NATIONAL BOARD FOR TECHNICAL EDUCATION
NIGERIAN JOURNAL OF TECHNICAL EDUCATION
Volume 22 Nos. 1 2024
ISSN No. 2992-3522

# SOLVING POLYTECHNIC EXAMINATION TIME TABLING PROBLEM USING GENERAL TREE PERMUTATION ALGORITHM

ISAH, Adamu Dagah.[1]
dagahtech@gmail.com
Umar, Ibrahim Kontagora[1]
ibrahimumarkontagora@gmail.com
GONNA, Isah Sheshi[1]
sheshiinfo@gmail.com
Department of Computer Science
Niger State Polytechnic Zungeru, Niger State, Nigeria
ADAMU, Mamman[2]
bejian2004@gmail.com
HOSEA,Kolo[2]
audutsadu@gmail.com
[2]Department of Computer Science
Federal Polytechnic Bida, Nigeria.

**Abstract**

*Examination timetable preparation is a tedious task to undertake most especially if it is to be undertaken manually because much human effort and time will be required. Hence, the used of General Tree Permutation (GTP) algorithm in preparing polytechnic examination timetable to reduce time wastage and effort required in preparing the timetable.The main motivation for using GTP is its ability to depict various ways a given set of items can be orderly arranged to minimize cost and maximize gain. GTP algorithm proposed in this paper orderly arrange the scheduling of timetable by automatic generation of a path for GTP at a time to form and ordered arrangement in examination timetabling.Theresults of implementation of the algorithm demonstrated that it can reduce stress in examination timetable preparation and generation.*

**Keywords:** Examination Timetable, General Tree Algorithm, Scheduling

## 1. Introduction

Examination timetable preparation is an intractable problem and have been considered to be complex and time-consuming required by educational institutions every (Ayob et al., 2011; Aldeeb et al., 2021). Typically, formulation of examination timetable follows a manually based repetitive approach educational institutions throughout the world which do not meet required standards of stakeholders in education because it requires huge time

based on their complex nature. In most cases, feasible solutions are not attained with a lot of conflicts in the solution (Aldeeb et al., 2015; Islam et al., 2016) and a big extent of effortrequiredto undertake and completethem within limited days (Aldeeb et al., 2021).Thus, proffering a better solution to examination timetabling issues lead to quality-based timetable with significant bearing on standard of the associated educational (Aldeeb et al., 2015). Examination timetabling problems have been defined as problem of assigning set of exams to given number of timeslots and rooms subject to set of constrains (usually divided into hard and soft constraints). Hard constraints related to the schedule violations, where no student should have two exams the same time while, soft constraints (schedule quality) arenot essential but should be satisfied. soft constraints refer to spreading exams evenly as possible throughout the schedule exams which are deemed to have the most students scheduled earliest as possible to allow sufficient marking time. Timetables that meet hard constraints are called feasible timetable (Ayob et al., 2007; Ayob et al., 2011; Aldeeb et al., 2021; Mandal et al., 2020).

Timetabling problem can be capacitated or Un-capacitated.Un-capacitated timetabling problem doesnot consider room capacities while capacitated timetabling problem include Hard constraint that the number of students allocated to specific room during a scheduled period must not exceed the capacity of the room (Aldeeb et al., 2015). The focus of this paper is on Un-capacitated Examination timetabling Problem.

## 2. Review of related works

Burke and Newall (1999) proposed A Multi-Stage Memetic Algorithm to avoid optimizing large problems. They make use of the original memetic algorithm (Burke et al., 1996; and Paechter *et al.*, 1995) requiring longer time to optimize problem involving large events by only applying the algorithm tosubset of total events a time. The algorithm fixes events in the timetable before considering next subset events and schedule those on top of already scheduled previous stage to reduced time taken in finding solution. However, attempting to find best subset size to use with this method, there appears to be other optimal sizethat is perhaps related to the number of events in the problem under considerations.

Mandal *et al.*, (2020) use partial exam assignment approach in solving examination time tabling problem, Partial Graph Heuristic orderings with a Modified Great Deluge algorithm (PGH-mGD). Their method partially schedules group of exams using graph heuristic orderings with exam

assignment value$v$. The partially scheduled exam is improvedusing modified GD algorithm. This continues with selection of next exams to be schedule based on $v$and graph heuristic orderings. This continues until the scheduling exams are complete.

Cowling *et al.*,(2002)report of survey on views from students and invigilators on exam time tabling software, reviled, that an automated time tabling system that can produce feasible and high-quality time tables is required.

Abdullah and Turabieh (2008) use genetic algorithm which is analogy of evolutionary process in nature with local search to generate university lecture time table aimed at enhancing the local search algorithm.

Ayob *et al.*,(2011) proposed an intelligent system to generate examination timetable for Universiti Kebangsaan Malaysia (UKM) involving phases. The phase that generates initial feasible solution for all predefinedhard constraints at the university using graph coloring method and thephasethat enhance timetable quality generated by applying meta-heuristics algorithm to minimize soft constraints violation.

Aldeeb *et al.*,(2015) proposed intelligent water drop algorithm to solve un-capacitated university examination time tabling problem by simulating the dynamic of the river flow system subject to a set of hard and soft constraint. The results of their analysis were not at the same level with that previously reported in literature when similar datasets were used. The authors were however of the opinion that local search-based algorithm can further improve the results they obtained.

Aldeeb *et al.*,(2021) in their study, they enhanced intelligent water drop (IWD) algorithm efficiently handle university examination timetabling problem. Enhancing exploitation capability intelligent water drop algorithm. Aldeeb *et al.*,(2015) hybridized local search optimizer (LSO) with Intelligent Water Drop algorithm for solving university examination time tabling problem to strike the right balance between exploration and exploitation. In all the timetabling algorithms and solutions, finding efficient optimal solution in which both hard and soft constraints are put into consideration with little delay are major challenges researchers have tried to address in their paper. This paper aims to complement the author's efforts by producing General Tree Permutation (GTP) algorithm for preparing a quality polytechnic examination timetable. The motivation for using GTP is its ability to depict various ways a given set of items can

be orderly arranged to minimize cost and maximize gain when constrains are applied since timetable problems are subject to hard and soft constraints.

### 3. Examination Timetabling

Time table problems are in many real-life circumstances. The term is described scheduling of courses, examination, sports, and meetings with the aim of satisfying requirements with limited time identified as been responsible for the problem (Islam et al., 2016; Qu et al., 2009).

Qu*et al.*,(2009)sees examination timetabling problems as assigning set of exams$E = e_1, e_2, e_3 \ldots . e_*$to limited number ordered time slots (time periods) $T = t_1, t_2, t_3 \ldots . t_T$ rooms capacity in certain times lot $C = c_1, c_2, c_3 \ldots . c_t$ subject to set of constraints. Constraints in the timetabling related studies are hard constraints and soft constraints. It is required that hard constraintsare not violated under any circumstances, for student not having multiple exams at the same time, room assigned more than its capacity (in the case of capacitated timetable). While Soft constraint related to schedule quality which is not critical.soft constraints give more time for making and discourage multiple exams for students at the same time or day (Aldeeb et al., 2021; Islam et al., 2016; Qu et al., 2009).

### 4. General Tree Permutation Description and Formulation

A general tree is anon-linear structure in which its objects (nodes) belong to a level in the hierarchy with one of the objects designated as the root and placed at the top of hierarchy connecting other nodes (child nodes) through arcs/edges. Permutation of objects (i.e. different ways a given set of set of objects/items $X = \{x_1, x_2, x_3 \ldots . x_n\}$can be arrange into slots) can be depicted using a permutation tree technique with each path from the root to the leaf node forming an ordered arrangement/schedule (Shola, 2008) as follow: For each node $x$starting from the parent node just created, create a child node for the node for each item in $X$ not yet on the path from the root to that node $x$. Repeat this procedure for each child nodes just created. Since generating all the possible path with the smallest cost may not be possible because of the memory problem when large number of items are involved, to have a schedule, a path of a tree must be generated at a time and that with the minimum cost should be taken. This can be obtained by application of constraints to the path generated (Shola, 2008). The Tree Permutation for the elements in object$X$ given above is given in figure 1

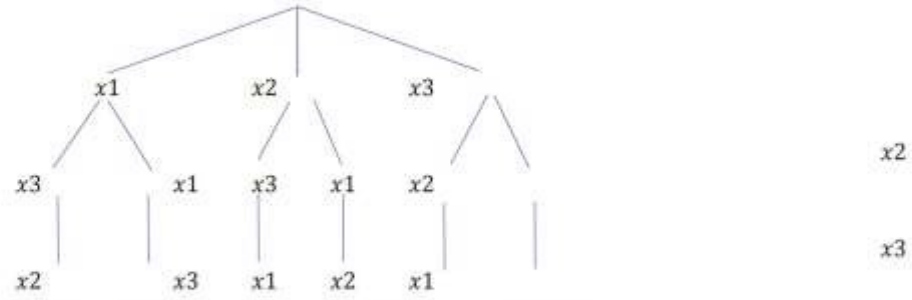



**Figure 1: General Tree Permutation of Element in Object *X***

The permutation tree in Figure 1, produce six path/schedule listed below: $x1, x2, x3$ $x1, x3x2,$ $x2, x1, x3$ $x2, x3x1,$ $x3, x1, x2$ *and* $x3, x2, x1$. The main motivation for using General Tree Permutation is its ability to depict various ways a given set of items can be orderly arranged to minimize cost and maximize gain. General Tree Permutation algorithm proposed in this paper orderly arrange the scheduling of timetable by automatic generation of a path for General Tree Permutation at a time to form and ordered arrangement in examination timetabling. Examination timetable preparation is a tedious task to undertake most especially if it is to be undertaken manually because much human effort and time will be required. Hence, the used of General Tree Permutation algorithm in preparing polytechnic examination timetable to reduce time wastage and effort required in preparing the timetable.

### 4.1. Proposed General Tree Permutation Algorithm for Polytechnic Examination Timetable

Here, we discussed General Tree Permutation Algorithm for Polytechnic Examination Timetable and the proposed algorithm. The proposed algorithm will be called GTP algorithm.

### 4.2 GTP Algorithm for Un-Capacitated Polytechnic Examination Timetabling Problem

Polytechnic Examination Timetable is the schedule of polytechnic examinations intogiven number of time slot ina way that there is no clash in the schedule courses during examination. It required that no studentshould take more than one examination in a single period. Tree Permutation technique is customized here to meet requirement for our desired polytechnic examination timetable problem. The notation of GTP algorithm formulations is given in Table 1.

**Table 1: Notation used in GTP Algorithm**

| Symbols | Definition |
|---|---|
| T | The Total Number of TimeSlot |
| E | The total number of Exams |
| t | Set of slots,$t = (1, 2, 3, \ldots T)$ |
| e | Set of exams$e = (1, 2, \ldots E)$ |
| $X$ | A timetable solution is given by$X = (e_i) \quad i = 1 \ldots E)$ |

### 4.2.1 Generating Schedule of Course for Examination Using GTP

To avoid memory problem which may arises in tree permutation when large number of items are involved and generation of solution in batches which may be subject to delay, this paper proposes automatic random generation of a path (a complete schedule) using Algorithm 1 below. This will help solve the problem within a minimal time with a feasible solution using Algorithm 2than using manual system:

1. initialize static variables
2. determine number of require T
3. while $t_i$ picked < T
4. automatically randomly pick $e_i$ from $E$ dataset
5. if$e_i$ already picked and schedule
6. drop$e_i$ and automatically randomly pick next from the $E$ to consider for schedule
7. else if $e_i$ not already schedule
8. determine it category from $E$ dataset
9. update timetable solution/tree path
10. update loop continuation parameters
11. endif
12. end while

**Algorithm 1: General Tree Permutation GTP Algorithm**

### 4.2.2 Improving Generated Examination Timetable Solution

To improve the quality of our solution, hard and soft constraints are applied to the tree path/timetable solution generated to give a feasible timetable solution using Algorithm 2 belowbase on the notations in Table 2.

**Table 2: Notation Used to Improve the Quality of Timetable Solution Generated Using GTP Algorithm**

| Symbols | Definition |
|---|---|
| GC | A General Course offer by students in more than one department |
| G | General course specific to students in a department |
| DC | A departmental course offers by students in other department(s) |
| DR | A Departmental course similar to course code required by students in another department |
| D | Departmental course specific to a department |
| *O* | Others |

1. Search through the path/timetable solution
2. If there is/are GC then
3. Move them to top of the tree path/schedule
4. endif
5. if there is/are G then
6. move them next to GC if GC is top and move same number of common department DR/D/O with G to same level on the schedule list
7. else
8. move G to top of the tree schedule and move same number of common department DR/D/O with G to same level on the schedule list
9. endif
10. If there is/are DC then
11. move them next to G on the schedule list
12. endif
13. move inremaining DR/D/O as generated by GTP to complete the solution

**Algorithm 2: Proposed GTP OptimizerAlgorithm**

**5. Experimental results**

We experiment with our algorithm by implementing it in Visual Studio 2019 using Visual Basic programming language with the National Board for Technical Education (NBTE) course specifications of both first and second semester for NDI Computer Science and NDII Statistics as our dataset. The main factor in evaluation of our solution is the ability of GTP algorithm to produce a quality solution that can be used for generation of feasible examination timetable. Our implementation of GTP algorithm involve two phases. The first

phase involves generating a schedule for a semester while the second phase involve application of constraints (hard and soft constraint) to optimize the schedule for an examination. Figure 2(a) give the result of a generated schedule of an examination for NDI Computer Science and Statistic Departments First Semester before application of constraints and Figure 2(b) after the application of constraints. Figure 3 (a) is the result of a generated schedule of examination for NDII Computer Science and Statistic Departments Second Semester before application of constraints and Figure 3(b) after the application of constraints.

**Timetable Generated Through GTP Before Applying Constraint**

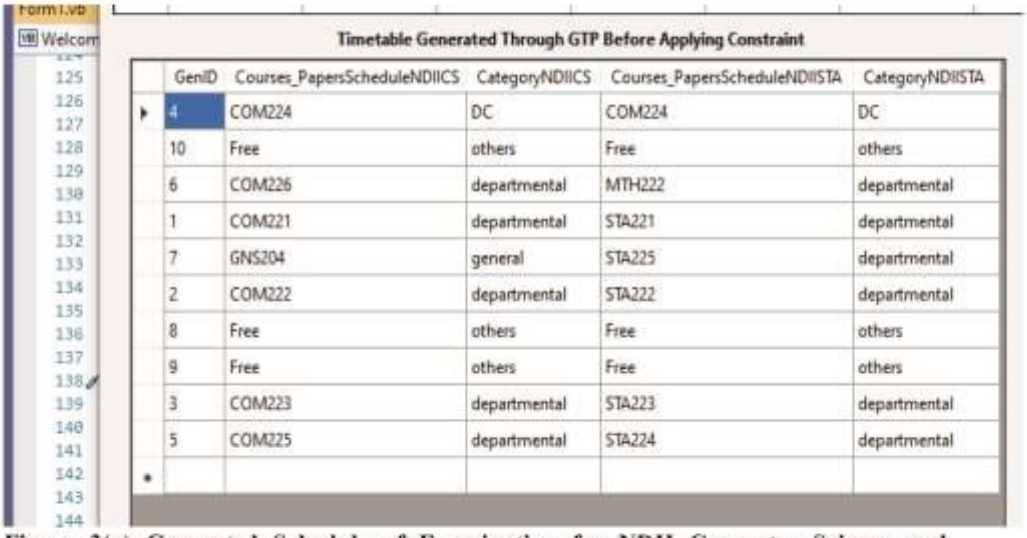
| GenID | Courses_PapersScheduleNDICS | CategoryNDICS | Courses_PapersScheduleNDISTA | CategoryNDISTA |
|---|---|---|---|---|
| 3 | COM113 | departmental | STA112 | departmental |
| 6 | GNS101 | general | Free | others |
| 4 | COM114 | departmental | MTH112 | required |
| 9 | Free | others | Free | others |
| 7 | GNS111 | GC | GNS111 | GC |
| 1 | COM111 | DC | COM111 | DC |
| 2 | COM112 | departmental | STA111 | departmental |
| 8 | MTH111 | DC | MTH111 | DC |
| 5 | COM115 | departmental | STA113 | departmental |
| 10 | Free | others | Free | others |

**Figure 2 (a): Generated schedule of Examination for NDI Computer Science and Statistic Departments first semester before application of constraints**

**After Applying Constraint**

| | Courses_PapersNDICS | Courses_PapersNDISTA |
|---|---|---|
| ▶ | GNS111 | GNS111 |
| | GNS101 | STA112 |
| | COM111 | COM111 |
| | MTH111 | MTH111 |
| | COM113 | Free |
| | COM114 | MTH112 |
| | Free | Free |
| | COM112 | STA111 |
| | COM115 | STA113 |
| | Free | Free |
| * | | |

**Figure 2 (b): After Application of Constraints to Schedule in Figure 2(a)**



**Timetable Generated Through GTP Before Applying Constraint**

| | GenID | Courses_PapersScheduleNDIICS | CategoryNDIICS | Courses_PapersScheduleNDIISTA | CategoryNDIISTA |
|---|---|---|---|---|---|
| ▶ | 4 | COM224 | DC | COM224 | DC |
| | 10 | Free | others | Free | others |
| | 6 | COM226 | departmental | MTH222 | departmental |
| | 1 | COM221 | departmental | STA221 | departmental |
| | 7 | GNS204 | general | STA225 | departmental |
| | 2 | COM222 | departmental | STA222 | departmental |
| | 8 | Free | others | Free | others |
| | 9 | Free | others | Free | others |
| | 3 | COM223 | departmental | STA223 | departmental |
| | 5 | COM225 | departmental | STA224 | departmental |
| * | | | | | |

**Figure 3(a) Generated Schedule of Examination for NDII Computer Science and Statistic Departments Second Semester before application of constraints**

| | After Applying Constraint | |
|---|---|---|
| | Courses_PapersNDIICS | Courses_PapersNDIISTA |
| ▶ | GNS204 | Free |
| | COM224 | COM224 |
| | Free | MTH222 |
| | COM226 | STA221 |
| | COM221 | STA225 |
| | COM222 | STA222 |
| | Free | Free |
| | Free | Free |
| | COM223 | STA223 |
| | COM225 | STA224 |
| * | | |

**Figure 3 (b): After Application of Constraints to Schedule in Figure 3 (a)**

## 6. Conclusion

A GTP algorithm has been developed with two phases.The Phase for generation of an initial solution not satisfy hard/soft constrains. Phase to improv the generated solution in the earlier phase through GTP Optimizer algorithm (Algorithm 2). The results obtained using GTP, shows that the algorithm is capable of solving polytechnic examination timetabling problem. The algorithm will be tested using data set compare performance with similar algorithm.

## 7. References

Abdullah, S., & Turabieh, H. (2008, November). Generating university course timetable using genetic algorithms and local search. In 2008 Third International Conference on Convergence and Hybrid Information Technology (Vol. 1, pp. 254-260). IEEE.

Aldeeb, B. A., Al-Betar, M. A., Norwawi, N. M., Alissa, K. A., Alsmadi, M. K., Hazaymeh, A. A., & Alzaqebah, M. (2022). Hybrid intelligent water Drops algorithm for examination timetabling problem. Journal of King Saud University-Computer and Information Sciences, 34(8), 4847-4859.

Aldeeb, B. A., Al-Betar, M. A., Abdelmajeed, A. O., Younes, M. J., AlKenani, M., Alomoush, W., ... & Alqahtani, M. A. (2019). A comprehensive review of uncapacitated university examination timetabling problem. International Journal of Applied

Engineering Research, 14(24), 4524-4547.

Aldeeb, B. A., Norwawi, N. M., Al-Betar, M. A., & Jali, M. Z. B. (2015). Solving university examination timetabling problem using intelligent water drops algorithm. In Swarm, Evolutionary, and Memetic Computing: 5th International Conference, SEMCCO 2014, Bhubaneswar, India, December 18-20, 2014, Revised Selected Papers 5 (pp. 187-200). Springer International Publishing.

Ayob, M., Hamdan, A. R., Abdullah, S., Othman, Z., Nazri, M. Z. A., Abd Razak, K., ... & Sabar, N. R. (2011). Intelligent examination timetabling software. Procedia-Social and Behavioral Sciences, 18, 600-608.

Ayob, M., Malik, A. M. A., Abdullah, S., Hamdan, A. R., Kendall, G., & Qu, R. (2007). Solving a practical examination timetabling problem: a case study. In Computational Science and Its Applications–ICCSA 2007: International Conference, Kuala Lumpur, Malaysia, August 26-29, 2007. Proceedings. Part III 7 (pp. 611-624). Springer Berlin Heidelberg.

Burke, E. K., & Newall, J. P. (1999). A multistage evolutionary algorithm for the timetable problem. IEEE transactions on evolutionary computation, 3(1), 63-74.

Cowling, P., Kendall, G., & Hussin, N. M. (2002, August). A survey and case study of practical examination time tabling problems. In Proceedings of the 4th International Conference on the Practice and Theory of Automated Timetabling PATAT02 (pp. 258-261).

Islam, T., Shahriar, Z., Perves, M. A., & Hasan, M. (2016). University timetable generator using tabu search. Journal of Computer and Communications, 4(16), 28-37.

Mandal, A. K., Kahar, M. N. M., & Kendall, G. (2020). Addressing examination timetabling problem using a partial exams approach in constructive and improvement. Computation, 8(2), 46.

Shola, P. B. (2008). Data structure and algorithm using C++ (First edition). Reflect publishing house

Qu, R., Burke, E. K., McCollum, B., Merlot, L. T., & Lee, S. Y. (2009). A survey of search methodologies and automated system development for examination timetabling. Journal of scheduling, 12, 55-89.