International Journal of Engineering and Techniques - Volume 12 Issue 2, March-April - 2026

Optimization of Resource Allocation of University Innovation and Entrepreneurship Education

U. Sriya Reddy, Sandhya Rani Padhy, Salina Pradhan, Amita Patro, Prof. Ashish Kumar Dass

reddysriya257@gmail.com, sandhyaranipadhy1204@gmail.com, salinapradhan2003@gmail.com, amitapatro305@gmail.com, ashishkumardass@nist.edu

Department of Computer Science and Engineering, NIST University, Berhampur

1. Abstract:

In various educational institutes and universities, it has always been quite difficult and a source of headache while constructing no or less conflict timetables especially for a large mass of teachers and students who are divided into different batches, branches and sections. Timetabling is categorized as NP-hard as there exists massive number of combinations and large search space. Not only teachers and students, but also managing other common or general resources such as Labs, Auditoriums, Conference Halls, is not that easy. Thus, this paper proposes an automated timetable generation and resource allocation system which is based on Genetic Algorithm and Constraint Satisfaction Problem techniques.

Keywords—Genetic Algorithm, Constraint Satisfaction Problem, Evolutionary Algorithm, Resource Utilization, Timetabling, Meta-heuristics

2. Introduction

Indeed, in an institution, a timetable itself is a crucial resource as it involves classes, rooms, teachers, sections under a timeline and associated people are bound to follow it. Designing a timetable manually, is time-consuming, sometimes inefficient and often involves conflicts. For example, a teacher is assigned with more than one class at the same time. Similarly, assigning a room to more than one section at the same time. As a result, the timetable is being modified multiple times with no guarantee of eliminating further mismatches and achieving optimal usage of resources. In addition to class scheduling, an institution also need to manage shared or general resources such as auditoriums, conference halls, and laboratories in such a way that no two events are scheduled at the same time, thereby avoiding conflicts. Along with all above mentioned,

we also need to take care of two types of constraints and they are Hard Constraints and Soft Constraints. The former one involves strict rules and conditions that need to be followed in any condition such as, no overlapping of classes for teachers or students, no multiple assignments of the same room at the same time, where as, the latter involves preferences that are desirable but not compulsory, as anyhow it may be adjusted. It includes balanced workload distribution, priority-based allocation of resources when multiple events are scheduled.

Thus, this paper aims to introduce a system that leverages Genetic Algorithm and Constraint Satisfaction Problem techniques that automates allocation of various available resources in an optimal manner

3. Literature Review

Over the years, several approaches and solutions have been proposed to address the scheduling problem. Many of them have used techniques such as Genetic Algorithms and Constraint Satisfaction Problem methods to generate efficient and conflict-free timetables. In [1], a Genetic Algorithm-based is used for timetable generation, where the system has capabilities for input of the various courses, halls of lectures, departments, lecturers and the specification of a few constraints from which the timetable is constructed and aims to reduce high cost and slow turnaround. In [2], a Genetic Algorithm-based approach was proposed to solve conflicts that occur while scheduling an examination at a university. In [3], a detailed study of automated timetabling methods was presented, emphasizing the role of heuristic and metaheuristic techniques. In [4], introduction to the concept of genetic algorithms is being done and explained their applications in solving optimization problems. In [5], a hybrid approach consisting of genetic and greedy algorithms was proposed to solve the NP-hard examination timetabling problem.

In [6], examination timetabling techniques were explored to generate schedules that are capable of handling uncertainties and variations. In [7], a review of university timetabling methods was done, covering various optimization techniques that includes genetic algorithms, integer programming, and other metaheuristic approaches. In [8], genetic algorithms were enhanced using graph neural networks. In [9], metaheuristic algorithms were applied to timetabling in railway systems.

4. Proposed System

Our proposed system combines ideas and principles of Constraint Satisfaction Problem with a genetic Algorithm framework. We have used the Genetic Algorithm to explore the solution space and optimize outcomes and CSP principles to enforce constraints like faculty availability, room suitability, and conflict handling. We implement this through domain filtering and fitness evaluation, penalizing invalid assignments and rewarding feasible solutions.

Additionally, our system manages shared institutional resources or general resources such as conference rooms, auditoriums, seminar halls, and research laboratories by using a priority-based scheduling mechanism. Events like guest lectures and expert talks receive higher priority than regular activities such as club workshops. When scheduling conflicts occur for shared resources, we may reschedule lower-priority events to make room for higher-priority activities.

Initially, our system collects considered inputs for the timetable generation. Inputs include allocated time (start time and end time), academic year and semesters, number of faculties available to teach, names of theory subjects, number of rooms available and its types. Once these foundational inputs are collected, each entity is assigned a unique identifier within the genetic string. This process, known as Encoding, represents a single potential timetable as a Chromosome. The Genetic Algorithm performs crossover by combining good parts of generated timetable and mutation i.e., randomly introducing any change to check for a better fit.

4.1 Advantages of our Proposed System

- Elimination of manual timetable making.
- Allocation of Resources in an Optimal way.
- Less or no conflicts among teachers, rooms, labs.
- Incorporates faculty availability and preferred time slots, ensuring more practical and convenient scheduling.
- Support for priority-based scheduling of shared resources.

5. Proposed System Architecture

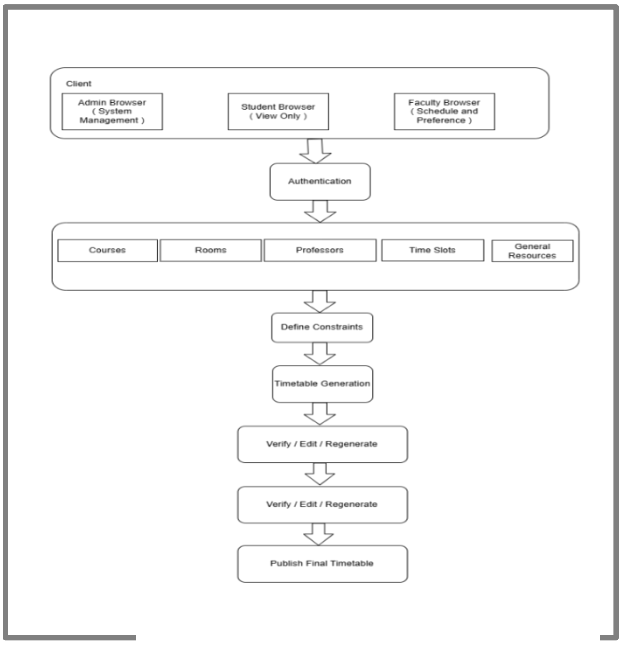

Fig 1: Workflow of the System

6. Genetic Algorithm

A Genetic Algorithm is a meta-heuristic i.e., a higher level algorithm that guides other heuristics for efficient solution. It is inspired by Charles Darwin's theory of evolution which states that species change their characteristics and properties with respect to time through natural selection and organisms who are fittest compared to others survive and reproduce further. Genetic Algorithm belongs to class of Evolutionary Algorithms and it has large search space that allows exploring a vast number of combinations

In Genetic Algorithm, a problem is treated as a set of solutions, represented as chromosomes and each of which represents potential solution of the given problem. Chromosomes are evaluated through fitness function which is a numerical score. A solution is desirable if it has a high score. As a result, it is being selected for the next generation. If not so, then the chromosome is discarded off.

New generation of chromosomes is generated with the help of three genetic operators i.e., Selection, Crossover, and Mutation.

- Selection: as the name suggests, selects the fittest chromosomes for reproduction and weakest ones are eliminated.
- Crossover: creates a new population by synthesizing genetic materials or genomic structures of two parent chromosomes. It introduces variation in chromosomes.
- Mutation: used in changing the values of a gene randomly, resulting in a new solution.

Above mentioned operators balance the two most important goals: Exploitation and Exploration.

Fitness Function: In Genetic Algorithm, implementation of "survival of fitness" is achieved through the fitness function as it serves the primary objective of evaluating the quality of the chromosome. It is inversely proportional to the number of violations of constraints. i.e., its value decreases as the amount of constraints increases.

Termination: The iterative process continues until a specific termination criterion is met. It usually stops under the following conditions:

- Optimal Solution Reached: A chromosome achieves a perfect fitness score.
- Generational Limit: The algorithm completes a pre-defined number of iterations.

At the end, the highest-ranking chromosome is identified as the optimal, or near-optimal, solution within the given search space as it has survived multiple cycles of competitive selection.

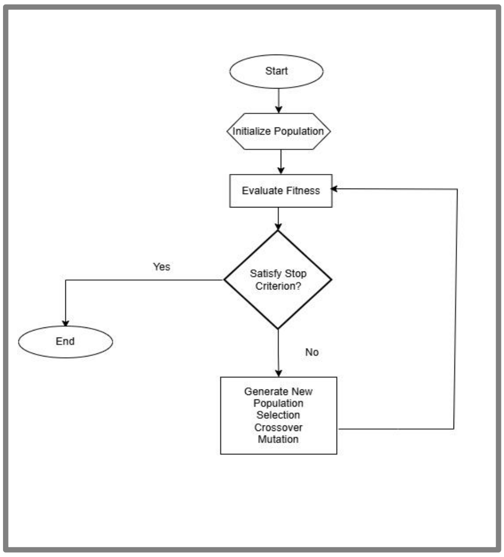

Fig 2: Flowchart of Genetic Algorithm

7. Constraint Satisfaction Problem

The Constraint Satisfaction Problem (CSP) serves as the foundational layer of our proposed resource allocation system, ensuring that all generated schedules satisfy essential and required rules. In this framework, the scheduling problem is formally modelled as:

(CSP: V, D, C)

where:

V is the set of variables including a task, event or class session.
D describes the set of domains defining all potential values.
C is the set of restrictions.

The system's constraints are categorized as hard and soft to ensure both correctness and quality. By integrating CSP logic directly into the Genetic Algorithm's fitness evaluation, the system effectively prunes the search space ensuring the evolutionary process is guided towards a globally consistent solution.

Hard constraints must be strictly adhered to under all situations and conditions. They guarantee the generated schedule is accurate because high penalty values follow invalid assignments. Some of the hard constraints are listed as :

- No faculty member is assigned to more than one session at a time.
- No room or venue is allocated to multiple sessions simultaneously.
- Room capacity must be sufficient for the number of participants.
- Laboratory or specialized sessions must be assigned to appropriate facilities.
- Faculty workload must remain within predefined limits.
- All subjects or sessions must meet their required number of hours.

Any violation of above constraints results in an infeasible schedule.

Soft constraints are preference-based conditions that enhance the usefulness and quality of the produced schedule. They ensure efficient resource utilization, balance workload distribution for a faculty and reduce idle time between courses for both instructors and students. Some of the soft constraints are listed as:

- The Idle time is to be minimized between sessions for faculty and students.
- Ensuring balanced workload distribution across faculty members.
- Reduce frequent changes in room or venue assignments.
- Maximize utilization of available resources and infrastructure.
- Prefer suitable time slots for different types of sessions.

Constraint Satisfaction Problem (CSP) repair is used for fundamental feasibility. In CSP, a collection of variables is established with a domain and each variable is assigned different constraints. Solving CSP requires assigning a value to a variable to satisfy all constraints.
Directly incorporating CSP logic into the fitness assessment of the Genetic Algorithm effectively reduces the search space. For effective local repair and feasibility verification CSP repair is also employed. It ensures that the evolutionary process is directed towards a globally consistent solution rather than being erratic.

8. Results and discussion

The system provides a seamless user experience by utilizing a modern web stack that includes React 18 and TypeScript for the front-end. This ensures responsiveness across all devices and type safety. TanStack Query is used for intelligent caching, which ensures data appears instantly upon a user's return after navigation away. The back-end is managed by Supabase for data persistence and security. To ensure the integrity of the academic schedule, Row-Level Security (RLS) is implemented, creating a strict logical barrier between user types. This allows administrators and faculty to manage resources while restricting students to a read-only view. The core scheduling logic is powered by a Genetic Algorithm executed within Supabase Edge Functions (Deno), which enables the heavy computational task of evolving schedules and checking constraints. To maintain data integrity, the system performs real-time validation against predefined constraints. If any conflict arises that cannot be resolved, the system provides immediate feedback to the administrator, allowing for manual repair or regeneration.

The Genetic Algorithm (GA) provides global exploration while the CSP repairer enforces hard rules after each evolutionary step. Over generations, fitness improvements reflect reductions in critical violations of constraints, which are retained through mutations and stagnation. The hybrid model ensures that the timetable strictly adheres to the given constraints. The Genetic Algorithm and Constraint Satisfaction Problem timetabling model uses a combination of hard constraint satisfaction metrics and soft constraints optimization metrics, demonstrating its effectiveness.

The steady increase in the overall fitness score across generations demonstrates the model's ability to balance broad exploration. It not only accelerates the path towards a feasible timetable but also ensures that the final output is robust enough to handle the unpredictable, high-density rules of a university or any educational institution.

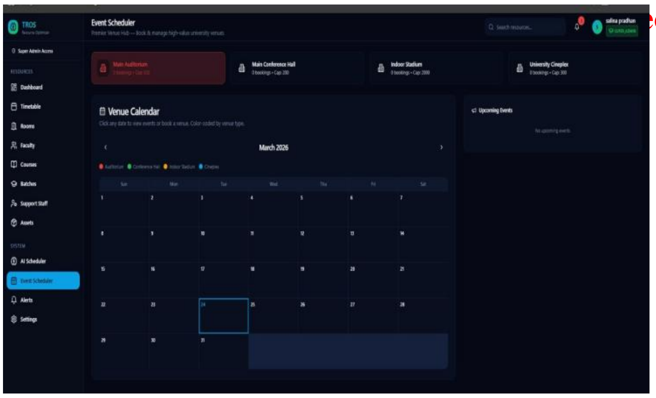

Fig 3: Event Scheduler and Venue Management

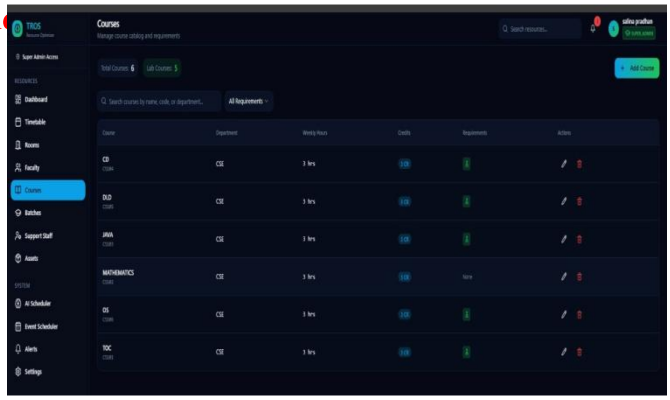

Fig 4: Course Curriculum and Requirement Configuration

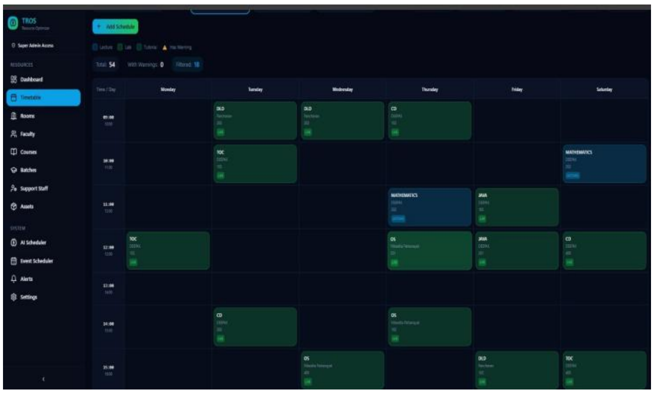

Fig 5: Timetable Generation

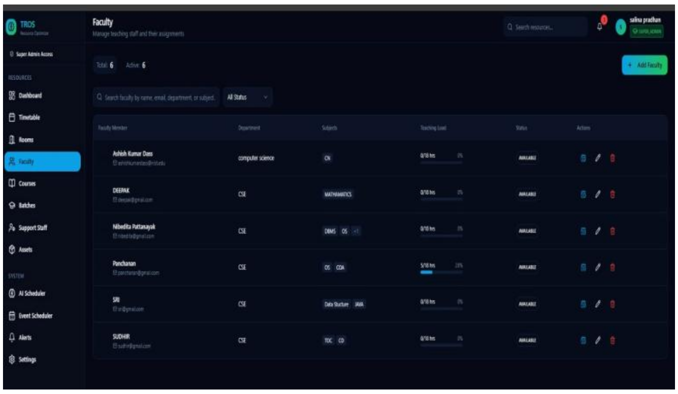

Fig 6: Faculty Management interface.

9. Application

Academic Timetabling is the direct application of our proposed system. It solves the complex task of assigning lecturers, student batches, classrooms and subjects while avoiding overlaps. Our system also considers general or shared resources such as Laboratories, Conference Halls, Auditoriums, and Seminar Halls and ensures maximum utilization of all above mentioned.

10. Conclusion

The hybrid approach of Genetic Algorithm and Constraint Satisfaction Problem is one of the best ways to demonstrate and solve optimized allocation of resource as it can be implemented in real world. Timetabling is a just subset of our system's capabilities and effectively minimizes conflicts, saves significant administrative time and provides flexibility and fairness that manual scheduling cannot achieve. It also ensures that specialized resources are utilized in a correct manner.

Beyond academic environments the core GA-CSP architecture can be further improved to solve other real world problems that ask for a proper scheduling and optimization.

References

[1] S. Thakare, P. Deshmukh, and R. Patil, "Automated Timetable Generation Using Genetic Algorithm," International Journal of Engineering Research & Technology (IJERT), vol. 9, no. 6, pp. 120–124, 2020.

[2] M. F. Mohammed, A. A. Ahmed, and S. K. Ali, "Examinations Timetabling System Based on a Genetic Algorithm," in Proc. ICEMIS, 2022, pp. 1–6.

[3] S. Adedayo, O. Olabode, and T. Adewale, "Development of an Automated Timetable Generator Using Genetic Algorithm," LAUTECH Journal of Computing and Informatics, vol. 5, no. 1, pp. 45–52, 2023.

[4] A. R. Mahlous and M. Mahlous, "Student Timetabling Genetic Algorithm Accounting for Student Preferences," PeerJ Computer Science, vol. 9, e1234, 2023.

[5] S. J. Agbolade, K. O. Adebayo, and M. O. Olatunji, "Optimisation of University Examination Timetable Using Hybridised Genetic and Greedy Algorithms," International Journal of Computer Applications, vol. 185, no. 12, pp. 25–32, 2024.

[6] B. Bassimir and R. Wanka, "On the Computation of Robust Examination Timetables," Journal of Scheduling, vol. 27, no. 2, pp. 233–247, 2024.

[7] M. A. Alghamdi, "A Review of University Timetabling

[8] L. Cornei, A. Popescu, and D. Ionescu, "Enhancing Genetic Algorithms with Graph Neural Networks: A Timetabling Case Study," arXiv preprint arXiv:2602.08619, 2026.

[9] D. Muñoz-Valero, J. Torres, and P. García, "Metaheuristic Algorithms for Timetabling in Railway Systems," arXiv preprint arXiv:2504.17455, 2025.