Investigating Constraint-Based Reasoning for University Timetabling Problem

Ho Sheau Fen @ Irene, Safaai-Deris, and Siti Zaiton-Mohd Hashim

***Abstract*—University Course Timetabling Problems (UCTP) are a NP-complete, real-world problems, with many soft and hard constraints involved. The difficulties in generating a good timetable solution are different in almost every institution. This paper investigates the constraint-based reasoning algorithm to solve this complex UCTP. The approach used constraints-based reasoning to search for the best preference value base on the student capacity for each lecture. The proposed algorithm has been tested using a real world data from Faculty of Computer Science and Information System, University of Technology Malaysia and Faculty of Science at Ibb University – Yemen.**

***Index Terms*—University course timetabling problem, constraint-based reasoning.**

I. INTRODUCTION

Scheduling problems are in the core of many real-world applications. They occur in the areas of production planning, timetabling and personnel planning.

The university course timetabling problem (UCTP) is part of the scheduling problem. It is a large, highly constrained and much more complicated problem. Problems of university course timetable (UCT) are greatly differing from university to university. The main task of UCTP is to allocate a number of lectures into a set of timeslots and rooms while satisfying the desire constraints. In UCTP there are divided into two different types of constraints: the hard constraints and the soft constraints. Hard constraints are constraints that needed to be satisfied irrespective of any environment. Violating hard constraints will cause the timetable solution infeasible, while violation of soft constraints will not cause the solution to lose its infeasibility.

Over the past few years, a wide variety of techniques have been proposed in solving the UCTP and its variants. Several techniques have been developed for automated timetables generations [4]-[6]. Other popular techniques including graph coloring algorithms [4], simulated annealing [7], tabu search [8] and genetic algorithms [9]-[10]. Constraint-based Programming [11]-[13] have been investigated to solve timetabling problems recently.

This paper investigates the constraint-based reasoning (CBR) to solve UCTP. The proposed algorithm is tested using real world data of Faculty of Computer Science and Information System, University of Technology Malaysia and Faculty of Science at Ibb University – Yemen.

II. UNIVERSITY COURSE TIMETABLING PROBLEMS

The UCTP for Faculty of Computer Science and Information System, University of Technology Malaysia and Information System, University of Technology Malaysia consists of a set of lectures to be scheduled in 35 timeslots (five days of nine hours each), a set of rooms in which the lectures take place and a set of students who attend the lectures, while Faculty of Science at Ibb University – Yemen consists of a set of lectures to be scheduled in 18 timeslots (six days of three hours each), a set of rooms in which the lectures take place and a set of students who attend the lectures. Each student attends a number of lectures and each room has a maximum capacity. A feasible timetable is one in which all lectures have been assigned to timeslots and rooms while all hard constraints are satisfied. The hard constraints that we are concerned with are as follows:

- Non-clashing subject constraints - same lecture should not be assigned to the same timeslot.
- Non-clashing room constraints – one room should not be assigned to more than one lecture for the same timeslot.
- Room capacity constraints – the number of students of a lecture assigned to a room should be less than or equal to the capacity of the room.
- Room and time-domain constraints – rooms or timeslots assigned to lectures must be within the range of the domain.

In addition, the soft constraints that we are focusing are as follows:

- The scheduled time of the lecture should fall within the preference sets as much as possible.
- The scheduled room of the lecture should fall within the preference sets as much as possible.

Infeasible timetable solutions are worthless and the objective is to maximize the usage of good timeslots and

Manuscript received December 30, 2008. This work was supported by the Malaysian Ministry of Science, Technology, and Innovation (MOSTI) and University Technology of Malaysia Research Management Centre (RMC) for their support (research grant number: Vote 79240).

Sheau Fen @ Irene-Ho is with the University of Technology Malaysia, Faculty of Computer Science & Information System, 81310 Johor, Malaysia (phone: +6019-8331338; e-mail: ireneluv@ hotmail.com).

Safaai-Deris is with University of Technology Malaysia, Faculty of Computer Science & Information System, 81310 Johor, Malaysia (e-mail: safaai@utm.my).

Siti Zaiton-Mohd Hashim is with the University of Technology Malaysia, Faculty of Computer Science & Information System, 81310 Johor, Malaysia (e-mail: sitizaiton@utm.my).

rooms. Early timeslots and near rooms are being categorized as good timeslots and rooms. UCTP are constructed in such a way that an optimal timetable solution exists with maximum usage of good timeslots and rooms. Figure 1 and 2 shows the structure of the weekly university course timetable (UCT) for both universities.

Time	Monday	Tuesday	Wednesday	Thursday	Friday
8:00 – 8:50	Subject Room	.	.	.	.
9:00 – 9:50	.	.	.	Subject Room	.
10:00 – 10:50	.	Subject Room	.	.	.
11:00 – 11:50	.	.	.	.	.
12:00 – 12:50	.	.	.	.	
13:00 – 13:50					
14:00 – 14:50	.	.		.	
15:00 – 15:50	.	.		.	.
16:00 – 16:50	.	.		.	.

Reserved timeslots for non-academic activities and lunch hours

Fig. 1. Structure of weekly UCT of Faculty of Computer Science and Information System, University of Technology Malaysia

Time	Saturday	Sunday	Monday	Tuesday	Wednesday	Thursday
8:00 – 11.00	Subject Room	.	.	.	.	.
11:00 – 14:00	.	.	.	Subject Room	.	.
14:00 – 17:00	.	Subject Room	.	.	.	Subject Room

Fig. 2. Structure of weekly UCT of Information System, University of Technology Malaysia and Faculty of Science at Ibb University – Yemen

III. DATA PREPARATION

Search in constraint programming (CP) paradigm tries to explore as much as possible the information about the current iteration before it moves to other iteration. Based on the CP philosophy, the timetable data is prepared before the search begins. Next, they are used for constraint propagation to provide domain reduction [1]. The UCTP data can be divided as follows:

- Arrays of list for the rooms with its maximum capacity included.
- Arrays of lists for the rooms and timeslots preferences (good timeslots and rooms will have higher preferences value).
- Arrays of list for the lessons with number of students attached.

The algorithm will run from the first lecture till the last lecture and finally stop when all the lectures are being allocated.

IV. CONSTRAINTS SATISFACTION PROBLEMS

Constraints satisfaction problems (CSPs) are a decision problem that defined as a set of objects whose state must satisfy a number of constraints [2]. From a general point of view, CSP is classified into two main groups:

- *Complete methods* aim at exploring the whole search in order to find all the solutions or to detect that the CSP is not consistent. Concerning complete resolution technique methods, our concern is to focus on backtracking search which is one of the techniques in constraint programming.
- *Incomplete methods* mainly rely to the use of heuristics providing a more efficient exploration of interesting areas of the search space in order to find some solutions. For example the local search (LS) techniques.

V. CONSTRAINTS PROCESSING ALGORITHMS

The main objective of constraint-processing procedures is to search for good timeslots and rooms for the particular lecture. If the room and timeslot value is consistent, then it will be selected for that particular event. If it is not consistent, a backtracking procedure will be performing to select next room and timeslot based on preferred score [3]. Figure 3 shows the example of backtracking search tree when inconsistent value occurred.

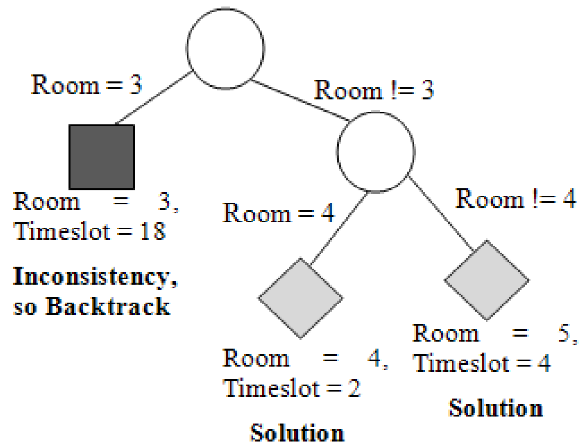

Fig. 3. Example of backtracking search tree

VI. FITNESS FUNCTION

The fitness function used was to focus on optimizing the preferences [12]. The need of this fitness function was to optimize the utilization of good timeslots and rooms. This is because room has different kind of facilities, accessibilities and locations. As for timeslots, all good timeslots should be selected first before the less favorable timeslots are chosen. Thus, the fitness function $f(t)$ of a timetable t is given as follow:

$$f(t) = \sum_{i=1}^{n} \Big(P\big(T(S_i)\big) + P\big(R(S_i)\big) \Big)$$

Where $P(T(S_i))$ are timeslot preferences for subjects S_i, $i=1,2,\ldots,n$ and $P(R(S_i))$ are room preferences for subjects S_i, $i=1,2,\ldots,n$. The fitness function $f(t)$ for each particle (solution) is computed by the total sum of preference value of timeslot $P(T(S_i))$ and preference value of room $P(R(S_i))$ for subject S_i, $i=1,2,\ldots,n$. This function enables a search for a near-optimal and feasible solution with the highest preference values of $f(t)$ and is designed to maximize the total sum of rooms and timeslots preference values.

VII. EXPERIMENTAL RESULTS

The proposed algorithm has been tested using timetable data from of Faculty of Computer Science and Information System, University of Technology Malaysia and Faculty of Science at Ibb University – Yemen. Table I and Table II summarize the information for both universities. Table III and Table IV show the preferences and ordering for timeslots and rooms of Faculty of Computer Science and Information System, University of Technology Malaysia while Table V and Table VI shows the preferences and ordering for timeslots and rooms of Faculty of Science at Ibb University – Yemen. The implementation has been developed in C. Experimental results are obtained by using notebook which has Intel Core 2 Duo 2.2 GHz processor and 2 GB RAM.

Table I. Summary of the Faculty of Computer Science and Information System, University of Technology Malaysia

Resources	Value
No. of Subjects	183
No. of Rooms	21
Total Timeslots	45
Total Timeslots Reserved	10
Total Timeslots Available	35

Table I. Summary of the Faculty of Science at Ibb University – Yemen

Resources	Value
No. of Subjects	136
No. of Lessons	226
No. of Rooms	16
Total Timeslots	18

Table III. Timeslots ordering and preferences

Timeslot no.	No. of lectures per week	Preference score	Order
1,2,3,4	4	3	1,2,3,4
5,6,7,8	4	2	5,6,7,8
9,10,11	3	3	9,10,11
12,13,14	3	3	12,13,14
15,16,17	3	3	15,16,17
18,19,20	3	2	18,19,20
21,22,23	3	1	21,22,23
24,25,26	3	1	24,25,26
27,28,29	3	1	27,28,29
30,31	2	3	30,31
32,33	2	3	32,33
34,35	2	2	34,35

Table IV. Rooms ordering and preferences

Room no.	Capacity (student no.)	Preference score	Order
1	20	2	1
2	20	1	2
3	30	2	3
4	30	2	4
5	30	2	5
6	30	1	6
7	30	1	7
8	50	1	8
9	60	2	9
10	60	2	10
11	60	2	11
12	60	2	12
13	60	2	13
14	60	1	14
15	60	1	15
16	60	1	16
17	60	1	17
18	60	1	18
19	150	2	19
20	150	1	20
21	150	1	21

Table V. Timeslots ordering and preferences

Timeslot no.	Preference score	Order
1	2	1
2	2	2
3	2	3
4	2	4
5	2	5
6	2	6
7	3	7
8	3	8
9	3	9
10	3	10
11	3	11
12	3	12
13	1	13
14	1	14
15	1	15
16	1	16
17	1	17
18	1	18

Table VI. Rooms ordering and preferences

Room no.	Capacity (student no.)	Preference score	Order
1	5	4	1
2	10	4	2
3	15	4	3
4	25	4	4
5	70	2	5
6	75	2	6
7	75	2	7
8	75	2	8
9	75	2	9
10	75	2	10
11	75	2	11
12	120	1	12
13	120	1	13
14	140	1	14
15	150	1	15
16	170	1	16

The results on the percentage of timeslots and rooms usage of the proposed algorithm are shown in Figure 4 and Figure 5 for Faculty of Computer Science and Information System, University of Technology Malaysia while Figure 6 and

Figure 7 show the percentage of timeslots and rooms usage for Faculty of Science at Ibb University – Yemen.

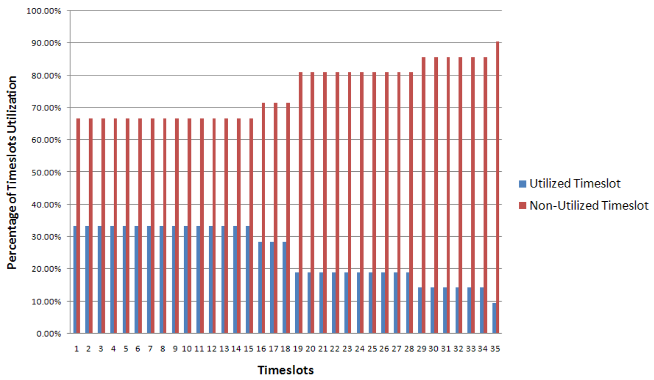

Fig. 4. Percentage of Timeslots utilization

As we can see from Figure 4, the first bar of each timeslot illustrates the utilized timeslots while the second bar illustrates the non-utilize timeslots. From the above figure, all the early timeslots (higher preference value) are being chosen first to be allocating for the university lectures. Overall a 24.90% of timeslots are being utilized while a 75.10% of timeslots are still available for future usage if there is any increment of lectures.

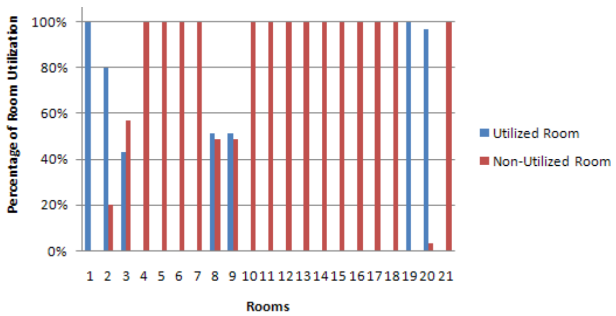

Fig. 5. Percentage of rooms utilization

Figure 5 shows that certain rooms are not being utilized. Referred to Table IV for room 9, we can see that from room 9 till room 18, the capacities for the rooms are 60. As shown from above figure, all the lectures allocation goes to room 9 since it is the earliest and nearest room with the higher preference values. Thus, the rooms behind such as room 10 till 18 will not be chosen unless room 9 is fully utilized. This can save the resources for future usage.

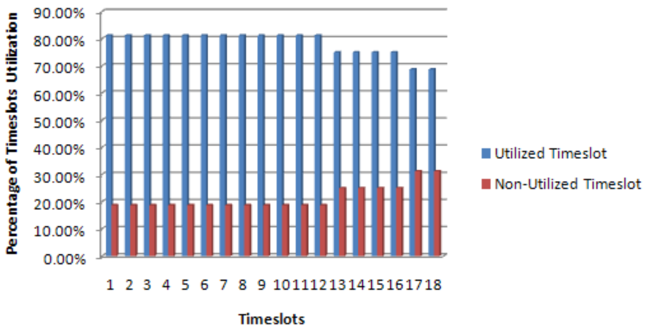

Fig. 6. Percentage of Timeslots utilization

Based on the Figure 6, we can see that the overall timeslots usages are 78.47% while the overall of non-utilized timeslots are 21.53%.

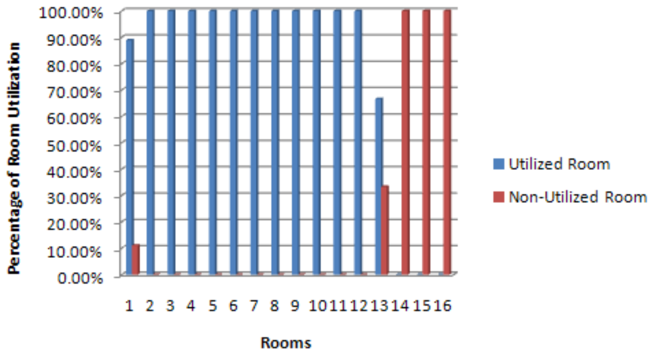

Fig. 7. Percentage of rooms utilization

Figure 7 shows that rooms 2 to 12 are fully utilized. This is mainly because of different total number of students to attend the lectures while the last three timeslots are not utilized at all. Basically, this is because there is no total number of students that is larger than the room capacity of room 14 to 16.

VIII. CONCLUSION

This paper is investigating the constraint-based reasoning approaches in solving the UCTP. Overall, the experiments show that there are still a lot of enhancement can be made towards solving such complex problem.

Future work will look into possibility of producing a better solution utilizing the constraints-based reasoning approaches with other possible techniques. A hybrid technique might be considered. The proposed algorithm is capable in finding feasible and near-optimal solution.

ACKNOWLEDGMENT

Malaysian Ministry of Science, Technology, and Innovation (MOSTI) and University Technology of Malaysia Research Management Centre (RMC) for their support (research grant number: Vote 79240).

REFERENCES

[1] W. Legierski, "Constraint-based Techniques for the University Course Timetabling Problem," in *CPDC 2005*, pp. 59-63.
[2] L. Tony, M. Eric, and S. Frederic, "Hybridization Strategies for Local Search and Constraint Propagation," *Proceeding of the 4th Workshop on Cooperative Solvers in Constraint Programming*, September 2004.
[3] W. Legierski, "Constraint-based reasoning for timetabling," *Artificial Intelligence Method: AI-METH*, Gliwice, Poland, 2002.
[4] E. K. Burke, D.G. Elliman, and R. Weare, "Automated Scheduling of University Exams," *Proceedings of the IEEE Colloquium on Resource Scheduling for Large Scale Planning System*, June 1993.
[5] A. Schaerf, "A Survey of Automated Tiemtabling," *Artificial Intelligence Review*, No. 13, 1999, pp. 87-127.
[6] E. K. Burke, J. Kingston, K. Jackson, and R. Weare, "Automated University Timetabling: The State of the Art," *The Computer Journal 40 (9)*, 1997, pp. 565-571.
[7] A. Calaor, and E. Hermosilla, Y. Augusta, and JR, Bobby O. Corpus., "Parallel Hybrid Adventures with Simulated Annealing and Genetic Algorithms," *Proceedings of the International Symposium on Parallel Architectures, Algorithms and Networks*, 2002.
[8] S. C. Chu, and , H. L. Fang, "Genetic Algorithms vs. Tabu Search in Timetable Scheduling," *Knowledge-Based Intelligent Information Engineering Systems*, Third International Conference, Adelaide, SA, Australia, 1999, pp. 492-495.

[9] B. Sigl, M. Golub, and V. Mornar, "Solving Timetable Scheduling Problem Using Genetic Algorithms," *25th International Conference Information Technology Interfaces*, Cavtat, Croatia, June 16-19, 2003.

[10] M. Omar el., R. N. Ainon, and R. Zainuddin, "Using a Genetic Algorithm Optimizer Tool to Generate Good Quality Timetables," *Proceedings of the 10th IEEE International Conference, Electronics, Circuits and Systems*, Vol.3, 14-17 Dec. 2003, pp. 1300- 1303.

[11] S. Deris, S. Omatu, H. Ohta, and P. Saad, "Incorporating Constraint Propagation in Genetic Algorithm for University Timetable Planning," *Engineering Applications of Artificial Intelligence 12*, 1999, pp. 231-253.

[12] S. Deris, S. Omatu, and H. Ohta, "Timetable Planning using the Constraint-based Reasoning," *Computer & Operations Research*, 27, 2000, pp. 819-840.

[13] Z. Lixi. and L. SimKim., "Constructing university timetable using constraint satisfaction programming approach," *Proceedings of the 2005 International Conference on Computational Intelligence for Modeling, Control and Automation, and International Conference on Intelligent Agents, Web Technologies and Internet Commerce* , 2005.