# University Course and Examination Timetables Scheduling using a Multi-Agents System

Hicham Talib BETAOUAF
Abou bakr Belkaid University
hichem.betaouaf@univ-tlemcen.dz

Hacene Rabab BOUKLI
*Abou bakr Belkaid University*
ra.boukli@gmail.com

***Abstract*— Scheduling is one of the important topics conducted over the years in several studies. One part of the schedule is known as university timetables. The most universities are faced with this NP-hard problem. The timetabling process must be performed on a regular basis for every academic semester and is daunting and time consuming. The aim of this research was to create an integrated method for creating a timetable of university courses in our department using a multi-agent approach and genetic algorithms. This method allows faculty members and department heads to plan, monitor, screen, and edit all events related to their classes. The timetable is created collaboratively and cooperatively by three agents: Capture Agent, Processing Agent, and Distributing Agent. Several simulations were conducted to assess system performance by evaluating the human and material constraint satisfaction rate.**

***Keywords— Course timetable scheduling, Multi-agent, Intelligent agents, Genetic algorithms***

## I. Introduction

Scheduling is a combined and universal real-life problem. Too many researchers have recently tried to resolve the problem using a variety of approaches, including organizational analysis, meta-heuristic methods, intelligent new techniques and multi-agent system-based approach, in order to find an optimal solution [1, 2]. Scheduling is a general problem. The university course timetabling problem (UCTTP) is an NP-hard problem with no optimal solution [3]. Many universities schedule it manually, and the number of students and courses is becoming extremely difficult. The development and administration of schedules necessitates a significant amount of human labor.

### *A. Problem definition*

The UCTTP's objective is to find a way to assign to whole pedagogical activities the predefined time frames and rooms in order to fulfill any constraints in the problem. Students, teachers and classes, including the theoretical and practical rooms, include classroom facilities and equipment. The allocation of resources for the events must meet some hard and soft constraints to improve and boost the quality of viable scheduling [1].

Some basic definitions can be given as following:

- Activity: a planned event for a lesson, a class, and an instructor.
- Activity type: to define the learning form, such as lectures, tutorials, practicals, and so on.
- Timeslot: a period of time during which an activity takes place. (e.g.: Tuesday – from 8:30 a.m. to 10:00 a.m.)
- Classroom: the location where an activity takes place. It could be a lecture hall, a training room, a laboratory, and so on.
- Constraint: is a kind of limitation to be taken into account when planning an activity. There are hard constraints (e.g., a space is not accessible within a certain time slot) and soft constraints (like a preferred timeslot for an activity type). Violating soft constraints does not disrupt the timeline's consistency, but it is preferable that they be met to the greatest extent possible. In contrast, satisfying hard constraints is mandatory when creating the timetable.
- Conflict: can exist between two events if they share the same resource, such as when two activities are planned for the same instructor at the same time.

So, the key aim of UCTTP is to plan all pedagogical activities of a department in time and space in strict compliance with the hard constraints while maintaining the soft constraints as far as possible within a defined time frame.

### *B. Related works*

Many methods are used to study the topic of UCTTP. Firstly, operational research methods include the methodology based on graph coloring theory, which [4] used to solve the first timetabling problem. However, it was not able to solve pre-assigned sessions. Several attempts have been made to overcome the problems faced. One of the most recent is presented by [5]. The classroom scheduling has been carried out using the graph coloring method to reduce the number of penalties and to create high quality schedules in relation to the manual schedules. Reference [6] reduced the cost of finding the lowest number of colors needed for coloring a graph. The computer-based system is based on constraint satisfaction programming (CSP). This method aims to find a consistent set of values in which each value can be assigned to the variables and meet the predefined constraints [7]. In population-based method, firstly, we have a number of people or initial solutions where this set of people is called initial population. A selection mechanism is used for selecting the best solution(s) from the current population at each iteration of population-based metaheuristic approaches. In [8], authors proposed to solve the problems of the UCTTP with a novel genetic algorithm (GA). This approach has the effect of minimizing the number of soft constraints breached, maximizing the usage of available rooms and reducing workload for teachers. Reference [9] have been presented an approach to solve UCTTP problem based on three genetic algorithms. Fuzzy logic is used in this algorithm to determine the number of soft constraint violations.

Multi agent systems have a more general concept and for all types of current systems [10]. UCTTP problem includes a set of courses in fixed timeslots in a circulating week. To generate course timetables, [11] have used distributed multi agent architecture.

Most of the work cited above has proposed systems which automatically handle conflicts based on priority rules. These systems are particularly disadvantageous for some universities such as ours, where the conflict may increase rapidly in relation to the number of teachers within one department. This will induce considerable demand for faculty members and department heads to intervene in the solution of difficult conflicts. The department's dimension is a critical element to be considered in the UCTTP.

Our work aims at establishing a framework to enable our department administration to automatically organize courses by dividing the UCCTP into two separated problems: instructor assignment and activities scheduling.

## II. Materials and Methods

We define the principal elements of our system in this section. Then we define the architecture of the system and at the end of the section we illustrate the whole process which produces the timetable through collaboration and cooperation between the intelligent agents.

### A. System Components

There are two main components in our system:

#### 1) Head of Department Module

This module enables the Head of Department (HoD) to input the following information in our system:

- List of subjects for each degree to be taught in one semester.
- List of courses and groups with the number of students for each year and for each specialty.
- Availability of rooms.
- Optional information such as:
  - Specific days and times for planning the courses. (e.g., a course can be arranged between Monday and Thursday, as Friday is booked only for seminars or extra-faculty activities)
  - Any time or space restriction in respect of a particular activity or category. (e.g., lecture activity must be planned in the auditorium.)

Furthermore, this module can help HoD ask for reports on scheduling and obtain details about conflicts and/or delays in relation to the actions of faculty members.

#### 2) Faculty member Module

With this module, the faculty member can select what to teach from the given list of courses for the following semester. The interface gives him an insight into the list of already chosen courses for better option guidance. The preferred times for the chosen courses may also be specified.

### B. System Architecture

We choose a multi-agent structure consisting of three distinct and collaborative agents inspired by [12] and [13]. (see Fig. 1)

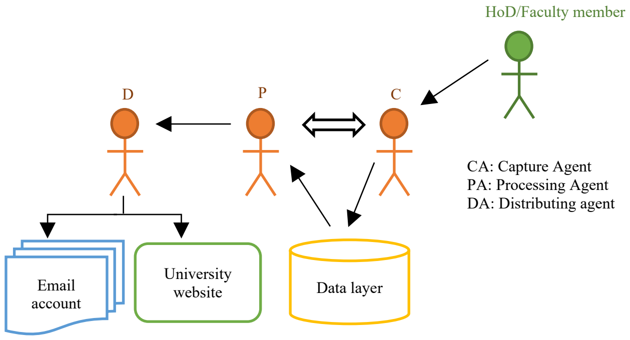


Fig. 1. System Architecture.

#### 1) Capture Agent (CA)

The principal role of the agent is to collect and store data from teachers and/or the HoD in the systems database. The CA may also be instructed by the PA to store or modify data in a database.

#### 2) Processing Agent (PA)

This is our system's most important agent. It continuously monitors the environment, schedules all operations using the CA's data, and monitors conflicts and delays. The PA also produces reports that the CA demands.

#### 3) Distributing Agent (DA)

The PA will submit the valid schedule to the DA if the schedule is validated by HoD. The DA distributes the schedule on the website and/or sends reports, notifications and warnings to faculty members' accounts or inboxes. If conflict or delay has not been sorted, the DA can also inform the HoD or a faculty member, on orders from the PA.

### C. Timetable Development Process

Once the HoD has provided the fundamental information, the PA asks DA to send an email to members of the faculty asking them to choose a course via the faculty member module. The CA stores data and the PA begins its scheduling process.

This agent goes through two phases. Firstly, the instructor is allocated for each task, and then a space and time zone are assigned for all activities. This division between the 2 subprocesses is necessary in order to reduce the complexity of the problem in our system. Experimentally, it is known that after the first step a solution is achieved, the second phase is simpler to overcome.

#### 1) Teachers allocation subprocess

When a teacher selects the courses that are to be provided and the CA stores data in a database, the PA examines whether the other selections in the database are conflicting. The PA will send a message to the DA to alert the faculty members concerned to a meeting to address the issue if a conflict is identified. It's a normal clash. When a conflict continues after the first warning, the PA will send a new message to DA to alert HoD to a "severe clash". The HoD intervenes for a definitive decision and requests DA to inform the interested parties of the decision.

PA is also responsible for monitoring the environment and avoiding delay. Therefore, when the time has elapsed, the PA detects a delay in the choices of a faculty member, the PA requests the DA to email the faculty members concerned to remind them to offer the course (s). Following that date, the PA will send the DA a message alerting the HoD of delay and activities without a faculty assigned.

The HoD calls the involved parties, settles on the remaining activities and confirms the distribution of tasks.

*2) Rooms and timeslots allocation subprocess*

The PA launches the sub-process rooms and time frames once the teacher's allocation has been completed. This procedure also consists of two sub-operations, time (day and hour), and location (classroom) assignment for time and computational saving. The first phase is the most important and difficult. The constraints are much harder and the calculation effort is much greater in this first phase. The room allocation is conceptually the same task as in the first step, following the appropriate timetable, and can be resolved in the same way [14].

The aim of this phase is to plan all activities in time and space in line with any hard constraint and to maximize soft constraints satisfaction.

This part of the process is focused on the evolutionary and genetic algorithms outlined in the next section. Since it is necessary to satisfy any hard constraints, if no solution can be found, PA sends an alarm message to the HoD via DA.

In case a satisfactory solution is found, the resulting schedule is sent to HoD by PA to check and validate it via DA. The HoD is entitled to order the DA to inform the faculty members, depending on the importance and satisfaction level of the soft constraints, if their time constraints cannot be met and offers them the available time ranges.

The PA sends a final timetable to the DA for publication after confirmation.

*3) Timetable Development Algorithms*

The timetable creation process is achieved by cooperation between the three agents in our system, as mentioned in the previous section. In comparison to the two agents of CA and DA who adopt basic data collection and distribution algorithms, the PA employs more sophisticated conflict management and schedule generation algorithms.

In fact, we used an algorithm based on the approach in [14] that uses genetic and evolutionary algorithms to assign resources (temporal and spatial) to activities in our proposed system. The population has a number of individuals or chromosomes, each of which has a single chromosome. Each chromosome has a set of genes (the smallest information carrying unit of a chromosome). There is a gene in the chromosome for each activity. The gene is the planned period of activity. A chromosome is essentially a range of genes, each reflects the beginning of a day and a period of activity.

The fitness function evaluates the optimality of the chromosome. The hard fitness and soft fitness of each chromosome are the number of unmet constraints (hard, respectively soft). The key role of the population is the method of growth. This describes the strategy by mutating, intersecting and simply propagating old chromosomes to create a new generation with the old population.

Inspired by [12] and adapted to our method, the algorithms of the three agents can be seen as follows:

*a) Algorithm of CA:*

1. Gets information relevant to course(s), rooms and timing from HoD.
2. Saves the information in database.
3. Gets data relevant to course(s) offering of a faculty member and GO TO 2.

End

*b) Algorithm of DA:*

1. Gets instructions from PA.
2. Publishes data on web site or/and email to the concerned faculty members.

End

*c) Algorithm of PA:*

1. If date = date to send a message to faculty members to offer course(s)
   i. Prepare a message for offering course(s)
   ii. Call DA to send it to all concerned faculty members
   End If
2. While (course(s) offering = under process or delay! = null or clash! = null) Begin
   2.1. If clash=normal
      i. Compiles a message to have a meeting to resolve the clash
      ii. Call DA to send it to all concerned faculty members
      End If
   2.2. If clash=severe clash
      i. Compiles a message to refer the case to HoD
      ii. Call DA to send it to the HoD
      End If
   2.3. If delay=normal
      i. Compiles a reminder message
      ii. Call DA to send it to the HoD
      End If
   2.4. If delay = severe
      i. Compiles a message to refer the case to HoD
      ii. Call DA to send it to the HoD
      End If

   End While
3. Get all required data of task repartition from database.
4. Call DA to send it to the HoD for verification and validation.
5. If validation != true
   i. Prepare a message as per instructions of HoD
   ii. Call DA to publish the message/send it to all concerned faculty members
   iii. Go to step 2
   End If
6. Launch the time and space allocation process based on genetic and evolutionary algorithms
7. If number of unsatisfied hard constraints != 0

i. Compiles a message to refer the case to HoD
ii. Call DA to send it to the HoD
iii. Go to step 5
End If

8. Get all required data from database.
9. Give it a shape of timetable
10. Call DA to send it to the HoD for verification and validation.
11. If validation = true

Call DA to publish the timetable on the web site/send it to all concerned faculty members

Else Go to step 6

End if

End

### D. *Examination Scheduling*

In the same way as for the teaching schedules, the system also allows the automatic programming of exams. The HoD inserts via the CA, the information and constraints related to the exams (e.g.: the number of groups, the subjects to be examined, the number of tests...etc.).

Teachers can also choose the preferred time slots for proctoring via the CA. The PA then proceeds to schedule the exams following the same algorithms described above. In case of severe conflict, the HoD intervenes to make a decision.

Once the schedules are finalized and validated, the DA posts the schedule on the university website and sends invitations to the proctors containing their proctoring schedule.

### E. *Tools and Technologies*

A spiral model [15] was used to design the web application. Changes and updates have been made in the various project cycles when appropriate. A combination of tools, including Ms Access, was used to create and graphically manage databases with simplicity and efficiency. It also provides interesting tools to create ways to enter data in the database. It may also be possible to generate reports with a few clicks. We have used open-source, Genetic and Evolutionary algorithms tools for saving time and energy, which is called FET [14] for time-frames and space assignment method.

## III. Results and Discussion

The system mentioned in the last section was designed to help our department administer schedules at the start of each academic semester. To facilitate the first test phase, two key modules of the system have been created. We present a few main interfaces of our framework in this section.

The example of the collecting data interface in the HoD interactive module is shown in Fig. 2. The HoD can edit the teachers assigned for each course through this gui. There is also a list of teachers who have picked the subject for the next semester and teacher list who taught the course last year. It also includes more details on the course.

This interface enables the manager to obtain all the required information to make a decision on teacher selection for each activity. It also requires all courses to be validated before rooms and time schedules are assigned.

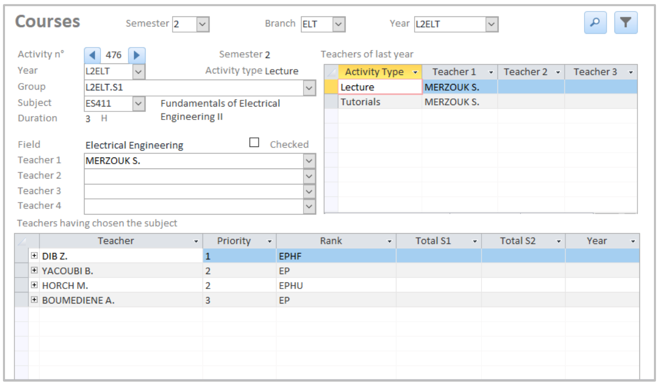


Fig. 2. Example of data capture Interface.

The instructor interface is another essential interface. This gui helps teachers to choose the lessons during the semester. Fig. 3 provides an example of this gui.

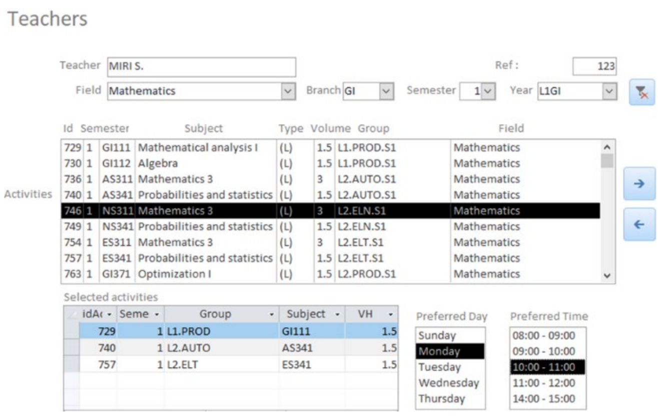


Fig. 3. Example of teacher interface.

This gui allows teachers to pick or exclude those they want to teach from all activities that relate to their field of expertise. It also gives the opportunity to choose desired times and days for any task. The app also helps you to customize the task view thanks to the filters at the top of the interface.

The results for a group of students are shown in Fig. 4. This is a view of the timetable reported on the website of the university. Every box corresponds to a subject code activity, the form of teaching, the teacher(s), the room in which the activity is scheduled.

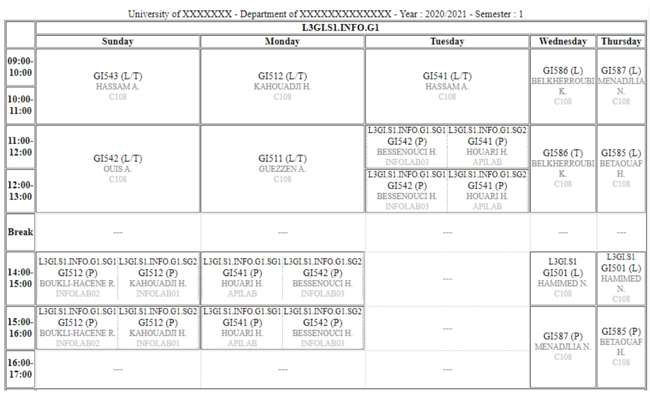


Fig. 4. Example of Timetable result.

University of Tlemcen - Department of Electrical and Electronic Engineering - Year : 2020/2021 - Semester : 1

| | KADDOURI N. | | | | |
|---|---|---|---|---|---|
| | Sunday | Monday | Tuesday | Wednesday | Thursday |
| 09:00-10:00 | --- | L2ELT.S1.ELT.G1<br>ES321 (L/T)<br>B304 | --- | L2AUTO.S1.AUTO.G1<br>AS321 (L/T)<br>B302 | --- |
| 10:00-11:00 | --- | | --- | | --- |
| 11:00-12:00 | --- | L2ELT.S1.ELT.G2<br>ES321 (L/T)<br>B308 | --- | --- | --- |
| 12:00-13:00 | --- | | --- | --- | --- |
| Break | --- | --- | --- | --- | --- |
| 14:00-15:00 | --- | --- | --- | L2AUTO.S1.AUTO.G1.SG1<br>AS343 (P)<br>LFELN1 | L2AUTO.S1.AUTO.G1.SG2<br>AS343 (P)<br>LFELN1 |
| 15:00-16:00 | --- | --- | --- | | |
| 16:00-17:00 | --- | --- | --- | --- | --- |

Fig. 5. Example of Teacher timetable.

The view of a teacher schedule is shown in Fig. 5. It is the individual schedule that each faculty member receives by email. The view is similar to the student view but permits the instructor to only see the sessions he/she participates in. Each box is the activity characterized by the students' group, the subject code, teaching form, and the room to schedule the activity.

In the same way, a room occupational schedule may be shown in order to allow managers to know the time available for a better plan. Fig. 6 demonstrates a timetable for a room. Each box corresponds to an activity of the group, the teacher(s), the code of the subject and the form of teaching concerned.

University of XXXXXXX - Department of XXXXXXXXXXXXX - Year : 2020/2021 - Semester : 1

| | APILAB | | | | |
|---|---|---|---|---|---|
| | Sunday | Monday | Tuesday | Wednesday | Thursday |
| 09:00-10:00 | --- | --- | L3GI.S1.AUTO.G1.SG2<br>HOUARI H.<br>GI541 (P) | --- | --- |
| 10:00-11:00 | --- | --- | | --- | --- |
| 11:00-12:00 | --- | --- | L3GI.S1.INFO.G1.SG2<br>HOUARI H.<br>GI541 (P) | --- | --- |
| 12:00-13:00 | --- | --- | | --- | --- |
| Break | --- | --- | --- | --- | --- |
| 14:00-15:00 | L3GI.S1.AUTO.G1.SG1<br>HOUARI H.<br>GI541 (P) | L3GI.S1.INFO.G1.SG1<br>HOUARI H.<br>GI541 (P) | --- | --- | --- |
| 15:00-16:00 | | | --- | --- | --- |
| 16:00-17:00 | --- | --- | --- | --- | --- |

Fig. 6. Example of rooms timetables.

We only showed the most important interface examples in our application. Other configuration and display interfaces were developed so that schedules could be well handled interactively and information shared.

In the last three years, this method has been evaluated by various branches of departments in three groups (small, medium and large) of various sizes. The results are shown in TABLE I.

TABLE I. RESULTS OF TIMETABLE DEVELOPMENT PROCESS

| Characteristics | Department size | | |
|---|---|---|---|
| | ***Small*** | ***medium*** | ***Large*** |
| Number of students | 100 | 400 | 1000 |
| Number of teachers | 15 | 60 | 150 |
| Number of courses | 45 | 100 | 300 |
| Hard constraints | 125 | 193 | 372 |
| Soft constraints | 65 | 131 | 312 |
| Number of conflicts | 0 | 8 | 27 |
| Satisfaction rate | **100** | **97.53** | **96.05** |

These findings show clearly that the larger the department, the more constraints (Hard/Soft) increase, which makes it much harder and more difficult to solve the issue. We note also that the system will build a schedule that fulfills all constraints for small numbers. Unfortunately, the effects of our method cannot be compared to previous systems in order to determine its efficacy. But we can infer from the results that satisfaction is still quite satisfactory with more than 95 per cent of conflicts relative to the overall number of constraints.

This version was developed for testing the behavior of the system and evaluating its performance. A full version of the platform based on an integrated system and a web interface has just been started.

## IV. CONCLUSION

We selected the UCTTP analysis because it is a complex problem that is faced, practically, by all university managers at the beginning of each year or semester. Many solutions have been proposed in the past but none of them has been able to provide an optimal solution to this problem. Moreover, the proposed solutions are either complex and difficult to apply, or local and limited to certain universities and cannot be used in other universities which may have their own specificities.

We have therefore proposed to design a system that helps the managers of our department to prepare and generate at the beginning of each semester, the schedules of the different trainings provided to the department. The proposed system is based on a multi-agent system, where the three agents (capture agent (CA), processing agent (PA) and distribution agent (DA)) work in collaboration with the aim of capturing data and information related to the courses, dealing with conflicts and delays in the choices of teachers, validating the schedules and disseminating them to students and teachers. This system has been tested during the last 3 years in our department and has given satisfactory results not only in terms of saving time and effort but also in terms of satisfaction of constraints and quality of timetable.

The results obtained in this work have encouraged us to continue this project in order to integrate other modules and functionalities such as the management of examinations and proctoring. Although the interfaces of our system are easy to handle by the various users, we considered it necessary to develop a new interface integrating all the applications for a better ergonomics and a good fluidity of the information.

## References


[1] H. Babaei, J. Karimpour, A. J. C. Hadidi, and I. Engineering, "A survey of approaches for university course timetabling problem," vol. 86, pp. 43-59, 2015.

[2] M. B. Rasheed, N. Javaid, M. S. A. Malik, M. Asif, M. K. Hanif, and M. H. J. I. A. Chaudary, "Intelligent multi-agent based multilayered control system for opportunistic load scheduling in smart buildings," vol. 7, pp. 23990-24006, 2019.

[3] D. J. P. P. E. E. Marx, "Graph colouring problems and their applications in scheduling," vol. 48, no. 1-2, pp. 11-16, 2004.

[4] D. J. Welsh and M. B. J. T. C. J. Powell, "An upper bound for the chromatic number of a graph and its application to timetabling problems," vol. 10, no. 1, pp. 85-86, 1967.

[5] M. Al-Mouhamed and A. Dandashi, "Graph Coloring for class scheduling," in *IEEE/ACS Internation Conference on Computer Systems and Applications (AICCSA)*, 2010, pp. 1-4.

[6] G. Asham, M. M. Soliman, A. J. A. I. T. N. A. Ramadan, and I. Practical Applications, "Trans genetic coloring approach for timetabling problem," pp. 17-25, 2011.

[7] M. R. R. Lewis, "Metaheuristics for university course timetabling," Napier University, 2006.

[8] O. M. K. Alsmadi, S. Za'er, D. I. Abu-Al-Nadi, and A. Algsoon, "A novel genetic algorithm technique for solving university course timetabling problems," in *International Workshop on Systems, Signal Processing and their Applications, WOSSPA*, 2011, pp. 195-198: IEEE.

[9] M. S. Kohshori and M. S. J. I. J. o. C. S. I. Abadeh, "Hybrid genetic algorithms for university course timetabling," vol. 9, no. 2, p. 446, 2012.

[10] S. Srinivasan, J. Singh, and V. J. I. J. o. C. S. I. Kumar, "Multi-agent based decision support system using data mining and case based reasoning," vol. 8, no. 4, p. 340, 2011.

[11] L. Di Gaspero, S. Mizzaro, and A. Schaerf, "A multiagent architecture for distributed course timetabling," in *Proceedings of the 5th International Conference on the Practice and Theory of Automated Timetabling (PATAT-2004)*, 2004, pp. 471-474.

[12] M. Tariq, M. Mirza, R. J. I. J. o. M. S. Akbar, and Engg, "Multi-agent Based University Time Table Scheduling System (MUTSS)," vol. 1, no. 1, pp. 33-39, 2010.

[13] S. M. Cheema, R. Shafiq, S. Saleem, S. Z. H. Shah, and A. Baqir, "Multi-agent Oriented Framework for University Course Timetable Scheduling," in *International Conference on Intelligent Technologies and Applications*, 2019, pp. 791-799: Springer.

[14] L. Lalescu and V. J. l. A. h. l. r. l. f. Dirr, "FET Free Timetabling Software," 2017.

[15] B. Boehm, J. A. Lane, S. Koolmanojwong, and R. Turner, The incremental commitment spiral model: Principles and practices for successful systems and software. Addison-Wesley Professional, 2014.