of goals, or to complete a specific task. MAS used to solve problems which are difficult or impossible for an individual agent or monolithic system to solve. Agents are autonomous programs which can understand an environment, take actions depending upon the current status of the environment using its knowledge base and also learn so as to act in the future. In order to solve complex problems agents have to cooperate and exhibit some level of autonomy. Agents cooperate with each other to solve large and complex collaborative problems. Because the majority of work is completed through distributing the tasks among the cooperative agents, the decision support module is responsible on the most of the work. So the pros and cons in decision support module directly affect on the success of tasks completion and problem solving. In [2], we proposed a Distributed Multi-Agent Intelligent System (DMAIS,) a general purpose agent framework, in which several interacting intelligent agents cooperate with some auxiliary agents to pursue some set of goals or tasks that are beyond their individual capabilities. There are some modules must be considered in designing the Distributed Multi-Agent Intelligent System (DMAIS) that help in developing in the agent systems. There are many researches considered some modules when designing multi-agent systems (MAS) in different ways. Each autonomous agent in DMAIS must be able to decide how to behave in various situations, so in this paper our main concern is to propose an efficient decision support module that helps in the improvement of DMAIS performance.

Agents have attractive characteristics like: autonomy, reactivity, reasoning capability and social ability. These characteristics ensure that agent-based technologies are responsible for enhancing the decision support system capabilities beyond the capabilities of the old model. Any active decision support can be facilitated by the autonomy, reactivity and social ability of agents. Furthermore, the artificial view of agents can contribute towards stronger collaborative relationships between a human and a decision support system. These enormous interests of researchers in investigating the decision support in Multi-Agent Systems is due to the great benefits of combining the decision support technology and Agent-based technology with taking the advantage of the agent characteristics. In this sense, many researches [3-7] recognized the promise that agent -based technologies holds for enhancing DSS capabilities.

Decision support module is a vital module in the success of DMAIS framework, due to the fact that any decision based on a foundation of knowledge and reasoning can lead agents into successful cooperation. So the performance of any multi-agent system depends directly with the right decisions that the agents made. Obviously, complex decision making tasks cannot be achieved by a single agent. Rather, it's achieved by efforts coordination of multiple agents possess different sets of expertise, attributes and assignment. This coordination among agents, which provide satisfactory solutions to problems among agents, needs many decisions that agents are required to make before this coordination can take place. So the Decision support module role is to allow agents to make the decision needed, which can help in the improvement of the DMAIS framework. In this paper, we

Sally M. El-Ghamrawy is with the Computers and Systems Department,Faculty of Engineering,Mansoura University, Egypt(email: Sally@mans.edu.eg)

Ali I. El-Desouky is with the Computers and Systems Department,Faculty of Engineering,Mansoura University, Egypt(email: ali_eldesouky@yahoo.com)

MostafaSaleh is with the Computers and Systems Department,Faculty of Engineering,Mansoura University, Egypt

extended the work done in DMAIS [2] by proposing a Decision support module in it, this module is concerned about taking the right decisions needed to allocate a specific task to a specific agent, by using the Task Allocation Decision Maker sub-module (TADM). So we spotted on these decisions due to the great importance for them in the improvement of agents' coordination in DMAIS framework. The rest of the paper is organized as following. Section 2 demonstrates the proposed decision support module. Section 3 discusses the related work on task allocation in Multi-agent Systems. An algorithm for the Task Allocation Decision Maker (TADM) is proposed in section 4, showing the scenario of our algorithm in allocating tasks to specific agent/s. section 5 shows the proposed granular rough model that used in TADM. Section 6shows the experimental evaluation and the results obtained after implementing the proposed algorithm. Section 7 summarizes major contribution of the paper and proposes the topics for future research.

II. Decision Support Module

In the DMAIS framework [2], the decision support module takes the information collected in the coordination and negotiation module, as shown in fig 1, so that it can help in the decision support process.

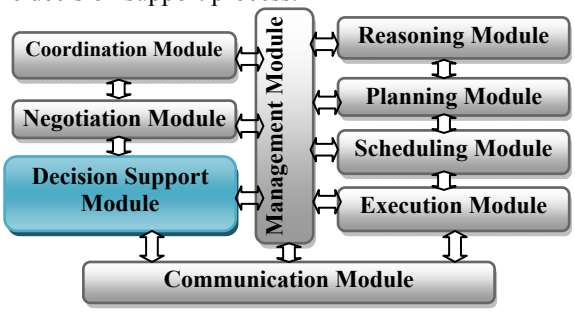

Figure 1. The Decision Support Module in DMAIS Framework

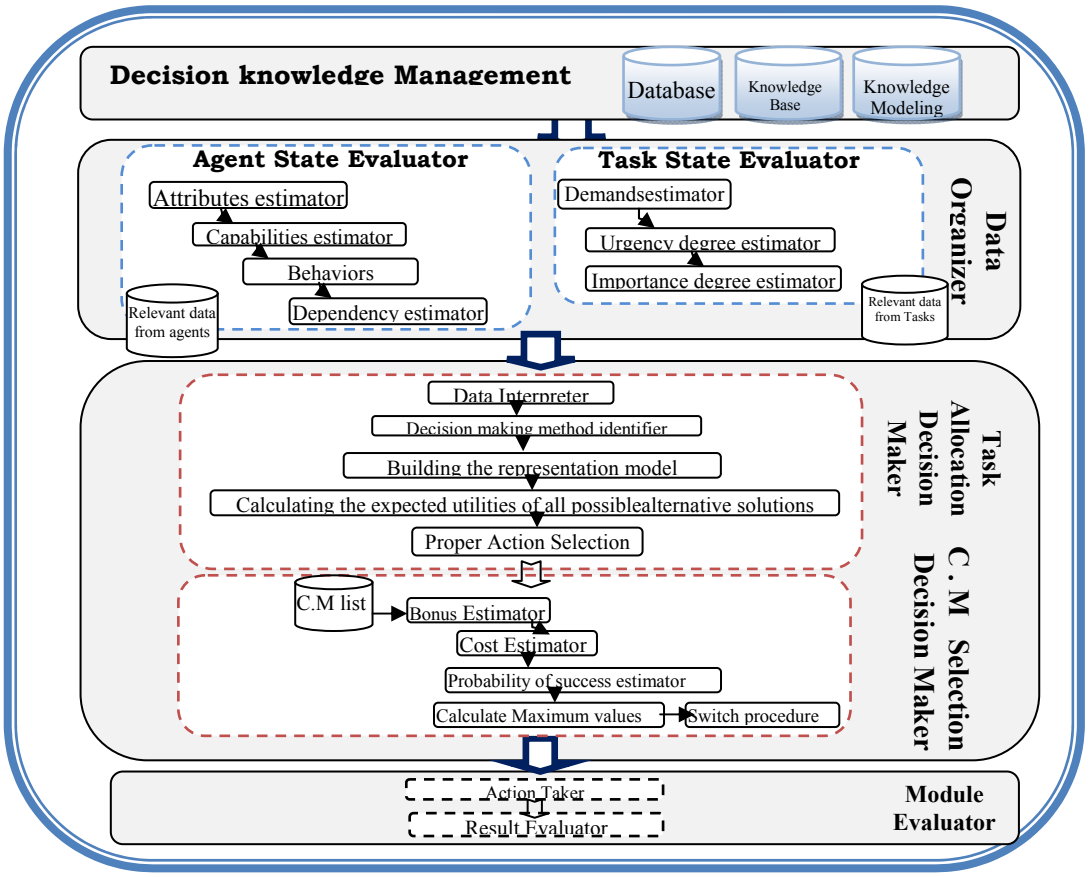

Figure 2. Architecture of the Decision Support Module

The Decision support module is activated when a decision is needed from the agent, we concerned about two main decisions that the agent may face: (1) the decision needed to allocate a specific task to a specific agent, making an effective decision for the task allocation problem is a critical job for any multi-agent system; it helps our DMAIS framework to complete its tasks and missions through cooperation among agents. The Task Allocation Decision Maker sub-module (TADM) is responsible for making this decision. (2) The decision needed to select appropriate coordination mechanism, when agents need to coordinate with another agent/s to accomplish a specific task.

The decision support module contains four main phases, as shown in figure 2, first is the decision knowledge management that contains:

- *The database*: contains the data that directly related to the decision problem (i.e. the performance measures, the values of nature states).
- The *knowledge base*: contains the descriptions for roles and structures of document and some knowledge of the problem itself, (i.e. Guide for how to select decision alternatives or how to interpret outputs).
- *The knowledge modeling*: is a repository contains the decision problem formal models and the algorithms and methodologies for developing outcomes from the formal models. Also contains different process models,

each model is represented as a set of process and event objects.

Second phase is the data organizer, which organizes agent's attributes using the agent state evaluator and organizes task's preferences by using task state evaluator The agent state evaluator estimates the attributes and capabilities of agents, in our work we concerned on specific attributes and characteristics of the agent that will help in the decision making procedure, as shown in fig 3, where:

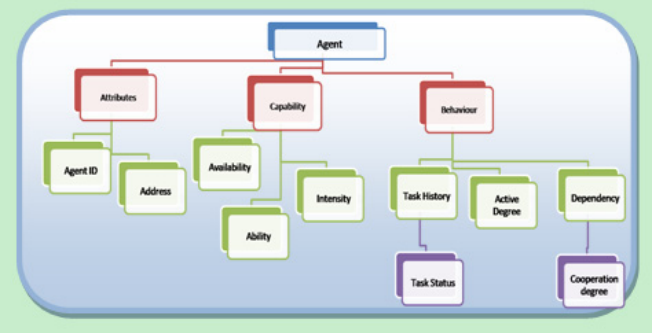

Figure 3. Attributes, Capabilities and behaviors of Agents

Agent attributes is defined as a tuple

$$<AgentID(A), Addrs(A)> \quad (1)$$

Where AgentID(A) is the identity of the agent, Addrs(A) records the IP address of agent A.Agent Capability is defined as a tuple

$$<Availability(A), Ability(A), Intensity(A)> \quad (2)$$

Where Availability(A) is the ratio of total number of successful agents, capable of accomplishing the task, to the total number of agents in the system,

$$AVLB(T)= SUC(A) / TOT (A) \quad (3)$$

Ability(A) indicates number of agents capable of fulfilling the task.Intensity(A) refers to the number of tasks that the agents can accomplish per time unit.

Agent Behaviors is defined as a tuple

$$< Task History(A), Active degree(A), Dependency(A)> \quad (4)$$

Where task history(A) stores the number of accomplished tasks that the agent(A) performed. Active degree(A) indicates the activity degree of a specific agent, this degree varies from agent to another according to many parameters (e.g. number of finished tasks, number of cooperation process).Agent Dependency(A) is concerned with the Cooperation Degree of an Agent(A) with respect to other agents in the system.

Third phase is divided into two main Sub-modules: (1)The Task Allocation Decision Maker (TADM) ,this sub-module will be discussed in details in the next sections, and (2) The Coordination Mechanism Selection Decision Maker (CMSDM), that takes the decision needed to select appropriate coordination mechanism, when agents need to coordinate with another agent/s to accomplish a specific task, it will be discussed in our future work. Both of them are responsible on making the proper decisions according to the input obtained from the Data organizer phase. The module evaluator phase is the fourth phase, it has two main goals: first, take the action that the third phase decided to be taken. Second, evaluate the results of taking this action.

III. Related Work

The task allocation in Multi-Agent Systems represent a problem which occupied to a large extends the researchers of decision support and artificial intelligence until our days. Tasks allocation is defined, in [8], as the ability of agents to self-organize in groups of agents in order to perform one or more tasks which are impossible to perform individually. In our context, it is a problem of assigning responsibility and problem solving resources to an agent. There are two main benefits for Minimizing task interdependencies in the coordination process between the agents: First, improving the problem solving efficiency by decreasing communication overhead among the agents. Second, improving the chances for solution consistency by minimizing potential conflicts. The issue of task allocation was one of the earliest problems to be worked on in Distributed Artificial Intelligence (DAI) research. In this sense, several authors studied the problem related to Task Allocation especially in MAS. The researches in task allocation can be classified in to two main parts, centralized and distributed, based on utility/cost functions.

The researches that investigate the task allocation problem in a centralized manner: Zheng and Koenig [9] presented reaction functions for task allocation to cooperativeagents. The objective is to find a solution with a small teamcost and each target to be assigned to the exact number ofdifferent agents. This work assumed that there is a centralplanner to allocate tasks to agents. Kraus et al. [10] proposed an auction based protocol which enables agents to formcoalitionswith time constrains. This protocol assumed eachagent knows the capabilities of all others, and one manageris responsible for allocating tasks to all coalitions. Pinedo[11] proposed a job shop scheduling treats the task allocation mostly in a centralized manner, and also ignores the communication cost. There are many drawbacks in the centralized task allocation like single point failure and bad scalability. To conquer these disadvantages, Task allocation in distributed environments has also been investigated. Davis and Smith [12] was the first in investigating a classic distributed task allocation in the multi–agent system using Contract Net Protocol (CNP), in which agents negotiate to assign tasks among themselves. Most of the subsequent literature on distributed task allocation is based on either contract net protocol or auctions [13]. The authors in [14] and [15] developed distributed algorithms with low communication complexity for forming coalitions in large-scale multi-agent systems. Abdallah and Lesser [16] provided a decision theoretic model in order to limit the interactions between agents and mediators. Mediators in this research mean the agents which receive the task and have connections to other agents. Mediators have to decompose the task into subtasks and negotiate with other agents to obtain commitments to execute these subtasks. However, their work concentrated on modelling the decision process of a single mediator. Sander et al. [17] presented a scalable and distributed task allocation protocol. The algorithm adopted in this protocol is based on computation geometry techniques but the prerequisite of this approach is that agents' and tasks' geographical positions are known. Weerdtet al. [18] proposed a distributed task allocation protocol in social networks. This protocol only allows neighbouring agents to help with a task which might result in high probability of abandon of tasks when neighbours cannot offer sufficient resources. Dayong et al. [19]

proposed an Efficient Task Allocation Protocol (ETAP) protocol based on the Contract Net approach, but more suitable for dealing with task allocation problems in P2P multi-agent systems with a decentralized manner. It enables agents to allocate tasks not only to their neighbours but also to commit unfinished tasks to their neighbours for reallocation. In this way, the agents can have more opportunities to achieve solution of their tasks Brahmi et al. [20] developed a decentralized and scalable method for complex task allocation for Massive Multi-Agent System, distributing the process of computing the optimal allocation among all agents based on the hypothesis: non conflict will be generated in the task allocation processes. Indeed, while being based on its Galois Sub-Hierarchy (GSH) and cooperation with other agents, each agent chooses the appropriate sub-task that ensures the global allocation optimality. Cheng and Wellman [21] used a market based protocol for distributed task allocation.

IV. Task Allocation Decision Maker (TADM)

The task allocation Decision Maker (TADM) main goal is to take the proper decisions of allocating tasks to the right agents. The first step in allocating tasks to specific agents is to take the decision whether the agents are capable of executing part or the entire task. The allocation decisions are made independently by each agent

The decisions needed for the Task allocation problem are mainly concerned about allocating the tasks to number of agents, whether these agents can complete its tasks by themselves or not. If agents can't achieve the task by themselves, they attempt to give a decision to specify other agents which have the appropriate capabilities and assign the task, or part of the task, to those agents. Fig 4 depicts the scenario of allocating tasks to specific agent/s in the TADM.

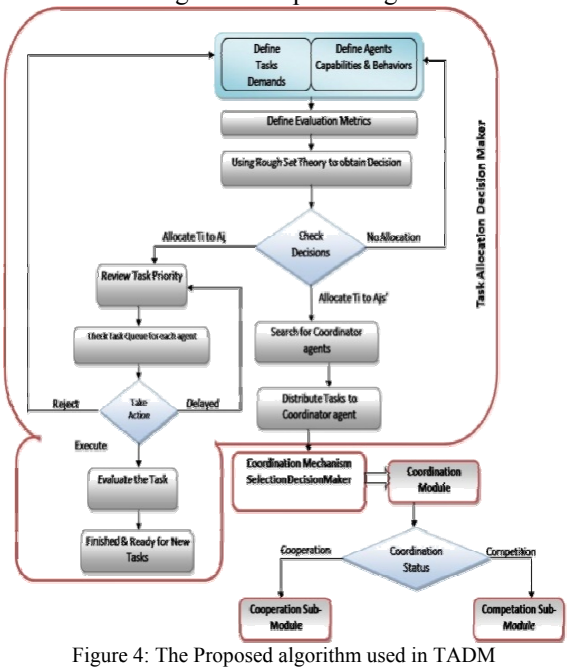

Figure 4: The Proposed algorithm used in TADM

First the capabilities and behaviors of agents must be defined, and also the demands of tasks, it's taken from the data organizer as an input to the TADMThe metrics that facilitate the decision making process must be evaluated, and then we used the rough set theory for classifying these metrics to obtain the decision of allocating tasks to agent/s. Then this decision is tested, it may be one of three decisions: first, if the allocation cannot be done, in this situation the capabilities of agents and demands of tasks must be defined all over again, it may encounter any kind of error. second, if the decision is to allocate the task to only one agent, then task priority must be reviewed to check the task queue for each agent, and depend on this queue an action must be taken whether to delay, reject or execute this task. Third, if the decision is to allocate the task to more than one agent, then the coordinated agents must registered, and the sub-tasks must be distributed among those coordinated agents according to the algorithm in figure 5.

TADM algorithm

```
1.  Agent Ai randomly select task T
2.  AinCharge= Ai
3.  For each a(i) in G
4.                  Ai=Send request ( )
5.                  Ai=Wait response( )
6.    If time waited > expired time then
7.                        Exit for
8.                        End if
9.                  Ai=Receive response( )
10.                 Ai=Process response( )
11.                 Ai=Store response()
12.                 Max-value=0
13.   If helpfulness-value> Max-value then
14.                       Max-value = helpfulness-value
15.                       Nominated-agent= a(i)
16.                       End if
17. Next
18. Xx:
19. Ai=Check (Nominated-agent( ))
20. If helpfulness-value(Ai)< Nominated-agent then
21. Ai= send Response(Nominated-agent)
22.   If reply (Nominated-agent) =1 then
23.                     AinCharge= Nominated-agent
24.                     Else
25.                     Ai= find_scnd_highst( )
26.                     Go to xx
27.                     End if
28. Else
29. AinCharge= Ai
30.    End if
```

Figure 5: The Pseudo code for TADM algorithm

V. Granular Computing And Rough Sets Theory

Decision-making is a difficult process due to factors, such as information about incompleteness, imprecision, and subjectivity which tend to be present in real-life situations to lesser or greater degree. In addition, each agent in any Multi-Agent Systems has incomplete information or capabilities for solving the problem, thus it has a limited viewpoint and has no control on the system. These factors indicate that the decision making in Multi-Agent System can best take place in a fuzzy environment. Theories of fuzzy sets [23-25] and rough sets [26,27] have been diagnosed to model these vagueness and uncertainty, and manipulating the imperfect knowledge. In decision support module, there must be some kind of support to extract the knowledge and represent any uncertainty that might occur. This fact motivated us to use the GrC models with RST tools to analysis the data in decision support module. The main goal of using Granular Rough theory in our proposed decision support module is to combine the advantages of using RST concepts with using the benefits of granulations. Figure 6 shows the proposed Granular Rough Model (**GRM**), the concepts used in Granular Rough theory in the decision support module, is also outlined, to implement the TADM algorithm. There are four main concepts used in **GRM** model, each concept illustrated in details in the next

subsections, showing the algorithms proposed within them.

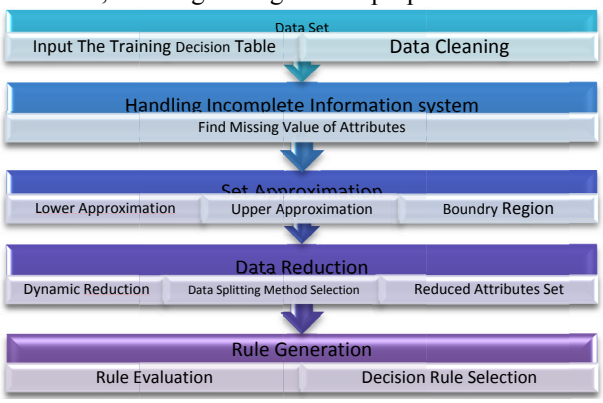

Figure 6: Granular Rough Model (GRM) Used in Decision Support Module

1. Decision Table

The first phase in **GRM** model presented is the data set. The data set used is represented in a table called information system in which each row in the table represents a case (e.g. the agent record) and each column represents an attribute. Objects are attracted by Granular rough theory as knowledge expression system, a Multi-Agent information system can be formulated as:

$$I_s = < A_g, U, A, V, f > \quad \ldots (1)$$

Where A_g is set of agents viewpoints, U is non-empty set of objects called the universe, A is non-empty finite set of attributes and V is called value set of attributes A and f is information function. In our context, I_s is expressed as a Multi-Agent information table, whose columns are labeled by attributes and rows are labeled by objects of interest. The entries of the table are attribute values.

Rough set theory base on philosophy of classifications so information system should be expressed by dividing non-empty finite set of attributes A into two subsets condition attribute C and decision attribute D (this process is called supervised learning) and information system in this case called decision table. A decision table is an information system in the form is given by:

$$DT = \{U, C \cup \{D\}, V\} \quad \ldots (2)$$

Where C is a non-empty finite set set of condition attributes that characterizes a decision category and $D \notin C$ is decision attribute or the thematic featuresuch that $c : U \rightarrow V_c$ for every $c \in C$, V_c is a value of attribute c. This is achieved by means of information granulation or indiscernibility is at the heart of rough set theory. A finer granulation means more definable concept. For $R \subseteq C$ the granule of knowledge about a forest with respect to indiscernibility relation can be represented as: $I_s = < U, A, V >$

$$IND_{IS}(R) = \{(x, x') \in U^2 \mid \forall c \in Rc(x) = c(x')\} \quad \ldots (3)$$

Where $IND_{IS}(R)$ is called the R-indiscernibility relation. If $(x, x') \in IND_{IS}(R)$, then objects x and x' are indiscernible from each other by attributes from R. The equivalence classes of the R-indiscernibility relation are denoted by $[x]_R$. The notion of indiscernibility is fundamental to RST. Informally, two objects in a decision table are indiscernible if one cannot distinguish between them on the basis of a given set of attributes. Hence, indiscernibility is a function of the set of attributes under consideration. The indiscernibility relation for a given attribute set is mathematically an equivalence relation. The equivalence relation is a binary relation $R \subseteq X \times X$ which is:Reflexive (xRx for any object x) or Symmetric (if xRy then yRx) or Transitive (if xRy and yRz then xRz).In the DMAS proposed in the previous chapter, suppose U denotes the whole MAS. When an agent needs a decision to be taken, the proposed decision support module proposed is activated, and then the attributes used in TADM algorithms to describe the process of taking the decision for TADM algorithms and the decision attributes D = {d} is illustrated in tables 1.

For the TADM algorithm, suppose U_1 denotes the TADM algorithm, there are nine attributes form an equivalent relation:

$U_1 : R = \{RTa_1, RTa_2, RTa_3, Rac_1, Rac_2, Rac_3, Rab_1, Rab_2, Rab_3\}$

TABLEI :THE ATTRIBUTES MEANING FOR TASKS AND AGENTS AND DECISION

Attribute Value	Demands RTa1	Urgency RTa2	Importance RTa3
1	Basic	Urgent	High
2	Average	Normal	Normal
3	Expert	Low	Low

Attribute Value	Availability Rac1	Ability Rac2	Intensity Rac3
1	High	High	Satisfactory
2	Medium	Medium	Medium
3	Low	Low	unsatisfactory

Attribute Value	Task History Rab1	Active Degree Rab2	Cooperation Degree Rab3
1	Expert	High	High
2	Average	Normal	Medium
3	Poor	Low	Low

Attribute Value	Decision Attributes D
1	Allocate Ti to Aj
2	Allocate Ti to Ajs'
3	Cant allocate

For TADM algorithm, the decision can be one of three possibilities: (1) The agent decides to allocate the task completely. (2) The agent decides that it can allocate only a part of the task. (3) The agent decides that it can't allocate the task at all. According to task allocation situations, a sample of decision table for TADM algorithms is obtained, as shown in table 2.In real life, there are lots of incomplete information systems. It means that sometimes the attribute values for an object are missing. RST introduced many ways to handle these incomplete information systems IIS. This concept is used in second phase of **GRM** model using the concept of a granulation because it gives a great help in MAS.This because each agent in any MAS may has incomplete information or capabilities for solving the problem.

TABLE II: DECISION TABLE FOR TADM ALGORITHMS

U1	RT a1	RT a1	RT a3	Rac 1	Rac 2	Rac 3	Rab 1	Rab 2	Rab 3	D
1	1	1	1	1	1	1	1	1	1	**1**
2	1	1	1	1	1	2	1	1	2	**1**
3	1	1	1	1	1	3	1	1	3	**2**
4	1	1	1	1	1	1	1	1	1	**2**
5	1	1	1	2	1	2	1	1	2	**1**

Thus sometimes the information for an agent can be incomplete. The agents can be processed at the level of information granules that is the most suitable from their local points of view. Information granules can come at various levels of granularity. Each agent could exploit a certain formalism of information granulation engaging with RST.So there must be an algorithm to deal with missing or null attributes. In this subsection a Missing Value Estimator (**MVE**) algorithm is proposed to handle the IIS based on

information granulation. Missing or null values will increase the uncertainty of information system and the induced rules will not be trusted. These missing values are represented by the set of all possible values for the attribute. To indicate such a situation, a distinguished value, a so-called null value is usually assigned to those attributes. Let $I_s = < U, A, V >$ is the information system, for every $a \in A$, there is a mapping a, $a{:}U \rightarrow V_a$, where V_a is called the value set of a. If $a \in A$contains a null value (will be denoted as *) for at least one attribute $a \in A$, then $I_s$is called an IIS, otherwise it is complete information systems. Recently, the researchers presented many ways to handle the IIS; these researches can be classified into two main approaches :(1) *Indirectly handle the IIS(Data Reparation)*: This approach is to transforming an IIS to a complete system using estimation methods (probabilistic and statistical techniques). These estimation methods mainly work as estimating the null value based on the appearing frequency of other values with same attribute. The value gained by these methods maybe not archive the best efficiency to the classification of decision attribute because it changes the original information and lead to a lower support degree and confidence degree. That is to say, this approach is to replace unknown value of attributes by either specific subsets of values or statistical values [28-30]. (2) *Directly handle the IIS (Model Extension)*: This approach is to extend the concepts of RST on complete information systems for handling IIS. It does not require the changes in the original system and still is capable of reducing dispensable knowledge efficiently. By using knowledge reduction that eliminates only the information which is not essential from the point of view of classification or decision making or by relaxing the requirement of indiscernibility relation of reflexivity, symmetry and transitivity, i.e., the indiscernibility relation is extended to inequivalence relations that can process IISs directly [31-35]. A Missing Value Estimator (**MVE**) algorithm is proposed, as shown in figure 7, to handle the IIS based on information granulation. Suppose that $I_s = < U, A, V >$is the IIS and let the attribute set is$R \subseteq C$. Based on Kryszkiewicz , in order to deal directly with IIS we can define the tolerance relation as follows:

$$SIM\ (R) = \{(u, v) \in U \times U | \forall\ c_i \in R, c_i(u) = c_i(v) \vee c_i(u) = * \vee c_i(v) = *\} \ldots (4)$$

Let $SIM_{R_i}(U_i)$ express the information granular for the biggest object satisfy the condition.Divide the information table into granules to i condition attributes. If the one of the i condition attribute values of the object is *, it will not be divided into any granules. Information granularity means we can observe and analysis the same problem from very different granularities. GQ_R is called the information granularity of information system $I_s = < U, A, V >$for the $R \subseteq C$.

$$Q_R = \{SIM_1(U_1), SIM_2(U_2), \ldots . SIM_{R_i}(U_{|u|})\} \ldots (5)$$

Then the definition of information granularity GQ_R will be:

$$GQ_R = \frac{1}{|U|^2} \sum_{i=1}^{|U|} |SIM_{R_i}(U_i)| \ \ldots (6)$$

Missing Value Estimator (MVE) Algorithm Based on Information Granulation

Input: Associate $R_i = c \cup c_i \quad c = \emptyset$
I=1,2,……n where n is the number of condition attributes of information table
$I_s = \{c_1, c_2, \ldots . c_n\}$
Output: Fill the missing values in the information table.

```
1.   R = {R1, R2, ….Rn}
2.   xx:
3.   Calculate SIM_Ri(Ui)
4.   Calculate GQ_R
5.   Foreach c in ci
6.   {
7.   Select c= cx
8.   }
9.   Next
10.  Yy:
11.  Foreach c in cx
12.  {   Find cx (Ui) = *
13.  Foreachcx (Uj) = *
14.  X=Get CandValue(SIM_Ri(Ui))
15.  Temp(k) = Get NullValue(cx (Ui))
16.  ForeachTemp(k)
17.      Calculate Qulty(c) in Predct(d)
18.  Next
19.  MaxValueOf(Qulty(c))
20.  Go to yy
21.  }
22.  Next
23.  }
24.  Next
25.  Assign C = cx , Ri = c ∪ ci,  I=1,2….n , R={R1, R2, ….Rn}
26.  Is = Is − cx
27.  IfIs = ∅then
28.      Exit
29.  Else
30.      Go to mm
31.  End if
```

Figure 7: The Pseudo Code of MVE Based on Information Granulation

2. Set Approximation

As mentioned in the introduction of RST, set approximation is the kernel concept in RST. The starting point of RST is the indiscernibility relation, generated by information about objects of interest. The indiscernibility relation is intended to express the fact that due to the lack of knowledge we are unable to discern some objects employing the available information. That means that, it is a must to consider clusters of indiscernible objects, as fundamental concepts of RST because there is no ability to deal with single objects. The equivalence classes of the indiscernibility relation are the basic building blocks from which sets of cases or objects can be assembled. Sets of interest to assemble in a supervised learning setting would typically be the sets of cases with the same value for the outcome variable. The set approximation concept add a great benefit in applying it in the decision support module , this is because any decision about the presence or absence of a given category is approximated by lower and upper approximation of decision concept.

In this extension, a notion of granulation order is introduced by using the Granular rough theory, in third phase of **GRM** model, to propose a Lower approximation Generator (**LAG**) algorithm and Upper approximations Generator (**UAG**) algorithm, these approximations of a target set under a granulation order is defined in an incomplete information system. Suppose that $I_s = < U, A, V >$ is IIS and $R \subseteq C$. And as showed in the previous sub-section, $SIM_{R_i}(U_i)$ is a tolerance relation on U, It can be easily shown that:

$$SIM_{R_i}(U_i) = \cap_{c \in R}\ SIM_{R_i}\{c\} \ldots (7)$$

Let $Max_{R_i}(U_i)$ denote the set $\{v \in U | (u, v) \in SIM_{R_i}(U_i)\}$,

$\mathrm{Max_{R_i}(U_i)}$ is the maximal set of objects which are possibly indistinguishable by R with u. Let $U/SIM_{R_i}(U_i)$ denote the family sets $\{Max_{R_i}(U_i)|u \in U\}$, the classification or the knowledge induced by R. A member $\mathrm{Max_{R_i}(U_i)}$ from $U/SIM_{R_i}(U_i)$ will be called a tolerance class or a granule of information. It should be noticed that, in general the tolerance classes in $\mathrm{U/SIM_{R_i}(U_i)}$do not constitute a partition of U, but they constitute a cover of U, i.e., $\mathrm{Max_{R_i}(U_i)} \neq \emptyset$ for every $\mathrm{u \in U}$, and $\mathrm{\cup_{u \in U} Max_{R_i}(U_i) = U}$. In an incomplete information system, a cover $U/\mathrm{SIM_{R_i}(U_i)}$ of U induced by the tolerance relation $\mathrm{SIM_{R_i}(U_i)}$, provides a granulation world for describing a concept X. So a sequence of attribute sets $R_i \in 2^n (n = 1, 2... N)$can determine a sequence of granulation worlds, from the most rough one to the most fine one. Based on RST approximations, lower and upper approximations are generated under a granulation order in IIS in the proposed **LAG** and **UAG** algorithms, as shown in figure 8(a) and 8 (b), respectively.

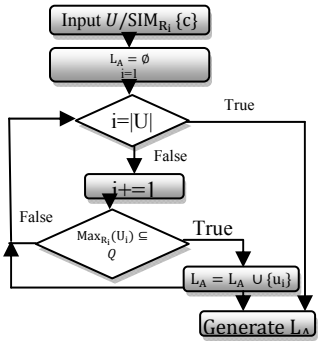

Figure 8 (a): LAG algorithm

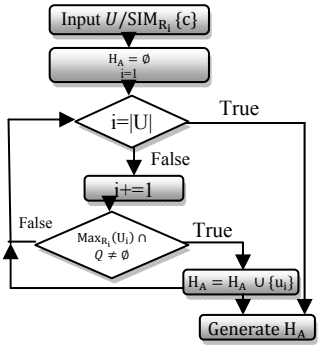

Figure 8(b): UAG algorithm

Let Q is subset of U, $\mathrm{Q \subseteq U}$, $c \in C$, and $U/\mathrm{SIM_{R_i}}\{c\}$,that denote classification under granulation in IIS ,is defined as:

$$U/\mathrm{SIM_{R_i}}\{c\} = \{\mathrm{Max_{R_{c1}}(u_1), Max_{R_{c2}}(u_2) \ldots, Max_{R_{|c|}}(u_{|U|})}\} \ldots (8)$$

The lower and upper approximation to Q from$\mathrm{U/SIM_{R_i}(c_i)}$ is defined as:

$$\mathrm{lower} = \mathrm{R_{*_{ci}}Q} = \{\mathrm{u \in U | Max_{R_{ci}}(u_i) \subseteq Q}\} \ldots\ldots\ldots (9)$$

$$\mathrm{Upper} = \mathrm{R^*{}_{ci}Q} = \{\mathrm{u \in U | Max_{R_{ci}}(u_i) \cap Q \neq \emptyset}\} \ldots (10)$$

The R-boundary region of Q from $\mathrm{U/SIM_{R_i}(c_i)}$equals the difference between the upper and lower approximations.

$$BN_{ci}(Q) = \mathrm{R^*{}_{ci}Q - R_{*_{ci}}Q} \ldots (11)$$

3. Data Reduction Phase

Attribute reduction can be viewed as the strongest and most characteristic results in RST to distinguish itself from other theories. Since, in real world it is difficult to identify all possible causal attributes, it will be necessary to establish a methodology to identify the critical attributes by eliminating redundant attributes using feature reduction or knowledge compression methods in rough set knowledge systems. This can be achieved by removing the attributes whose removal will not change the indiscernibility relation. Attribute reduction aims to find minimal subsets of attributes, each one of them has the same discrimination power as the entire attributes. A minimal subset of attributes that enables the same classification of elements of the universe as the whole set of attributes is called a reduction, if it cannot be further reduced without affecting the essential information [36]. The RST provides different methods for finding which attributes separates one class or classification from another. RST generates reduced information table, which implies that the number of evaluation criteria is reduced with no information loss through rough set approach. And then, this reduced information is used to develop classification rules.

The main objective of this section is to propose an Attribute Reducts Generator algorithm ARG, based on information granulation, to be embedded in the fourth phase of GRM model. Based on RST, a lot of attributes reduction algorithms has been developed recently [37], which can be classified into two categories: (1) Attribute reduction from the view of algebra [38,39]. (2)Attribute reduction from the view of information [40,41]. Algebra view only focuses on the positive region of a decision system, while information view also tries to keep the distribution of inconsistent instances in attribute space besides that. In other words, information view restricts attribute reduction more rigorously. Therefore, a reducible attribute in information view is also reducible in algebra view, but not vice versa. The challenging issues of these methods are multiple computations for equivalent classes and huge number of objects.

Although algebra and information view's definitions are different [39], both of them need to compute equivalent classes. To compute the equivalent classes, let the decision table is DT= (U, C $\cup${D}, V) where $\mathrm{d \notin c}$ is a distinguished attribute called decision. The value set of conditions c is $\{c_1, c_2, \ldots . c_i\}$, the value set of decisions d is $\{d_1, d_2, \ldots . d_j\}$, and The value set of objects o is $\{o_1, o_2, \ldots . o_k\}$ .The notation $c_i(o_k)$ is used to represent the value of a condition attribute ci $\in$ C for an object $o_k \in$ U. Similarly, the notation $d_j(o_k)$ represents the value of the decision attribute d for an object o_k . The concept of indiscernibility relation is adopted to partition the object set U into disjoint subsets, this partition that includes o_k is denoted by $[o_k]_R$, denoted by U/R or $\mathrm{IND_R}$, as shown:

$$\mathrm{IND_R = U/R} = \{[o_k]_{R/o_k} \in U\} \ldots \ldots\ldots\ldots\ldots (12)$$

Where

$$[o_k]_{R/o_k} = \{o_k \,|\forall c \in R_{c_i(o_k)} = c_i(o_k)\} \ldots (13)$$

The equivalence classes based on the decision attribute are denoted by$\mathrm{U/\{d_j\}}$, as shown:

$$\mathrm{IND_{\{d_j\}} = U/\{d_j\} = \{[o_k]_{\{d_j\}} | o_k \in U\}} \ldots (14)$$

As described in equation (9) and (10), the objects in $\mathrm{R_{*_{ci}}Q}$ can be classified with certainty as members of Q on the basis of knowledge in R, while the objects in $\mathrm{R^*{}_{ci}Q}$ can be only classified as possible members of Q on the basis of knowledge in R. $\mathrm{Pos_{c_i}(\{d_j\})}$ is the positive region of the partition $\mathrm{U/\{d_j\}}$ with respect to C, it can be defined as the set of all elements of U that can be uniquely classified to blocks of the partition $\mathrm{U/\{d_j\}}$.

$$\mathrm{Pos_{c_i}(\{d_j\}) = \cup_{Q \in U/\{d\}} R_{*_{ci}}Q} \ldots\ldots\ldots (15)$$

A reduct is a minimal subset of attributes from C that preserves the positive region and the ability to perform classifications as the entire attributes set C. A subset $\mathrm{R \subseteq C}$ is a reduct of C with respect to d_j, iff R is minimal, and:

$$\mathrm{Pos_R(\{d_j\}) = Pos_C(\{d_j\})} \ldots \ldots\ldots\ldots (16)$$

The core is the most important subset of attributes; it is

included in every reduct. The Core corresponding to this part of information cannot be removed without loss in the knowledge that can be derived from it, it can be defined as:

$$\mathrm{Core}(\mathrm{A}, \mathrm{d_j}) = \cap\, \mathrm{RED}(\mathrm{A}, \mathrm{d_j}) \ldots\ldots\ldots\ldots (17)$$

Where $\mathrm{RED}(\mathrm{A}, \mathrm{d_j})$ is the set of all reducts of A relative to $\mathrm{d_j}$.Decision Table DT=(U, C∩D), where U is a non-empty, finite set called the universal, C is condition attributes set, D is decision attributes set, R = (U', C∪D), U'∪U, R is called sub-decision table of DT. Let p(DT) denote the set of all sub-decision system of DT, F ⊆ (DT) is called aF family of decision tableDT, RED(DT) denotes the set which contains all reducts of decision table DT, and RED(R) denotes the set which includes allreducts of sub-decision table R.

A decision system at least exists one reduct, which is just itself, so the set of reduct is not empty. In many cases, a given decision table may exist several reducts, as it the case in our decision tables (table2). Each reduct can product a rule set, and it is difficult to justify which is the best rule set. Therefore it is a important to search the most stable reduct, dynamic reduct is proposed in this case. The Dynamic reduct of decision table DT, DR (DT, F), is describes as:

$$\mathrm{DR}(\mathrm{DT}, \mathrm{F}) = \mathrm{RED}(\mathrm{DT}, \mathrm{d_j}) \cap \underset{\mathrm{R}\in\mathrm{F}}{} \mathrm{RED}(\mathrm{R}, \mathrm{d_j}) \ldots (18)$$

Any element of DR(DT, F) is called an F-dynamic reduct of DT, which describes the most stable reducts in decision table. From the definition of dynamic reducts, it follows that a relative reduct of DT is dynamic if it is also a reduct of all sub-tables from a given family F. This notation can be sometimes too restrictive, so there is a more general notion of dynamic reduct is applied, it is called (F, ε)-dynamic reducts, Where ε is the dynamic reduct threshold and it ranges $1 \geq \varepsilon \geq 0$. The set DRε(A, F) of all (F, ε)-dynamic reducts is defined by:

$$\mathrm{DR}_\varepsilon(\mathrm{DT}, \mathrm{F}) = \{\mathrm{R} \in \mathrm{RED}(\mathrm{DT}, \mathrm{d_j}) : \frac{|\{\mathrm{R}\in\mathrm{F}:\mathrm{R}\in\mathrm{RED}(\mathrm{R}, \mathrm{d_j})\}|}{|\mathrm{F}|} \geq 1 - \varepsilon \ldots (19)$$

Significance of an attribute can be evaluated by measuring effect of removing the attribute from decision table. Let C and D be sets of condition and decision attributes respectively and let a be a condition attribute, i.e., $a \in C$. The number $\gamma(C, D)$ expresses a degree of consistency of the decision table, or the degree of dependency between attributes C and D, or accuracy of approximation of U/D by C. If the attribute a removed, the coefficient $\gamma(C, D)$ must changes. The difference between $\gamma(C, D)$ and $\gamma(C - \{a\}, D)$ is normalized and the significance of the attribute a, is defined as:

$$SigF_{(C,D)}(a) = \frac{(\gamma(C,D) - \gamma(C - \{a\}, D))}{\gamma(C,D)} = 1 - \frac{\gamma(C - \{a\}, D)}{\gamma(C,D)} \ldots (20)$$

It can be denoted by SigF(a) and it ranges $0 \leq \mathrm{SigF(a)} \leq 1$. The more important is the attribute a, the greater is the number SigF(a). $\mathrm{SigF}(\mathrm{DT_J}, \mathrm{a})$ denotes the significance factor of attribute a within all attributes in sub-decision table $\mathrm{DT_J}$. The Attribute Reducts Generator algorithm ARG, shown in figure 9, used to generate reducts for our decision tables. The scenario of using this algorithm is: First, the reducts for the whole decision table DT are calculated by this algorithm. Then this decision table is randomly divided into new sub-system$\mathrm{DT_s}$, and the dynamic reducts are computed for each sub-table $\mathrm{DT_s}$, using ARG algorithm too.

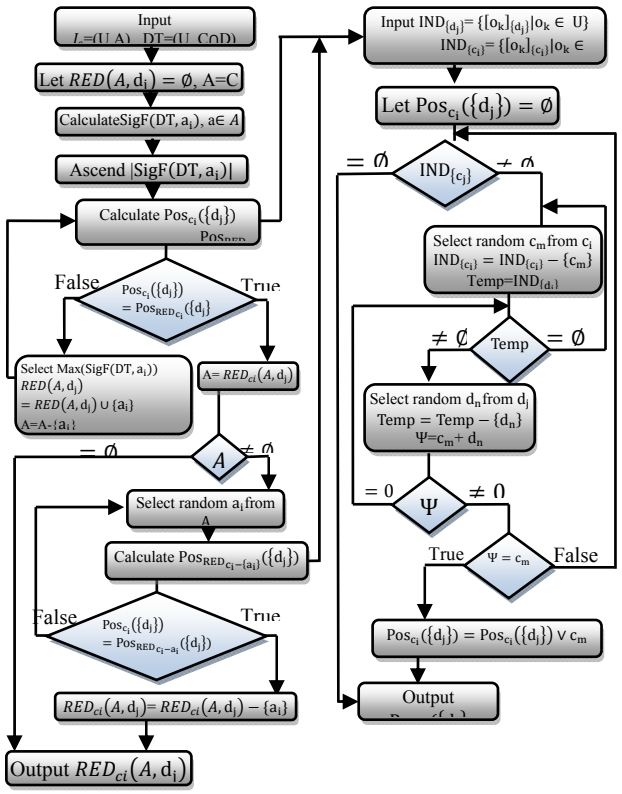

Figure 9: The Attribute Reducts Generator Algorithm (ARG)

The above algorithm generates $\mathrm{RED_{ci}}(\mathrm{A}, \mathrm{d_j})$, is the set of all reducts of A relative to $\mathrm{d_j}$ by incrementally removing the least informative attribute from it till there is no change in the value of dependency function. The positive region of c with respect to d, $\mathrm{Pos_{c_i}}(\{\mathrm{d_j}\})$, is also calculated to help in generating the reducts of our decision tables.

4. Rule Generation Phase

The basic notations of rule generation concept are introduced in the fifth phase of the **GRM** model. The rules generation based on granular rough theory, is the output of this model. According to the condition attributes of decision table, the rules are generated. However, not all condition attributes may need to be checked in the sense that some condition attributes are essential to classify and the other attributes are redundant. When the reduced decision table DT_{red} = {U, C ∪D, V} is obtained from the previous phase. Once reducts are found, generating a set of decision rules from this decision table can be an easy process. Every reduced decision table describes some decisions (actions, results etc.), when some conditions are satisfied. In other words, each row of the decision table specifies a decision rule which determines decisions in terms of conditions [41]. A decision rule expressed as a logical statement: **IF** conjunction of elementaryconditions**THEN** disjunction of elementary decisions. The decision rule constructed is denoted from a subset B ⊆ C of condition attribute, the set D of decision attributes and an object x ∈ U by (B, x) → (D, x), or in short B →$_x$ D. Decision table is a finite set of "if ... Then" decision rules. With every decision rule three coefficients are associated: the certainty, the coverage, and the strength factors of the rule. These factors are well-known criteria for evaluating decision rules. The Supp(B →$_x$ D) is the support of the decision rule B →$_x$ D and defined as:

$$\mathrm{Supp}(\mathrm{B} \rightarrow_x \mathrm{D}) = |[\mathrm{x_i}]_{\mathrm{B_i}} \cap \mathrm{d_i}| \ldots (21)$$

For any decision rule $B \rightarrow_x D$, the degree $Cer(B \rightarrow_x D)$ of certainty, its interpreted as the frequency of objects having the property D in the set of objects having the property D, and is defined as:

$$Cer(B \rightarrow_x D) = \frac{Supp(B \rightarrow_x D)}{|[x_i]_{B_i}|} \ldots (22)$$

For any decision rule $B \rightarrow_x D$, the degree $Cov(B \rightarrow_x D)$ of coverage, its interpreted as the frequency of objects having the property B in the set of objects having the property D, its defined as follows:

$$Cov(B \rightarrow_x D) = \frac{Supp(B \rightarrow_x D)}{|d_i|} \ldots (23)$$

Where the set d_i: is the decision class such that $x \in Di$. By checking values of all condition attributes, we can classify all discernible elements in a given decision table to those correct decision classes. Every decision rule is characterized by the strength. The strength means the number of object satisfying the condition part of the rule, i.e. covered by the rule, and belonging to the suggested decision class. If the rule is approximate, the strength is computed for each possible decision class separately. Generally speaking, stronger rules are more general because their condition parts are shorter and less specialized. The strength of the decision rule can be defined as:

$$\sigma(B \rightarrow_x D) = \frac{Supp(B \rightarrow_x D)}{|d_i|} \ldots (24)$$

The strength, certainty and coverage factors can be interpreted either as probabilities (objective), or as a degree of truth. Moreover, they can be also interpreted as a deterministic flow distribution in flow graphs associated with decision algorithms. This leads to a new look on Bayes' theorem [42] and its applications in reasoning from data, without referring to its probabilistic character. The coefficients can be computed from the data or can be a subjective assessment. It is shown that these coefficients satisfy Bayes' formula. Bayesian inference methodology consists in updating prior probabilities by means of data to posterior probabilities, which express updated knowledge when data become available.

VI. Experiments Evaluation

To evaluate the performance of TADM, we compare it with Efficient Task Allocation Protocol ETAP [18] and with the Greedy Distributed Allocation Protocol (GDAP) [19]. In order to validate the effectiveness of TADM algorithm and compare it with ETAP and GDAP two experiments are performed; each experiment has its own goal and settings. In each experiment to evaluate the experiment results two metrics are evaluated the Efficiency Ratio and Run Time, which can be defined as follows:

The Efficiency Ratio is the ratio between summation efficiency of finished tasks and the total efficiency expected of tasks. The efficiency of a task can be calculated as follows:

$$EFF_{(T)} = Reward_{(T)} / Resource_{(T)} \ldots (25)$$

Where: $Reward_{(T)}$ is the rewards gained from successfully finishing the task. $Resource_{(T)}$is the resources required for accomplishing the task. Run Time is the time of performing TADM algorithm in the network under pre-defined settings. The unit of Run Time is millisecond. To investigate the effects of TADM, and compare it with ETAP and GDAP, a multi-agent system has been implemented to provide a testing platform. The whole system is implemented on a 6 Pc's with an Intel Pentium 4 processor at 300GHz, with 3GB of Ram, connected with network Ethernet 512Mbps. A network of cooperative agents is designed, in which most agent team are connected to each other,the generation of this network can follow the approach proposed in [22]. For each experiment, there are unified settings have to be specified.

In Experiment 1: The main goal of this experiment is to demonstrate the scalability of TADM algorithm and compare it with ETAP and GDAP, as shown in figure 10 and figure 11, using same environment and settings, as follows: The average number of agent's team is fixed at 8. The number of agents range from 100 to 600, depending on the specific test. The number of tasks is range from 60 to 360. The number of resources types is range from 10 to 60.

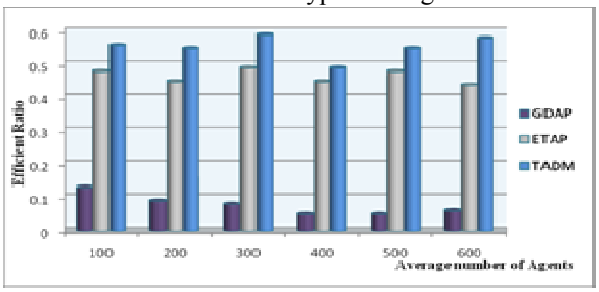

Figure 10: The Efficient Ratio on different number of agents

From figure 10, it is shown that when the number of agents is increasing, the Efficiency Ratio of TADM is much higher and more stable than of ETAP and GDAP that is continually descending with the increasing of agents.

Figure 11 shows the Run Time of TADM, GDAP and ETAP when the numberof agents in the network change. It can be noticed that ETAP spends more time when there are more agents in the network, this is because ETAP make many reallocation steps which results in time and communication overhead rising. On the other hand, the time consumption of GDAP is steady during the entire test process and keeps a lower level than ETAP, this because GDAP relies on neighbouring agents only.

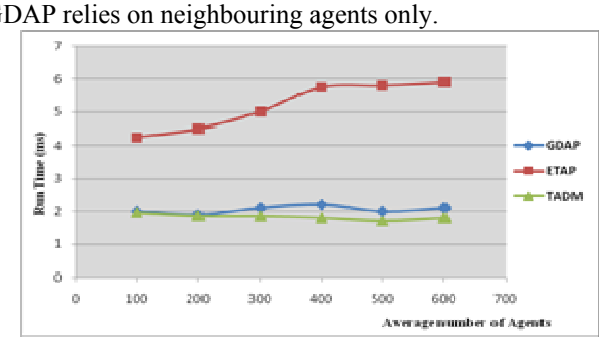

Figure 11: The Run Time on different number of agents

While TADM avoid these two drawbacks from the recent systems, and this experiment shows that the TADM performance is faster than that of GDAP and ETAP.

In Experiment 2: The main goal of this experiment is to test the influence of team grouping on TADM (i.e. show how different average number of agent's team influences the performance of TADM) and compare it with ETAP and GDAP, as shown in figure 12 and figure 13, using same environment and settings, as follows: The average number of agent's team is fixed at 8. The number of agents and tasks are 50 and 30 separately. The average number of

resources for each type is 30 and the average number of resources required by each task is also 30. The number of resources types is 5.

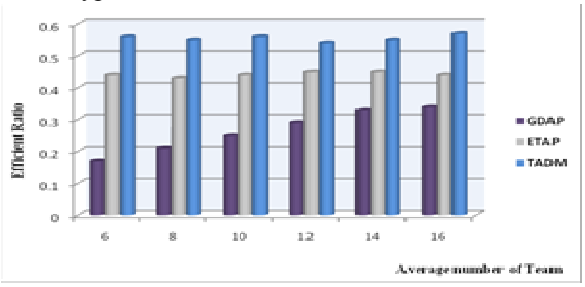

Figure 12: The Efficient Ratio on different number of agent's team

Figure 12 demonstrated that The Efficiency Ratio of TADM is much higher and more stable than that of GDAP and ETAP. GDAP performance is very low this is because task allocation in GDAP is only depending on neighbours of the agent.On the other hand, ETAP has better performance because it relies not only on neighbours of the agent, but also other agents if needed. The TADM has the higher performance and more stable than GDAP and ETAP, these results ensures and validates our algorithm.

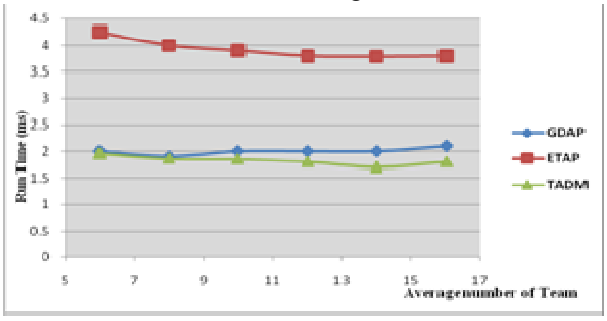

Figure 13: The Run Time on different number of agents' team

The Run Time of GDAP, ETAP, TADM in different number of agents' team is depicts figure 13. It's obviously that the Run Time of ETAP is higher than that of GDAP. As ETAP has to reallocate tasks when resources from neighbours are insufficient this lead to increase the reallocation steps and more time spending. While GDAP is steady due to its considering for only neighbours which could decrease the time and communication cost during task allocation process.

But TADM takes the benefits of agent's team and also uses any other agent if needed, and in the same time reduces the steps taken to allocate the task which lead to decrease the time and communication cost during task allocation process. So this is why TADM has better Run Time that GDAP and ETAP.

VII. Conclusion and Future Work

In this paper, a decision support module is proposed which enables any agent to make decisions needed, focused on two main decisions: first, is the decision needed to allocate a specific task to a specific agent. Second, the decision needed to select appropriate coordination mechanism. And a survey of recentalgorithms in task allocation in Multi-agent Systems is discussed. In addition an algorithm for the Task Allocation Decision Maker (TADM) is proposed showing the scenario of allocating tasks to specific agent/s. a Granular Rough Model (GRM) is proposed to generate the rules that help in the performance of TADM.Finally, a preliminary experiment is then conducted, indicating that TADMhas the scalabilityadvantage comparing to most recent systems. We plan to propose new algorithm for the coordination mechanism selection in the decision support module. Also, we intend to propose a coordination module in our DMAIS framework.

References

[1] Gruver, W., "Technologies and Applications of Distributed Intelligent Systems", IEEE MTT-Chapter Presentation, Waterloo, Canada, 2004.
[2] A. El-Desouky and S. El-Ghamrawy. A Framework for Distributed Decision Support Multi-Agent Intelligent System. The 2008 International Conference on Artificial Intelligence ICAI08, WORLDCOMP'08 July14-17, 2008, Las Vegas, Nevada, USA.
[3] T. Bui and J. Lee,"An agent-based framework for building decision support systems"Decision Support Systems ,vol. 25 , 1999, pp.225-237.
[4] S.D. Pinson, J.A. Louca and P. Moraitis, "A distributed decision support system for strategic planning," Decision Support Systems , vol. 20 1, 1997, pp. 35–51.
[5] Song Qiang, Lingxia Liu, "The Implement of Blackboard-Based Multi-agent Intelligent Decision Support System," iccea, vol. 1, pp.572-575, 2010 Second International Conference on Computer Engineering and Applications, 2010
[6] Rustam and Bijan, "Pluralistic multi-agent decision support system: a framework and an empirical test, " Information & Management, vol. 41, Sept., 2004, pp. 883-898.
[7] Wang Fuzhong, "A Decision Support System for Logistics Distribution Network Planning Based on Multi-agent Systems," dcabes, pp.214-218, 2010 Ninth International Symposium on Distributed Computing and Applications to Business, Engineering and Science, 2010
[8] Tosic, P.T., Agha, G.A.: Maximal Clique Based Distributed Group Formation For Task Allocation in Large-Scale Multi-Agent Systems. In: Ishida, T., Gasser, L., Nakashima, H. (eds.) MMAS 2005. LNCS (LNAI), vol. 3446, pp. 104–120. Springer, Heidelberg (2005).
[9] X. Zheng and S. Koenig. Reaction functions for task allocation to cooperative agents. In Proceedings of 7th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2008), pages 559–566, Estoril, Portugal, May 2008
[10] S. Kraus, O. Shehory, and G. Taase. Coalition formation with uncertain heterogeneous information. In Proceedings of 2th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2003), pages 1–8, Melbourne, Australia, July 2003.
[11] Pinedo, M., "Scheduling: Theory, Algorithms and Systems", Prentice Hall, 1995.
[12] Davis, R., and Smith, R., "Neogitiation as a metaphor for distributed problem solving," Artificial Intelligence, 20, 63–100, 1983.
[13] R. McAfee, and J. McMillan, "Auctions and bidding," Journal of Economic Literature, 25, 699–738, 1987.
[14] K. Lerman and O. Shehory. Coalition formation for largescale electronic markets. In Proceedings of 4th International Conference on Multi-Agent Systems, pages 167–174, Boston, Massachusetts, USA, July 2000. IEEE Computer Society.
[15] O. Shehory and S. Kraus. Methods for task allocation via agent coalition formation. Artificial Intelligence, 101:165–200, May 1998
[16] S. Abdallah and V. Lesser. Modeling task allocation using a decision theoretic model. In Proceedings of 4th Automous Agents and Multiagent Systems (AAMAS 2005), pages 719–726, Utrecht, Netherlands, July 2005.
[17] P. V. Sander, D. Peleshchuk, and B. J. Grosz. A scalable, distributed algorithm for efficient task allocation. In Proceedings of 1st International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2002), pages 1191–1198, Bologna, Italy, July 2002.
[18] M. D. Weerdt, Y. Zhang, and T. Klos. Distributed task allocation in social networks. In Proceedings of 6th Automous Agents and Multiagent Systems (AAMAS 2007), pages 500–507, Honolulu, Hawaii, USA, May 2007.
[19] Dayong Ye, QuanBai, Minjie Zhang, Khin Than Win, ZhiqiShen, "An Efficient Task Allocation Protocol for P2P Multi-agent Systems," ispa, pp.11-18, 2009 IEEE International Symposium on Parallel and Distributed Processing with Applications, 2009.
[20] ZakiBrahmi, Mohamed Gammoudi, MalekGhenima: Cooperative Agents Based-Decentralized and Scalable Complex Task Allocation Approach Pro Massive Multi-Agents System. ACIIDS 2010: 420-430
[21] Cheng, J., andWellman, M. , "TheWALRAS algorithm: A convergent distributed implementation of general equilibrium outcomes," Computational Economics, 12, 1–24 ,1998.
[22] D. J. Watts and S. H. Strogatz. Collective dynamics of small world' networks. Nature, 393:440–442, 1998
[23] Liu Yunxiang, Sun Jigui, Fuzzy logic and fuzzy reasoning based on Triangular norms, Journal ofJlin University(Scied,) vol.41 No.1 2003.

[24] L.A. Zadeh, Toward a restructuring of the foundations of fuzzy logic (FL), Abstract of BISC Seminar, Computer Science Division, EECS, University of California, Berkeley, 1997.
[25] L.A. Zadeh, Fuzzy sets, Information and Control, 8:338-353 (1965).
[26] Z. Pawlak, "Rough Sets", International Journal of Computer and Information Sciences, Vol.11.p.341-356, (1982) .
[27] Z.Z. Shi, Z. Zheng, Z.Q. Meng. Image Segmentation- Oriented Tolerance Granular Space Model. IEEE International Conference on Granular Computing, 26- 28 Aug. 2008,Hangzhou, 566-571.
[28] M.R. Chmielewski, J.W. Grzymala-Busse, N.W. Peterson, S. Than, The rule induction system LERS - A version for personal computers, Found. Comput. Decision Sci. 18 (3/4) (1993) 181-212..
[29] Grzymala-Busse J W, Hu M. A comparison of several approaches to missing attribute values in data mining. In Proceedings of the Second International Conference on Rough Sets and Current Trends in Computing (RSCTC'2000). Lecture Notes in Computer Science, Berlin: Springer, 2001, 2005: 378–385
[30] Grzymala-Busse J W. Rough set strategies to data with missing attribute values. In: Proceedings of the Workshop on Foundations and New Directions in Data Mining, associated with the 3rd IEEE International Conference on Data Mining, 2003, 56–63
[31] Wang G Y. Extension of rough set under incomplete information systems. Journal of Computer Research and Development, 2002, 39(10): 1238–1243 (in Chinese)
[32] Wang G Y. Domain-oriented data-driven data mining based on rough sets. In: Proceedings of International Forum on Theory of GrC from Rough Set Perspective (IFTGrCRSP2006), 2006, 46–46.
[33] Kryszkiewicz M. Rough set approach to incomplete information systems. Information Sciences, 1998, 112(1): 39–49
[34] Xiaoping YangAn Improved Model of Rough Sets on Incomplete Information Systems, 2009.
[35] Wang G Y. Rough Set Theory and Knowledge Acquisition. Xi'an: Xi'an Jiaotong University Press, 2001, 38–39 (in Chinese)
[36] Wang,G.Y., Zhao,J., An,J.J., Wu,Y.: Theoretical Study on Attribute Reduction of Rough Set Theory: in Algebra View and Information View. Third International Conference on Cognitive Informatics. (2004) 148-155
[37] Bosse, E., Jousseleme, A.L., Grenier, D.: A new distance between two bodies of evidence. In: Information Fusion, vol. 2, pp. 91–101 (2001)
[38] 3. Elouedi, Z., Mellouli, K., Smets, P.: Assessing sensor reliability for multisensor data fusion within the transferable belief model. IEEE Trans. Syst. Man cyben. 34(1), 782–787 (2004)
[39] Fixen, D.: The modified Dempster-Shafer approach to classification. IEEE Trans. Syst. Man Cybern. A 27(1), 96–104 (1997); In: Workshop of Data Mining, the 3rd International Conference on Data Mining, Melbourne, Florida, vol. 5663
[40] 5. Modrzejewski, M.: Feature selection using rough sets theory. In: Proceedings of the 1th International Conference on Machine Learning, pp. 213–226 (1993)
[41] A. Skowron, C. Rauszer: The discernibility matrices and functions in information systems, in: R. Słowiński (ed.), Intelligent Decision Support. Handbook of Applications and Advances of the Rough Set Theory, Kluwer Academic Publishers, Dordrecht, 1992, 311-362
[42] Bayes' Theorem Revised – The Rough Set View, Lecture Notes in Computer Science, 2001, Volume 2253/2001, 240-250, DOI: 10.1007/3-540-45548-5_27