MATP: A Multi-agent Model for the University Timetabling Problem

Houssem Eddine Nouri and Olfa Belkahla Driss

Abstract This paper proposes a multi-agent model for solving the university course timetabling problem. It is composed of cooperating agents enabling highly distributed processing of the problem and incorporating constraints that have not been considered by previous works. The aim of our model is to provide a best solution satisfying hard and soft constraints while reducing temporal complexity. To analyze the efficiency of our model, we give experimental results based on real instances of the Higher Business School of Tunis by analyzing the variation effect of the lecture and teacher numbers on the messages number and the CPU execution time, and the variation effect of the assignment priority score on the percentage of teacher's preferences satisfaction.

Keywords University course timetabling problem · Multi-agent system · Negotiation · Messaging exchange system

1 Introduction

The timetabling problem is an instance of the personal scheduling problems which has become more diffused in our real life. 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.

H.E. Nouri (✉) · O.B. Driss
Stratégies d'Optimisation et Informatique intelligentE,
Institut Supérieur de Gestion de Tunis, Université de Tunis,
41, Avenue de La Liberté, Cité Bouchoucha, Bardo, Tunis, Tunisia
e-mail: houssemeddine.nouri@gmail.com

O.B. Driss
e-mail: olfa.belkahla@isg.rnu.tn

R. Silhavy et al. (eds.), *Software Engineering Perspectives and Application in Intelligent Systems*, Advances in Intelligent Systems and Computing 465,
DOI 10.1007/978-3-319-33622-0_2

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. Our aim is to get a best solution for this problem satisfying several hard and soft constraints while minimizing the temporal complexity.

This paper is organized as follows. In Sect. 2, we present how the university course timetabling problem is solved by previous works as well as its hard and soft constraints. We detail then in Sect. 3 our contribution based on multi-agent systems. Section 4 is devoted to the presentation of a real case study (instances of the Higher Business School of Tunis) in order to test our model as well as a set of scenarios 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 based on the theory of graphs [6, 12], the integer linear programming [8] and the techniques of constraint satisfaction problem [1, 13, 15]. 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 meta-heuristics such as the tabu search [16], 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 [3, 10–11, 14], 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 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 should be satisfied. These constraints are often classified into two categories, the first includes hard constraints and the second category includes constraints often called soft constraints:

Hard constraints: 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, that we have named MATP a Multi-Agent model for university Timetabling Problem, 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 and to act consequently, 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 Fig. 1, with three classes of agents. The first class is composed of agents that we have named TA, "Teacher Agents", divided into three categories of teachers: *C1*: Professor, Associate-professor; *C2*: Assistant-professor, Assistant; *C3*: Contractual. The second class is composed of agents that we called CA, "Classroom Agents", divided into three types of class-rooms ("Course", "Tutorial Class", 'Practical Class",) related to the type of the lecture session. The third class contains three agents: two "Interface Agents" that we called IA1 and IA2 and one "History Agent" that we named HA.

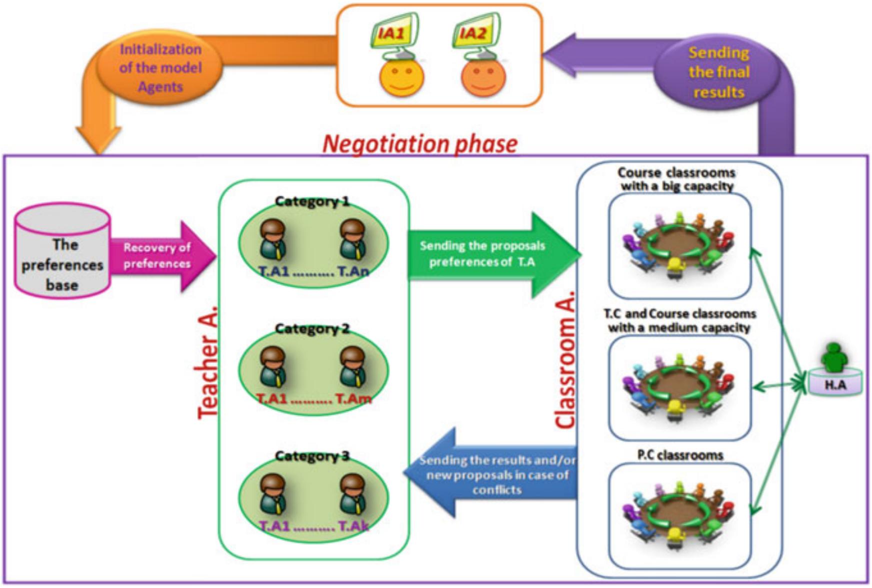

Fig. 1 Multi-agent model for university timetabling problem

3.2 Global Steps of MATP

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

Initialization phase. In this phase, we present the role of the agent IA1 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.

Negotiation phase. This phase is the kernel of our model. It is based on a messaging exchange system between the two agent's classes TA and CA in order to have in each case an agreement between them, respecting all the hard constraints of this problem. The first class of agents TA starts the negotiation process by sending all their allocation propositions (which were recovered from their preferences base) to the CA agents in order to get a better reservation of the most suitable classrooms and the most favorite time periods of the day.

The second class of agents CA will receive and analyze the TA agent's preferences. In fact, this class will ask the HA agent to verify the existence of duplication of time periods for a same TA agent in each reception of propositions. Thus, it allows either to validate, or to give a new proposition in the case of conflict. The CA may have 1 or n TA propositions asking the same period in the same day, and generating conflicts between them, see Fig. 2. That's why, we have added a

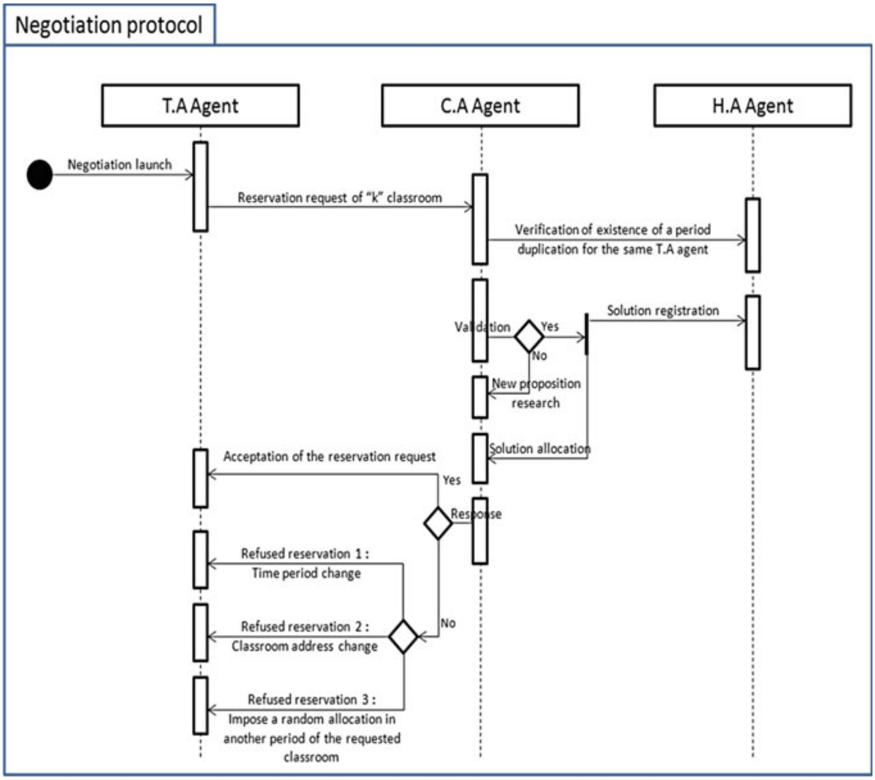

Fig. 2 Interaction protocol diagram

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 TA agent with a category ***i*** where:

- Priority 1, ***VS1max*** (Rank of teacher): This score is given for each agent belonging to the first category of teacher agents TA having a rank of "Professor" or "Associate-professor".
- Priority 2, ***VS2max*** ("Course"): This score is given for each agent belonging to the second category of teacher agents TA and asking a lecture session with type "course".

- Priority 3, ***VS3max*** ("TC" or "PC"): This score is given for each agent belonging to the third category of teacher agents TA and asking a lecture session with type "TC" (Tutorial Class) or "PC" (Practical Class).

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 TA agents.

Transmission of final results. Whenever a TA agent receives all solutions in response to its messages, it finishes its negotiation phase and transmits its final results to IA1 agent generating the form of teacher's timetable. Then the agent IA2 ends the process by generating the final timetable of the different classrooms.

3.3 Agent Behaviour

Interface agent behaviour. The behaviour of the IA1 is to initialize all the other agents of our model. Then, it moves to an inactive state pending the reception of the final TA agent messages to generate them in the form of a solution for the teacher timetabling problem. For the IA2 behaviour, this latter has to generate a solution for the classroom timetabling problem after the end of the negotiation process.

Teacher agent behaviour. A TA agent possesses a group of lectures (which can be a course, TD or TP) that it seeks to assign them to classrooms in the most favourite periods of the day. In fact, each TA begins its negotiation phase by sending its proposals to CA agents requesting the most preferred classrooms and teaching periods. Then, he receives a response message from CA:

- Verification of 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.
```

Classroom agents behaviour. A CA 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:

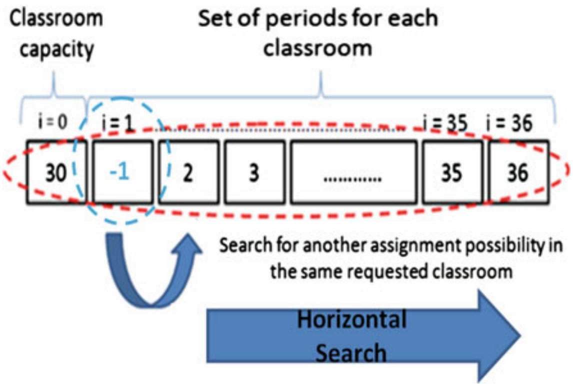

Fig. 3 Vertical assignment search

- Verification of duplication part:

```
Begin
If The requested period hasn't been duplicated for the
same TA agent.
   Then Go to the validation step.
   Else Change the requested period and send a message of
   a refused reservation 1 to TA agent.
End.
```

- Validation of proposition part (Figs 3 and 4):

```
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
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
   TA agent, see Figure 3.
Else-If The requested period has been reserved and
VSi > VSimax.
   Then Horizontal assignment search: impose a random as-
   signment in another available period of the requested
   classroom and send a message of a refused reservation 3
   to TA agent, see Figure 4.
End.
```

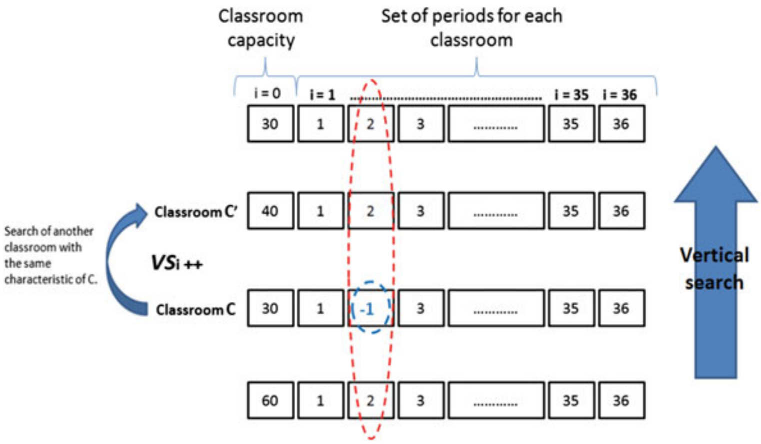

Fig. 4 Horizontal assignment search

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.

Number and types of teachers.

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

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 TC and course classrooms having a medium capacity of students (B1, B2, B3, B4, D1…D24, C2…C11, I8… I21). Category 3: all PC classrooms (I1, I2, I3, I4, I5, I6, I7).

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 [4] 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.

4.3 Experimental Results

To evaluate the efficiency of our model, we realized a two test scenarios, where we have analyzed:

- The variation effect of the lecture and teacher numbers on the message number and the CPU execution time.
- The variation effect of the assignment priority score on the percentage of Teachers preferences satisfaction.

The variation effect of the lecture and teacher numbers on the message number and the CPU execution time. For this first test scenario, we analyzed the effect of varying the number of lectures and teachers on the messages number and the execution time. In fact, we realized 4 tests where we incremented in each case the lectures number as well as the teachers while fixing the assignment priority score (SV1 = 4, SV2 = 5, SV3 = 5: random choice) for the three TA agents categories until to the end of those tests.

According to the results, see Table 1, we can visualize the rapid increase of the message number and this is due to the increment of the lecture number for each test case, requiring more communication flows between agents to satisfy all the reservation requests in the negotiation phase. In other hand, the CPU time has known a small increase in each test, but its variation has always remained in second by 5.688 (for 540 lectures) to 8.531 (for 868 lectures) with an average of 6.976 s for this bound.

Table 1 Variation effect of the lecture and teacher number on the message number

Scenario 1	Lecture number	Teacher number	Assignment percentage (teacher—lecture—classroom) (%)	Message number	CPU time (s)
Test 1	540	85	100	2732	5.688
Test 2	660	100	100	3438	6.547
Test 3	728	111	100	3880	7.14
Test 4	868	131	100	4584	8.531

Table 2 Variation effect of the priority score VSi on the teachers preferences satisfaction

Scenario 2.1	VS1	VS2	VS3	Message number	CPU time (s)	Teachers preferences satisfaction in %
Test 1	2	3	3	4114	8.265	64.97
Test 2	4	5	5	4483	8.531	67.51
Test 3	4	7	7	4746	8.688	71.19
Test 4	4	10	10	5238	9.078	72.46

Table 3 Percentage of preferences satisfaction by category

Scenario 2.2	Preferences satisfaction in % (category 1)	Preferences satisfaction in % (category 2)	Preferences satisfaction in % (category 3)	Teachers preferences satisfaction in %
Test 1	100	77.27	49	64.97
Test 2	100	77.27	54.50	67.51
Test 3	100	77.80	62	71.19
Test 4	100	78.18	64.25	72.46

Moreover, the allocation percentage of the three resources "teacher—lecture—classroom" reached the 100 % for the 4 test cases. In fact, 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" combinations to the different classrooms.

The variation effect of the assignment priority score on the percentage of Teachers preferences satisfaction. For this second test scenario, we chose to analyze the variation of the percentage of teachers preferences satisfaction by increasing in each time the assignment priority score for each category of Teacher Agents and fixing the lectures number to 868 and teachers to 131.

In view of these results, see Table 2, we distinguished that the first category of teachers has had the largest percentage of preferences satisfaction compared to the other categories, see Table 3. In fact, it knew a total satisfaction of all its allocation needs counting a percentage of 100 % from the first test case with a score of 2 assignment possibilities, this percentage remained unchanged until the last test.

The second category has had a percentage of 77.27 % with 3 assignment possibilities for the first test. This percentage has remained unchanged in the second test and then it knew a small improvement of 0.45 % by increasing its score to 7 assignment possibilities in the third test. Passing to the last test, it has been improved again by 0.46 % by increasing its score to 10 possibilities.

The third category has had a percentage of 49 % with 3 assignment possibilities for the first test and then it knew a small improvement of 5.5 % by increasing its score to 5 assignment possibilities. And this percentage was changed again in the third test by modifying the assignment score to 7 possibilities with an increase of 7.5 %. Passing to the last test, it knew another improvement of 2.25 % by increasing its score to 10 possibilities.

From these interpretations, we can conclude that the variation of the percentage of the preferences satisfaction for the three teacher's categories strongly depends on the assignment priority score SVi. In fact, see Table 3, the increase of the SVi will generate an improvement of the percentage of the teacher's preferences satisfaction by giving new assignment possibilities to other classrooms having the same characteristics of the requested one.

5 Conclusion and Perspectives

This paper proposed a multi-agent model based on cooperative agents, named MATP (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. The performances of the proposed model are exhibited through experimental results based on a real case study (instance of the higher business school of Tunis) by analyzing the variation effect of the lectures and teachers number on the messages number and the CPU execution time and the variation effect of the assignment priority score on the percentage of teachers preferences satisfaction.

Despite the encouraging results, some improvements are still possible. We can improve the intelligence level and the individual learning of our system agents, in such way that agents have a deliberative behavior such as B.D.I (Belief Desire Intention). Moreover, this model can be adapted to solve other forms of the personnel timetabling problem (hospitals, protection services and emergency, etc.) in the future works.

References

1. Abbas, A., Tsang, E.P.K.: Software engineering aspects of constraint based timetabling: a case study. Inform. Softw. Technol. J. **46**, 359–372 (2004)
2. Adewumi, A.O., Sawyerr, B.A., Ali, M.M.: A heuristic solution to the university timetabling problem. Eng. Comput.: Int. J. Comput.-Aided Eng. Softw. **26**(8), 972–984 (2008)
3. Babkin, E., Adbulrab, H., Babkina, T.: AgentTime: a distributed multi-agent software system for university's timetabling. In: Proceedings of the 4th European Conference on Complex Systems, a satellite Conference Emergent Properties in Natural and Artificial Complex Systems, pp. 10–22 (2007)
4. Bellifemine, F., Poggi, A., Rimassa, G.: JADE—A FIPA-compliant agent framework. In: Proceedings of the 4th International Conference and Exhibition on the Practical Application of Intelligent Agents and Multi-Agent Technology, pp. 97–108 (1999)
5. Burke, E.K., Petrovic, S.: Recent research directions in automated timetabling. Eur. J. Oper. Res. **140**(2), 266–280 (2002)
6. Burke, E.K., Marecek, J., Parkes, A.J., Rudová, H.: A supernodal formulation of vertex coloring with applications in course timetabling. Ann. Oper. Res. **179**, 105–130 (2010)

7. Ceschia, S., Di Gaspero, L., Schaerf, A.: Design, engineering, and experimental analysis of a simulated annealing approach to the post-enrolment course timetabling problem. Comput. Oper. Res. **39**, 1615–1624 (2011)
8. Daskalaki, S., Birbas, T., Housos, E.: An integer programming formulation for a case study in university timetabling. Eur. J. Oper. Res. **153**, 117–135 (2004)
9. Ferber, J.: Multi-agent Systems—An Introduction to Distributed Artificial Intelligence. Addison- Wesley (1999). ISBN: 0-201-36048-9
10. Henry-Obit, J., Landa-Silva, D., Ouelhadj, D., Khan-Vun, T., Rayner, A.: Designing a Multi-agent approach system for distributed course timetabling. In: Proceedings of the Hybrid Intelligent Systems Conference, pp. 103–108. IEEE Press (2011)
11. Oprea, M.: Multi-Agent system for university course timetable scheduling. In: The 1st International Conference on Virtual Learning, pp. 231–237 (2006)
12. Redl, T.A.: University timetabling via graph coloring: an alternative approach. Congr. Numer. **187**, 174–186 (2007)
13. Sheaufen, I.H., Safaai, D., Siti Zaiton, M.H.: Investigating constraint-based reasoning for university timetabling problem. In: Proceedings of the International Multi-Conference of Engineers and Computer Scientists vol. 1, pp. 139–143 (2009)
14. Xiang, Y., Zhang, W.: Distributed university timetabling with multiply sectioned constraint networks. In: the Twenty-First International FLAIRS Conference, pp. 567–571 (2008)
15. Zhang, L., Lau, S.: Constructing university timetable using constraint satisfaction programming approach. In: Proceedings of the International Conference on Computational Intelligence for Modelling, Control and Automation and International Conference on Intelligent Agents, Web Technologies and Internet Commerce, vol. 2, pp. 55–60. IEEE Press (2005)
16. Zhipeng, L., Hao, J.K.: Adaptive tabu search for course timetabling. Eur. J. Oper. Res. **200**, 235–244 (2008)

http://www.springer.com/978-3-319-33620-6

Software Engineering Perspectives and Application in Intelligent Systems
Proceedings of the 5th Computer Science On-line Conference 2016 (CSOC2016), Vol 2
Silhavy, R.; Senkerik, R.; Oplatkova, Z.K.; Silhavy, P.; Prokopova, Z. (Eds.)
2016, XIV, 484 p. 187 illus., 134 illus. in color., Softcover
ISBN: 978-3-319-33620-6