Journal of Sciences, Computing and Applied Engineering Research (JSCAER), Vol. 1, No.3, pp. 1-15

Published Online (https://jcaes.net), September 12, 2025 by SciTech Network Press ISSN: 3092-8648

An Adaptive Fuzzy Ant Colony Optimization-Based System for Scheduling University Lecture Timetable

[1*]Samuel A. Robinson, [2]Moses E. Ekpenyong, [3]Udoinyang G. Inyang, [4]Daniel E. Asuquo, [5]Anthony A. Imianvan, [6]Kingsley U. Joseph

[1]Department of Cyber Security, Faculty of Computing, University of Uyo, Uyo, Nigeria, samuelrobinson@uniuyo.edu.ng
[2]Department of Computer Science, Faculty of Computing, University of Uyo, Uyo, moseseekpenyong@uniuyo.edu.ng
[3]Department of Data Science, Faculty of Computing, University of Uyo, udoinyanyinyang@uniuyo.edu.ng
[4]Department of Information Systems, Faculty of Computing, University of Uyo, Uyo, danielasuquo@uniuyo.edu.ng
[5]Department of Computer Science, Faculty of Physical Sciences, University of Benin, Benin City, Nigeria, tonyvanni@uniben.edu
[6]Department of Information Systems, Faculty of Computing, University of Uyo, Uyo, Nigeria, kingsleyjoseph@uniuyo.edu.ng
[1,3,4,6]TETFund Center of Excellence in Computational Intelligence Research, UNIUYO, Uyo, Nigeria
***Corresponding Author's email**: samuelrobinson@uniuyo.edu.ng

Abstract: University lecture timetabling is a complex optimization problem involving multiple constraints and resource limitations. This study presents a hybridized Ant Colony Optimization (ACO) and Fuzzy Logic (FL) based system to enhance scheduling efficiency and accuracy. By integrating Constraints Logic Programming, the proposed system effectively handles both hard and soft constraints, optimizing lecture schedules while minimizing clashes and resource conflicts. FL is employed to manage uncertainties in the search space, improving the adaptability and decision-making capabilities of ACO. The study also incorporates a knowledge base to store and process timetabling constraints, ensuring logical and structured allocations. Scatter plot analysis reveals that certain hard constraints, such that no lectures should be scheduled after 6:00 pm (H3) and the course must be assigned to either the first or second semester (H4) exhibit high stability and significantly influence timetable optimization. The system is implemented using MATLAB R2015a, Microsoft SQL Server, and Python, with results demonstrating improved scheduling efficiency compared to conventional methods. The proposed model enhances knowledge assimilation, reduces lecture delays, and contributes to more effective university timetabling solutions.

Keywords: Lecture Timetabling, Hard and Soft Constraints, Scheduling Efficiency, Hybrid Algorithm, Decision Support System.

1.0 Introduction

Companies, institutions, and organizations strive to allocate resources efficiently to minimize costs while maximizing productivity and profitability. This is accomplished through strategic planning, which involves structuring operations, setting guidelines, and implementing policies that drive businesses toward their objectives. Resource allocation plays a crucial role in optimizing available assets to achieve specific goals, such as in-flight scheduling, timetable scheduling, and online reservations. Tertiary institutions serve as the foundation of the educational sector, fostering knowledge acquisition and contributing significantly to global economic growth. Lectures and examinations are essential for assessing students' academic progress and performance. However, one of the major challenges faced by educational institutions is university lecture timetabling. [1] classify university timetabling as a non-deterministic polynomial (NP)-hard problem, meaning that the computational time required to solve it increases exponentially with the problem size. Various approaches, including sequential, cluster-based, and meta-heuristic methods, have been employed to address this issue [2],[3][4]. Despite these efforts, existing timetabling methods often result in poor knowledge retention among students, lecture delays, insufficient lecture periods, and scheduling conflicts, ultimately leading to the production of underprepared graduates entering the workforce annually [5][6]. The constraints in university lecture timetabling are categorized into hard and soft constraints [7]. Hard constraints must be strictly adhered to for academic activities to function smoothly, while soft constraints support and enhance hard constraints to improve the effectiveness of timetable scheduling. This study aims to satisfy hard constraints while minimizing the violation of soft constraints in university lecture timetabling. Effective scheduling impacts teaching quality and student learning by aligning course timeslots with preferences. Various methods, including nonlinear programming, coloring theory, evolutionary algorithms, genetic algorithm (GA), particle swarm optimization (PSO), tableau search (TS), and ant colony optimization (ACO), have been developed to address this challenge[8]. These algorithms aim to find optimal solutions within the search space. Among them, ACO has demonstrated strong performance due to its minimal parameter requirements, parallelism, and positive feedback, which enable the rapid discovery of optimal solutions. However, ACO presents challenges such as difficulty in theoretical analysis and uncertain convergence time. To overcome these limitations, fuzzy logic (FL) is introduced to manage uncertainties and vagueness in the search space. FL mathematically represents imprecision and noisy data, offering an inference system that mimics human reasoning. Despite its strengths, conventional knowledge representation techniques struggle to express fuzzy concepts effectively, as classical probability and first-order logic are inadequate for handling commonsense knowledge.

To address these limitations, ontology is integrated into the timetabling framework, providing a structured visualization of domain knowledge. This research proposes a hybrid approach combining ACO, FL, and ontology for university timetabling. Constraints Logic Programming (CLP) is used to incorporate both hard and soft constraints into the system, ensuring optimal scheduling. In this hybrid model, FL decision variables refine ACO's search process. By integrating these techniques, the proposed system enhances timetabling efficiency, effectively resolving classroom constraints, and optimizing university scheduling.

The rest of the paper is organized as follows: Section 2 reviews existing literature on lecture timetable scheduling while Section 3 presents the methodology, and Section 4 presents results and discussion while section 5 concludes the paper with direction for future research.

2.0 Related Works

While students and lecturers assess timetable quality based on their individual preferences, there are also standardized criteria that apply collectively such as minimizing workloads and idle time. [9] adopted a multi-level multi-criteria approach to addressing the university timetabling problem. They demonstrated that computational results yield the procedure's ability to produce high-quality schedules. Genetic algorithms (GAs) are widely used optimization tools, inspired by biological evolution and the survival of the fittest [10][11]. Ideally, GAs explores the search space by evaluating multiple potential solutions and can be easily hybridized to form knowledge-augmented GA models [12]. Several studies have attempted to improve lecture timetabling. [13] developed a scheduling system using a modified quicksort algorithm but failed to account for university management staff preferences. [14] introduced an integer programming approach for final exam scheduling, aiming to maximize student study time. However, the model struggled with handling uncertainties and did not consider university management preferences. [15] applied a fuzzy multiple heuristic approach to optimize exam scheduling. The order of exam placement significantly impacted the final schedule, with difficult exams placed first to satisfy constraints. However, this method often resulted in resource omissions, venue clashes, and lacked adaptability to changes. These challenges highlight the need for more flexible and robust scheduling solutions to optimize university timetabling. [16] implemented an ACO-based approach in parallel to solve university class scheduling as a constraint satisfaction problem. Their findings showed that Ant Colony System (ACS) with pheromones outperformed ACS without pheromones, producing good timetabling solutions.

However, the optimal number of ants remained uncertain, and the study relied on limited artificial data (50 records, three classrooms), which may have affected results. Future research should determine the optimal number of ants for hybridizing ACO with First-Order Logic (FOL), Fuzzy Logic (FL), and Ontology while comparing it with other heuristics. [17] explored three types of ACO parallelism: ant-level, data-level, and functional parallelism. Most ACO implementations use ant-level parallelism, which could be further investigated for efficiency improvements. [18] proposed 'Ant-Solver', an ACO-based approach for solving Constraint Satisfaction Problems (CSPs). This algorithm followed the standard ACO scheme but optimized parameter influences through local search techniques. Key parameters included pheromone factor weight, quality factor weight, persistence rate, and the number of ants. By integrating repair-based local search, 'Ant-Solver' directed ants toward promising solutions. The choice between fast but less effective local searches and slower but more robust methods remains crucial in optimizing CSP resolution. To enhance the 'Ant-Solver', a preprocessing step is introduced to improve search space exploration at minimal cost, allowing solutions to be found earlier. This step gathers a representative set of local minima, from which the best 'N' is selected to initialize pheromone trails. [19] explored agent technology for timetabling, highlighting its distributed nature and performance dependency on distributiveness. The study emphasized the need for open timetabling systems that interact with existing software, store data in databases, and integrate results into these systems.

The research suggested that systems should interpret user preferences and provide accessible feedback. However, the current system's scope was too limited for practical use, necessitating integration with running systems for broader applicability. [20] applied constriction particle swarm optimization (PSO) with local search to solve university course timetabling problems. The aim was to incorporate teacher and class preferences to enhance scheduling efficiency and prevent negative impacts on learning. By encoding particles based on timeslots rather than study hours and introducing an interchange heuristic, the approach improved solution quality. Additionally, an interchange local search mechanism prevented premature convergence. The study successfully addressed conflicts in teacher, class, and classroom schedules, leading to improved satisfaction among stakeholders. However, further refinement is necessary to enhance adaptability and integration into real-world applications. This study is best implemented on a system with limited resources, requiring hybridization for reliability and cost reduction. [21] introduced a university course timetabling approach using Constraint Logic Programming (CLP) with soft constraints.

The key advantages of this method include its declarative problem descriptions through logical constraints and a constraint propagation technique that minimizes the search space. Unlike rigid scheduling frameworks, university timetabling presents additional requirements and preferences, which were validated using datasets from different semesters. [22] used a hybrid metaheuristic combining an electromagnetic-like mechanism (EM) and the great deluge algorithm (GD) for the University Course Timetabling Problem. EM is a global optimization algorithm based on attraction-repulsion physics, while GD is a local search method that accepts worse solutions within a set boundary. This paper improves the approach by using the dynamic force from the attraction-repulsion mechanism to update the boundary level during the search process. However, their approach, tested on a population size of 50, lacks scalability. Although [23] reviewed machine learning (ML)-assisted metaheuristics with promising results on the use of ML models such as Deep Neural Networks, Hopfield Networks, and Self-Organizing Maps, the present study focuses on the use of ACO and FL methods. Current meta-heuristic approaches struggle with adaptability and computational efficiency in scheduling.

Integrating ACO with FL enhances scheduling accuracy by addressing uncertainty and complexity. Our approach adopts a model that connects to a database, managing huge connection records. Through a simple interface, users select a database management system (DBMS), view accessible databases, and execute structured query language (SQL) queries locally or across connected nodes.

3.0 Methodology

This study focuses on a faculty at the University of Uyo, consisting of seven departments, each offering a diverse range of courses. Each semester, these courses must be distributed unevenly across 50-time slots per week and allocated to approximately 40 lecture rooms, including laboratories and workshops. The faculty's operations are divided into administrative and academic activities. While some courses require laboratory sessions, most are conducted in regular lecture rooms. Scheduling these courses involves assigning them within a defined timeframe while managing limited resources, which presents constraints in constructing the lecture timetable. To optimize scheduling, factors such as time slots, available venues, student enrollment, class sizes, and course requirements must be carefully considered. Effective timetable analysis requires evaluating key resources, including periods, student numbers, venue capacities, courses, and lecturers.

3.1 Model Formulation

The current method of timetabling scheduling in the University of Uyo is a manual approach and this leads to the clashing of lecture venues and allocation of large populations of students to a smaller hall. The University Lecture Timetable Problem (ULTP) consists of five sets: C, F, T,L, and S. Set C contains all subject instances, or course events, each having a specific faculty and department designation, maximum student capacity, unit equivalent value, type classifications, and feature requirements. Additionally, every course has a list of other Faculties in whose buildings it can also be scheduled if there is no more room for it in its own designated faculty.

F is the set of all teaching personnel, each one having a minimum, maximum, and targeted unit load values as expressed in Equation (1).

$$F = \{F_{i1}, F_{i2}, F_{i3} \ldots, F_{ij}\} \tag{1}$$

where, F_{ij} is the lecturer of the i_{th} department and j_{th} faculty, T contains the timeslot schedules classified under multiple types expressed in Equation (2).

$$T = \{t_1, t_2, t_3, \ldots, t_q\} \tag{2}$$

T is the total number of timeslots while L is the set of all rooms or locations where course events can take place as expressed in Equation (3).

$$L = \{r_i\}, \text{where}, i = 1,2,3,4,5, \ldots. \tag{3}$$

The total number of students S in each department i at each level j is expressed in Equation (4).

$$S = \{s_{i1}, s_{i2}, s_{i3} \ldots, s_{ij}\} \tag{4}$$

The problem is to assign every course event to a faculty, timeslot, and location so that the hard constraints are satisfied.

3.2 Materials and Methods

The architecture of the proposed Fuzzy ACO-based system consists of the Knowledge Base (KB) comprising the rule base, database, constraint base, and Ontology; Inference Engine housing the ACO module and FL module; and User Interface as depicted in Figure 1.

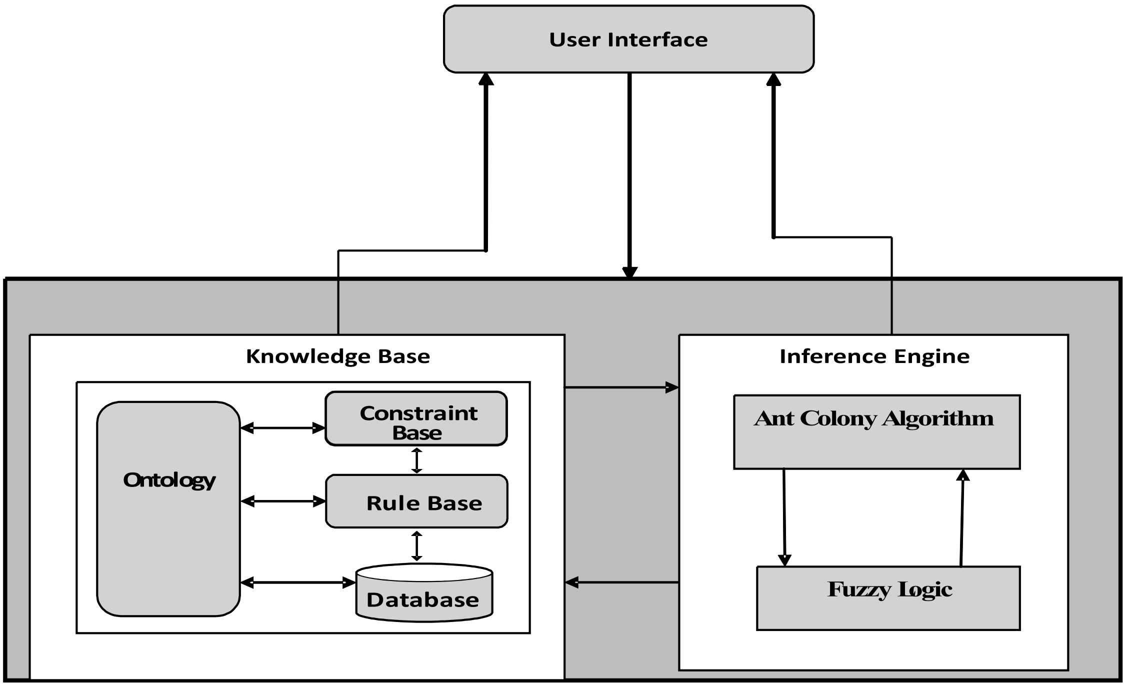

Figure 1: Proposed Fuzzy-ACO Lecture Timetable Scheduling Framework

3.2.1 Ontology Modeling of Timetabling System

The ontology outlines entities, classes, instances, and their interrelationships, providing a formal and explicit specification of the university timetabling system. By establishing a shared understanding of the environment, the system becomes more adaptable for users. All components are interconnected and interdependent, each maintaining a complete copy of the intelligent system. Figures 2–5 illustrate the system's modules and their interactions. The key components of the timetabling system ontology include the department, course, student, and lecture venue

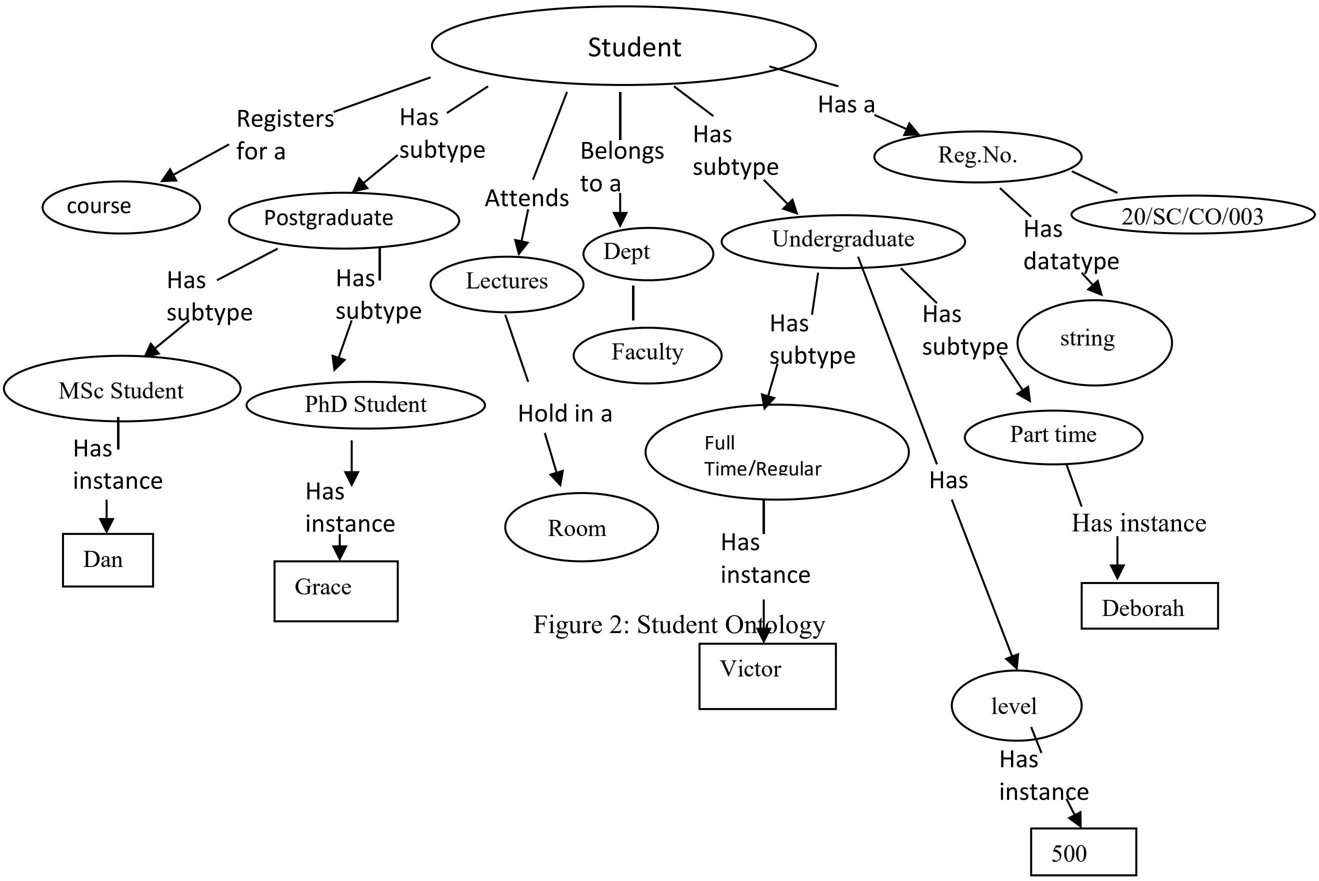

Figure 2: Student Ontology

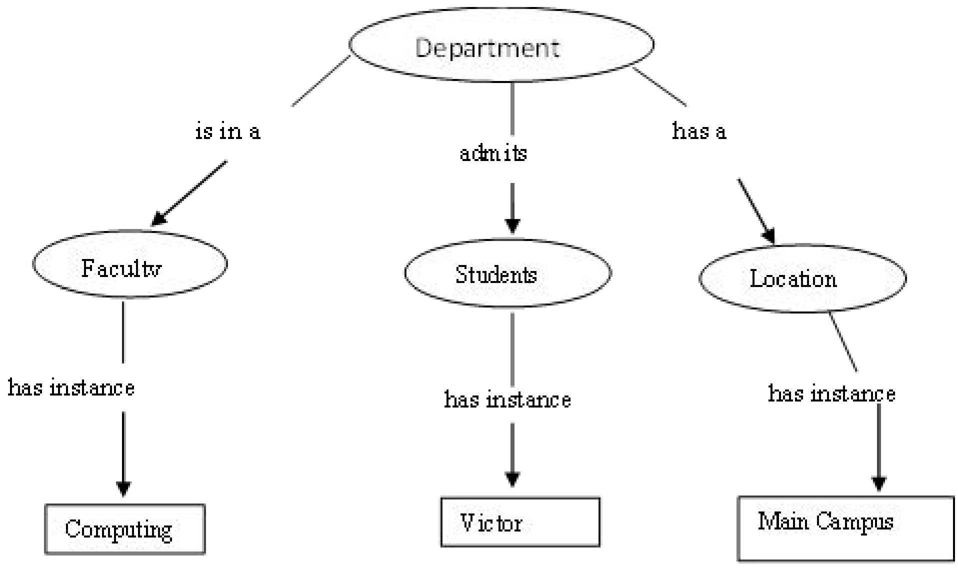

Figure 3: Department Ontology

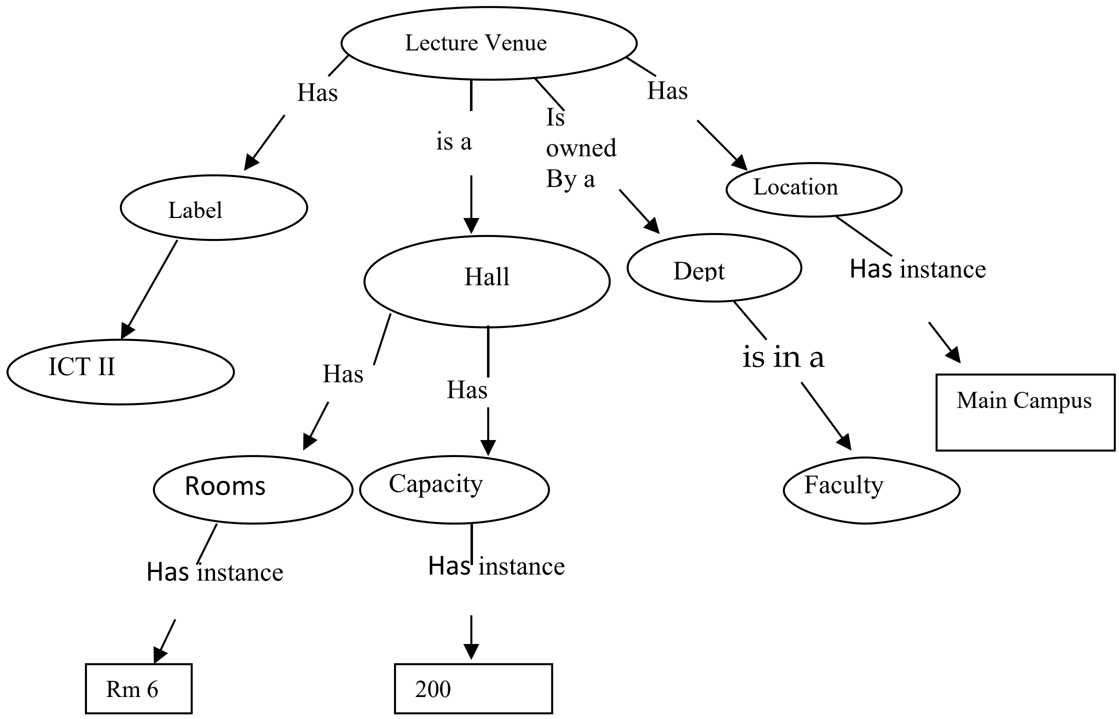

Figure 4: Lecture Venue Ontology

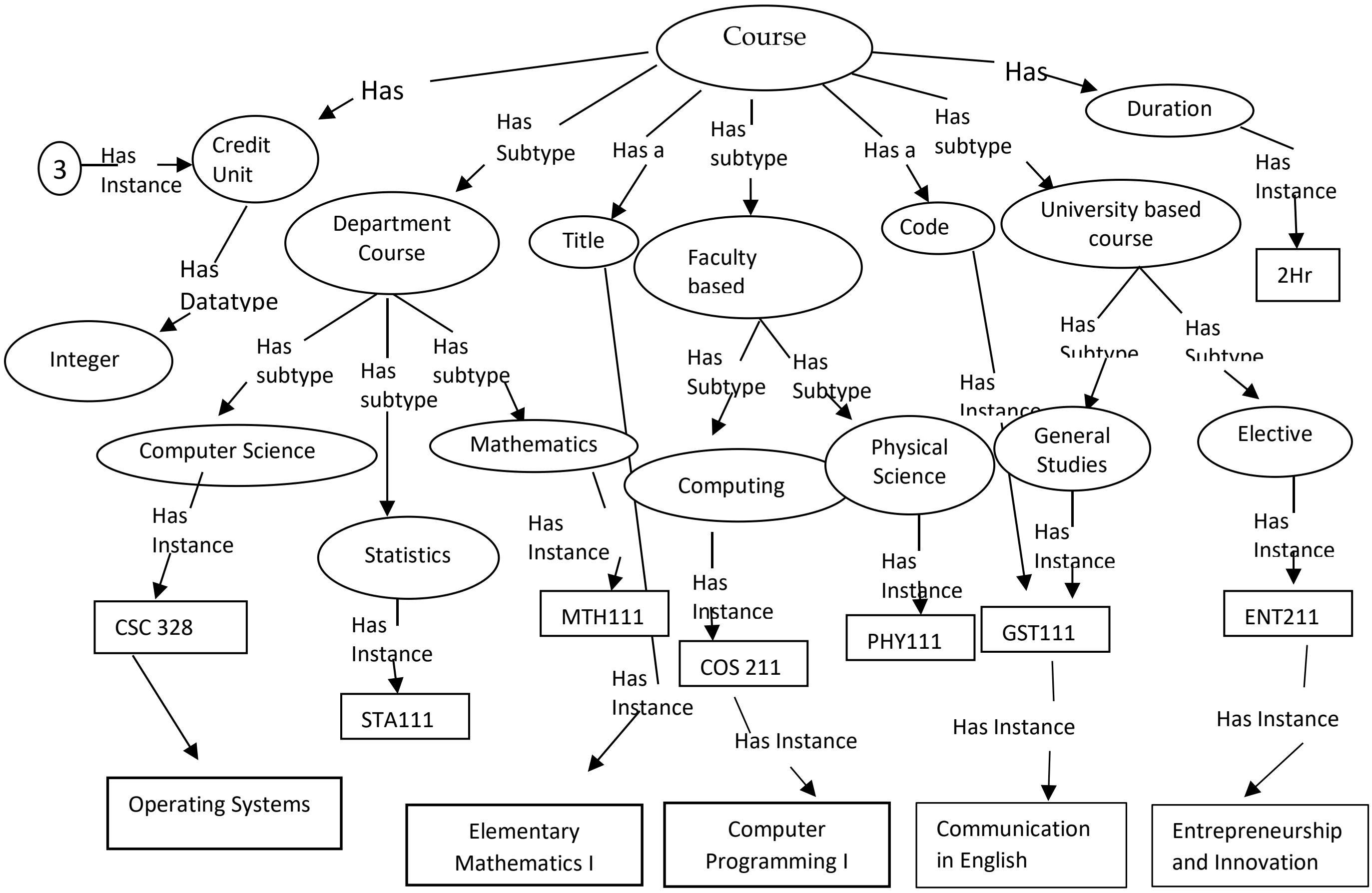

Figure 5. Course Ontology

3.2.2 Knowledge Base

This module permits connections to the database and keeps the registers of these connections. Employing a simple interface, the user selects the database management system (DBMS), and a list of accessible databases is exhibited. Once a database has been selected, a query can be made with the SQL locally and sent to the other nodes connected to the system. The primary data collection method is used to gather information on venues, lecturers, time slots, courses, and student numbers from the various departments in the faculty under study. The KB functions as a specialized database that collects organizes, and stores essential timetable resources, including students, courses, departments, constraints, and lecturers. These resources often contain ambiguities and lack tolerance for imprecision. Within the KB, a constraint base component stores all constraints relevant to timetabling. In lecture scheduling, courses are assigned to specific classrooms and time slots within a week, while students and lecturers are allocated accordingly to ensure classes proceed smoothly. The process of assigning resources to satisfy constraints is systematically organized and stored in the constraint base. This constraint base is categorized into two main types: hard constraints, which must be strictly followed, and soft constraints, which help enhance the scheduling process.

3.2.3 ACO Algorithm and FL Modeling of Constraints

The ACO algorithm is inspired by the foraging behavior of real ants, which find the shortest path between food sources and their nest. This method has been successfully applied to various combinatorial optimization problems. ACO relies on a probabilistic solution construction mechanism based on stigmergy. In artificial intelligence (AI) and optimization algorithms like ACO, stigmergy is used to coordinate agent behavior by reinforcing successful paths or solutions through virtual pheromone updates. This mechanism enables decentralized problem-solving and emergent intelligence in complex systems. The algorithm incorporates a pheromone trail that includes both a proportional rule and a pseudorandom rule. The pseudorandom rule helps capture noisy and imprecise data from the Knowledge Base (KB), sort and prioritize it based on events, and pass it to the proportional rule. The proportional rule then organizes, categorizes, and allocates the data into appropriate storage spaces while assigning suitable timeslots. The algorithm optimizes the processed data and forwards the refined results, now structured but lacking tolerance for imprecision, to the FL system. Figure 6 illustrates the steps required to implement the ACO algorithm

```
Input problem data

        parameters' setting and initialization
        sort course(C) according to the priority of events and student sizes
        while iteration <= max iteration do
            for k = 1 to m do
             create ant k (where k = 1,2,…m)
            set timetable k is empty
            for C = 1 to C do
        choose timeslot (t) using pseudorandom/proportional rules
          if ACS then update local pheromone
          end for
          withdraw ant
           end for
record best solution
update global pheromone based on the type of ACO methods
if Max Min Ant System then update pheromone local
(Max-min) on the trails
end while
Output the best solution
```

Figure 6: ACO Foraging Algorithm

Fuzzy sets help manage uncertainty and vagueness in final timetable scheduling by accommodating varying levels of membership. Key constraints include ensuring the number of students does not exceed classroom capacity (H1), lecturers have a minimum of 4 and a maximum of 12 working hours per week (S2), curriculum-related lectures are scheduled consecutively (S3), and all lecturers of a course are assigned to a single classroom (S4). These constraints are categorized into hard and soft constraints:

i. Hard Constraints

These are mandatory rules that must be followed for a valid timetable. Examples include:

H1: Lectures must be scheduled between 8:00 AM and 6:00 PM.

H2: Undergraduate lectures can last a maximum of 2 hours, while postgraduate lectures can last up to 3 hours. H3: No lectures should be scheduled after 6:00 PM.

H4: A course must be assigned to either the first or second semester.

H5: No two courses for the same student group should overlap.

H6: Two-credit-unit courses should be taught in one session or twice a week.

H7: All lectures for the same group must be held in the same room.

H8: A lecturer and a classroom can only be assigned to one lecture at a time.

H9: Students must have at least a one-hour break between lectures.

ii. Soft Constraints

These are desirable but can be adjusted if necessary. Examples include:

S1: Students should not have only one course in a day.

S2: The number of students in a class should not exceed the classroom's seating capacity.

S3: Students should not have to attend more than two consecutive lectures in a day.

S4: Courses should not be scheduled in the last time slot of the day.

Since real-world timetabling is complex, soft constraints may sometimes need to be relaxed to generate feasible schedules. FOL is used in representing the timetable constraints in the KB. The FOL modeling of the hard constraints are as follows:
Let Lecture = L_{ij}, Time = $T_{ij,}$ Course = C_{ij}, Student = S_{ij}, Group = G_{nm}, Room = R_{ij}, Semester = S_i =1, 2, Credit Unit = Cu_{ij}

3.2.4 FOL Modeling of Hard Constraints

$\forall L: holds\ (L, not - before\ (t, 8) \bigwedge not\ after\ (T, 6)\ )$

$\forall L: hasmax(L, 2hours)$

$\forall L:$ ¬ held but_ after (L, 6 pm)

$$\forall c\ \Rightarrow (valid\ (C, S1) \bigwedge (C, S2)\ )$$

$$\forall 2Cu\ (2\ credit\ unit\ course, 2Cu) \Rightarrow taught\ (2Cu, one\ day) \vee \text{(2Cu, twice a week)}$$

$\forall L, G\ scheduled\ \Rightarrow (groups, G)$

$\forall R\ (room, R) \Rightarrow host\ (R, lecture)$

$\forall S\ (Students) \bigwedge$ (lecture, L) ⇒ attend (S (1, L))

FOL Modeling of Soft Constraints

For modeling of the soft constraints is as follows;

Let Lecture = L_{ij}, Time = T_{ij}, Course = C_{ij}, Student = S_{ij}, Group = G_{nm}, Room = R_{ij}, Semester = S_i =1, 2, Credit Unit = Cu_{ij}

$\forall S\ (Student, x) \Rightarrow have\ (x, 1\ hour\ break)$

$\forall 3Cu$: $(3\ credit\ unit\ course, 3Cu) \Rightarrow taught\ (3C, 1\ day)$(3Cu, 2day)

$$\exists\ L \forall FL\ Fixed\ (L, FL)$$

3.2.5 Rule Base and Inference Engine

The rule base contains a set of predefined rules that guide the inference engine in decision-making. These rules evaluate various factors, such as student size, course type, venue capacity, and location, to determine the optimality of course allocations.

a) If student size is small, the course is departmental, the venue has limited capacity, and it is located on the main campus, then the allocation is 'highly optimal';
b) If student size is large, the course is departmental, and the venue is small but on the main campus, then the allocation is 'not optimal';
c) If student size is very large under the same conditions, the allocation remains 'not optimal';
d) If student size is small, the course is university-wide, and the venue is small and on the main campus, then the allocation is 'highly optimal';
e) If the student size is large, the course is university-wide, and the venue is on the main campus, the allocation is 'moderately optimal';
f) If student size is very large with the same conditions, the allocation is still 'moderately optimal;

If the course belongs to a faculty and the venue is small, regardless of student size, the allocation is 'not optimal'

a) If the student size is very small, the course is departmental, and the venue has medium capacity on the main campus, then the allocation is 'highly optimal'.

This component of the decision support system synthesizes constraints and rules stored in the KB, allowing logical reasoning to produce an effective timetable.

4.0 Results and Discussion

The proposed system was developed using various tools, including MATLAB R2015a, Microsoft SQL Server Database, and Python. The results are presented in Figures 7-8. Upon launching the timetable system, a splash screen appears, followed by a login form. After a successful login on a form that emphasizes user authentication, requiring fields for entering a username and password, the main form is displayed with a menu for Parameters Entry, ACO Analysis**,** ACO-Fuzzy Analysis, Generated Timetable, and Exit button as shown in Figure 7.

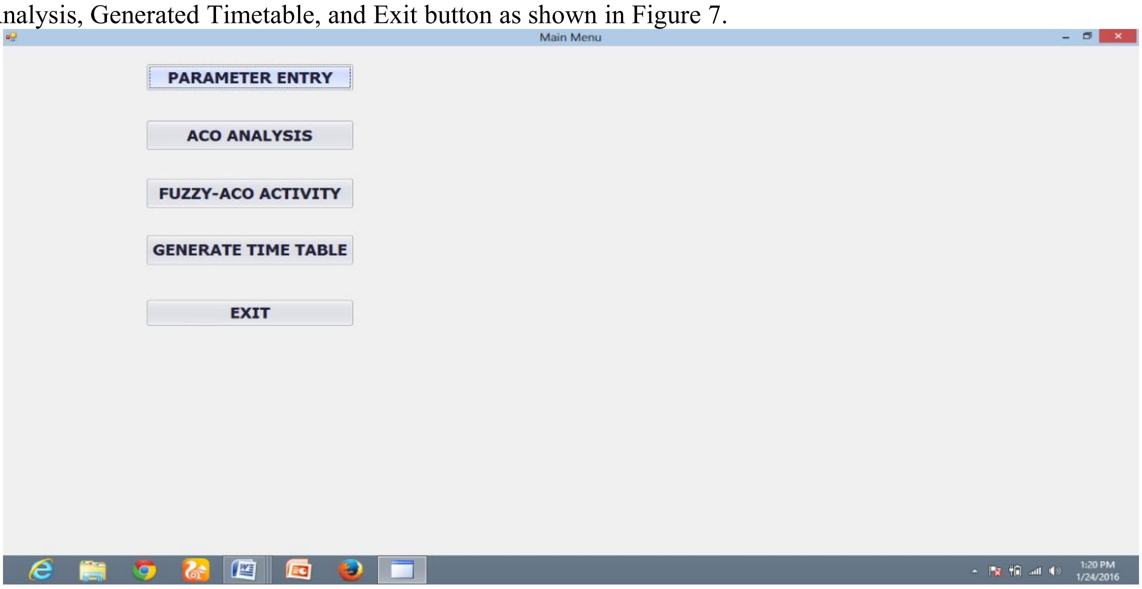

Figure 7: Main Menu of the Proposed System

A click on the ACO analysis command displays an interface that contains timetable date, preview, print, drop all, transDate, Description, Semester, epoch, type, hard and soft constraints, thread hold values (HT and ST), departmental timetable, lecturer timetable, and venue and course capacity. It analyzes the generated timetable and determines the one with the highest optimality. The departmental timetable displays the generated timetable for each of the departments in the faculty resolving all the conflicts in the system is displayed in Figure 8.

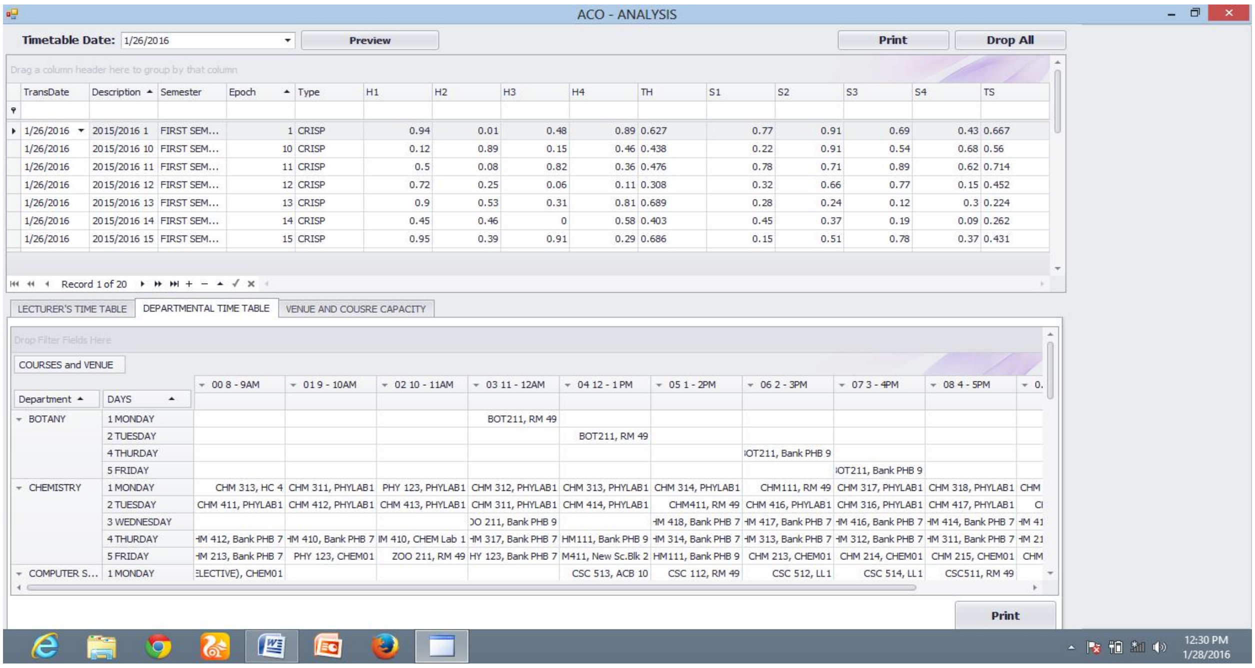

Figure 8: Departmental Timetable

Table 1: Crisp Values Data Analysis for Optimality of Generated Timetable

TransDate	Description	Semester	Epoch	Type	H1	H2	H3	H4	**TH**	S1	S2	S3	S4	**TS**
1/26/2025	2024/2025	1	1	CRISP	0.94	0.01	0.48	0.89	**0.627**	0.77	0.91	0.69	0.43	**0.667**
1/26/2025	2024/2025	1	2	CRISP	0.38	0.78	0.71	0.34	**0.597**	0.16	0.22	0.73	0.67	**0.424**
1/26/2025	2024/2025	1	3	CRISP	0.45	0.84	0.72	0.25	**0.611**	0.89	0.73	0.68	0.02	**0.552**
1/26/2025	2024/2025	1	4	CRISP	0.08	0.19	0.49	0.37	**0.305**	1.00	0.71	0.12	0.51	**0.557**
1/26/2025	2024/2025	1	5	CRISP	0.81	0.67	0.39	0.07	**0.524**	0.61	0.59	0.08	0.59	**0.445**
1/26/2025	2024/2025	1	6	CRISP	0.14	0.74	0.89	0.45	**0.600**	0.94	0.36	0.65	0.01	**0.467**
1/26/2025	2024/2025	1	7	CRISP	0.62	0.55	0.68	0.03	**0.508**	0.82	0.46	0.93	0.95	**0.752**
1/26/2025	2024/2025	1	8	CRISP	0.66	0.07	0.43	0.87	**0.549**	0.91	0.82	0.68	0.63	**0.724**
1/26/2025	2024/2025	1	9	CRISP	0.07	0.12	0.29	0.58	**0.286**	0.61	0.75	0.44	0.44	**0.533**
1/26/2025	2024/2025	1	10	CRISP	0.12	0.89	0.15	0.46	**0.438**	0.22	0.91	0.54	0.68	**0.56**
1/26/2025	2024/2025	1	11	CRISP	0.5	0.08	0.82	0.36	**0.476**	0.78	0.71	0.89	0.62	**0.714**
1/26/2025	2024/2025	1	12	CRISP	0.72	0.25	0.06	0.11	**0.308**	0.32	0.66	0.77	0.15	**0.452**
1/26/2025	2024/2025	**1**	**13**	**CRISP**	**0.9**	**0.93**	**0.71**	**0.81**	0.839	**0.67**	**0.83**	**0.95**	**0.79**	0.810
1/26/2025	2024/2025	1	14	CRISP	0.45	0.46	0	0.58	**0.403**	0.45	0.37	0.19	0.09	**0.262**
1/26/2025	2024/2025	1	15	CRISP	0.95	0.39	0.91	0.29	**0.686**	0.15	0.51	0.78	0.37	**0.431**
1/26/2025	2024/2025	1	16	CRISP	0.42	0.38	0.3	0.58	**0.454**	0.4	0.41	0.17	0.74	**0.41**
1/26/2025	2024/2025	1	17	CRISP	1	0.27	0.81	0.29	**0.641**	0.11	0.39	0.99	0.81	**0.548**

1/26/2025	2024/2025	1	18	CRISP	0.05	0.97	0.49	0.35	**0.503**	0.95	0.8	0.1	0.21	**0.49**
1/26/2025	2024/2025	1	19	CRISP	0.13	0.89	0.05	0.34	**0.381**	0.27	0.88	0.74	0.34	**0.531**
1/26/2025	2024/2025	1	20	CRISP	0.4	0.28	0.05	0.71	**0.389**	0.48	0.68	0.45	0.11	**0.41**

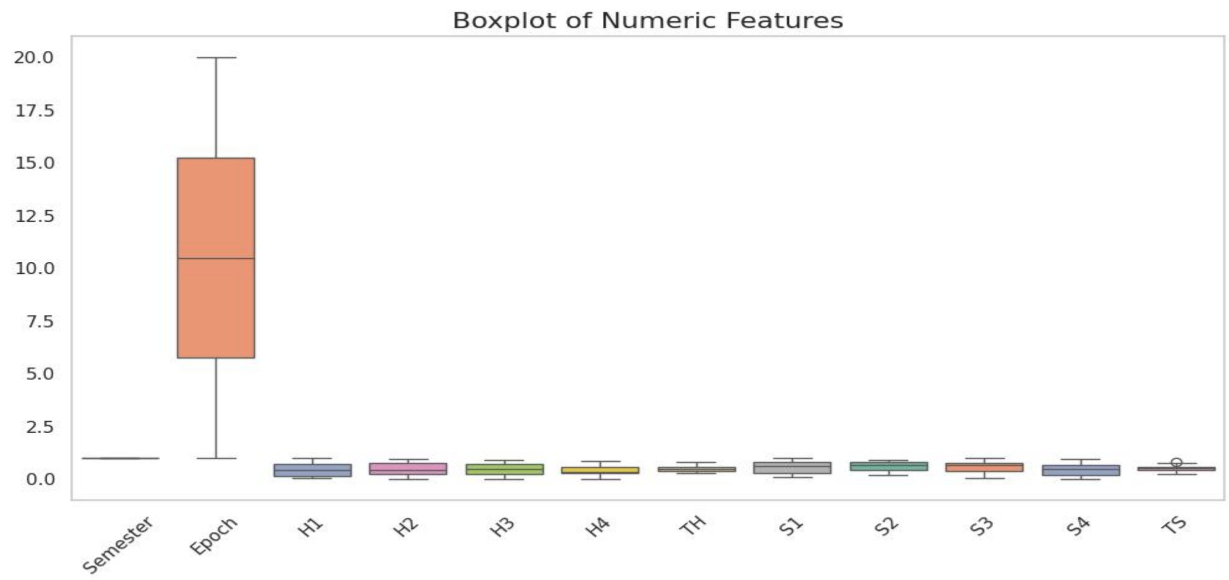

Figure 9: Visualization of Different Constraints on the Scheduled Timetable

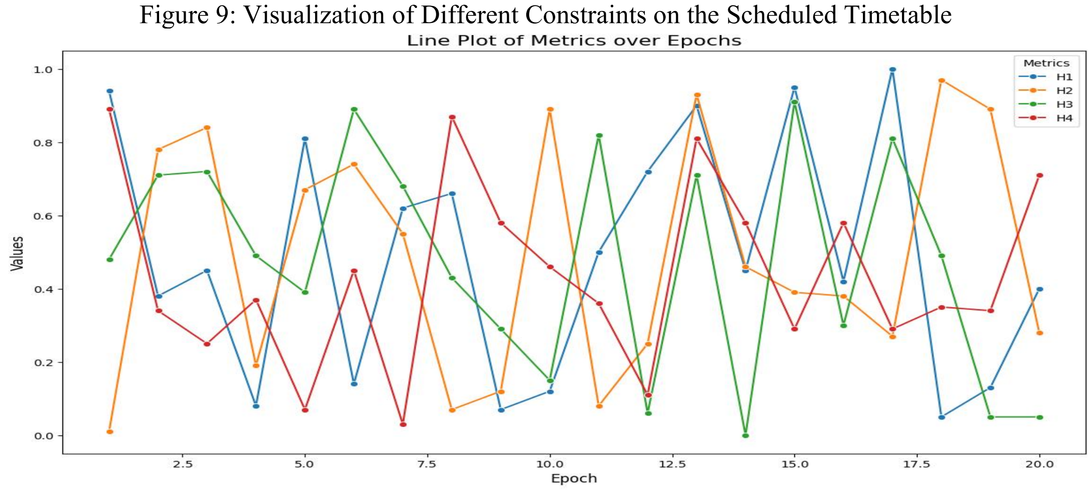

Figure 12: Effect of Hard Constraints (H1-H4) on Timetable Feasibility Over Epochs

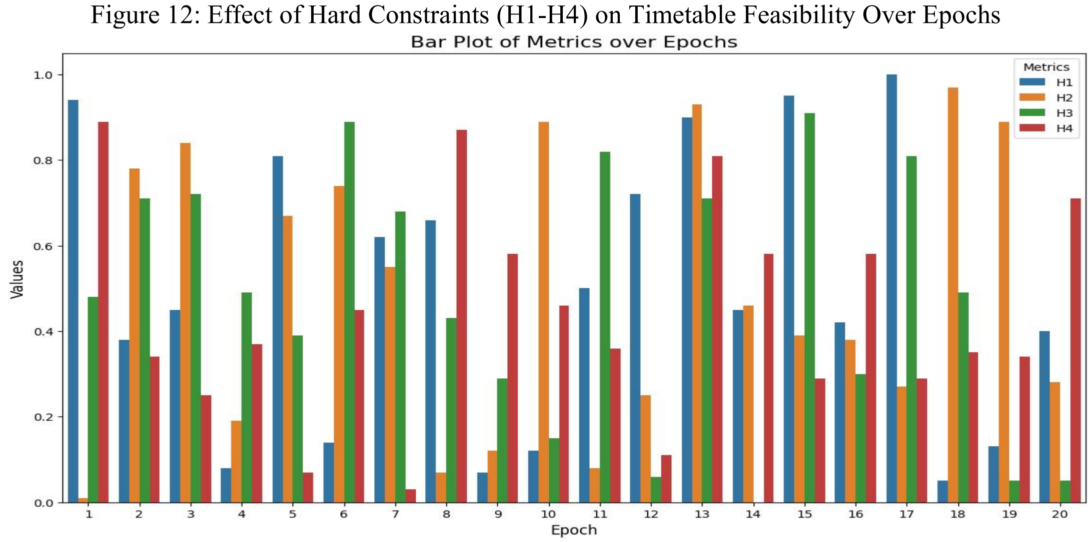

Figure 13: Effect of Hard Constraints (H1-H4) on Timetable Feasibility Over Epochs

Figure 14 shows the relationships between iterations (Epoch), hard constraints (H1-H4), soft constraints (S1-S4), and their satisfaction (TH and TS). The Epoch generatesa slight negative correlation with most constraints.Hard constraints such as H1 and H3 yield a moderate positive correlation of 0.32 while H3 and TH are strongly correlated of 0.71. H4, however, operates more independently, with weaker correlations to the other constraints. For the soft constraints, there is a moderate positive correlation of 0.30 to 0.56, with S3 and S4 showing the strongest relationship of 0.56, indicating a strong relationship between lecturer working hours with preference foracademic staff in managerial positions. The satisfaction of hard constraints (TH) is driven mainly by H1 of 0.60 and H3 of 0.71, while TS (total satisfaction of soft constraints) is closely tied to S3 of 0.56 and S4 of 0.55. The moderate correlation of 0.37 between TH and TS highlights some interdependence between hard and soft constraints. Iteration 13 yields the optimal timetable, achieving the highest TH of 0.839 and TS of 0.810.

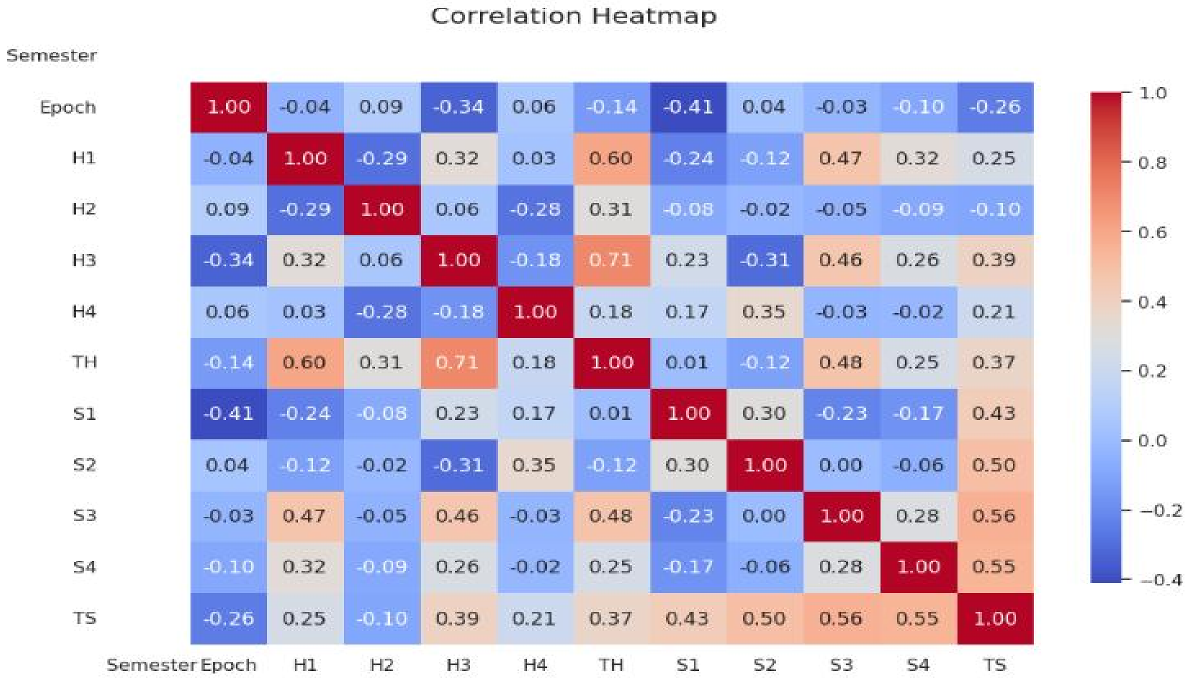

Figure 14: Correlation Analysis of Hard and Soft Constraints

Table 2 shows the result of Fuzzy type analysis on the generated timetable that satisfies the hard and soft constraints which produces five linguistic variables (i.e. high (H), Medium (M), Very High (VH), Low (L), and Very Low (VL). From the analysis, the one with the highest linguistic variable is the one that satisfies the hard and soft constraints to a larger extent and is adopted as the optimal timetable generated. Hence iteration 13 is the one that produced the optimal timetable for the semester. Figure 15 illustrates the relationships between the hard constraints (H1–H4) and their classification based on total hard constraints (TH) values, categorized as Very High (VH), High (H), Medium (M), Low (L), and Very Low (VL). The density distributions along the diagonals highlight the concentration of TH values across different levels of the constraints. A strong clustering of higher TH classifications (H and VH) is evident when H3 and H4 values are above 0.7, emphasizing their significant influence on achieving optimal hard constraint satisfaction. Similarly, higher TH values are observed when H1 and H2 values exceed 0.5, although their impact appears to be more complementary. The relationship between H2 and H3 is notable, with strong clustering of VH and H classifications when both constraints exhibit high values, indicating their interdependence. Additionally, H1 and H4 demonstrate clustering of higher TH values when H4 nears its upper limit (above 0.8), reinforcing its critical role in satisfying hard constraints. The distribution patterns show that H1 values are broadly spread, with peaks near 0.4 and 0.9 corresponding to Medium and High TH categories, while H2 and H3 with sharper peaks around 0.7–0.9, aligning with higher TH classifications. H4, in particular, shows a strong concentration near 0.9 and is predominantly associated with Very High TH values. The H3 and H4 are the most influential in achieving higher TH classifications, while H1 and H2 help in obtaining optimal lecture scheduling. Table 3 presents a comparison of the proposed approach with previous works.

Table 2: Fuzzy Type Analysis of Generated Timetable Schedule

TransDate	Description	Semester	Epoch	Type	H1	H2	H3	H4	**TH**	S1	S2	S3	S4	**TS**
1/26/2025	2024/2025	1	1	FUZZY	0.94	0.01	0.48	0.89	**H**	0.77	0.91	0.69	0.43	**H**
1/26/2025	2024/2025	1	2	FUZZY	0.38	0.78	0.71	0.34	**M**	0.16	0.22	0.73	0.67	**L**
1/26/2025	2024/2025	1	3	FUZZY	0.45	0.84	0.72	0.25	**H**	0.89	0.73	0.68	0.02	**M**
1/26/2025	2024/2025	1	4	FUZZY	0.08	0.19	0.49	0.37	**L**	1.00	0.71	0.12	0.51	**M**
1/26/2025	2024/2025	1	5	FUZZY	0.81	0.67	0.39	0.07	**M**	0.61	0.59	0.08	0.59	**L**
1/26/2025	2024/2025	1	6	FUZZY	0.14	0.74	0.89	0.45	**H**	0.94	0.36	0.65	0.01	**L**
1/26/2025	2024/2025	1	7	FUZZY	0.62	0.55	0.68	0.03	**M**	0.82	0.46	0.93	0.95	**H**
1/26/2025	2024/2025	1	8	FUZZY	0.66	0.07	0.43	0.87	**M**	0.91	0.82	0.68	0.63	**H**
1/26/2025	2024/2025	1	9	FUZZY	0.07	0.12	0.29	0.58	**VL**	0.61	0.75	0.44	0.44	**M**
1/26/2025	2024/2025	1	10	FUZZY	0.12	0.89	0.15	0.46	**L**	0.22	0.91	0.54	0.68	**M**
1/26/2025	2024/2025	1	11	FUZZY	0.5	0.08	0.82	0.36	**L**	0.78	0.71	0.89	0.62	**H**
1/26/2025	2024/2025	1	12	FUZZY	0.72	0.25	0.06	0.11	**L**	0.32	0.66	0.77	0.15	**L**
1/26/2025	2024/2025	**1**	**13**	**FUZZY**	**0.9**	**0.93**	**0.71**	**0.81**	**VH**	**0.67**	**0.83**	**0.95**	**0.79**	**VH**
1/26/2025	2024/2025	1	14	FUZZY	0.45	0.46	0	0.58	**L**	0.45	0.37	0.19	0.09	**VL**
1/26/2025	2024/2025	1	15	FUZZY	0.95	0.39	0.91	0.29	**H**	0.15	0.51	0.78	0.37	**L**
1/26/2025	2024/2025	1	16	FUZZY	0.42	0.38	0.3	0.58	**L**	0.4	0.41	0.17	0.74	**L**
1/26/2025	2024/2025	1	17	FUZZY	1	0.27	0.81	0.29	**H**	0.11	0.39	0.99	0.81	**M**
1/26/2025	2024/2025	1	18	FUZZY	0.05	0.97	0.49	0.35	**M**	0.95	0.8	0.1	0.21	**L**
1/26/2025	2024/2025	1	19	FUZZY	0.13	0.89	0.05	0.34	**L**	0.27	0.88	0.74	0.34	**M**
1/26/2025	2024/2025	1	20	FUZZY	0.4	0.28	0.05	0.71	**L**	0.48	0.68	0.45	0.11	**L**

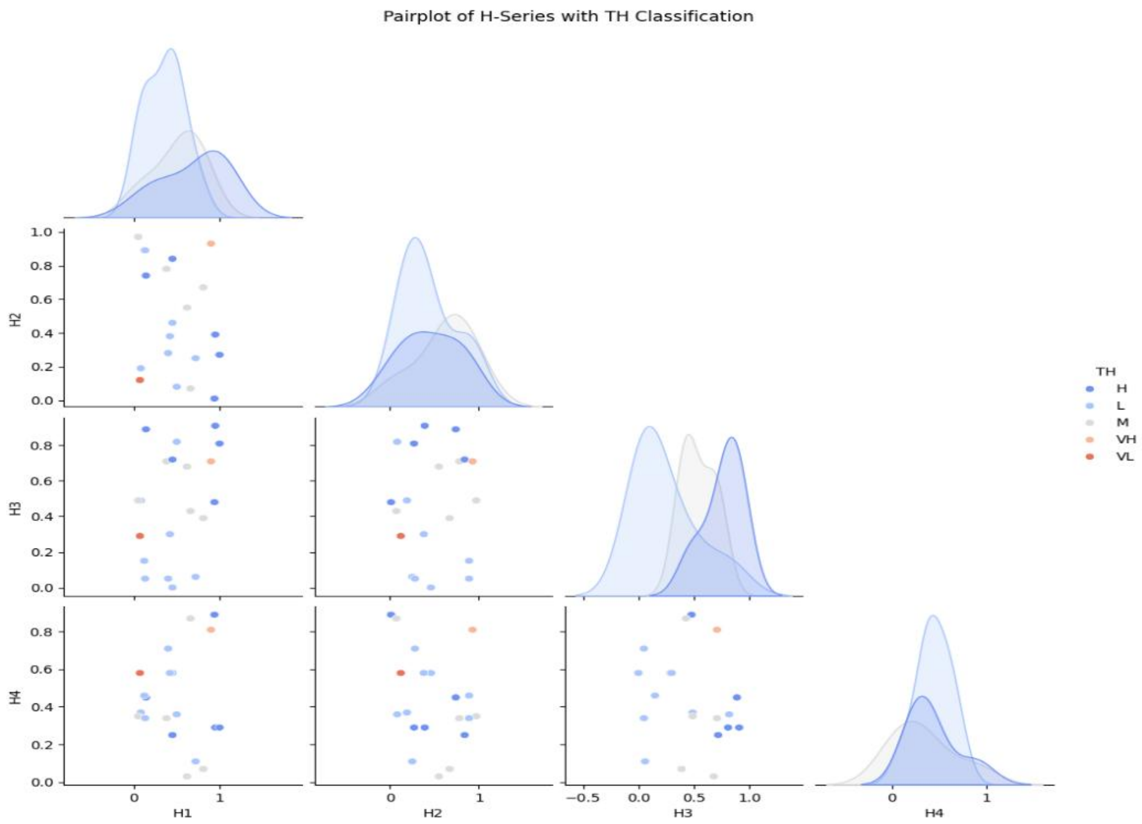

Figure 15: Pair-wise Relationships of Hard Constraints (H1–H4) with TH Classification

The table 3 gives the summary of the comparison of the existing work with the proposed ACO-FL timetabling models.

Table 3. Comparison of the Fuzzy ACO Approach with Existing Works

Reference	Method	Weakness	ACO-FL
[13]	Modified Quicksort	Lacks adaptability to preferences	It incorporates user constraints
[14]	Integer Programming	Poor uncertainty handling	It adapts to dynamic constraints
[15]	Fuzzy Heuristic	Resource omission, venue clashes	It ensures structured allocation of resources
[20]	PSO	Prone to premature convergence	It avoids premature convergence
[21]	Logical Scheduling	Rigid framework	It offers more flexibility

5.0 Conclusion

The study presents a hybrid approach utilizing ACO, FL, and Ontology to tackle the complexities of university lecture timetabling. By integrating CLP, the system incorporates both hard and soft constraints to optimize scheduling, enhance resource utilization, and minimize clashes and delays. The study emphasizes the role of FL in reducing uncertainties within the search space, thereby improving the performance of the ACO algorithm. Scatter plot analysis illustrates the dynamic behavior of hard constraints over time, with H3 and H4 demonstrating the highest stability and impact in achieving optimal satisfaction levels. The hybrid fuzzy-ACO-based system proves more effective in handling classroom constraints compared to existing methods, resulting in more efficient and optimized timetabling. This system design contributes significantly to higher education by promoting better knowledge assimilation, minimizing lecture delays, and enhancing overall academic performance.

Acknowledgment
The authors are grateful to Tertiary Education Trust Fund (TETFund), Nigeria for funding this research.

References

[1] Abdipoor, S., Yaakob, R., Goh, S. L., & Abdullah, S. (2023). Meta-heuristic approaches for the university course timetabling problem. Intelligent Systems with Applications, 19, Article 200253. https://doi.org/10.1016/j.iswa.2023.200253

[2] Asmuni, H. (2008). Fuzzy methodologies for automated university timetabling solution construction and evaluation (Doctoral dissertation). University of Nottingham.

[3] Asmuni, H., Burke, E. K., & Garibaldi, J. M. (2005). Fuzzy multiple heuristic exploratory pattern of the Argentine ant. Journal of Insect Behaviour, 3(2), 456–470.

[4] Bashab, A., et al. (2020). A systematic mapping study on solving university timetabling problems using meta-heuristic algorithms. Neural Computing and Applications, 32(23), 17397–17432.

[5] Bashab, A., et al. (2022). Optimization techniques in university timetabling problem: Constraints, methodologies, benchmarks, and open issues. *Computers, Materials & Continua, 74(3), 6461–6484.

[6] Chen, R.-M., & Shih, H.-F. (2013). Solving university course timetabling problems using constriction particle swarm optimization with local search. Algorithms, 6(2), 227–244. https://doi.org/10.3390/a6020227

[7] Daniel, P. G., Maruf, A. O., & Modi, B. (2018). Paperless master timetable scheduling system. International Journal of Applied Science and Technology, 8(2), 58–68. https://doi.org/10.30845/ijast.v8n2p7

[8] Imianvan, A., & Robinson, S. (2024, April). Optimizing 5G IoT Connectivity for Path Loss Reduction through Ant Colony Optimization and Support Vector Regression. In 2024 International Conference on Science, Engineering and Business for Driving Sustainable Development Goals (SEB4SDG) (pp. 1-10). IEEE.

[9] De Causmaecker, P., Demeester, P., Lu, Y., & Vanden Berghe, G. (2003). Using web standards for timetabling. In E. K. Burke & P. De Causmaecker (Eds.), Practice and theory of automated timetabling IV (Lecture Notes in Computer Science, vol. 2740, pp. 238–257). Springer.

[10] Dorigo, M., & Gambardella, L. M. (1997). Ant colony system: A cooperative learning approach to the traveling salesman problem. IEEE Transactions on Evolutionary Computation, 1(1), 53–66.

[11] Imianvan, A. A., & Robinson, S. A. (2024). Enhancing 5G Internet of things (IoT) connectivity through comprehensive path loss modelling: A systematic review. LAUTECH Journal of Computing and Informatics, 4(2), 31–48.

[12] Dunke, F., & Nickel, S. (2023). A meta-heuristic for customized multi-level multi-criteria university timetabling. Annals of Operations Research, 328, 1313–1348.

[13] Hana, H., & Keith, K. (2016). University course timetabling using constraint logic programming with soft constraints. In Proceedings of the 2016 International Conference on Artificial Intelligence, (pp. 310–328).

[14] Jamil, N. H. A., Rahmin, N. A. A., & Nawawi, A. (2024). Final examination timetable scheduling using integer programming with average time slot gap minimization. Journal of Quality Measurement and Analysis,20(1),145158https://doi.org/10.17576/jqma.2001.2024.1

[15] Kristiansen, S., & Stidsen, T. R. (2013). A comprehensive study of educational timetabling – A survey (DTU Management Engineering Report No. 8.2013).
[16] Pellegrini, P., Favaretto, D., & Moretti, E. (2008). Exploration in stochastic algorithms: An application on ant system. In Nature Inspired Cooperative Strategies for Optimization (NICSO 2008) (pp. xx–xx).
[17] Robinson, S. A., Inyang, U. G., & Imaivan, A. A. (2016). Ant colony optimization framework for university lecture timetable scheduling. Journal of the Nigerian Association of Mathematical Physics, 38(2), 147–154.
[18] Sigl, B., Golub, M., & Mornar, V. (2003). Solving timetable scheduling problem using genetic algorithms. In Proceedings of the 25th International Conference on Information Technology Interfaces (ITI) (pp. 519–524). IEEE.
[19] Solnon, C. (2002). Ants can solve constraint satisfaction problems. IEEE Transactions on Evolutionary Computation, 6(4), 347–357.
[20] Socha, K., Knowles, J., & Sampels, M. (2002). A max-min ant system for the university course timetabling problem. In Ant Algorithms: Third International Workshop, ANTS 2002 (pp. 1–13). Berlin: Springer.
[21] Szénási, S., & Légrádi, G. (2024). Machine learning aided metaheuristics: A comprehensive review of hybrid local search methods. Expert Systems with Applications, 258, Article 125192. https://doi.org/10.1016/j.eswa.2024.125192
[22] Abdullah, S., Turabieh, H., McCollum, B., & McMullan, P. (2010). A hybrid metaheuristic approach to the university course timetabling problem. Journal of Heuristics, 18, 1–23. https://doi.org/10.1007/s10732-010-9154-y
[23] Asuquo, D. E., & Robinson, S. A. (2018). A genetic algorithm-based framework for soft handoff optimization in wireless networks. Studies in Engineering and Technology, 5(1), 1–14. https://doi.org/10.11114/set.v5i1.2590