ACADEMIC World Academic Union

ISSN 1746-7659, England, UK
Journal of Information and Computing Science
Vol. 10, No. 3, 2015, *pp*. 220-227

Solving the Examination Timetabling Problem Using a Two-Phase Heuristic: The case of Sokoine University of Agriculture

Egbert Mujuni[#1], Allen Mushi[#2]
[#1]Department of Mathematics, University of Dar es salaam
[1]emujuni@udsm.ac.tz
[#2]Department of Mathematics, University of Dar es salaam
[2]allen.mushi@gmail.com

(*Received February 06, 2015, accepted June 07, 2015*)

Abstract. Examination timetabling is an important operational problem in any academic institution. The problem involves assigning examinations and candidates to time periods and examination rooms while satisfying a set of specific constraints. An increased number of student enrolments, a wider variety of courses, and the growing flexibility of students' curricula have contributed to the growing challenge in preparing examination timetables. Since examination timetabling problems differ from one institution to another, in this paper we develop and investigate the impact of a two-phase heuristic that combines Graph-Colouring and Simulated Annealing at Sokoine University of Agriculture (SUA) in Tanzania. Computational results are presented which shows great improvement over the previous work on the same problem.

Keywords: Heuristics, Timetabling, Optimization, Graph Colouring, Simulated Annealing

1. Introduction

Examinations timetabling problem involves the assignment of examinations and candidates to time period and rooms while satisfying a set of specific constraints. This paper extends the work of Selemani et al. [14] in which a graph colouring based algorithm was designed and implemented to solve the examination timetabling problem at Sokoine University of Agriculture (SUA) using data for Semester I and II in the year 2011. The algorithm produced a collision-free examinations timetable. The results obtained showed that examinations for Semester I could be scheduled in 13 timeslots instead of the planned 30 slots. Thus, 17 timeslots could be saved. Similarly, 11 timeslots could be saved in Semester II. Unfortunately, the work of Selemani et al did not consider the importance of having gaps between examinations per each student. The aim of this work is to develop and implement an algorithm that produces a collision-free examination timetable in which large examinations are scheduled as early as possible and each candidate examinations are spread as much as possible within the planning horizon. We first give an overview of the examinations timetabling problem.

Timetabling is one of the most important administrative activities that take place at least once a year in all academic institutions. Timetabling can be classified into three groups, namely: School, university courses and university examinations timetabling. School timetabling is about scheduling of school classes in fixed rooms. University timetabling is concerned with determining which course should be assigned to which lecturer on which day and timeslot while satisfying a number of constraints. The examination timetabling problem consists of assigning examinations to periods and classrooms in such a way that all examinations are scheduled within given timeslots and a number of specific constraints are satisfied. Timetabling is hard, complex and time consuming task and is classified as NP-Hard [3, 9].

Formally, the examination timetabling problem can be defined as an assignment of a set of examinations $E = \{e_1, e_2, \dots, e_m\}$ into a limited number of ordered timeslots (time periods) $T = \{t_1, t_2, \dots, t_n\}$ and rooms of certain capacity in each timeslot $C = \{c_1, c_2, \dots, c_n\}$, subject to a set of constraints [11]. Constraints are usually divided into hard and soft. Hard constraints have to be satisfied in order to have a feasible timetable. Common hard constraints include: (i) exams which have common students cannot be scheduled into the same timeslot, and (ii) there must be enough seating capacity in the room for the number of students scheduled in it at any given timeslot. Soft constraints tend to be institution specific, and are given different priorities (weightings) in different institutions [11]. Soft constraints do not have to be satisfied but they are desired to be satisfied as much as possible.

Because of its importance, from both practical and theoretical point of view, the examination timetabling problem has attracted attention of researchers across the world from both Operations Research and Artificial Intelligence communities for many decades. The examination timetabling problem is known to be varying from one institution to another [9]. Unfortunately, the fact that it is NP-Hard implies that it is unlikely that the problem can be solved optimally in polynomial time. Thus, a large number of researchers have focused on finding heuristic algorithms to solve the problem. Below we present a brief survey of previous work on the problem.

A comprehensive survey can be found in [5] and [11]. Graph based heuristics were among the earliest approaches to be used for the timetabling problem. Welsh and Powell in [17] pointed out the connection between the timetabling and graph colouring problems. They presented a graph colouring heurist based on vertices ordering. Burke et al. in [2] presented graph colouring and room allocation algorithms for the university timetabling problem. A graph colouring algorithm is used to split the examination into non-conflicting clusters and the room allocation algorithm is used to place examinations into rooms. Recently, Sabar et al in [12] investigated a graph coloring constructive hyper-heuristic for solving examination timetabling problems. They utilized the hierarchical hybridizations of largest degree, saturation degree, largest colored degree and largest enrolment graph colouring heuristics to produce four ordered lists. For each list, the difficulty index of scheduling the first examination was calculated by considering its order in all lists to obtain a combined evaluation of its difficulty. The most difficult examination to schedule was scheduled first. To improve the effectiveness of timeslot selection, a roulette wheel selection mechanism was included in the algorithm to probabilistically select an appropriate timeslot for the chosen examination. They tested the proposed approach on the most widely used un-capacitated Carter benchmarks, and on the examination timetable dataset from the 2007 International Timetabling Competition. The graph coloring constructive hyper-heuristic produced good results and outperformed other approaches on some of the benchmark instances. Selemani et al. [14] designed and implemented a graph coloring based heuristic to solve the examinations timetabling problem at Sokoine University of Agriculture (SUA) in Tanzania. In average, the developed algorithm saved as much as 46.7% of timeslots in the dataset.

Simulated annealing, introduced by Kirkpatrick et al. [6], is a popular local search meta-heuristic used to solve optimization problems. The key feature of simulated annealing is that it provides a means to escape local optima by allowing hill-climbing moves (i.e., moves which worsen the objective function value) in hope of finding a global optimum. Simulated Annealing has been successfully applied to the examination timetabling problems (see, e.g., [3], [7], [11], [15]). A study by Mushi [10] investigated the use of simulated annealing in the course timetabling in the University of Dar es Salaam, Tanzania. The computational results indicated that Simulated Annealing and steepest descent combination performs better, faster and with much less effort compared to manually generated results.

In this paper, we present a two-phase hybrid heuristic algorithm which combines both the efficiencies of graph-colouring and Simulated Annealing algorithms. In the first phase we use a graph-colouring based heuristic to generate a feasible solution with minimum number of timeslots. This phase is implemented in such a way that those examinations with large number of students are scheduled first. The second phase uses simulated annealing heuristic to improve the quality of the solution by spreading the examinations evenly. A survey of Cowling et al [5] indicated that almost all students prefer to have at least a gap between exams. This constraint is usually called back-to-back constraint. The aim of the back-to-back constraint is to give the students enough revision time between examinations.

The remainder of the paper is organized as follows. We first present the Examinations Timetabling Problem at the Sokoine University of Agriculture (SUA), giving the main features of the problem. Then we give the general description of a two-phase heuristic. Lastly, we present a summary of results and conclusions.

2. Problem description

The examination session at SUA is fixed to 3 weeks, with 2 examination sessions per day. An examination week is made up of five days, from Monday to Friday. Thus, the total number of timeslots during the examination session is 30. Examinations are held in 59 rooms whose capacities range from 15 to 500. An examination with many students can be scheduled in more than one room. Similarly, a room can have more than one examination scheduled in it provided sufficient room space is available. Examinations timetable is currently manually generated by the central timetable office. Examination invigilators are normally scheduled by individual departments after the release of the main timetable. Thus, the central timetable office is involved with assignments of examinations to the timeslots and rooms while satisfying a given set of constraints. Important constraints are given below.

2.1 Hard Constraints

i. A student cannot do more than one examination at the same time.
ii. A room cannot be assigned more candidates than its capacity

2.2 Soft Constraints

i. Examinations with large number of candidates should be scheduled as early as possible.
ii. Spread examinations for each student as even as possible so as to have as many gaps as possible between examinations.
iii. As much as possible, students should not be scheduled to sit more than one examination in a day

3. The Algorithm

The algorithm constitutes two phases as previously explained. In the first phase we construct an initial feasible solution. This feasible solution is gradually improved in the second phase.

3.1 Phase I: Seek an Initial Solution

In this phase we employ a sequential graph colouring heuristic to construct an initial solution. The relationship between the examination timetabling and graph colouring problems is widely discussed in the literature (See [8], [12] and [13]). Basically, the examination timetabling problem is represented as a graph colouring problem where the vertices represent examinations, the edges represent conflicts between two examinations and colour of the vertices represent different timeslots in the timetable. In order to construct a conflict-free timetable using graph based algorithm, a list of examinations is generated according to the "difficulty" in scheduling each examination which depends on certain criteria. Some common graph based heuristics are *Largest Enrolment First, Largest Degree First, Saturation Degree First, and Largest Coloured Degree First* [11].

Largest Enrolment First heuristic is selected for this phase mainly because with this heuristic, examinations with many students are scheduled first. This gives more time for examiners to mark the large number of scripts associated with the examination. For an examination e, let ρ(e) be the number of students enrolled in e. A pseudocode for Largest Enrolment First implemented in this work is given in Figure 1.

Input: *A set of examinations*
$E = \{e_1, e_2, \dots, e_m\}$
1. $F \coloneqq E$
2. *Choose an examination* $e \in F$ *with* $\rho(e)$ *largest*
3. *Let* s *be the smallest timeslot such that* e *does not conflict with any examination scheduled in* s
4. *Schedule* e *into timeslot* s
5. *Set* $F \coloneqq F - \{e\}$

If $F \neq \emptyset$ *Go to Step 2.*

Figure 1: Pseudocode for the Largest Enrolment First heuristic

3.2 Phase II: Spread Conflicting Examinations

In the first phase we generated a conflict-free examination timetable in which each examination is scheduled in a room(s) big enough to accommodate it. In addition, examinations with big number of students are scheduled earlier. Therefore, the main objective of this phase is to distribute the examinations evenly within the planning horizon. However, since examinations with large examinations are scheduled earlier in Phase I, in distributing the examinations we need to ensure these large examinations are held at the beginning of the examination session. We use simulated annealing heuristic to minimize an objective function that measures the interval between conflicting examinations.

Before stating the mathematical model we define the following notation:

E : Set of examinations
S : Set of students
T : Set of timeslots

$x_{se} = 1$ if student s is doing exam e, and 0 otherwise.
$y_{et} = 1$ if exam e is scheduled at timeslot t, and 0 otherwise.
$z_{st} = 1$ if student s is doing exam that is scheduled at timeslot t, and 0 otherwise.
$D_{std} = 1$ if student s is scheduled for examinations in both timeslots t and $t + d$, and 0 otherwise.

Cartel et al [4] proposed proximity cost functions to assess the quality of a solution, when two examinations for the same student are placed too close together; these costs are non-increasing with the timeslot difference. A penalty cost λ_d is charged whenever one student has to write two examinations scheduled d units apart. Here we apply the following penalty costs as used in Cartel et al [4]: $\lambda_1 = 8$, $\lambda_2 = 4$, $\lambda_3 = 2$, $\lambda_4 = 1$ and λ_d=0 for $d \geq 5$. Using this notation, mathematical model of this phase becomes:

$$\text{Min } f = \sum_{s \in S} \sum_{t \in T} \sum_{d=1\ldots5} \lambda_d D_{std} \tag{1}$$
$$\text{S.t } \sum_{t \in T} y_{et} = 1 \quad \forall e \in E \tag{2}$$
$$\sum_{e \in E} x_{se} y_{et} \leq 1 \quad \forall t \in T, \forall s \in S \tag{3}$$
$$x_{se} + y_{et} - z_{st} \leq 1 \quad \forall s \in S, \forall e \in E, \forall t \in T \tag{4}$$
$$z_{st} + z_{st+d} - D_{std} \leq 1 \quad d = 1 \ldots 5, \forall s \in S, \forall t \in T \tag{5}$$
$$\sum_{e \in E} y_{et} \rho(e) \leq \text{TotalAvailableRooms}, \forall t \in T \tag{6}$$

The objective cost function (1) spreads conflicting exams as much as possible. Equation (2) ensures that each exam is scheduled in precisely one timeslot. Equation (3) ensures that no student may have two exams in the same timeslot. Equation (4) represents the relationship between x_{se}, y_{et} and z_{st}. Similarly, Equation (5) represent the relationship between z_{st}, z_{st+d} and D_{std}. Finally, Equation (6) ensures that there is enough room space for each timeslot.

The mathematical model is solved by a Simulated Annealing heuristic. Simulated Annealing algorithms are devised so as to avoid being trapped in poor local optima by accepting bad moves according to a probability function. The method imitates the physical annealing process in metallurgy; Starting from a randomly generated solution, a neighbouring solution is sampled and compared with the current one according to an appropriate cost function. If the cost function is improved, the sampled solution is accepted. In addition, a worse solution may be accepted with certain probability which depends on a parameter T. The parameter T come from the field of statistical thermodynamics, referred to as the temperature and is decreased or cooled during each run. The heuristic terminates when either the optimal solution is obtained or the initial temperature decreases to the lowest point (called freezing point). A pseudocode for simulated annealing as used in this work is given in Figure 2.

In order to use the Simulated Annealing heuristic an appropriate neighbourhood, a good initial temperature, an effective decreasing rate for the temperature and stopping criteria have to be specified. For a given feasible solution S_0, we take the neighbourhood $N(S_0)$ to be the set of all feasible solutions that can be obtained from S_0 by moving one exam from its current allocated time-slot to another. The most commonly reported successful cooling schedule is the Geometric function (Ref) and is applied in this work. This is defined as $f(x) = \alpha x$, where $0 < \alpha \leq 1$. After experimentation, the parameters of Simulated Annealing as applied in this work are given in Table 1.

```
Input: Initial Feasible Solution S_0
Initialize: T = T_0, Freeze
While (T > Freeze)
{
Get a candidate solution S ∈ N(S_0)
If Sis feasible
{
 Δ= f(S) − f(S_0)
If Δ< 0
    S_0 = S
Else { Generate a random x ∈ (0,1)
   If (x <  e^(−Δ/T))
     S_0 = S
   Else
     Reject S
    }
}
T = αT
}
```

Figure 2: Pseudocode for the Simulated Annealing heuristic

Table 1: Parameters for Simulated Annealing

Parameter	Value
Cooling rate (α)	0.98
Initial Temperature (T_0)	1000
Freezing Temperature ($Freeze$)	0.0005

4. Computational Results

The algorithm was tested on an examination timetabling problem previously solved by manual methods at SUA for semester 1 and 2 of the 2011/2012 academic year. The algorithm was implemented using Microsoft Visual C++ 2010 Express Edition. We ran the algorithm on a 2GHz machine with 1.87 GB RAM and Windows 7. Table 2 gives characteristics of data set.

Table 2: Characteristics of input data

Semester	Number of exams	Average number of students per exam	Number of Rooms
I	334	129	59
II	377	114	59

4.1 Results after Phase I

In this phase, the examinations for Semester I and Semester II were scheduled using 13 and 17 timeslots, respectively (see Tables 3). This clearly indicates that many timeslots were not used, and many students had examinations in consecutive timeslots. A good feature of this phase is that the algorithm scheduled large examinations early in the timetable.

4.2 Results after Phase II

The timetable produced by phase I is used as the initial solution for the Simulated Annealing phase. This phase is used to improve the quality of the timetable.

Table 4 gives the results after Simulated Annealing phase. As stated in the previous section, the cost function in (1) measures the spread of conflicting exams. The objective is to minimise the number of students having consecutive examinations. Unlike in Phase I, all timeslots are used in Phase II. The need to satisfy all student gaps requirements has therefore a trade-off by using more timeslots.

Table 3: Results for Phase I

Time slot	Semester I		Semester II	
	Number of Exams per Timeslot	**Number of Students per Timeslot**	**Number of Exams per Timeslot**	**Number of Students per Timeslot**
1	33	4416	28	4235
2	30	4272	27	4235
3	31	4411	35	4182
4	37	4435	36	4115
5	36	4253	34	4075
6	36	3815	41	4201
7	29	4012	37	3928
8	29	3561	39	3607
9	31	2606	38	3048
10	22	2345	34	2342
11	11	1048	17	990
12	8	1086	5	367
13	1	98	2	202
14	-	-	1	34
15	-	-	1	34
16	-	-	1	34
17	-	-	1	34

Table 4: Results for Phase II

Timeslot	Semester I		Semester II	
	Number of Exams per Timeslot	**Number of Students per Timeslot**	**Number of Exams per Timeslot**	**Number of Students per Timeslot**
1	7	2269	15	3040
2	11	2685	11	2723
3	2	936	6	1346
4	8	1048	7	1317
5	9	2008	6	1364
6	7	1298	10	1141
7	13	1405	10	914
8	4	453	16	1083
9	13	1127	17	1621
10	9	405	13	822
11	14	1671	9	759
12	13	1941	19	1265
13	8	475	11	615
14	16	1701	10	1501

15	3	457	15	1519
16	12	1391	10	904
17	8	998	9	755
18	15	1252	11	878
19	11	832	14	1277
20	7	1213	10	529
21	11	1593	10	1147
22	12	1175	15	1294
23	11	986	10	1094
24	10	788	8	805
25	13	1564	14	1356
26	9	778	10	759
27	8	896	7	887
28	6	829	6	437
29	18	1607	19	1430
30	14	1065	16	1833

Figure 3 represents the improvements in the cost function by iterations. The graph shows that there is a sharp drop in the objective function, for both data of semester I and II, within the first 100 iterations followed by a slow convergence. The cost function has an impact on evenly spreading all examinations. This is a normal behaviour for global heuristics techniques, where improvement in solution quality is expected to slow down when approaching convergence.

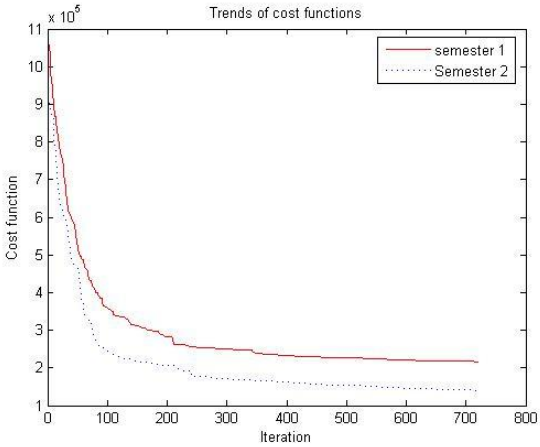

Figure 3: Trends of cost function

5. Concluding Remarks AND Further Research Directions

As previously stated, this paper extends previous work by Selemani et al [14] which developed an algorithm for examinations timetabling at SUA using graph colouring. In this paper a new soft constraint which considers gaps between examination sessions per student has been introduced. Furthermore, the algorithm used is a combination of graph colouring and simulated annealing algorithms in a two-phase approach.

The algorithm has been able to generate a better timetable that takes care of gaps between examinations while ensuring that all large examinations are scheduled first. We have been able to schedule a timetable within 13 slots for semester I and 17 slots for semester II in phase I. However, the timetable in this phase did not consider the gaps between examinations per student. Phase II managed to spread examinations in such a way that all students were scheduled with no consecutive examinations. This had a consequence of extending the timeslots used to cover the whole planning horizon (30 timeslots). Although we have used all timeslots, we have been able to generate a timetable that is more user-friendly to students and therefore of higher quality.

As suggestions for further research, it is worth investigating the use of other natural-inspired heuristics for this problem. This is a case study for SUA, but there are many other educational institutions in Tanzania which are using manual methods for timetabling and can be investigated. The paper did not consider optimization of room space; it only ensured that examinations are scheduled in rooms which can accommodate the examinations. Optimal room allocation strategy is another area of further research. Usually there are many soft constraints for a particular problem, including specific restrictions on timeslot windows for given examinations, special room preferences, minimal use of some slots especially late evenings and many others. Extension of this model to include more soft constraints will increase further the quality of the timetable.

6. References

[1] Aarts E., Korst J.H.M, Laarhoven P. and Van J. M., *A Quantitative Analysis of the Simulated Annealing Algorithm, A Case Study for the Travelling Salesman Problem*, Journal of Statistical Physics, Vol. 50, pages 187-206, 1988.

[2] Burke, E.K. and Elliman, D.G., *A University Timetabling System Based On Graph Colouring and Constraint Manipulation*, Journal of Research On Computing In Education, Vol. 27 (1), pages 1-18, 1994.

[3] Burke, E.K., Bykov, Y., Newall, J. and Petrovic, S., *A Time- Predefined Local Search Approach to Exam Timetabling Problems*, IIE Transactions, 36, pages 509-528, 2004.

[4] Carter M.W., Laporte G. and Lee S.Y., *Examination timetabling: Algorithmic strategies and applications*, Journal of Operational Research Society, Vol. 47(3), pages 373-383, 1996.

[5] Cowling P., Kendall G. and Hussin N. M., A *survey and case study of practical examination timetabling problems*, In Proceedings of the 4th International Conference on the Practice and Theory of Automated Timetabling PATAT02, Pages 258-261, 2002.

[6] Kirkpatrick S., Gelatt C.D. and Vecchi M.P., *Optimization by Simulated Annealing*, Science, 220, pages 671-680, 1983.

[7] Merlot, L.T.G., Boland, N., Hughes, B.D. and Stuckey, P.J., *A Hybrid Algorithm for the Examination Timetabling Problem*, In: Burke EK and De Causmaecker P (Eds). Practice and Theory of Automated Timetabling IV: Selected Papers from 4th International Conference. Lecture Notes in Computer Science, Vol. 2740, Springer-Verlag, Heidelberg, pages 207-231, 2003.

[8] Malkawi, M., Hassan M. A. And Hassan O.A., *A New Exam Scheduling Algorithm Using Graph Colouring*. The International Arab Journal of Information Technology, Vol. 5 (1): pp 80-86, 2008.

[9] Mushi, A. R., *Mathematical Programming Formulations for the Examinations Timetable Problem: the Case of University of Dar es Salaam*, African Journal of Science and Technology (Science and Engineering Series), Vol. 5 (2): pages 34-40, 2004.

[10] Mushi, A.R., *Two Phase Heuristic Algorithm for the University Course Timetabling Problem: the Case of University of Dar es Salaam*, Tanz. J. Sci. Vol. 37, pages 73-83, 2011.

[11] Qu R., Burke E. K., McCollum B., Merlot L. T. G. and Lee S. Y. *A survey of Search Methodologies and Automated System Development for Examination Timetabling*, Journal of Scheduling, Vol. 12(1), pages 55-89, 2009.

[12] Sabar N. R., Ayob M., Kendall G. and Qu R., *Roulette Wheel Graph Colouring for Solving Examination Timetabling Problems*, COCOA 2009, pages 463-470, 2009.

[13] Sabar N.R., Ayob M., G. Kendall G., Qu R., *A Graph Coloring Constructive Hyper-Heuristic for Examination Timetabling Problems*, Applied Intelligence, Vol. 37(1), pages 1-11, 2012.

[14] Selemani M.A., Mujuni E. and Mushi A., *An Examination Scheduling Algorithm Using Graph Colouring - The case of Sokoine University of Agriculture*, Int. Jour. Comp. Eng. Appl., Vol. II, Issue I/III pages 116-127, 2013

[15] Thompson, J.M. and Dowsland, K.A., *A Robust Simulated Annealing Based Examination Timetabling System*, Computers & Operations Research, Vol 25, pages 637-648, 1998.

[16] Vishwakarma, Kumar K., Dubey H.M., Pandit M, and Panigrahi B. K, *Simulated Annealing Approach for Solving Economic Load Dispatch Problems with valve point loading effects*, International Journal of Engineering, Science and Technology Vol. 4(4) , pages 60-72, 2013.

[17] Welsh, D.J.A and Powell, M.B., *An Upper Bound for the Chromatic Number of a Graph and Its Applications to Timetabling Problems*, Comp. Journal. Vol. 10(1), pages 85-86, 1967.