International Journal of Computer Engineering & Applications, Vol. II, Issue I/III

www.ijcea.com ISSN 2321-3469



# AN EXAMINATION SCHEDULING ALGORITHM USING GRAPH COLOURING – THE CASE OF SOKOINE UNIVERSITY OF AGRICULTURE

**Mohamed Abdallah Selemani [1], Egbert Mujuni [2], and Allen Mushi [3]**

[1]*Department of Mathematics, Sokoine University of Agriculture, Morogoro, Tanzania*

[2,3] *Department of Mathematics, University of Dar es Salaam, Tanzania*

**ABSTRACT:**

**This paper presents a graph coloring based algorithm for Examinations Timetabling Problem at Sokoine University of Agriculture (SUA) in Tanzania. A Recursive Largest First algorithm for graph coloring is applied to find timeslots. We present a summary of results which indicates good performance.**

**Keywords:** ***Examinations Timetabling, Combinatorial Optimization, Graph Coloring, Recursive Largest First*, Heuristic Algorithms**

## I. INTRODUCTION

The University Examination Timetabling Problem has attracted significant research interest over the years. Basically, this is the problem of finding a schedule of a set of examinations within a given period of time while satisfying constraints over resources such as examination space and conflicts between examinations and rooms. These constraints are normally divided into hard and soft, where hard constraints must be satisfied for a feasible timetable, while soft constraints are desired to be satisfied as much as possible. Despite of four decades of research, no polynomial algorithm is known for the optimal solution of this problem within reasonable time. This problem is therefore known to be NP-hard as recorded by Burke et al [1], and El-Mohamed et al [2]. Owing to the importance of this problem, from both practical and theoretical point of view, numerous studies have been conducted including Abdullah, Burke and Mushi [4, 5]. The main difference between various studies is the set of assumptions and constraints which differs between applications. Due to the complexity of the problem, most of these studies concentrate on the heuristic algorithms which try to find an approximate solution. Some of

these include Tabu Search [6], Simulated Annealing [2, 7] and Graph Colouring [8].

Graph based heuristics were among the earliest approaches to be used for the timetabling problem [9, 10]. They are used as constructive heuristics, constructing solutions by ordering exams that have not yet been scheduled, according to the perceived difficulty in scheduling that exam into a feasible timeslot. The difficulty of an exam can be represented in various ways such as the degree of conflict it has with other exams and the number of student enrolments.

The Graph Colouring Problem (GCP) and its relationship to timetabling is widely discussed in the literature [4, 8, 10, 11]. Burke [4] 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. Sabar [12] presented a simple graph based heuristic that employs a roulette wheel selection mechanism for solving examination timetabling problems. They arrange exams in descending order of the number of conflicts (degree) that they have with other exams. The difficulty of each exam to be scheduled is estimated based on the degree of exams in conflict. The degree determines the size of a segment in a roulette wheel, with a larger degree giving a larger segment. The roulette wheel selection mechanism selects an exam if the generated random number falls within the exam's segment. This overcomes the problem of repeatedly choosing and scheduling the same sequence of exams. They utilized the proposed Roulette Wheel Graph Colouring heuristic on the uncapacitated Carter's benchmark datasets. Results showed that this simple heuristic is capable of producing feasible solutions for all 13 instances in the benchmark set.

Despite the fact that literature on timetabling problem is always growing, the review of literature revealed that, there exist no study based on graph colouring approach for timetabling problem has been done before in Tanzanian situation and this prompted the study.

The remainder of the paper is organized as follows. We close this section by giving some basic definitions and notations on graph theory. Then we 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 the Graph Colouring algorithm and its adaptation for the timetabling problem in the case study. Lastly, we present a summary of results and conclusions.

Graph Theory Notation: For graph theoretic terminology not defined in this paper, we refer the reader to standard text books (see [13, 14]). In this paper we consider the connected simple graphs $G=(V,E)$. The set of neighbours of a vertex $v$ in a graph $G$ is denoted by $N_G(v)$ (or simply $N(v)$). The degree of a vertex $v$ of a graph $G$ denoted by $\deg(v)$, is the number of edges in $G$ incident to $v$ i.e., $\deg(v)=|N(v)|$. A (proper) colouring of a graph $G$ is an assignment of the elements of some set $C$ (called colours) to the vertices of $G$ one colour to each vertex, so that adjacent vertices are assigned different colours. In other words, we say that a colouring is a function $f:V\to C$ where $f(u)\neq f(v)$ for all edges $uv\in E$. If $|C|=k$ then $f$ is a $k$—colouring of $G$.

## II. EXAMINATIONS TIMETABLING AT SUA

At SUA, the examinations period is fixed to 3 weeks, with 2 examination sessions per day. An examination week is made up of five days, from Monday to Friday. This brings the total number of examination timeslots to 30 for the three weeks of examinations. Examinations are held in 59 rooms whose capacities range from 15 to 500. An examination with many students can be scheduled into more than one room. Similarly, a room can have more than examination scheduled in it if sufficient room space is available.

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. As stated earlier, these constraints are classified into hard and soft. The hard constraints must be satisfied in order to produce a feasible timetable, whilst violation of the soft constraints should be minimised. For some problems, it is not possible to find a solution which satisfies all soft and hard constraints. In such circumstances priority is given in satisfying all hard constraints and minimizes violation of soft constraints.

The main hard constraints encountered in this paper are:



1. No student can to sit two or more examinations simultaneously.
2. There must be enough seating capacity in the room for the number of students scheduled in it at any given time slot.

Some soft constraints for this problem are:

1. As much as possible, students should not be scheduled to sit more than one examination in a day,
2. Examinations with large number of candidates should be scheduled as early as possible.

## III. GRAPH COLOURING APPROACH

One of the most common models for examination timetabling problems is the graph colouring and is used as a basic model for this problem. Modelling examination timetabling problem in this way allows us to apply techniques and ideas successfully used for GCPs.

The basic structure of every examination timetabling problem contains a set of examinations, $C = \{c_1, ..., c_n\}$ together with constraints on which examinations clash and therefore cannot be scheduled together. Equating the set $C$ of examinations to set of vertices, $V = \{v_1, ..., v_n\}$ of a graph we can add an edge, $v_i v_j$ to the graph for every pair $c_i c_j$ of examinations if they clash.

The other important component of every examination timetabling problem is the timeslots into which all the examinations must be scheduled. Using graph representation, it is clear that any pair of examinations which share some students cannot be assigned to the same time slot. Therefore, examinations must be assigned to timeslots in such a way that this constraint holds. This is analogous to the well known problem of graph-colouring [9, 15] in which the vertices of a graph must be assigned a colour such that no two vertices sharing a common edge have the same colour. de Werra [16] gives a formulation of course timetabling as a graph colouring problem and also discusses the differences between course and examination timetabling with regard to this model. For examination timetabling, colours represent the different timeslots.

The related graph $k$ —colouring problem limits the number of colours to $k$ and requires only that a solution to the problem is found using $k$ or fewer colours. This is

equivalent to an examination timetabling problem in which the number of timeslots is fixed. Otherwise, the number of colours (timeslots) is a variable which is to be minimised. Both cases represent common Timetabling Problems (TTP). This model only takes account of the Boolean value of whether two examinations clash or not and treats all clashes equally.

A procedure that reduces the TTP into the graph colouring problem involves creating a conflict graph from the assembled input university examinations data, properly colouring the conflict graph, and transforming this colouring into a conflict-free exams timetable. From this conflict-free exam timetable one can then assign the exams to classrooms based on room capacity and availability. Next section discusses the Recursive Largest First algorithm for graph colouring problem as used in this paper.

**3.1 Recursive Largest First (RLF)**

The Recursive Largest First (RLF) algorithm of Leighton [17] is one of the widely used graph colouring algorithms. The algorithm colours vertices of one class at a time, adding vertices one at a time to the current class so as to reduce as much as possible the number of edges left in the uncoloured subgraph. The RLF algorithm is based on the following principle: The vertices with largest degree are coloured first so that at a given step, the subgraph induced by the uncoloured vertices is sparse. Hence the remaining graph becomes easier to colour.

RLF is selected for this work because of two main reasons; first, data shows that examinations with many conflicts are those with many students. Therefore, 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. Second, examination that is in conflict with a largest number of other examinations is normally considered to be more difficult to schedule and so should be handled first. A pseudocode for RLF in this work is given in Figure 1.

www.ijcea.com ISSN 2321-3469



***Input:*** *G=(V,E):* the undirected graph
Output: - *C:* the list of colored vertices
- *k:* the number of colors used.

1. *C:=∅; U:= ∅; V*:=V; k:=0*
2. *Choose a vertex x of maximum degree in the subgraph induced by V*. Increment k by 1 and proceed to Step 3.*
3. *Assign k to x. Move x from V* to C and all y∈V* that are adjacent to x from V* to U
   *If V* is empty, then proceed to 4. Otherwise check whether C=V. If so, stop with G colored with k colors. If not, then V*:=U, U:=∅ and return to Step 2*
4. *Choose a vertex x∈V* that has the maximum number of edges to vertices in U. Go to Step 3.*

**Figure 1: Pseudocode for RLF Algorithm**

The complexity of RLF is $O(n^3)$. One factor $O(n)$ is due to the determination of a vertex $x$ of maximal degree. Traversing the non-neighbours of $x$ in search for a vertex $y$ with a maximal number of common neighbours with $x$ may cost another $O(n^2)$ elementary operations.

Creating Exam Conflict Graph

Given a set of examinations to be scheduled, the conflict graph $G$ is constructed as follows. Suppose we have a set of $n$ examinations $C=\{c_1,..., c_n\}$ to be scheduled for a particular semester. Each examination $c_i$ will be represented by exactly one vertex $v_i$ in $G$. Therefore $G$ contains $n$ vertices, and $V(G)=\{v_1,..., v_n\}$. Finally, for every pair of vertices $v_iv_j (i \neq j)$, we add an edge connecting them if their corresponding examinations $c_i$, and $c_j$ have some common students.

An example of the conflict graph $G$ for an instance of the examination timetable problem that is involving 7 exams with 10 students is given in Figure 2.

### 3.2 Finding Timeslots

Once the conflict graph $G$ has been constructed from a given set of examinations, we apply the RLF algorithm to colour vertices of $G$. Extend the obtained vertex colouring of $G$ to examination scheduling as follows: If a vertex $c_i$ is assigned colour $k$ we then schedule the corresponding examination $e_i$ into slot $k$. Since every pairs of adjacent vertices are coloured differently

for proper vertex colouring, any two conflicting examinations are assigned into two different slots. Thus, we use the idea of a vertex colouring of the conflict graph to construct a conflict-free examination timetable.

Note that a pair of non-adjacent vertices may be coloured with either the same colour or different colours. This implies that a pair of non-conflicting examinations can be assigned either the same slot or different slots.

$S(c_1) = \{s_1, s_2\}$
$S(c_2) = \{s_1, s_2, s_3\}$
$S(c_3) = \{s_2, s_3, s_9\}$
$S(c_4) = \{s_3, s_4, s_{10}\}$
$S(c_5) = \{s_4, s_5, s_5\}$
$S(c_6) = \{s_1, s_2, s_6, s_7, s_8\}$
$S(c_7) = \{s_8, s_9, s_{10}\}$

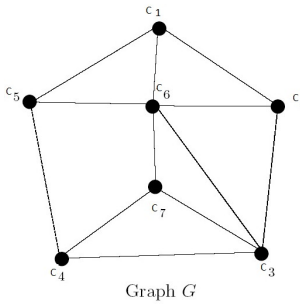



**Figure 2. Construction of a conflict graph $G$ from an instance of the examination timetabling problem. $S(c_i)$ denotes the set of students in the course.**

### 3.3 Finding Rooms for Each Examination

For each timeslot, we have a set of non-conflicting examinations that need to be assigned to a set of rooms. In allocating rooms we consider the followings:

No room can be assigned to examinations with more candidates than its capacity.

An examination with many students can be scheduled into more than one room.

A room can have more than one examinations scheduled in it if sufficient room space is available

To allocate a room to an examination, the set of rooms is searched to determine which room is best to allocate. First-fit, best-fit, and largest-fit are the most common strategies used to select a free room from the set of available rooms. The first-room strategy allocates the first room that is being enough. Searching starts at the beginning of the set of rooms and stop as soon as a room with capacity that is large enough has been found. With the best-fit strategy, we allocate the smallest room that is big enough. This strategy produces the smallest leftover space. The largest-fit strategy allocates the largest room. This strategy produces the largest leftover space, which may be more useful to other examinations than the smaller leftover from the best- room approach.

Given a total of $n$ rooms and timeslots $t$ , define:

$R_i$ =capacity of room $i$

$S_t$ =set of examinations in timeslot $t$

www.ijcea.com ISSN 2321-3469



$A(i,t,c)=$ number of students in an examination $c$ that have been assigned room $i$ in timeslot $t$.

The remaining capacity of room $i$ at timeslot $t$ is given by

$$R(i,t) = R_i - \sum_{c \in S_i} A(i,t,c)$$

For each timeslot $t$, we first set $R(i,t) = R_i$ for each room $i$, and sort the room in ascending order according to $R(i,t)$. We then sort the examinations in the slot in decreasing order according to the number of students in the examinations. The first examination in the list for each slot will be assigned to the best fit rooms. After each assignment, we update $A(i,t,c)$, $R(i,t)$ and sort the rooms according to $R(i,t)$ again. If the combined size of the examination in the timeslot is less than the capacity of all available rooms, we split the timeslot into two timeslots.

## 4. SUMMARY OF RESULTS

The algorithm was tested on an examination timetabling problem previously solved by manual methods at SUA for semester 1 and 2 of the 2010/2011 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. The size of the two-semester data is given in Table 1.

| Semester | Number of Exams | No. of Students | Number of Rooms |
|---|---|---|---|
| Semester I | 334 | 43,129 | 59 |
| Semester II | 377 | 43,077 | 59 |

**Table 1: Size of input data**

The following is the summary of performance of the graph colouring algorithm.

Semester I: The conflict graph $G$ has 334 vertices, 2104 edges, a maximum degree of 84. The colouring algorithm labelled the vertices of $G$ with 13 colours. The algorithm found the solution in 8.424 seconds.

Semester II: The conflict graph has 377 vertices, 2446 edges, a maximum degree of 76. The algorithm coloured the graph with 17 colours. The algorithm used 11.84 second to compute the solution.

The examination is assigned into examination timetable based on colour assigned. Thus, our results shows that examination timetable at SUA for semester I and Semester II (2010/11) could have been scheduled by using 13 and 17 timeslots, respectively. However, two compulsory examinations (MTH106 and SC101) have larger number of candidates. These courses had to be split into many rooms. For example, the course MTH106 with 2503 candidates required 11 rooms. So, putting such an examination course with others that have assigned the same colour could cause chaos in room allocation. Therefore, this examination was assigned its own timeslots and remaining exams were allocated into other slots. Since these examinations are taught in Semester II, all examinations in this semester were scheduled in 19 slots.

The algorithm scheduled large examinations early in the timetable. For examples, the largest courses MTH106 (with 2503 students) and SC101 (with 2293) where scheduled in timeslot 1 and 2, respectively.

The number of rooms allocated for a single examination has an implication on the manpower, such as invigilators, that are involved during the examination period. Thus, it is important to minimize this parameter. That is each examination should be assigned to a single room, unless the examination cannot fit in a single room. In this project we set four (4) as the maximum number of rooms per examination. However, since the combined capacity of the largest four rooms is 1198, this condition was relaxed for four examinations whose each enrolment is exceeding 1198. Table 2 shows results of room allocation distribution of examinations according to the number of rooms required.

| Semester | Number of rooms | Number of Examinations scheduled |
|---|---|---|
| I | 1 | 256 |
| | 2 | 68 |
| | 3 | 6 |
| | 4 | 3 |
| | >=5 | 1 |
| | Total | 334 |
| II | 1 | 330 |
| | 2 | 41 |
| | 3 | 3 |
| | 4 | 0 |
| | >=5 | 3 |
| | Total | 377 |

**Table 2: Examination distribution**

Table 2 indicates that about 76.6% and 87% of the examinations in Semester I and II, respectively, were scheduled in a single room. In addition, about 97% and 99.7% of examinations in semester I and II

respectively, were scheduled in a single or two rooms. All problems were solved within a few seconds which is tolerable in timetabling situation.

## 5. CONCLUSIONS AND FUTURE RESEARCH DIRECTIONS

The aim of this work was to design and implement a graph colouring based algorithm to solve the examinations timetabling problem at SUA. To the authors' best knowledge, the study that is based on graph colouring approach has not been implemented in Tanzanian academic timetabling problems. The algorithm produced a collision-free examination timetable in few seconds compared to the manually generated timetables, which usually take about three weeks to prepare. The results obtained shows 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. Basically, it has been shown that, with the automated timetabling, the examinations at SUA can be scheduled in a maximum of two weeks instead of the current 3 weeks, saving a full week's cost. Therefore, the graph colouring is a viable and good algorithm for the SUA examinations timetable.

In this work we used the best-fit strategy in room allocation. A competitive strategy is the largest-fit. This strategy allows multiple (small) examinations to be scheduled in a single big room. Thus, the largest-fit strategy can be used in optimizing the total number of invigilators. It would be interesting to investigate the impact of the largest-fit strategy in room allocation in relation to human resource utilization.

**REFERENCES**

[1] Burke, E. K., Bykov, Y., Newall, J., & Petrovic, S. (2004). A time-predefined local search approach to exam timetabling problems. *IIE Transactions on Operations Engineering,* 36(6): 509-528.

[2] El-mohamed M. A. S., Coddington P. Fox G. (1998). A Comparison of Annealing Techniques for Academic Course Scheduling, *Lecture Notes in Computer Science,* 1408: 92-114.

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

[4] Mushi, A. R (2004). 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)*, 5 (2): 34-40.

[6] White, G. M. and. Xie, B. S (2001). Examination Timetables and Tabu Search with Longer-term Memory. *Lecture Notes in Computer Science vol. 2079. The Practice and Theory of Automated Timetabling III:* Selected Papers (PATAT 2000), pages 85-103.

[7] Mushi A. R (2007). Simulated Annealing Algorithm for the Examinations Timetabling Problem. *African Journal Of Science and Technology (Science and Engineering Series)*, 8 (2): 24-32.

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

[9] Welsh, D.J.A and Powell, M.B (1967). An Upper Bound for the Chromatic Number of a Graph and Its Applications to Timetabling Problems. *Comp.46 Journal*, 10: 85-86.

[10] Redl, T. A. (2009). On using Graph Colouring to Create University Timetables with Essential and Preferential Conditions, *Advances in Marketing, Management and Finances*, *Proceedings of the 3rd International Conference on Management, Marketing and Finances (International Conference on Computational and Information Sciences*, University of Houston-Downtown), pages 162-167.

[11] Mehta, N. K (1981). The Application of a Graph Colouring Method to an Examination Scheduling Problem". *Interfaces,* Volume 11 (5). 57-65.

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

[13] Chartrand G. and Lesniak L., (2005) Graphs & digraphs, Chapman & Hall/CRC, Boca Raton, FL.

[14] Diestel R., (2000) Graph Theory, volume 173 of Graduate Texts in Mathematics. Springer Verlag, New York.

[15] Brelaz, D. (1979). New Methods to Colour the Vertices of a Graph. *ACM*, 22(4): 251-256.

[16] De Werra, D. (1985) An Introduction to timetabling. *European Journal of Operational Research,* 19: 151-162.

[17] Leighton, F. T. (1979). A graph colouring algorithm for large scheduling problems, *Journal of Research of the National Bureau of Standards* 84: 489-503.