

An Agent Based General Solution Model For the Course Timetabling Problem

Yan Yang
Electronic Systems Engineering
TRLabs Regina
University of Regina
1-306-7916141

roc_young@hotmail.com

Raman Paranjape
Electronic Systems Engineering
TRLabs Regina
University of Regina
1-306-7916137

raman.paranjape@uregina.ca

Luigi Benedicenti
Software Systems Engineering
TRLabs Regina
University of Regina
1-306-5854855

luigi.benedicenti@uregina.ca

ABSTRACT

This paper proposes a general solution model using agent technology in order to solve two critical open issues in the course timetabling problem: application-specific issues and dynamic-constraint issues. In our solution model, each agent in the system represents a constraint in the course timetabling problem. Compared to existing course timetabling solutions, the proposed agent solution model has much better generality, flexibility, dynamics, modularity and scalability.

Categories and Subject Descriptors

I.2.11 [Artificial Intelligence]: Distributed Artificial Intelligence-Multiagent systems, Intelligent agents.

General Terms: Algorithms, Design, Experimentation

Keywords: Multi-agent Architecture, Timetabling Problem

1. INTRODUCTION

There are fundamentally two open issues in the course timetabling problem [1-3, 8]. One is the application-specific issue in which a solution to one course timetabling problem may not be suitable for another one. The other is the dynamic-constraint issue in which the course timetabling problem becomes much harder when the constraints change dynamically. This paper proposes a general solution model using agent technology [4, 5, 9, 10] and attempts to solve the above issues using a system design approach. In this paper, the agent technology is based on the TEEMA (TRLabs Execution Environment for Mobile Agents) platform [7].

2. AN AGENT SOLUTION MODEL

No matter how different they are, any course timetabling problem will have certain common hard constraints: 1) no student has more than one course at the same time; 2) only one course is scheduled in a classroom at any one time; 3) the classroom

assigned to any course is large enough to hold all enrolled students. Identifying the common hard constraints provides the basis to establish a general solution model for the course timetabling problem using agent technology.

A course is an event involving course name, class time, classroom and people, i.e., students and instructors. There are a set of constraints which govern the relationships between these entities. In the proposed agent solution model, each agent represents a constraint in the course timetabling problem. The model includes the following agents: the **CHC** Agent which represents the Common Hard Constraints (identified above); the **HC** Agents which represent other additional Hard Constraints (such as no class scheduled during the lunch break time); the **SC** Agents which represent the Soft Constraints (such as no student assigned more than two classes consecutively). There is only one CHC Agent in the system. The number of HC and SC Agents depends on the specific course timetabling problem. The constraint agents are independent of one another. The system design is presented in three different modes of operation: the passive mode, the active mode and the parallel-active mode.

2.1 Passive Mode

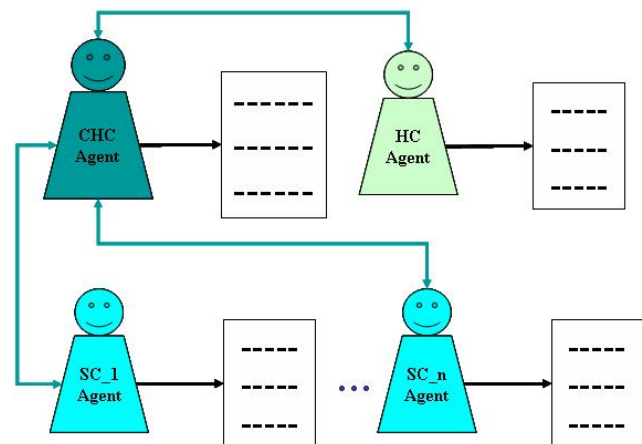


Figure 1. Passive mode of the general solution model

As shown in Figure 1, the passive mode general solution model is composed of one CHC Agent, one HC Agent and a number of SC Agents. The hard constraints can be represented by one HC agent (as shown in the figure) or a number of them (similar to the soft constraints). In order to simplify the illustration, only one HC

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. to copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

AAMAS'06, May 8-12, 2006, Hakodate, Hokkaido, Japan.

Copyright 2006 ACM 1-59593-303-4/06/0005...\$5.00.

Agent is used. Generally, there are more than one soft constraint in the course timetabling problem. So in the solution model, the soft constraint agents are represented by the SC_1 Agent through SC_n Agent and the value of n depends on the number of soft constraints. The notepad symbol in the figure represents a course timetable. The solid polyline with bi-directional arrows between agents represents their communication. These symbol conventions will also apply in subsequent figures.

During the course timetabling process, the CHC Agent first produces a basic feasible course timetable consistent with the common hard constraints. Thereafter it sends a copy of the basic feasible timetable to each of other constraint agents. Each constraint agent evaluates the timetable based on its own constraint and sets penalty points on the number of constraint violations. These evaluations are sent to the CHC Agent and it develops an overall evaluation result on the basic feasible timetable. Following this, the CHC Agent creates a small disturbance to the timetable and guarantees that the disturbance does not violate the common hard constraints. The small disturbance can be realized by either moving one course to another available position in the timetable or exchanging the positions between two or three courses. As a timetable improvement proposal, the disturbance is sent to the constraint agents for evaluation. The constraint agents evaluate the proposal and provide feedback to the CHC Agent. After developing a new overall evaluation result on the modified timetable, the CHC Agent decides if the modified timetable represents an improved schedule. If the modified timetable is accepted, the CHC Agent will adopt it as the current best timetable and inform other constraint agents to make the same change. In this way, the system can guarantee that at any moment the course timetable carried by each agent is exactly the same. If the modified timetable is rejected, the CHC Agent will discard it. The CHC Agent then continues looking for a new proposal attempting to further improve the timetable. This process is repeated until all the constraints violations are eliminated and a final course timetable satisfying all the constraints is found. On the other hand, if the constraints violation can not be reduced further, the course timetabling process will go to the active mode described in the next subsection.

In this timetabling mode, only CHC Agent makes proposals and all the other constraint agents passively receive and evaluate proposals. For this reason we call this the passive mode.

2.2 Active Mode

The active mode is presented in Figure 2. From the figure it can be seen that each constraint agent has connections to all of others. In addition, there is a token which is passed around between the constraint agents. Only the agent owning the token has the right to make proposals to try to improve the course timetable and the proposals are guaranteed not to violate its own constraint. If a proposal violates the common hard constraints, it will be rejected because all timetables must always be basic feasible course timetable, even if they do not reduce other constraint violations. Otherwise, the timetabling process is similar to that described in the previous section.

In the active mode, all the agents equally contribute to improve the course timetable based on their own constraints. This allows the constraint violations to be reduced more effectively and

efficiently. That is also the reason it is called the active mode.

If a solution can not be found after the token has been passed for a certain number of rounds among all the agents, the timetabling process is ended. Like other existing metaheuristics course timetabling solutions [6], the proposed method can not guarantee an optimal solution satisfying all the constraints. However, it has some key advantages over other existing methods which are identified in later sections.

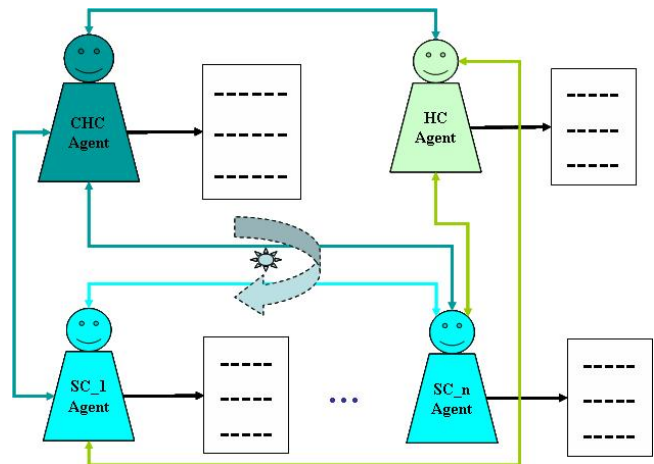


Figure 2. Active mode of the general solution model

2.3 Parallel-active Mode

The active mode can be further modified to make the token unnecessary allowing all the constraint agents to work in parallel. In this situation, the CHC Agent acts as a mediator to judge if the proposals are valid or not. Furthermore, each agent can adopt a different algorithm to create a hybrid and parallel course timetabling solution model. In such an operation mode, the system control would be much more complex, but it could be expected that the problem may be solved more quickly and more effectively.

3. EXPERIMENTAL RESULTS

The problem instances from Metaheuristic Network (<http://www.metaheuristics.org/>) are chosen to demonstrate the behavior of the new model. The problem description [6] is presented as follows:

The timetabling problem consists of a set of events to be scheduled in 45 timeslots (5 days of 9 hours each), a set of rooms in which events can take place, a set of students who attend the events, and a set of features satisfied by rooms and required by events. Each student attends a number of events and each room has a size. A feasible timetable is one in which all events have been assigned a timeslot and a room so that the following hard constraints are satisfied: 1) no student attends more than one event at the same time; 2) the room is big enough for all the attending students and satisfies all the features required by the event; 3) only one event is in each room at any timeslot.

Currently only the CHC Agent (representing the above three common hard constraints) and one SC agent (representing one of soft constraints in [6]: no student has a class in the last timeslot of

a day) have been implemented. The solution satisfying all the three common hard constraints and the mentioned soft constraint can be found through cooperation between the CHC Agent and SC Agent.

A portion of course timetabling results is shown in Figure 3. There are 45 timeslots (5 days of 9 hours each) and they are denoted by 0 through 44 in the implemented program. Timeslots No.8, 17, 26, 35, 44 are the last timeslots for each day of the week. Classrooms are denoted by integer numbers starting from 0. In the figure, the timeslot No.8 has no course scheduled. This is consistent with the soft constraint that students not have to attend the last timeslot of the day.

Time_Room	Room Size (No. of Students)	Event
=====		
MONDAY		
7_0	9(9)	E28
7_1	2(2)	E74
7_2	3(3)	E29
7_3	11(10)	E25
7_4	15(15)	E32
=====		
Time Slot No.8: No event scheduled on this time slot!		
=====		
TUESDAY		
9_0	9(9)	E31
9_1	2(2)	E83
9_2	3(3)	E88
9_3	11(11)	E37
9_4	15(15)	E43

Figure 3. A portion of course timetable

4. DISCUSSION AND FUTURE WORK

Although the quantitative assessment of the proposed solution model as compared with other existing course timetabling methods [6] has not been fully performed, some important qualitative advantages can be identified. Because each constraint agent is an independent module in the solution model, new constraint agents can be easily added into or removed from the timetabling system. This means that the dynamic-constraint issue is resolved through the system design. The proposed system design is extremely flexible, for example, course agents which represent individual courses can be added into the system. This would allow the proposed solution to adapt to the dynamic changes in schedule mandated even by individual courses.

The system design solves the application-specific issue in the course timetabling problem by allowing the user to select appropriate HC and SC Agents for their specific problem domain. The specific problem domain constraint agents make the solution model application-specific. Therefore the proposed solution is very general and flexible. The solution model's modularity means it can have any number of agents added into or removed from the system. This means there is a good scalability for the system as well.

The agents in the system can be mobile agents. In such a situation, the solution model can be used in the distributed environment and can take advantage of distributed problem solving. If the parallel- active mode (discussed in the subsection 2.3) is adopted, the proposed solution model will have the ability to solve problems in parallel.

To summarize, the advantages of the proposed solution model in this work over other existing approaches are as follows:

- Dynamic Problem Solving
- Good Flexibility and Generality
- Good Scalability and Modularity
- Distributed and Parallel Problem Solving

In order to explore the complete solution model, other soft constraint agent modules will be implemented. The parallel-active timetabling mode will also be investigated. Finally, quantitative comparison between the proposed solution model and other existing models will be made.

5. CONCLUSION

The proposed multi-agent general solution model for the course timetabling problem presented in this paper overcomes two key difficulties of the timetabling problem: application-specific issues and dynamic-constraint issues. The experimental results indicate that our model develops promising avenue of research in the course timetabling problem. It is reasonable to anticipate that the proposed solution model can be adapted to other scheduling problems.

6. REFERENCES

- [1] Andrea Schaerf. *A Survey of Automated Timetabling*. Artificial Intelligence Review, 13: 87-127, 1999.
- [2] Dominique de Werra. *An introduction to timetabling*. European Journal of Operational Research, 19: 151-162, 1985.
- [3] Edmund Kieran Burke and Sanja Petrovic. *Recent research directions in automated timetabling*. European Journal of Operational Research, 140: 266-280, 2002.
- [4] Hyacinth S. Nwana. *Software Agents: An Overview*. Knowledge Engineering Review, 11(3): 205-244, 1996.
- [5] Michael J. Wooldridge, Nicholas R. Jennings. *Intelligent Agents: Theory and Practice*. Knowledge Engineering Review, 10(2): 115-152, 1995.
- [6] Olivia Rossi-Doria, et al. *A Comparison of the Performance of Different Metaheuristics on the Timetabling Problem*. E. Burke and P. De Causmaecker (Eds.): PATAT 2002, LNCS 2740: 329-351, 2003.
- [7] Robert Martens, L. Benedicenti. *TEEMA TRLabs Execution Environment for Mobile Agents*. Technical Report, TRLabs Regina, Saskatchewan, Canada, 2001.
- [8] Sanja Petrovic and Edmund Burke. *University Timetabling*. J. Leung (Eds.): Ch.45 in the Handbook of Scheduling: Algorithms, Models, and Performance Analysis, CRC Press, 2004.
- [9] Yan Yang, Raman Paranjape and Luigi Benedicenti. *An Examination of Mobile Agents System Evolution in the Course-Scheduling Problem*. In Proceedings of the 2004 IEEE Canadian Conference on Electrical and Computer Engineering, Ontario, Canada, 5782-5785, May 2-5, 2004.
- [10] Yan Yang, Raman Paranjape, Luigi Benedicenti and Nancy Reed. *A Mobile Agent System for University Course Timetabling*. In Proceedings of the 2nd Indian International Conference on Artificial Intelligence, Pune, India, December 20-22 2005.