Implementation of Constraint Programming and Simulated Annealing for Examination Timetabling Problem

Tan Li June, Joe H. Obit, Yu-Beng Leau and Jetol Bolongkikit

Knowledge Technology Research Unit, Universiti Malaysia Sabah,
Labuan International Campus, Malaysia
june_tan93@hotmail.com, joehenryobit@gmail.com, lybeng@ums.edu.my
jetol@ums.edu.my

Abstract. Examination timetabling problems is the allocation of exams into feasible slots and rooms subject to a set of constraints. Constraints can be categorized into hard and soft constraints where hard constraints must be satisfied while soft constraints are not necessarily to satisfy but be minimized as much as possible in order to produce a good solution. Generally, UMSLIC produces exam timetable without considering soft constraints. Therefore, this paper proposes the application of two algorithms which are Constraint Programming and Simulated Annealing to produce a better solution. Constraint Programming is used to generate feasible solution while Simulated Annealing is applied to improve the quality of solution. Experiments have been conducted with two datasets and the results show that the proposed algorithm managed to improve the solution regardless the different problem instances.

Keywords: Examination timetabling, Constraint Programming, Simulated Annealing.

1 Introduction

Examination timetabling problem is one of the most concerns under domain of academic institution [17]. The problem is to assign events into timeslots and rooms by satisfying a set of constraints. Constraints are categorized as hard constraints and soft constraints. Hard constraints are used to determine the feasibility of the timetable. Thus, all hard constraints must be satisfied in any circumstances. For example, no student can sit for more than one exam simultaneously. While soft constraints are used to determine the quality of the timetable, however, soft constraints are not necessarily to be all satisfied but should be minimized as much as possible in order to produce a better quality timetable. For example, minimize the number of students with consecutive exams [22]. Examination timetabling problems have been widely studied with different approaches [3, 5, 9, 15, 17]. This research developed a model based on the datasets from Universiti Malaysia Sabah Labuan International Campus (UMSLIC) with a set of hard constraints and soft constraints. Technically, this research involves

R. Alfred et al. (eds.), *Computational Science and Technology*, Lecture Notes in Electrical Engineering 481, https://doi.org/10.1007/978-981-13-2622-6_18

two phases (1) Initialization phase: to generate feasible timetable by using constraint programming (2) Improvement phase: to enhance the quality of timetable with Simulated Annealing. In initialization phase, all hard constraints must be satisfied to generate feasible timetable as an initial solution. After generating the initial solution, next phase is to apply simulated annealing to improve it by reducing the soft constraints as much as possible.

The paper is organized as follows. After the introduction of the research in Section 1, Section 2 describes previous work regarding to the same problem domain. Section 3 provides explanation on the problem and the set of constraints of the problem are presented in mathematical formulas in Section 4. Section 5 explains the implementation of algorithms on the problem and the experimental results are discussed in Section 6. Section 7 concludes this research.

2 Related Works

Timetabling problems have been widely studied since 1960's with the implementation of different approaches [13, 19].The complexity of the problem make it more difficulty to produce an optimal solution or near to optimality. It is known that every institution has different policies and requirements therefore, not every approach applied in the literature can perform well in other problems event in the same problem domain dataset. Currently, UMSLIC uses CELCAT system to generate examination timetable. The timetable is feasible however, the system does not take any soft constraints into consideration which causes the low quality of timetable. For example, many students with consecutive exams in a day which causes students have limited time to do revision. As stated previously, this research involves two phases. The first phase implements constraint programming to produce feasible timetable and it will be further improved with the application of simulated annealing in the second phase. In [8], the course timetabling problem of UMSLIC had been studied and constraint programming algorithm was developed to solve all the hard constraints. The result shows the constraint programming is able to produce feasible timetable within a short period of time with several times of experiments.

Meanwhile, [20] studied on school timetabling problem by integrating Constraint Programing with operations research produced results are close to pre-defined optimal values. Overall, the solutions were produced in an acceptable duration. In [7], they implemented four different cooling strategies of Simulated Annealing to improve the quality of course timetable for every semester. They are linear cooling, exponential cooling, linear multiplicative cooling and geometric cooling. Among these four strategies, geometric cooling performed better than the rest which showed the best improvement from initial solution.

Research [19] proposed the combination of constraint programming and Simulated Annealing to solve examination timetabling problem of HoChiMinh City (HCMC) University of Technology. This research used Kempe chain to determine starting temperature which could improve the performance of Simulated Annealing in terms of its efficiency.

Besides, [12] implemented four hybrid heuristics to generate feasible solution for University course timetabling problem which are Sequential Heuristics (Largest Degree and Saturation Degree), Local Search, Tabu Search and Great Deluge. The results showed that all four hybrid heuristics could generate feasible solution but none of them perform outstandingly in terms of quality of solution such as this heuristic cannot scheduling students to have less than two consecutive course in a day. However, result from [12] shows that Sequential Heuristics could only produce feasible solution for small instances of the Socha et al [18] dataset.

In [14], Non-linear Great Deluge (NLGD) was proposed to compare its performance with the conventional Great Deluge (GD). NLGD algorithm modifies the conventional GD in the way of changing in water level with non-linear decay rate. The modification shows that NLGD outperforms over conventional GD and other algorithms in terms of the scheduling computational time and the improvement of the solutions. Research from [11] proposed the Evolutionary Non-Linear Great Deluge (ENGD) for course timetabling problem. [11] extends NLGD with effective operators and three neighborhood moves for solving the instances of Socha et al [18]. The results showed that ENGD perform effectively, obtaining best solution with zero penalties in small instances.

3 Problem Background

Examination timetabling problem is basically formed by four sets of parameter: exams, rooms, timeslots and constraints. This research aims to study examination timetabling problem of UMSLIC and schedule exams into room and timeslots subject to a set of constraints. Constraints are obtained from the Academic Service Division (BPA) of UMSLIC through several times of interview. Constraints are categorized into hard and soft constraints and shown as below:

Table 1. Hard constraints and soft constraints

Hard Constraints	Soft Constraints
HC_1: All exams must be assigned into available timeslots.	SC_1: Maximize the room utilization.
HC_2: No student can attend more than one exams simultaneously.	SC_2: Minimize the number of students with consecutive exams.
HC_3: The room capacity must be equal or greater than the total number of students taking the particular exams.	SC_3: Prioritize the exams with greater size in first two weeks.

UMSLIC uses CELCAT system to generate examination timetable which do not consider the soft constraints of the problem, hence, the timetable is feasible but low quality. Therefore, this research applies simulated annealing to further improve the solution quality by reducing the soft constraints violations as much as possible. In terms of resources, the duration of whole examination is three weeks, with 11 slots per week. Therefore, 33 slots in total and six rooms are available to accommodate the exams. However, some exams have large student enrollment which cannot be accommodated into the largest examination hall of UMSLIC. Therefore, it is necessary to assign those exams into multiple rooms. This research carried out experiments with two different semester datasets which are semester 2 session 2014/2015 and semester 1 session 2015/2016. Both datasets have different total number of students and exams. Table 2 summarizes the attributes of both datasets respectively.

Table 2. Summary of datasets

	Semester 2 session 2014/2015	**Semester 1 session 2015/2016**
Students	2248	2371
Exams	112	125
Timeslots	33	33
Rooms	6	6

4 Formulation Model

In order to achieve the objective of this research, formulation model has been developed to evaluate the feasibility and quality of the timetable. The formal model of examination is presented as below:

- $E = E_1, \ldots, E_n$ *where n is the total number of examinations*
- $T = T_1, \ldots, T_t$ *where t is the total number of timeslots*
- $R = R_1, \ldots, R_r$ *where r is the total number of rooms*
- $C = C_1, \ldots, C_c$ *where c is the total number of room combinations*
- $S = S_{E1}, \ldots, S_{En}$ *is the list of total number of students taking the exam E where n is the total number of examinations*

In this research, the feasibility and quality of the solution is evaluated based on the constraints violations. There are three hard constraints and three soft constraints in this research and each of them are assigned with different penalty cost. Table 3 shows the penalty cost of each constraint. For example, if a student is scheduled to attend more than one exam at the same time, the solution is violated with second hard constraints. Hence, the penalty cost of the solution is 100,000.

Table 3. Penalty cost of constraints violations

Weight	Penalty	Description
λ_1	100,000	Hard constraints, HC_1
λ_2	100,000	Hard constraints, HC_2
λ_3	100,000	Hard constraints, HC_3
λ_4	1	Soft constraints, SC_1
λ_5	1	Soft constraints, SC_2
λ_6	1	Soft constraints, SC_3

The penalty cost of the solution, F is presented in equation (1)

$$F = SC_1 + SC_2 + SC_3 \tag{1}$$

subject to the total of HC_1, HC_2, HC_3 must be zero, as stated previously, all hard constraints must be satisfied in order to produce a feasible solution. HC_1 is to ensure all exams are assigned and formulated as equation (2)

$$HC_1 = \lambda_1 \sum_{i=0}^{n} E_i \tag{2}$$

where n is the total number of exams, λ_1 is HC_1's weight, E_i is the number of exams that cannot be assigned. HC_2 is to prevent clashes happen in timetable and formulated as equation (3)

$$HC_2 = \lambda_2 \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} EE_{ij} \tag{3}$$

where n is the total number of exams, λ_2 is HC_2's weight, EE_{ij} is when exam i and j clash with each other. HC_3 is to assign exams into rooms based on the size of the exams and formulated as equation (4)

$$HC_3 = \lambda_3 \sum_{i=1}^{n} \sum_{j=1}^{r} ER_{ij} \tag{4}$$

where n is the total number of exams, r is the total number of rooms, λ_3 is HC_3's weight, ER_{ij} is the number when the size of exam i is larger than capacity of room j. There are three soft constraints in this problem, SC_1 is to reduce the extra space of the room and formulated as equation (5)

$$SC_1 = \lambda_4 \sum_{i=1}^{n} \sum_{j=1}^{r} ER_{ij} \tag{5}$$

where n is the total number of exams, r is the room size of the exam, λ_4 is SC_1's weight, ER_{ij} is the extra room space of the assignment of exam i into room j.
SC_2 is students should not sit for more than one subject a day and formulated as equation (6)

$$SC_2 = \lambda_5 \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} E_{ij} \tag{6}$$

where n is the total number of exams, λ_4 is SC_2's weight, E_{ij} is the time interval between exam i and exam j. SC_3 is to assign large exam into first two weeks with the evaluation of equation (8), in order to get large exam, equation (7) will be used to identify which exam is considered as a large exam

$$average_s = \frac{S}{n} \tag{7}$$

$$SC_3 = \lambda_6 \sum_{i=\frac{T}{2}}^{t} \sum_{j}^{n} TE_{ij} \tag{8}$$

where S is the total size of all exams, n is the total number of exams, t is the total number of timeslots, λ_5 is SC_3's weight, TE_{ij} is the number when exam i has larger value than $average_s$ and it is assigned later than first two weeks.

5 Implementation

The research work proposed in this paper approaches examination timetabling problem in two stages. For the first stage, Constraint Programming (CP) is applied to generate feasible solution by solving all the hard constraints and it is suitable for timetabling problem as it performs effectively when constraints of the problem are presented into numeric form [6]. For example, in [8], constraints are presented in binary matrices such as course conflict matrix to identify the conflict between courses. Research [20] implemented CP to solve the school timetabling problems which produce very good result.

In this research, all the exams are categorized into 4 parts: main exams, Promotion of Knowledge and Language (PPIB) exams, language exams and co-curriculum exams. Categorizing exams into different types is due to the reason that some exams have to be assigned into specific timeslot. For example, co-curriculum exams have to be on Saturday. Besides, a list of room combinations (RC) is also created for those exams have too many students which cannot be accommodated into the largest examination hall of UMSLIC. Therefore, room sharing is necessary for these exams.

At the beginning of the algorithm, a pool of unscheduled exams (UE) is generated to store all exams. The sequence of exams assignment is co-curriculum exams, language exams and others. CP will firstly select an exam according to the sequence of exams from UE and a timeslot at random. If the timeslot is feasible for the exam, it will select a room from RC at random. If the room is available and large enough to fit the exam, the exam will be assigned into that feasible slot and room. Exam will then moves from UE to pool of scheduled exams (SE) which indicates that exam is assigned successfully. If no room is available in that particular slot, the algorithm will select another slot until the exam is assigned. The whole step will iterate until UE is empty. However, it could happen when some exams are not able to fit into any slot due to limited room choice or clashes and those unassigned exams will move to pool of fail scheduled exams (FE) in order to empty UE and identify which exam is fail to assigned. Meanwhile, if UE is empty and there are still some exams in FE, neighbourhood search will take place in this situation to disturb the solution. An exam from SE, a slot and a room will be selected randomly to perform swapping. If the slot and room is feasible, the exam will move that slot and room. Hence, the solution is changed and proceeded with the scheduling of exams from FE. The search will stop until FE is empty.

In the second stage, Simulated Annealing (SA) is applied once the feasible solution is found to improve the timetable quality. SA algorithm is derived from the idea of cooling process in a physical system to study on the movement of the particles. The process of SA in timetabling problem can be represented with the elements of SA [4]. The physical system refers to the pool of solution and it is made up by smaller particles which refer to each solution. Meanwhile, the penalty cost of the timetable is represented by the system energy. In this research, the penalty cost of initial solution and initial temperature are defined. In cooling process of SA, the searching area is getting smaller as the temperature drops and the movement of particles becomes less active.
The result of [1] shows that SA performs outstandingly with the implementation of adaptive cooling and reheating scheme in SA for solving course timetabling problem of Syracuse University. In this research, SA with geometric cooling scheme is applied in second stage to improve the solution created by CP. The pseudocode of SA is shown in Fig.1:

Initial solution, $S_{initial}$ from CP
Initial temperature, $T_{initial} = 100°C$
Set $S_{current} = S_{initial}$
Set $T_{current} = T_{initial}$
while temperature>0.1 and iteration <100000
    S_{new} = NeighborhoodSearch($S_{current}$)
    $\delta = F(S_{new}) - F(S_{current})$
    if($\delta < 0$ or) or (exponential($-\delta/T_{current}$) < rand[0,1])
        $S_{current} = S_{new}$
    endif
    $T_{current}$ = CoolingScheme($T_{current}$)
endwhile

Fig. 1 Pseudocode of SA

In this phase, the initial temperature of SA in the experiments is set at $100°C$. After CP created a solution, it will be used as an initial solution ($S_{current}$). As the temperature decreases, the algorithm will search for a new solution (S_{new}). If the penalty cost of S_{new} is lower than $S_{current}$, $S_{current}$ will be updated as S_{new}. In order to ensure SA performs effectively, different numbers of iterations were tested: 1000, 10000 and 100000. 100000 iterations performs best among them. Therefore, the process will iterate until it reaches 100000 iterations or the temperature drops to $0°C$. The geometric cooling scheme is presented with equation (9):

$$t = \alpha t \tag{9}$$

where t is the temperature, α is the reduction parameter for geometric cooling, this research set it at 0.9999 according to [4].

When the temperature drops, the search space will decrease which could lead to the solution stuck in local optima [2]. Therefore, there is an acceptance criteria [1] known as Boltzmann probability, P with equation (10) which will accept worse solution with a certain probability.

$$P = e^{-\delta/t} \tag{10}$$

where δ is the difference of the penalty cost between current solution and new solution.

6 Experimental Results

This research is carried out with two different semester datasets of real world from UMSLIC as described in Table 2. Both datasets have different number of students and exams. Therefore, the research produced different result although they are solved by same algorithm.

The experiments in the initialization phase of this research which is carried out by using CP to solve all the hard constraints to produce feasible solutions. CP produced a feasible solution with less than one second. On the other side, UMSLIC system, CELCAT requires more than one week to produce a feasible solution. Therefore, in terms of time taken to produce a feasible solution, CP performs more effective than CELCAT. By taking soft constraints into consideration, the solution will be further improved with the application of SA. For example, shifts exams with smaller size into smaller room without causing any violation of hard constraints to reduce extra room space. With the neighbourhood movement, and the acceptance of the solution based on the theory of SA, this research will produce a lower penalty cost of solution. This research runs 50 times of experiment to obtain the average result as shown in Table 4.

Table 4. Experimental Results

	Semester 2 session 2014/2015	**Semester 1 session 2015/2016**
Average Cost	6848.0	11695.6
Highest Cost	7125	11888
Lowest Cost	6357	11462
Improve (%)	21.30%	16.12%

The results in table 4 show the first dataset (semester 2 session 2014/2015) has lower average penalty cost than the second dataset (semester 1 session 2015/2016). It means the violation of soft constraints in second dataset is higher than first dataset. The different improvement rate of two datasets shows SA performed less effective in second dataset during improvement phase. This indicates the size of the dataset affects the performance of the algorithm.

However, the results show that SA is able to improve the solution from CP for both datasets. For example, the extra room capacity is calculated into penalty cost. This can be explained by when the temperature decrease, SA will only accept the solution which has lower penalty cost. However, when there is no improvement, the worse solution can be accepted under certain probability as shown in equation (10). Meanwhile, in terms of performance, the solution in first dataset is much better because there are different number of students and exams in each semester. In second dataset, more students and exams need to be scheduled and thus, increase the difficulty to schedule. This can be summarized that SA produce good solution for this examination timetabling domain. However, it does not mean SA can produce good result in other domain as stated in [21].

7 Conclusion

This research had proposed the development of Constraint Programming (CP) and Simulated Annealing (SA) to produce a better quality examination timetable. A set of formal mathematical models are developed to obtain the violations of constraints in this research. CP is developed to solve all the hard constraints without considering the violation of soft constraints. In improvement stage, SA is applied to improve the quality of solutions. Results show that SA managed to search for the better quality solution in both instances of UMSLIC.

For future research, Non-Linear Great Deluge (NLGD) can be potentially implemented in this research according to the result of [10] for course timetabling problem. NLGD provides new method to control the speed and the shape of water level decay rate which could produce perform effectively in same problem domain.

References

1. Abdullah, S.: Heuristic Approaches For University Timetabling Problems. The School of Computer Science and Information Technology (2006).
2. Abramson, D.: A Very High Speed Architecture for Simulated Annealing. 27-36 (1992)
3. Abramson, D., Krishnamoorthy, M., & Dang, H.: Simulated Annealing Cooling Schedules for the School Timetabling problem. 1-17 (1997).
4. Aycan, E., & Ayav, T.: Solving the Course Scheduling Problem Using Simulated Annealing. (n.d).
5. Cupic, M., Golub, M., & Jakobovic, D.: Exam Timetabling Using Genetic Algorithm. 357-362 (2009).
6. Groves, G., & Wijck, W.: Constraint Programming and University Timetabling. 1-12 (n.d).
7. Kuan, Y. J., Joe, H. O., & Rayner, A.: Comparison of Simulated Annealing and Great Deluge Algorithms for University Course Timetabling Problems (UCTP). American Scientific Publishers, 4, 400-407 (2015).
8. Kuan, Y. J., Joe, H. O., & Rayner, A.: A Constraint Programming Approach to Solve University Course Timetabling Problem (UCTP). American Scientific Publishers, 400-407 (2016).
9. Kuan, Y. J., Joe, H. O., & Rayner, A.: The study of Genetic Algorithm approach for university course timetabling problem. International Conference on Computational Science and Technology, 454-463 (2017).

10. Landa-Silva, D., & Obit, J. H.: Great Deluge with Non-linear Decay Rate for Solving Course Timetabling Problems. IEEE, 11-18 (2008).
11. Landa-Silva, D., & Obit, J. H.: Evolutionary Non-linear Great Deluge for University Course Timetabling. Springer-Verlag Berlin Heidelberg, 269-276 (2009).
12. Landa-Silva, D., & Obit, J. H.: Comparing Hybrid Constructive Heuristics for University Course Timetabling. 222-225 (2011).
13. Malik, A. M., Ayob, M., & Hamdan, A. R.: Iterated Two-stage Multi-neighbourhood Tabu Search Approach for Examination Timetabling Problem. IEEE, 141-148 (2009).
14. Obit, J. H., & Landa-Silva, D.: Computational Study of Non-linear Great Deluge for University Course Timetabling. Springer-Verlag Berlin Heidelberg, 309-328 (2010).
15. Obit, J. H., Landa-Silva, D., & Ouelhadj, D.: Non-Linear Great Deluge with Learning Mechanism for the Course Timetabling Problem. Springer-Verlag Berlin Heidelberg, 1-10 (2009).
16. Ozcan, E., Misir, M., Ochoa, G., & Burke, E. K.: A Reinforcement Learning - Great-Deluge Hyper-Heuristic for Examination Timetabling . 1-25 (n.d.).
17. Rankhambe, M., & Mrs.S.Kavita.: Optimization of Examination Timetable Using Harmony Search Hyper-Heuristics (HSHH). Internation Journal of Computer Science and Information Technologies, 5, 6719-6723 (2014).
18. Socha, K., Knowles, J., & Samples, M. (2002). A MAX-MIN Ant System for the University Course Timetabling Problem. Springer-Verlag London, 1-13 (2002).
19. Tuan-Anh Duong, Kim-Hoa Lam.: Combining Constraint Programming and Simulated Annealing on University Exam Timetabling. 205-210 (n.d.).
20. Valouxis, C., & Housos, E.: Constraint programming approach for school timetabling. Elsevier Science Ltd., 1555-1572. (2003).
21. Wolpert, D. M.: No free lunch theorems for optimization. IEEE transactions on evolutionary computation, 67-82 (1997).
22. Wong, T., Cote, P., & Gely, P.: Final Exam Timetabling: A Practical Approach. IEEE, 726-731 (2002).