Solving a University Course Tmetabling Problem by Negotiation in a Multi-agent System

Olfa Belkahla Driss, Houssem Eddine Nouri
Stratégies d'Optimisation et Informatique intelligentE (SOIE)
High Institute of Management, University of Tunis
41, Street of Liberty Bouchoucha-City CP-2000-Bardo Tunis, Tunisia
Higher Business School of Tunis, University of Manouba
Olfa.belkahla@isg.rnu.tn, Houssemeddine.nouri@gmail.com

ABSTRACT: *This paper proposes a multi-agent model for solving the university course timetabling problem named M.A.T.P (Multi-Agent model for university Timetabling Problem). Our model is based on cooperative and negotiating agents enabling highly parallel and distributed processing of the problem and incorporating new constraints that have not been taken into account by previous works. The aim of our model is to provide a best solution satisfying hard and soft constraints while reducing temporal complexity. To evaluate the performance of our model, we choose to address a real case (instances of the Higher Business School of Tunis) where we realize a test scenario by analyzing the variation of the number of messages in terms of the assignment priority score and its effect on the execution time.*

Keywords: University Course Timetabling Problem, Multi-agent System, Negotiation, Lecture Scheduling, Messaging Exchange System

Received: 28 October 2012, Revised 13 December 2012, Accepted 17 December 2012

1. Introduction

Nowadays, the personal scheduling problems have become more diffused in our real life. They can be presented in many configurations or models depending on the specific organizational environment and the duration of the planning period. The timetabling problem is an instance of the personal scheduling problems, it is well known as an NP-complete problem. This problem is pervasive in all practical aspects of modern society. It plays a very important role in many types of organizations such as hospitals, transport companies, protection services and emergency and universities.

In our case, we focus more precisely on the problem of the university timetabling problem. Burke and his colleagues [5] note in this regard that this problem can be divided into two main categories: courses and exams. Different aspects separate these two categories. For example, we try to group the courses, but we prefer to move away exams from each other as possible. Or again, a course may take place at a given time in one classroom, while many exams may take place at the same time in the same classroom, or the same exam can be dispatched in many classrooms.

In this paper, we are interested to solve the university course timetabling problem. It can be defined as a set of university courses which take place throughout specific periods for five or six days in a week, directed by a limited number of teachers and classrooms requiring a better management in order to contain the large number of the registered students.

The goal of our model is to provide a best solution for this problem satisfying hard and soft constraints while reducing temporal complexity.

This paper is organized as follows. In section II, we introduce the university course timetabling problem as well as its hard and soft constraints. We detail, then in section III our contribution based on multi-agent systems. Section IV is devoted to the presentation of a real case study (instances of the Higher Business School of Tunis) in order to test our approach as well as a scenario evaluating its efficiency.

2. University course timetabling problem

Many researchers are facing this problem from several points of view and with different approaches using different paradigms of resolution. The first attempts of resolution methods were the *theory of graphs* [13] [6], the *integer linear programming* [8] and the techniques of *constraint satisfaction problem* [1], [16] and [14]. However, these methods have not given a solution dealing with all instances and constraints of this problem. That's why, they have given a way to other types of methods adapted to this type of problem, namely metaheuristics such as the *tabu search* [17], the *simulated annealing* [7] and the *genetic algorithms* [2]. This family of approximate search has mechanisms that allow a good general investigation of the search space. But generally, it is nondeterministic and gives no guarantee of optimality. This has allowed the appearance of new approaches based on the *multi-agent systems* [12], [3], [15], [11] and [10], but they did not succeed to well adapt this formalism to generate a solution satisfying all the problem constraints. That's why, and in this area we have proposed a new multi-agent model allowing to minimize the time complexity, to introduce a new details that have not been taken into account by previous work and to attend a good satisfaction of the teachers preferences.

In order to get a best solution for this problem, we must take into account all the constraints of the problem that must be satisfied. These constraints are often classified into two ategories, the first includes hard constraints and the second category includes constraints often called soft constraints.

Hard constraints: these are constraints that must be satisfied in any environment, because the violation of these constraints may cause the generation of an unsatisfiable solution.

• Two lectures cannot be programmed in the same classroom and at the same period of time,

• The lectures given by the same teacher cannot be programmed at the same period of time,

• A classroom can be assigned only to one lecture at the same period of time,

• A group lecture cannot takes place at the same period with another that is not a group lecture belonging to the same level of study,

• The number of students must be less than or equal to the capacity of the assigned classroom.

Soft constraints: the violation of these constraints has no effect on the generation of a satisfiable solution.

• The assignment of classrooms and periods of time must allow to satisfy at best the preferences of teachers,

• The assignment of classrooms to the different lectures must allow to satisfy at best some preferences.

In this work, we propose a multi-agent model based on cooperative agents, named M.A.T.P a **M**ulti-**A**gent model for university **T**imetabling **P**roblem, enabling highly parallel and distributed processing of the problem. Our model incorporates several constraints that have not been taken into account by previous works.

Multi-Agent Systems (M.A.S) are chosen because of their advantages in many different domains by means of the cooperation between a society of agents. In fact, each agent, concurrently and asynchronously, acquires information from its environment and from other agents to reason on the basis of these information and to act consequently, see the studies of [4] using M.A.S in scheduling. For more details on M.A.S, see the studies of [9].

3. Multi-agent model for university timetabling problem

3.1 Agent identification

We have equipped our multi-agent model MATP, see figure 1, with three classes of agents. The first class is composed of agents that we have named T.A, "*Teacher Agents*", divided into three categories of teachers: C1: Professor, Associateprofessor; C2: Assistant-professor, Assistant; C3: Contractual. The second class is composed of agents that we called C.A,

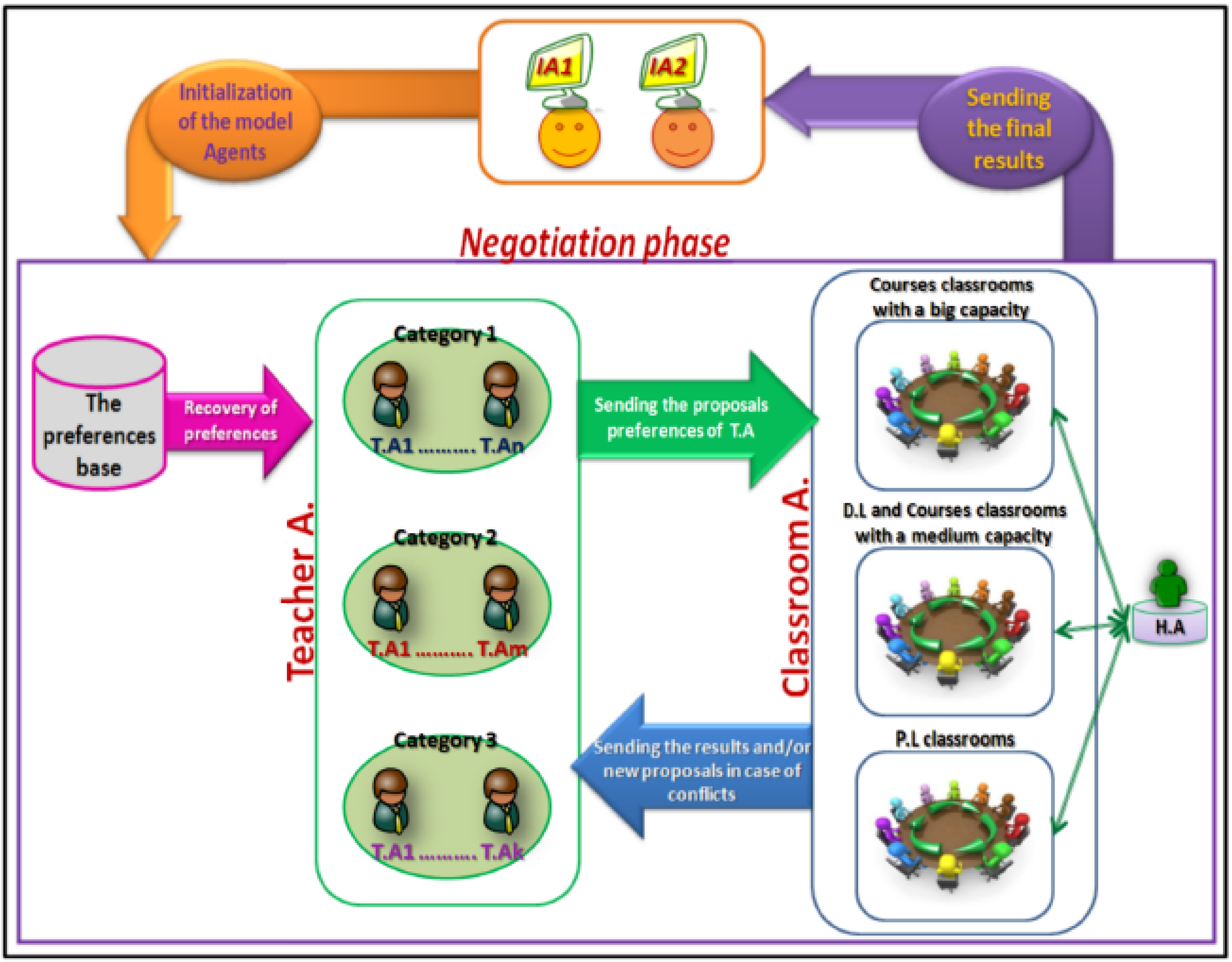


Figure 1. Multi-Agent model for university Timetabling Problem

"*Classroom Agents*", divided into three types of classrooms ("*Courses*", "*Practical Lectures*", "*Directed Lectures*") related to the type of the lecture session. The third class contains three agents: two "*Interface Agents*" that we called I.A1 and I.A2 and one "*History Agent*" that we named H.A.

3.2 Global steps of M.A.T.P

The steps of our model proceed in three phases: initialization, negotiation and transmission of final results.

3.2.1 Initialization phase

In this phase, we present the role of the agent I.A1 which initializes the execution of the system agents. In fact, it allows the implementation of all agents based on the initial parameters fixed at the start by the user.

3.2.2 Negotiation phase

This phase is the kernel of our model. It is based on a messaging exchange system between the two agent's classes T.A and C.A in order to have in each case an agreement between them, respecting all the hard constraints of this problem. The first class of agents T.A starts the negotiation process by sending all their allocation propositions (which were recovered from their preferences base) to the C.A agents in order to get a better reservation of the most suitable classrooms and the most favourite time periods of the day. ther hand, the second class of agents C.A will receive and analyze the T.A agent's preferences. In fact, this class will ask the H.A agent to verify the existence of duplication of time periods for a same T.A agent in each reception of propositions. Thus, it allows either to validate, or to give a new proposition in the case of conflict. The C.A may have 1 or n T.A propositions asking the same period in the same day, and generating conflicts between them, see figure 2. That's why, we have added a new hypothesis in which a classroom can be replaced by another one having the same characteristics, that we called the equivalence of classrooms (or vertical search) for the three categories of teacher agents. So we used a **V**ertical assignment priority **S**core ***VSi*** affected to each ***i*** category of teacher, where ***i*** $\in \{1, 2, 3\}$. This score ***VSi*** is incremental from zero to a maximum value ***VSimax***, where ***VSi*** $\in$ [0, ***VSimax***]. ***VSimax*** is the maximum value given by the user for this score that may have a T.A agent with a category ***i*** where:

• **Priority 1,** VS_{1max} **(Rank of teacher):** This score is given for each agent belonging to the first category of teacher agents T.A having a rank of "Professor" or "*Associateprofessor*".

• **Priority 2,** VS_{2max} **("*Course*")**: This score is given for each agent belonging to the second category of teacher agents T.A and asking a lecture session with type "*course*".

• **Priority 3,** VS_{3max} **("*D.L*" or "*P. L*")**: This score is given for each agent belonging to the third category of teacher agents T.A and asking a lecture session with type "*D.L*" or "*P. L*".

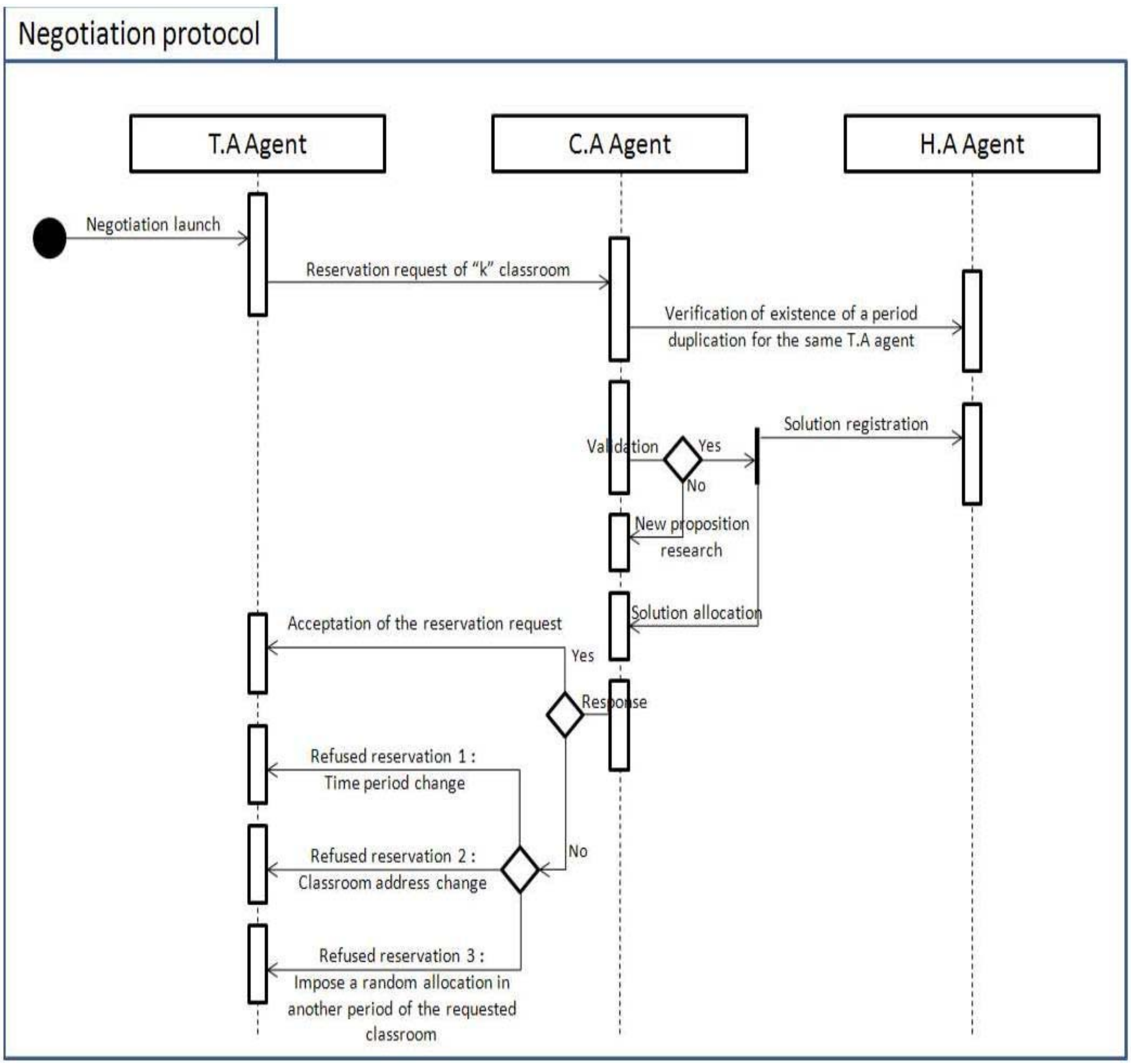


Figure 2. Interaction protocol diagram

Also, by integrating many types of criteria for acceptance of a reservation (capacity of students for each lecture session, the teacher's category, type of classroom to be reserved and type of lecture session), we will have a decrease in the percentage of appearance of conflicts between T.A agents.

3.2.3 Transmission of final results

Whenever a T.A agent receives all solutions in response to its messages, it finishes its negotiation phase and transmits its final results to I.A1 agent generating the form of teacher's timetable. Then the agent I.A2 ends the process by generating the final timetable of the different classrooms.

3.3 Agent knowledge

3.3.1 Interface agent 1

It has as acquaintances all T.A agents, all C.A agents and the H.A agent.

Its static knowledge is formed by the set of the system initialization parameters such as the number of T.A agents, the number of C.A agents and the assignment priority score for each category of T.A agents. Its dynamic knowledge consists of the construction of the solution for the teacher timetabling problem.

3.3.2 Interface agent 2

It has as acquaintances all C.A agents and the H.A agent.

Its static knowledge consists of the number of C.A agents. Its dynamic knowledge is formed by the construction of the solution for the classroom timetabling problem.

3.3.3 Teacher agents

They have as acquaintances I.A1, I.A2, H.A and all C.A agents.

Their static knowledge consist of their education rank, the type of lecture sessions, the type of the classrooms and the number of students for each lecture session. Their dynamic knowledge are formed by the assignment priority score, classrooms and education period for each lecture session.

3.3.4 Classroom agents

They have as acquaintances I.A1, I.A2, H.A and all T.A agents.

Their static knowledge consist of their education rank, the type of lecture sessions, the type of the classrooms and the number of students for each lecture session. Their dynamic knowledge consist of the assignment priority score, classrooms and education period for each lecture session.

3.3.5 History agent

It has as acquaintances all T.A agents and all C.A agents.

Its static knowledge is formed by the set of the fields prepared for the accepted solutions such as the teacher names, the lecture sessions and the education levels. Its dynamic knowledge is formed by saving the accepted reservation for each T.A agent.

3.4 Agent behaviour

3.4.1 Interface agents behaviour

The behaviour of the I.A1 is to initialize all the other agents of our model. Then, it moves to an inactive state pending the reception of the final T.A agent messages to generate them in the form of a solution for the teacher timetabling problem.

For the I.A2 behaviour, this latter has to generate a solution for the classroom timetabling problem after the end of the negotiation process.

3.4.2 Teacher agents behaviour

A T.A agent possesses a group of lectures (which can be a course, D.L or P.L) that it seeks to assign them to classrooms in

Verification of the reservation request

Begin

If Response = "*Accepted* "

Then Display a message of acceptance.

Else-If A refusal reservation message 1, 2 or 3 has been received.

Then Send another message to C.A containing the new update of the assignment propositions.

End.

the most favourite periods of the day. In fact, each T.A begins its negotiation phase by sending its proposals to C.A agents requesting the most preferred classrooms and teaching periods. Then, he receives a response message from C.A:

3.4.3 Classroom agents behaviour

A C.A agent contains an array of periods to search solutions for the requested periods. Thus, this type of agent is composed of a set of rules for the negotiation management:

Verification of duplication part

Begin

If The requested period hasn't been duplicated for the same T.A agent.

Then Go to the validation step.

Else Change the requested period and send a message of a refused reservation 1 to T.A agent.

End.

After finishing the verification part, the C.A agent passed to the next behaviour to validate the reservation request:

Validation of reservation part

Begin

If The requested period hasn't been reserved.

Then Send an acceptance message of the requested period to the T.A agent and record the solution in the memory of the H.A agent.

Else-If The requested period hasn't been reserved and the *VSi* <= *VSimax*.

Then *Vertical assignment search:* change the requested classroom address, increment the assignment priority score and send a message of a refused reservation 2 to T.A agent, see figure 3.

Else-If The requested period has been reserved and the *VSi* > *VSimax*.

Then *Horizontal assignment search:* impose a random assignment in another available period of the requested classroom and send a message of a refused reservation 3 to T.A agent, see figure 4.

End.

3.4.4 History agent behaviour

The H.A agent contains seven record tables (Period, Teacher, Lecture, Speciality, Level, Type-lecture group) for storing the set of the accepted propositions throughout the negotiation phase in order to offer to the C. A agents the opportunity to verify whenever existing a period duplication for the same T. A agent.

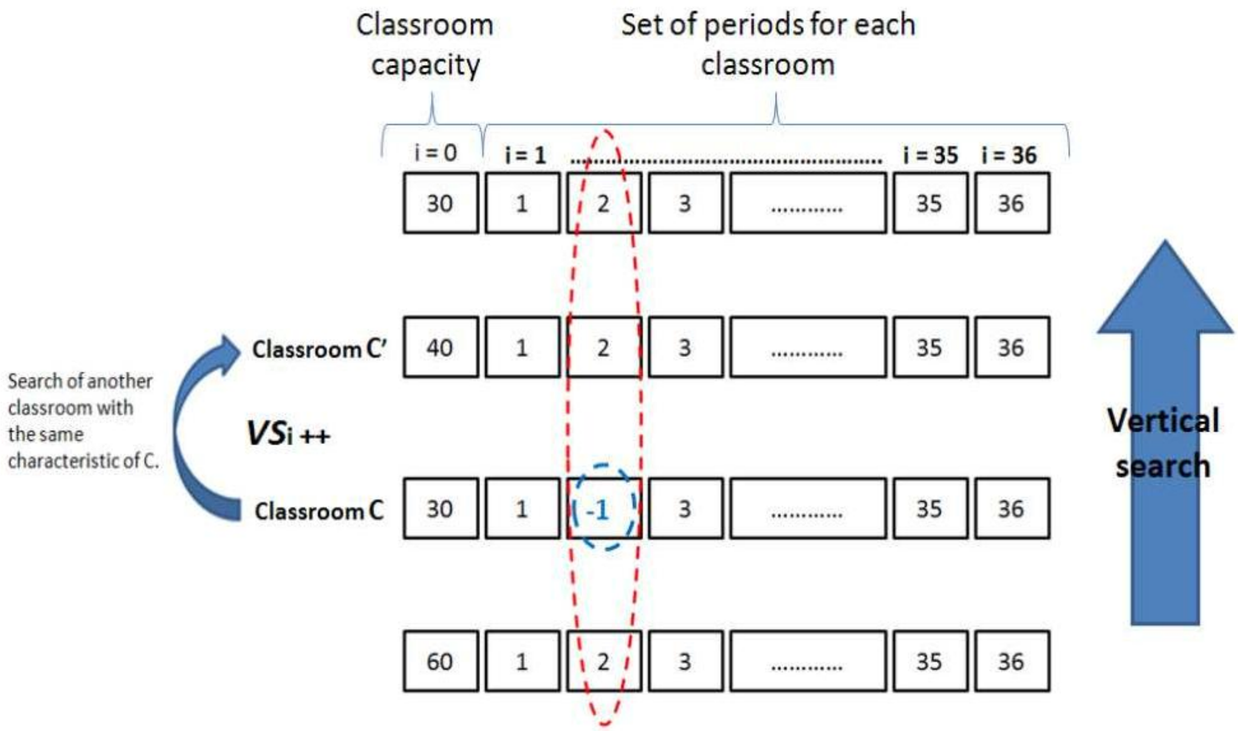


Figure 3. Vertical assignment search

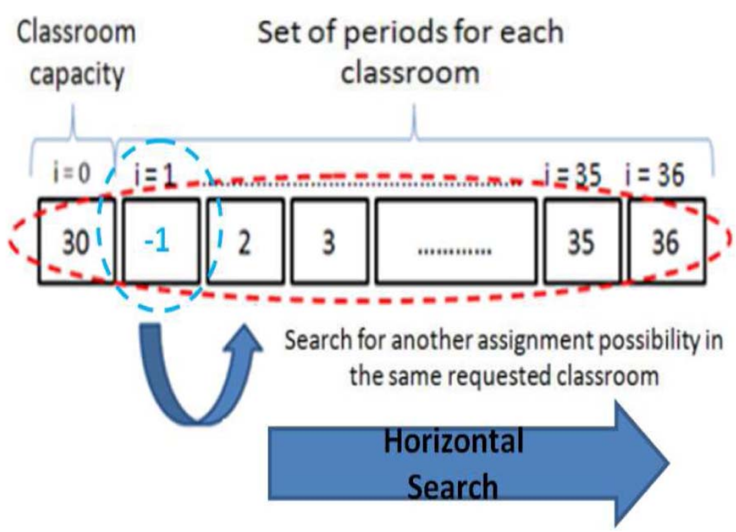


Figure 4. Horizontal assignment search

4. Experimentation

4.1 A case study

To test our approach, we have chosen to conduct our study on a real case where we used data instances of the Higher Business School of Tunis.

4.1.1 Number and types of teachers

- **5 Professors:** 10 lecture sessions per week.

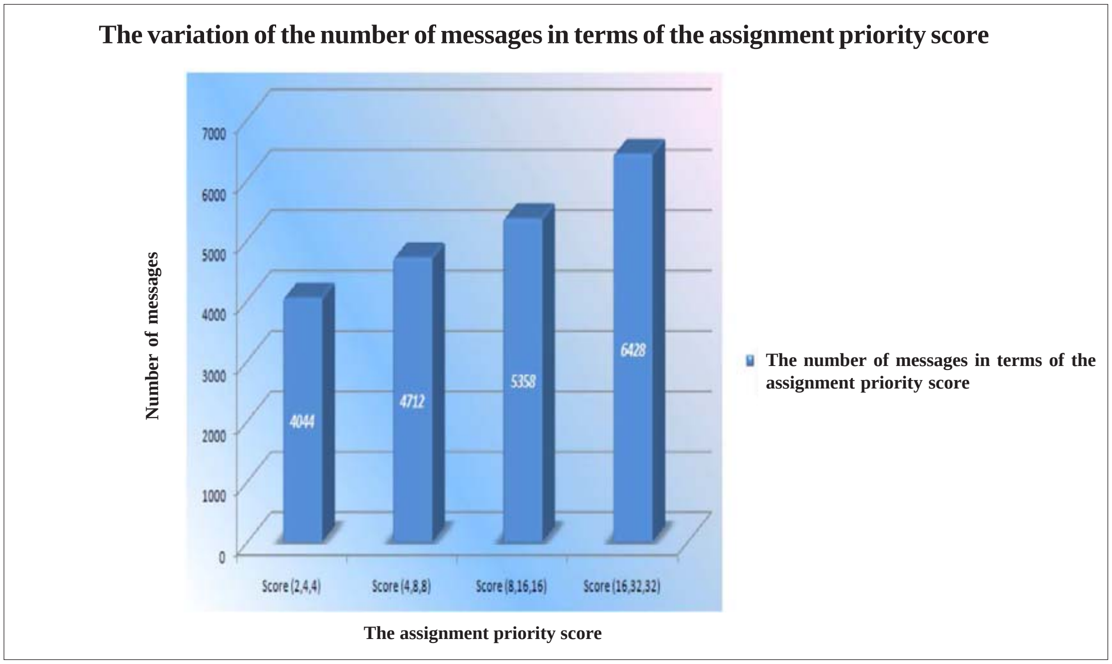


Figure 5. The variation of the number of messages in terms of the assignment priority score

- **6 Associate-professors:** 18 lecture sessions per week.
- **30 Assistant-professors:** 120 lecture sessions per week.
- **40 Assistants:** 320 lecture sessions per week.
- **50 Contractuals:** 400 lecture sessions per week.

→ Total number of teachers = 131 teachers.
→ Total number of lecture sessions = 868 sessions.

4.1.2 Number of classrooms

For the teaching classrooms, we have 64 classrooms belonging to 5 blocks A, B, C, D, and I of building, that we have chosen to group them into three categories:

• **Category 1:** all course classrooms having a big capacity of students (A1, A2, A3, B5, B6).

• **Category 2:** all D.L and course classrooms having an average capacity of students (B1, B2, B3, B4, D1…D24, C2…C11, I8…I21).

• **Category 3:** all P.L classrooms (I1, I2, I3, I4, I5, I6, I7).

4.1.3 Specialities and education levels

The school offers 25 specialities divided into 5 education levels: 1st year license, 2nd year license, 3rd year license, master M1, master M2.

4.2 Experimental design

Furthermore, we have chosen to use the famous multi-agent platform ***Jade*** to implement our model agents. Our choice was motivated by the benefits presenting this platform.

For the development of our model, we have chosen to use the object-oriented programming language ***Java*** with the Eclipse Helios IDE. This choice was imposed because the different agents in our system are implemented on the Jade multi-agent platform and this latter has been entirely developed in Java.

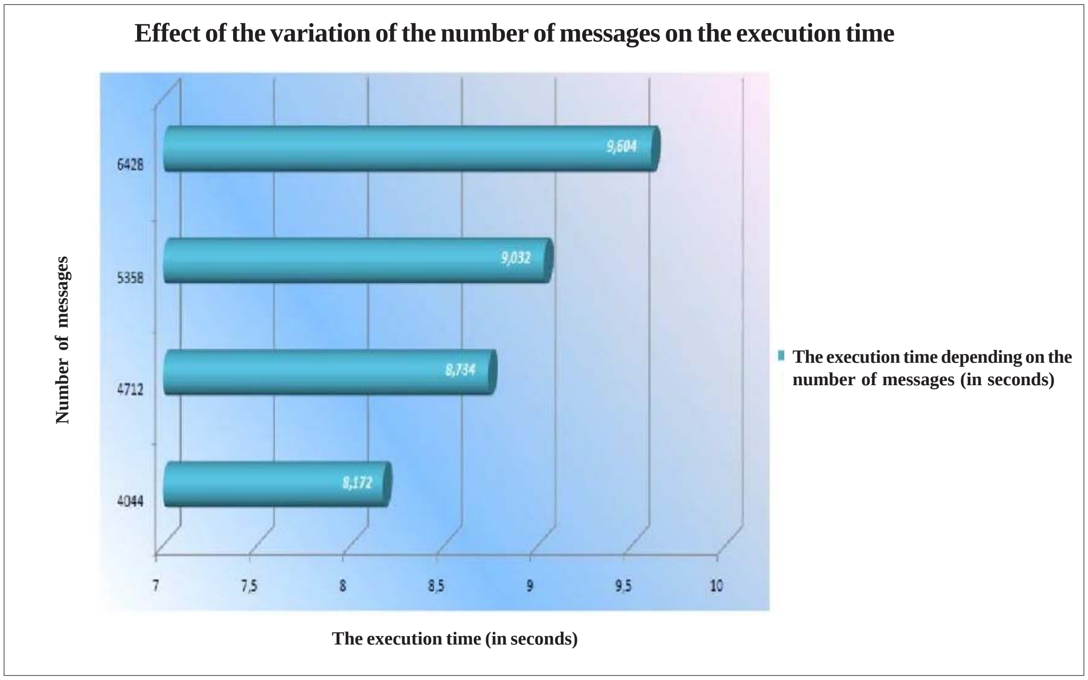
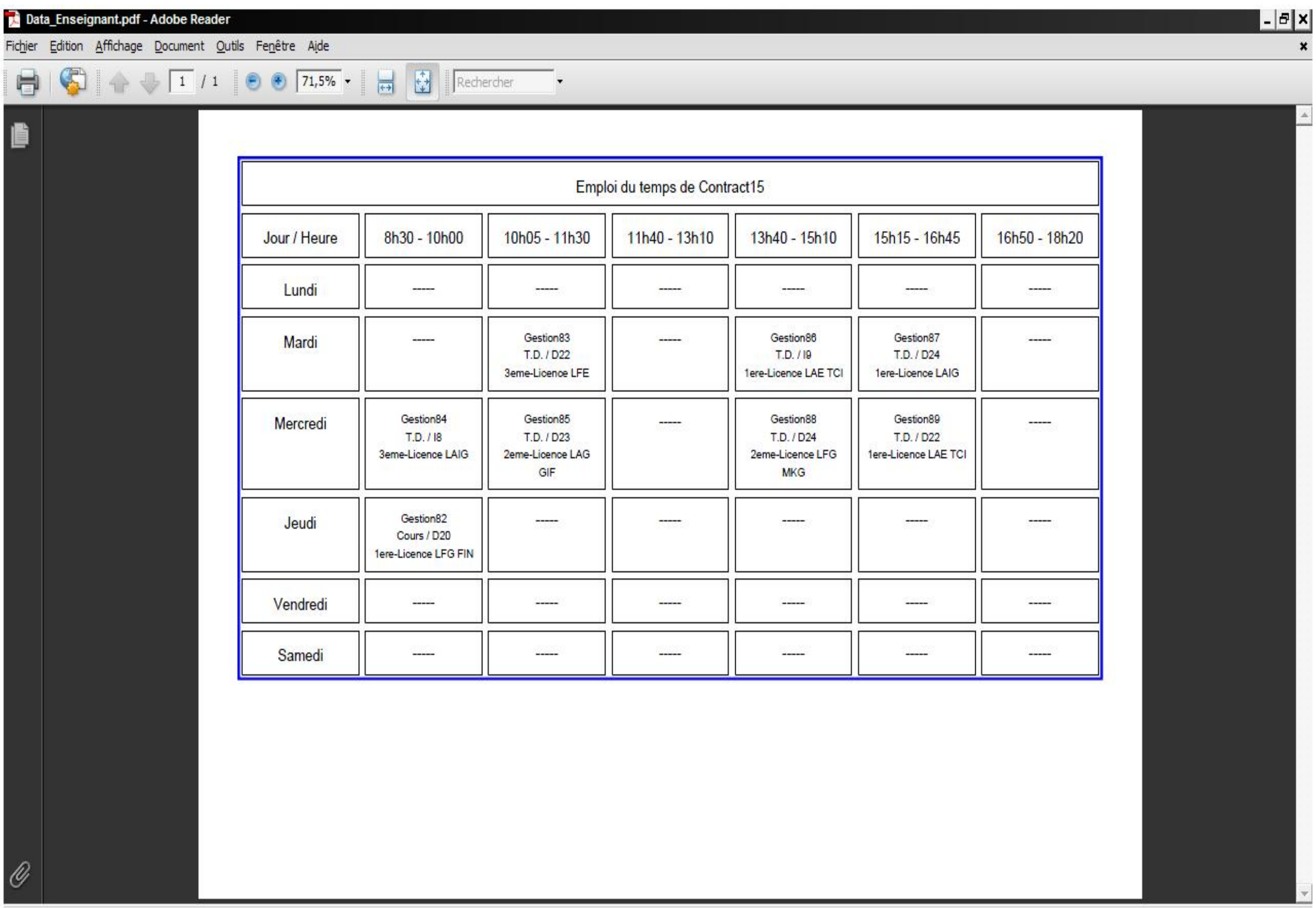


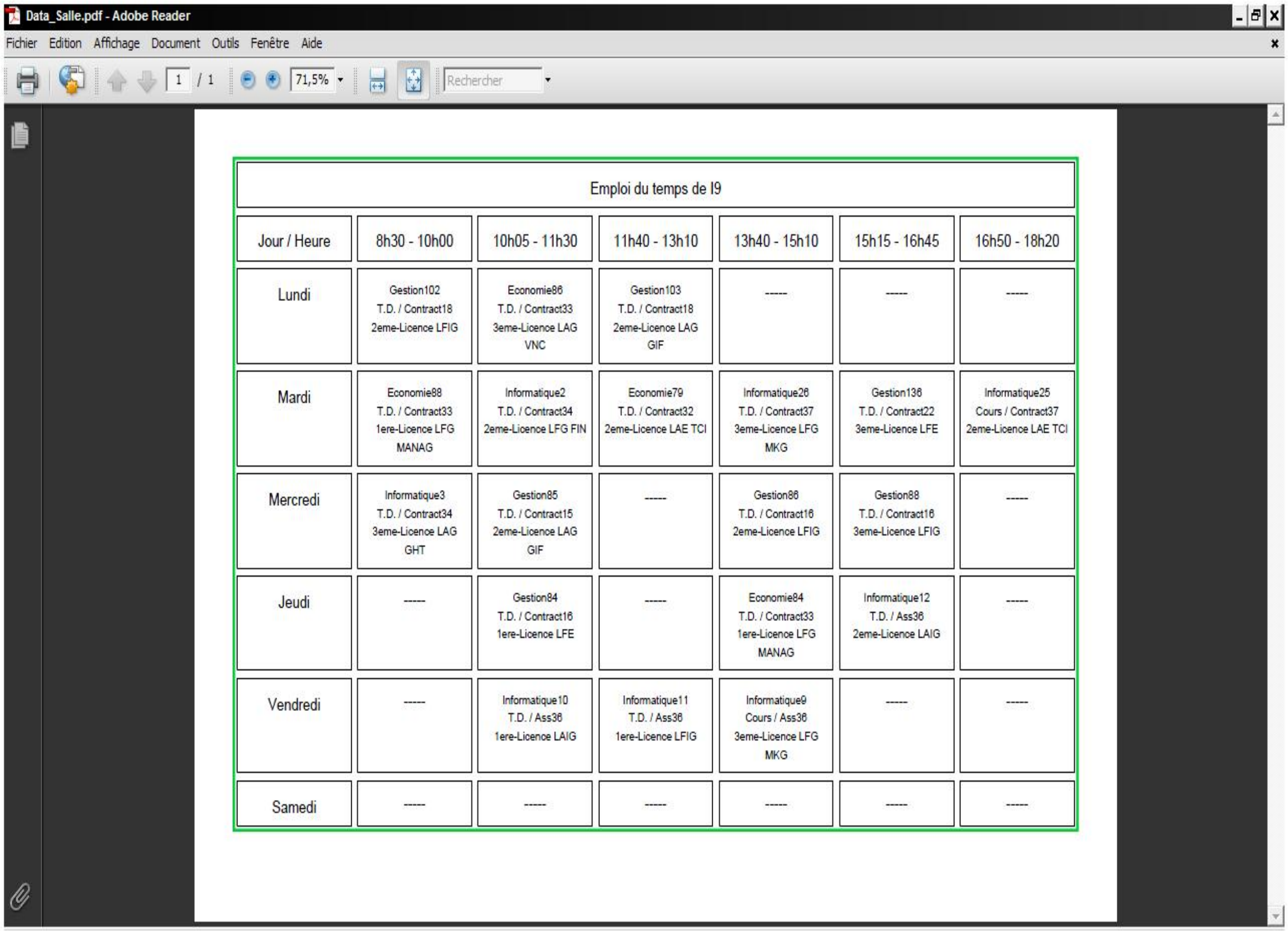
Figure 6. The effect of the variation of messages on the execution time



Emploi du temps de Contract15

Jour / Heure	8h30 - 10h00	10h05 - 11h30	11h40 - 13h10	13h40 - 15h10	15h15 - 16h45	16h50 - 18h20
Lundi	-----	-----	-----	-----	-----	-----
Mardi	-----	Gestion83 T.D. / D22 3eme-Licence LFE	-----	Gestion86 T.D. / I9 1ere-Licence LAE TCI	Gestion87 T.D. / D24 1ere-Licence LAIG	-----
Mercredi	Gestion84 T.D. / I8 3eme-Licence LAIG	Gestion85 T.D. / D23 2eme-Licence LAG GIF	-----	Gestion88 T.D. / D24 2eme-Licence LFG MKG	Gestion89 T.D. / D22 1ere-Licence LAE TCI	-----
Jeudi	Gestion82 Cours / D20 1ere-Licence LFG FIN	-----	-----	-----	-----	-----
Vendredi	-----	-----	-----	-----	-----	-----
Samedi	-----	-----	-----	-----	-----	-----

Figure 7. Teacher timetable



Emploi du temps de I9						
Jour / Heure	8h30 - 10h00	10h05 - 11h30	11h40 - 13h10	13h40 - 15h10	15h15 - 16h45	16h50 - 18h20
Lundi	Gestion102 T.D. / Contract18 2eme-Licence LFIG	Economie86 T.D. / Contract33 3eme-Licence LAG VNC	Gestion103 T.D. / Contract18 2eme-Licence LAG GIF	-----	-----	-----
Mardi	Economie88 T.D. / Contract33 1ere-Licence LFG MANAG	Informatique2 T.D. / Contract34 2eme-Licence LFG FIN	Economie79 T.D. / Contract32 2eme-Licence LAE TCI	Informatique26 T.D. / Contract37 3eme-Licence LFG MKG	Gestion136 T.D. / Contract22 3eme-Licence LFE	Informatique25 Cours / Contract37 2eme-Licence LAE TCI
Mercredi	Informatique3 T.D. / Contract34 3eme-Licence LAG GHT	Gestion85 T.D. / Contract15 2eme-Licence LAG GIF	-----	Gestion86 T.D. / Contract16 2eme-Licence LFIG	Gestion88 T.D. / Contract16 3eme-Licence LFIG	-----
Jeudi	-----	Gestion84 T.D. / Contract16 1ere-Licence LFE	-----	Economie84 T.D. / Contract33 1ere-Licence LFG MANAG	Informatique12 T.D. / Ass36 2eme-Licence LAIG	-----
Vendredi	-----	Informatique10 T.D. / Ass36 1ere-Licence LAIG	Informatique11 T.D. / Ass36 1ere-Licence LFIG	Informatique9 Cours / Ass36 3eme-Licence LFG MKG	-----	-----
Samedi	-----	-----	-----	-----	-----	-----

Figure 8. Classroom timetable

4.3 Experimental results

To analyze the efficiency of our model, we realized a test scenario that we presented in the form of two types of graphs. For this scenario, we have analyzed the variation of the number of messages in terms of the assignment priority score and the effect of the number of messages on the C.P.U time. In fact, we have chosen to make four successive tests of our generator by changing in each case the assignment priority score for the three categories of teachers:

• The first case was tested by an initial ***VSi*** of (2,4,4), two vertical assignment opportunities in the case of conflicts for the first teachers category, three possibilities for the second category and three possibilities for the third.

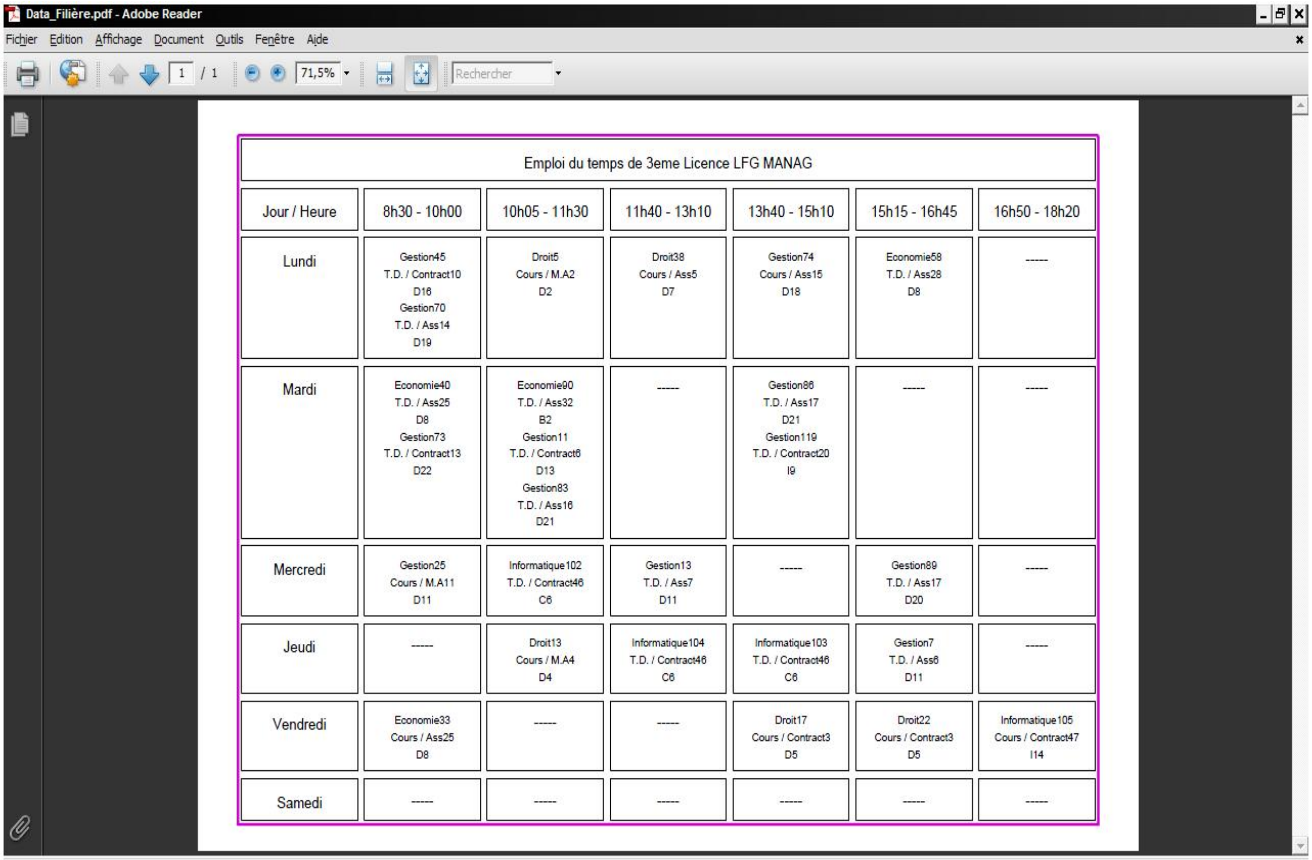
• For the second case, we changed the initial score to (4,8,8).

• For the third test case, the score was changed to (8,16,16).

• For the last test case, it was changed again to (16,32,32).

4.3.1 The variation of the number of messages

According to the figure 5, we have distinguished that the number of messages varies greatly depending on the assignment priority score (which will be taken by teachers ranking for the first category, by the lecture types "*Course*" for the second category and by the lecture types "*D.L*" or "*P.L*" for the third). In fact, this score allows to give new opportunities for vertical allocation (classroom equivalence) for each case where it was an unsatisfied request. Moreover, increasing the assignment priority score allows to influence on the variation of the number of messages for each new test case. Whenever a reservation request (or preference) hasn't been satisfied, the number of messages increases by searching other assignment opportunities in other equivalent classrooms and to more satisfy the teachers preferences. The execution process of our allocation algorithm between agents (TA, CA, HA) cannot be stopped only after a total assignment of all the "*teacher - lecture*"



Data_Filière.pdf - Adobe Reader

Emploi du temps de 3eme Licence LFG MANAG						
Jour / Heure	8h30 - 10h00	10h05 - 11h30	11h40 - 13h10	13h40 - 15h10	15h15 - 16h45	16h50 - 18h20
Lundi	Gestion45 T.D. / Contract10 D16 Gestion70 T.D. / Ass14 D19	Droit5 Cours / M.A2 D2	Droit38 Cours / Ass5 D7	Gestion74 Cours / Ass15 D18	Economie58 T.D. / Ass28 D8	-----
Mardi	Economie40 T.D. / Ass25 D8 Gestion73 T.D. / Contract13 D22	Economie90 T.D. / Ass32 B2 Gestion11 T.D. / Contract6 D13 Gestion83 T.D. / Ass16 D21	-----	Gestion86 T.D. / Ass17 D21 Gestion119 T.D. / Contract20 I9	-----	-----
Mercredi	Gestion25 Cours / M.A11 D11	Informatique102 T.D. / Contract46 C6	Gestion13 T.D. / Ass7 D11	-----	Gestion89 T.D. / Ass17 D20	-----
Jeudi	-----	Droit13 Cours / M.A4 D4	Informatique104 T.D. / Contract46 C6	Informatique103 T.D. / Contract46 C6	Gestion7 T.D. / Ass6 D11	-----
Vendredi	Economie33 Cours / Ass25 D8	-----	-----	Droit17 Cours / Contract3 D5	Droit22 Cours / Contract3 D5	Informatique105 Cours / Contract47 I14
Samedi	-----	-----	-----	-----	-----	-----

Figure 9. Speciality and education level timetable

combinations to the different classrooms.

4.3.2 The variation of the CPU time

In other hand, the messages variation will be accompanied by an increase of the execution time. The more the number of message increases, the more the C.P.U time increases. That's why and according to figure 6, we have concluded that the execution time depends strongly on the number of messages accumulated after each negotiation phase. But this time variation has always remained in second by 8,172 (for 2,4,4 ***VSi*** and 4044 messages) to 9.604 (for 16,32,32 ***VSi*** and 6428 messages) with an average of 8.885 seconds for this bound.

5. Timetable models

We presented the three possible types of timetabling generated by our M.A.T.P model:

- Teacher timetable.
- Classroom timetable.
- Speciality and education level timetable.

A first type of table for the teacher timetabling, see figure 7:

A second type of table for the classroom timetabling, see figure 8:

A third type of table for the speciality and education level timetabling, see figure 9:

6. Conclusion and perspectives

This paper proposed a multi-agent model based on cooperative agents, named M.A.T.P (Multi-Agent model for university

Timetabling Problem), to solve the university course timetabling problem, enabling highly parallel and distributed processing of this problem and incorporating new details that have not been considered by previous work. To evaluate our model, we choose to address a real case (instances of the Higher Business School of Tunis) where we have realized a test scenario by analyzing the variation of the number of messages in terms of the assignment priority score and its effect on the execution time. Our current researches are dealing with comparison between our model and other previous models in the literature in terms of the execution time and soft constraint satisfaction of the problem. Moreover, this model can be adapted to solve other forms of the personnel timetabling problem (hospitals, protection services and emergency, ...) in the future works.

References

[1] Abbas, A., Tsang, E., P., K. (2004). Software engineering aspects of constraintbased timetabling : a case study, *Information and Software Technology Journal*, 46, p. 359-372.

[2] Adewumi, A., O., Sawyerr, B., A., Ali, M., M. (2008). A heuristic solution to the university timetabling problem, Engineering Computations: *International Journal for Computer-Aided Engineering and Software*, 26 (8) 972-984.

[3] Babkin, E., Adbulrab, H., Babkina, T. (2007). AgentTime : A Distributed Multi-agent Software System for University's Timetabling, Emergent Properties in Natural and Artificial Complex Systems (ECCS'07) in European Conference on Complex Systems, p. 10-22.

[4] Belkahla Driss, O., Korbaa, O., Ghedira, K., Yim, P. (2007). A Distributed transient inter-production scheduling for flexible Manufacturing Journal Européen des Systems Automatisés, *JESA*., 41 (1) 101-123.

[5] Burke, E., K., Petrovic, S. (2002). Recent research directions in automated timetabling, *European Journal of Operational Research*, (EJOR), 140 (2) 266-280.

[6] Burke, E., K., Marecek, J., Parkes, A., J., Rudová, H. (2010). A supernodal formulation of vertex colouring with applications in course timetabling, *Ann Oper Res*, 179, p. 105-130.

[7] Ceschia, S., Di Gaspero, L., Schaerf, A. (2011). Design, Engineering, and Experimental Analysis of a Simulated Annealing Approach to the Post- Enrolment Course Timetabling Problem, *Computers and Operations Research*, 39, p. 1615-1624.

[8] Daskalaki, S., Birbas, T., Housos, E. (2004). An integer programming formulation for a case study in university timetabling, *European Journal of Operational Research*,153, p. 117-135.

[9] Ferber, J. (1999). Multi-Agent Systems: An introduction to Distributed Artificial Intelligence, Harlow, U.K, Addison Wesley, ISBN 0-201-36048-9.

[10] Henry-Obit, J., Landa-Silva, D., Ouelhadj, D., Khan-Vun, T., Rayner, A (2011). Designing a Multi-agent Approach System for Distributed Course Timetabling, 11th International Conference on Hybrid Intelligent Systems, p. 103-108, IEEE. Press.

[11] Nandhini, M., Kanmani, S (2009). Implementation of Class Timetabling Using Multi Agents, International Conference on Intelligent Agent & Multi-Agent Systems, p. 1-2, IEEE. Press.

[12] Oprea, M. (2006). Multi-Agent System for University Course Timetable Scheduling, The 1st International Conference on Virtual Learning, p. 231-237.

[13] Redl, T., A. (2007). University Timetabling via Graph Coloring: An Alternative Approach, University of Houston. Congressus Numerantium, 187, p. 174-186.

[14] Sheaufen, I., H., Safaai, D., Siti Zaiton, M., H. (2009). Investigating Constraint-Based Reasoning for University Timetabling Problem, *In*: Proceedings of the International MultiConference of Engineers and Computer Scientists VolI (IMECS), p. 139-143.

[15] Xiang, Y., Zhang, W. (2008). Distributed University Timetabling with Multiply Sectioned Constraint Networks, the Twenty-First International FLAIRS Conference, p. 567-571.

[16] Zhang, L., Lau, S. (2005). Constr ucting university timetable using constraint satisfaction programming approach, *International Conference on Computational Intelligence for Modelling, Control and Automation, and International Conference on Intelligent Agents,Web Technologies and Internet Commerce*, 2, p. 55-60, IEEE. Press.

[17] Zhipeng, L., Hao, J., K. (2008). Adaptive Tabu Search for Course Timetabling, *European journal of operational research* 200, p. 235 -244, Elsevier Press.