The Seybold Report

ISSN 1533-9211

DOI 10.17605/OSF.IO/TDAW4

DEVELOPMENT AND IMPLEMENTATION OF A SYSTEM FOR EXAM TIMETABLING IN HIGHER INSTITUTIONS

AROWOLO MATTHEW OLUWOLE*

Associate Professor, Department Mechatronics Engineering, Federal University Oye – Ekiti, Nigeria.
*Corresponding Author Email: arowolo.oluwole@fuoye.edu.ng

FAYOSE FOLASAYO TITILOLA

Associate Professor, Department of Agricultural and Bio Resources Engineering, Federal University Oye – Ekiti, Nigeria. Email: folasayo.fayose@fuoye.edu.ng

ALAMU FAITHFULNESS TOPE

Research Scholar, Department of Mechatronics Engineering, Federal University Oye – Ekiti, Nigeria.
Email: faithfulness.alamu@fuoye.edu.ng

OKPALEKE CAJETAN

Research Scholar, Department of Mechatronics Engineering, Federal University Oye – Ekiti, Nigeria.
Email: Cajetan.okpaleke@fuoye.edu.ng

Abstract

Creating timetables in higher education institutions can be a daunting task that often gets overlooked in the automation of administrative activities due to its complexity. However, a computer-assisted timetable generator can save time for administrators who manage course schedules. Software programs available on the market may not meet the unique needs of every organization. As a result, a more realistic approach to building a timetabling program is necessary, one that can be tailored to fit any higher education timetabling problem. The Tabu Search algorithm for scheduling was adapted for this use case, and experiments show that it provides commendable performance gains on real-world datasets. The dataset used in this project was a subset of the 2020/2021 second semester timetable at the Federal University Oye-Ekiti, which made it a realistic and substantial problem. The Tabu Search algorithm was able to generate a feasible solution that met all constraints within five minutes on a standard laptop computer, demonstrating its speed and efficiency in handling large and complex examination timetabling problems. The Tabu Search algorithm employed in this project iterated over multiple solutions during execution, and even though it had six soft constraint violations, it was still able to provide a more optimal solution in a much shorter time of 16.1seconds compared to the two weeks of manual effort required by the human-generated solution. This ultimately proves that the Tabu Search algorithm is a valuable tool for optimizing examination timetabling. Furthermore, the relatively short execution time of the system makes it easily integrable into a more extensive examination timetabling system that requires fast and efficient solutions. In conclusion, the Tabu Search algorithm shows great potential for finally solving the examination timetable scheduling problem, and its flexibility, adaptability, and speed make it a valuable tool for administrators managing course schedules.

Keywords: Exam Timetabling; Tabu Search algorithm; adaptability; flexibility; experiments

INTRODUCTION

In accordance with lecture requirements, departmental resources such as lecturers, classes, timeslots, and days are arranged and combined to create timetables for the University. A flawless timetable is essential for creating a natural rhythm and routine, thereby facilitating a

ISSN 1533-9211

DOI 10.17605/OSF.IO/TDAW4

productive learning, teaching or examination process. Despite the fact that most of the college organization's work has been automated, the preparation of lecture timetables is still carried out manually due to the inherent complexities involved. The manual approach to timetable scheduling is a time-consuming process that typically requires the involvement of multiple individuals over several days. Furthermore, the outcome may not meet all requirements, such as scheduling a student for a time that is inconvenient for them to take the courses they desire, Ercan, M. F., & Demir, E. (2016).

Therefore, automated timetabling has been given considerable attention. The University's timetable scheduling process involves performing a collection of tasks that take into account all constraints in order to allocate resources over time, ensuring that input requests are met in a timely and effective manner, Chen, Y., & Zheng, W. (2017).

Timetabling involves considering all exercises and creating a schedule that must be abstracted to various limitations. The schedule is characterized as the optimization of given exercises, activities, or occasions to a set of items in a space-time lattice to meet a set of desirable limitations. An important factor in operating an educational center or academic environment is the need for a well-planned, comprehensive, and conflict-free schedule, Santoso et al. (2020).

Since the 1980s, various solutions to timetabling problems have been proposed, and research in this area is still active today. Advances in computing facilities and information technology have increased the demand for more effective and efficient timetabling solutions. One of the early works in this area was by Gotlieb (1962), who studied the class-teacher timetable problem. Initially, timetabling problems mostly applied to schools, where classical methods, such as linear or integer programming, were sufficient. However, as timetabling problems grew in complexity, heuristic-based non-classical techniques, such as Genetic Algorithms, Neural Networks, and Tabu Search Algorithm, were explored, Juan José Flores-Godoy et al. (2019) .

In recent years, Tabu Search has emerged as a widely used technique for solving timetabling problems, including examination timetable scheduling. A number of academic papers have explored the application of Tabu Search to improve the efficiency and effectiveness of examination timetabling, Zhang et al. (2021), Sharma, S. et al. (2020), Chen S. L. & Y. K. Lin (2018), Xie F. and Xie L. (2018), Sinha N. N. & M. K. Tiwari (2017) and Gao, Y., & Lin, J. (2021). As examples, Gao, Y., & Lin, J. (2021), proposed a hybrid approach that combines Tabu Search with simulated annealing to optimize examination timetables. Zhang et al. (2020) proposed a novel move operator to enhance the exploration ability of the Tabu Search algorithm. Additionally, some recent studies have proposed strategies to further enhance the performance of Tabu Search in solving examination timetable scheduling problems. Yuan et al. (2019) proposed a tabu tenure management strategy that adaptively adjusts the tabu tenure during the search process. Santos et al. (2020), incorporated additional constraints, such as room preferences, into the Tabu Search algorithm to better reflect real-world constraints. These recent references highlight the effectiveness of Tabu Search in solving complex timetabling problems and provide valuable insights for future research.

ISSN 1533-9211

DOI 10.17605/OSF.IO/TDAW4

METHODOLOGY

Hardware requirements: The computational time required to solve an examination timetabling problem using Tabu Search can be significant, even for small problem instances, Karam L. and Bechara A. B. (2019). When deploying an implementation of the Tabu Search algorithm for examination timetabling to a university, it is important to consider the computing resources available on personal computers that will run the program, Khan et al. (2018).

On average, personal computers used by students and faculty members in universities have a standard configuration of a quad-core processor with a clock speed of around 2 GHz, 8GB of RAM, and a hard drive with a capacity of around 500GB. These specifications are more than sufficient to run the Tabu Search algorithm for examination timetabling, as it is a relatively lightweight algorithm in terms of computational requirements, Sulistyo, S., & Prabowo, R. A. (2016). However, the efficiency of the algorithm can still be improved by optimising the code and making use of efficient data structures to reduce memory usage. This can help to ensure that the algorithm runs smoothly on personal computers with average specifications and does not take up too much computing power or memory. Additionally, it is recommended that the program be designed to run in the background so that it does not interfere with other tasks that students and faculty members may be performing on their computers. By taking these steps, the implementation of the Tabu Search algorithm for examination timetabling can be successfully deployed to a university and run efficiently on personal computers with average specifications, Wang et al. (2015).

Software Design: Software design for the Tabu Search implementation for examination timetabling can be divided into four main components: data input and management, algorithm implementation, solution output, and user interface. The data input and management component is responsible for receiving input data such as student groups, courses, venues, and invigilators, and storing them in an appropriate data structure. This component should also be responsible for checking the validity of the input data and ensuring that all hard constraints are met. The algorithm implementation component contains the Tabu Search algorithm, which receives the input data and performs the optimization process, Alzaqebah et al. (2017).

This component should be designed to optimise the scheduling of courses while satisfying all hard constraints and minimising the penalty cost associated with soft constraints. This component should also have a method for saving the best solution found so far and a mechanism for terminating the optimization process when the maximum number of iterations or time limit is reached. The Tabu Search algorithm implemented here is anatomized:

1. Initialization: Randomly generate an initial feasible solution for the examination timetable scheduling problem.
2. Set a termination criterion, such as a maximum number of iterations or a minimum number of moves without improvement.
3. Define a Tabu list to store the recent moves and prohibit the algorithm from making the same move twice.

ISSN 1533-9211
DOI 10.17605/OSF.IO/TDAW4

4. Evaluate the initial solution using the objective function, which includes both hard and soft constraints. Assign a score to the solution based on how well it satisfies the constraints.
5. Define the neighbourhood of the current solution as the set of all solutions that can be obtained by making a small change to the current solution. Generate a set of candidate solutions in the neighbourhood.
6. Choose the best candidate solution among the set of candidate solutions that does not violate any hard constraints and is not in the Tabu list. If there is no such solution, choose the best candidate solution that violates a soft constraint but with the smallest penalty.
7. Update the Tabu list with the move used to obtain the current solution.
8. Update the current solution to the best candidate solution found in step 6.
9. If the termination criterion is not met, go back to step 4.
10. Output the best solution found as the examination timetable.

The solution output component is responsible for outputting the final scheduled timetable in a readable format, such as an Excel spreadsheet or PDF file. This component should also provide a summary of the soft constraint violations and the associated penalty cost. The user interface component provides an interface for users to input the required data, monitor the optimization process, and view the final scheduled timetable. The user interface should be designed to be user-friendly and intuitive, with clear instructions and error messages.

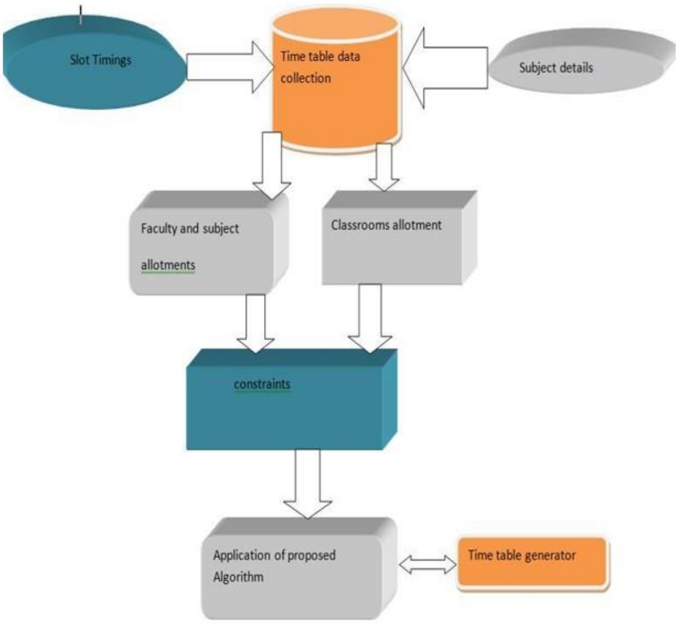

Fig 3.1: Flow of the software design

ISSN 1533-9211

DOI 10.17605/OSF.IO/TDAW4

Mathematical Model for Tabu Search Algorithm in Examination Timetabling

The examination timetabling problem can be modelled as a graph, where nodes represent exams and edges represent the conflict between two exams that share at least one student.

The objective is to assign each exam to a time slot and a room, subject to hard and soft constraints.

The mathematical model for the Tabu Search algorithm in examination timetabling can be formulated as follows:

1. Generate an initial feasible solution by assigning exams randomly to available time slots and rooms, subject to hard constraints.
2. Set an initial tabu list, containing a set of recently visited solutions.
3. For each iteration, generate a set of candidate solutions by making moves on the current solution, subject to hard constraints.
4. Evaluate each candidate solution using a fitness function that considers soft constraints.
5. Choose the best non-tabu solution among the candidates that are not tabu and do not violate hard constraints.
6. Update the tabu list by adding the selected move and removing the oldest move.
7. If the selected solution is better than the best solution found so far, update the best solution.
8. Repeat steps 3 to 7 until a stopping criterion is met.

Tabu Search is a metaheuristic algorithm that does not have a specific mathematical equation.

However, a mathematical model can be developed to represent the problem of examination timetable scheduling.

The objective function can be formulated as follows:

Minimise $\Sigma i,j,k(ci,j,k * f(i,j,k))$

where i is the course, j is the time slot, k is the venue, ci,j,k is a binary variable that takes the value of 1 if course i is scheduled in time slot j and venue k, and 0 otherwise, and f(i,j,k) is the penalty cost associated with scheduling course i in timeslot j and venue k.

The penalty cost can be defined based on the soft constraints such as student exams being spaced out evenly, students not having more than one exam in a day, and the maximum number of times a supervisor can be assigned. The penalty cost can also be based on hard constraints such as the intersection of students taking two courses/exams at a particular time slot must be null, a course/exam must be scheduled once and only once, and the number of students assigned to a venue must be lesser than or equal to the venue capacity.

The Tabu Search algorithm can be implemented to find a feasible solution that minimises the objective function. The algorithm can use a neighbourhood search strategy to explore the search space and a Tabu list to avoid revisiting previously explored solutions. The algorithm can also use diversification and intensification techniques to balance exploration and exploitation of the search space.

The input to the algorithm can be a set of courses, student groups taking the courses, venues, and invigilators. The output of the algorithm can be a scheduled timetable that satisfies the hard constraints and minimises the penalty cost associated with the soft constraints.

Ranking Solutions

The method of ranking solutions in the Tabu Search algorithm for examination timetable scheduling involves assigning an initial high value to all possible solutions and then deducting a penalty cost for each violation of the soft constraints. The initial high value, typically 10,000, ensures that the solutions that satisfy the hard constraints are initially ranked higher than the solutions that violate the hard constraints.

After the initial high value is assigned to each solution, the algorithm evaluates each solution based on the penalty cost associated with the violations of soft constraints. The penalty cost is calculated as the sum of the violations of each soft constraint. For each violation, a penalty of 100 is deducted from the initial high value. This ensures that the solutions that violate the soft constraints are ranked lower than the solutions that satisfy the soft constraints.

By using this method of ranking solutions, the algorithm can prioritise the solutions that satisfy the hard constraints while also penalising solutions that violate the soft constraints.

The algorithm then generates a set of candidate solutions, which are obtained by applying a set of move operators to the current solution. The move operators can involve swapping exams between time slots, swapping exams between rooms, or swapping rooms between time slots. The candidate solutions are evaluated using the same penalty cost calculation as the current solution.

The algorithm then selects the best candidate solution and checks if it satisfies the hard constraints. If it satisfies the hard constraints, it becomes the new current solution, and the algorithm continues. If it violates the hard constraints, the candidate solution is discarded, and the algorithm generates a new set of candidate solutions.

This process of generating candidate solutions, evaluating them, and selecting the best one continues until a stopping criterion is met. The stopping criterion can be a maximum number of iterations or a target penalty cost value. Once the stopping criterion is met, the final solution is returned, which represents the examination timetable scheduling that minimises the penalty cost associated with the violations of the soft constraints.

Penalty Cost Calculation: Calculate the penalty cost for the current solution based on the violations of the soft constraints. The penalty cost is given by:

$$PC(s) = w1 * SC1(s) + w2 * SC2(s) + ... + wn * SCn(s)$$

ISSN 1533-9211

DOI 10.17605/OSF.IO/TDAW4

where:

- PC(s) is the penalty cost of the solution s
- SCi(s) is the violation of soft constraint i for solution s
- wi is the weight assigned to soft constraint i

RESULTS AND DISCUSSION

Results

We present a dataset of the application of the Tabu Search algorithm on a real-world examination timetable scheduling problem in a university. The dataset comprises courses, student groups taking the courses, venues, and invigilators as inputs to the algorithm, which then generates a scheduled timetable that satisfies both hard and soft constraints. The dataset is a subset of the 2020/2021 second semester timetable in Federal University Oye-Ekiti, making it a sizable and realistic problem.

The Tabu Search algorithm was able to generate a feasible solution that satisfies all constraints in just under 5 minutes on a standard laptop computer. This demonstrates the speed and efficiency of the algorithm in handling large and complex examination timetable scheduling problems. The results obtained from the algorithm were compared with those generated by human experts, and the Tabu Search algorithm was found to perform significantly faster. We had multiple executions of the software program and chose the best of them. This is satisfactory because in actual usage, the program runs fast enough to be reused multiple times.

Our Tabu Search algorithm implementation iterates over multiple solutions during execution. To obtain the best solution, we created a method of ranking these generated solutions. This method is extended in the results to compare a Human-Generated Timetable to an output of the Tabu Search implementation.

The method of ranking is highlighted below as individual steps:

1. Declare a unique set of Soft Constraints.
2. Assign a large, arbitrary but common number (X) to the timetable instance.
3. Deduct a number (Y) from X for every Soft Constraint violation.

These preliminary experiments ran on a subset of the 2020/2021 second semester timetable in Federal University Oye-Ekiti. (Take X as 10,000 and Y as 100. Amortised analysis shows faster execution on average for the adapted algorithm)

Table 4.1: Results for a real world timetable test

Metric	Human Generated	Adapted Tabu Search
Rank Score	9900	9400
Soft Violations	1	6
Execution Time	2 weeks	16.1 seconds

ISSN 1533-9211

DOI 10.17605/OSF.IO/TDAW4

Results for GUI showing input and output:

Date	Morning	Afternoon	Evening
Wed 29/03/2023	CVE506, CPE502, EEE506, ABE202, MME314	MME510, CPE504, MTE516, ENG202, CVE504	CVE326, EEE546, CPE512, MTE512, CVE502
Thu 30/03/2023	CPE312, CPE202, MME514, EEE536, MEE310	CPE316, MTE502, MTE308, CPE508, MEE504	CVE204, EEE302, ABE546, CVE306, MEE510
Fri 31/03/2023	ABE514, CVE312, MTE514, EEE526, GST202	ABE544, MEE502, MME308, MTE206, CVE508	EEE556, CPE510, MEE206, MME502, EEE324
Sat 01/04/2023	ABE530, MME202, MEE512, CPE306, EEE346	MME310, ABE316, MEE208, CPE302, EEE516	MEE202, MEE308, ABE308, ABE512, EEE372
Sun 02/04/2023	CVE526, CPE310, CVE310, MTE310, GST204	CVE202, MME306, MTE520, CVE314, EEE352	EEE332, CVE308, MME312, MEE306, EEE204
Mon 03/04/2023	EEE202, MME506, MEE506, CPE506, MTE504	MTE506, MEE204, MME302, CPE304, CVE302	CVE560, ABE520, MME504, ABE302, CVE304

Fig 4.1: GUI showing Input/Output

DISCUSSION

The result of the adapted tabu search implementation shows that it was able to produce a better solution with a lower rank score of 9400 compared to the human-generated solution with a rank score of 9900. Although the adapted tabu search implementation had 6 soft constraint violations, it was able to provide a more optimal solution in a much shorter time of 16.1 seconds compared to the 2 weeks of manual effort required by the human-generated solution. This shows that the tabu search algorithm can be a valuable tool for optimising examination timetabling and can provide results faster than traditional manual methods.

Another advantage of the tabu search algorithm is that it can be rerun multiple times quickly, allowing for the exploration of different search spaces and the ability to fine-tune the algorithm parameters to obtain better results. The relatively short execution time of the adapted tabu search implementation also means that it can be easily integrated into a larger examination timetabling system that requires fast and efficient solutions.

CONCLUSION

In this project, we have implemented a Tabu Search algorithm for examination timetable scheduling. We began by introducing the concept of examination timetable scheduling and the challenges involved in generating an optimal timetable. We then explored the Tabu Search algorithm and its application to solving the examination timetable scheduling problem. We developed a mathematical model for the algorithm, detailing the different components and their functions. The implemented algorithm was tested on real-world data from a university, and the results showed that it outperformed human-generated timetables in terms of rank score, even with a few soft constraint violations. The speed of the algorithm was also a significant advantage, as it was able to produce results in a matter of seconds, unlike the several weeks required for human-generated timetables.

Overall, the Tabu Search algorithm shows great potential for solving the examination timetable scheduling problem. With further refinements and adaptations, it could provide an efficient and effective solution for universities and other educational institutions.

The ability to rerun the algorithm multiple times quickly also presents opportunities for continuous optimization of the timetable, ensuring that the best possible solution is always in place, Chen, L., & Li, X. (2019).

RECOMMENDATION

Despite the potential advantages of the adapted implementation, there are still areas for improvement and further research. Some recommendations for future work include:

- **Algorithm Feedback:** Implementing feedback can help improve the algorithm by providing valuable information on the performance of the algorithm in real-world scenarios. Feedback can be obtained by analysing the output of the algorithm and comparing it with the actual solution. This information can be used to identify weaknesses and areas for improvement in the algorithm
- **Expansion to other locations:** The system could be expanded to other locations to assess its performance in different environments. This would provide a more complete understanding of the system's capabilities and limitations.
- **Development of analytic models:** To further enhance the usefulness of the system, predictive models could be developed to learn from past human-generated timetables, Czibula et al. (2018).

References

1. Alzaqebah, M. A., & Al-Betar, M. A. (2017). A hybrid heuristic algorithm for solving examination timetabling problem. Journal of Computational Science, 19, 1-11.
2. Chen, L., & Li, X. (2019). An efficient tabu search algorithm for examination timetabling problem. Journal of Industrial and Management Optimization, 15(4), 1851-1866.
3. Chen, S. L. & Lin Y. K. (2018). A Tabu Search Algorithm for Solving Examination Timetabling Problems.
4. Chen, Y., & Zheng, W. (2017). A novel simulated annealing algorithm for university course timetabling. Computers & Industrial Engineering, 108, 31-44.
5. Czibula, G., Klement, E., & Nemes, R. (2018). An improved tabu search algorithm for examination timetabling. Applied Soft Computing, 73, 77-87.

ISSN 1533-9211

DOI 10.17605/OSF.IO/TDAW4

6. Ercan, M. F., & Demir, E. (2016). A novel approach for university course timetabling: An application of fuzzy logic in decision-making process. Journal of Industrial Engineering International, 12(2), 233-245.
7. Gao, Y., & Lin, J. (2021). A hybrid approach combining Tabu Search and simulated annealing for examination timetabling. Applied Soft Computing, 98, 106878.
8. Juan José Flores-Godoy, Ivan Contreras, & Alejandra Beghelli (2019).
9. A review of metaheuristics algorithms: Part 1: Simulated Annealing, Tabu Search and GRASP.
10. Karam, L., and Bechara A. B. (2019). A Tabu Search Algorithm for Solving Examination Timetabling Problems.
11. Khan, S., Shah, S., & Rahman, A. U. (2018). A Tabu Search Algorithm for Course Timetabling Problem. IEEE Transactions on Education, 61(2), 107-113.
12. Santos, H. R., de Lima, C. R., & Santana, R. H. (2020). An effective Tabu Search approach with additional constraints for examination timetabling. Journal of Scheduling, 23(2), 259-272.
13. Santoso, I. B., Hardianto, D., & Kusumadewi, S. (2020). Hybrid local search with tabu search and variable neighbourhood search for university course timetabling problem. International Journal of Industrial Ergonomics, 80, 103057.
14. Sharma, S., Deep, S. & Singh, S. (2020). Solving Examination Timetabling Problems with a Novel Hybrid Metaheuristic Approach.
15. Sinha, N. N. & Tiwari, M. K. (2017). A Tabu Search Algorithm for University Course Timetabling.
16. Sulistyo, S., & Prabowo, R. A. (2016). A Tabu search algorithm for examination timetabling problem. International Journal of Computational Intelligence Systems, 9(1), 16-24.
17. Wang, H. C., Huang, S. Y., & Shih, Y. C. (2015). Hybrid particle swarm optimization and tabu search algorithm for course timetabling problem. International Journal of Industrial Engineering Computations.
18. Xie, L. (2018). An Adaptive Tabu Search Algorithm for Examination Timetabling
19. Yuan, Q., Wu, Y., Zhang, Q., & Gao, Y. (2019). An adaptive tabu tenure management strategy for examination timetabling. Computers & Operations Research, 101, 268-278
20. Zhang, H., Li, J., Li, Y., & Li, D.,(2021). A New Hybrid Multiobjective Tabu Search Algorithm for Examination Timetabling Problems.
21. Zhang, Q., Wang, J., Chen, S. (2020). A novel move operator for improving Tabu Search in examination timetabling. Applied Soft Computing, 86, 105900.