Project Artificial Intelligence

1)

* Project idea: Solving a Faculty's Timetable Scheduling Problem using Differential Evolution algorithms
* Overview : The University course timetabling problem is known as a NP-hard problem. It is a complex problem wherein the problem size can become huge due to limited resources (e.g. amount of rooms, their capacities and number availability of lecturers) and the requirements for these resources. The university course timetabling problem involves assigning a given number of events to a limited number of timeslots and rooms under a given set of constraints; the objective is to satisfy the hard constraints and minimize the violation of soft constraints. In this paper, a Differential Evolution (DE) algorithm is proposed. DE algorithm relies on the mutation operation to reduce the convergence time while reducing the penalty cost of solution. The proposed algorithm is tested over eleven benchmark datasets (representing one large, five medium and five small problems). Experimental results show that our approach is able to generate competitive results when compared with previous available approaches. Possible extensions upon this simple approach are also discussed.

2) Main Functionalities

* User Enter Data of Courses
* User Enter Data of Rooms
* Select (time, day, room, course ,instructor) Random
* Calculate The Confusion
* Calculate The Fitness
* System generate Timetable
* Evolution Your Timetable
* Print Population
* Print The Fitness
* Print Timetable

Diagram

Description automatically generated

3) Similar Application in The Market

# 1-School timetabling

# Description : Universities and schools aim to provide high quality lesson schedules to their teachers and students. Depending on room capacity and availability, school timetabling decides when and where a lecture takes place. Optimize this planning problem with an AI constraint solver to improve teacher and student satisfaction. Reduce commute and gap hours while adhering to other constraints such as conflicting lessons, teacher availability, room availability and room capacity.

**Link:** [OptaPlanner - School timetabling](https://www.optaplanner.org/learn/useCases/schoolTimetabling.html)

**2-TimeTabler**

**Description:** ***TimeTabler***is a reliable and proven computer timetabling program that can schedule your timetable for you.

**Link:** [About TimeTabler | Timetable Software | TimeTabler](https://www.timetabler.com/scheduling/)

**3-Wise Timetable**

**Features:**

* Advanced assistance tools for interactive timetabling.
* Wise Timetable can be used to produce unlimited number of timetables.
* Comprehensive reports and one click notifying by email.
* Web publishing implemented for windows and unix web servers.
* Lecture execution is not restricted by time duration or weekly cycles.
* Your timetable can be spread over the whole year each week differently.
* Solely scheduling software capable of student level timetabling.
* Wise Timetable university version complies to Bologne convention.
* Interfacing to the other applications in your organization has never been easier.
* Compatible to all popular database systems (e.g. Oracle, MySql, MSSql, ...).
* Compatible with all Windows platforms.
* If application is not already localized in your language on your request we will gladly do so.
* Our users get guaranteed support promptly and professionally without time limits.

**Link:** [Wise Timetable: Complete scheduling solution (wtimetable.com)](https://wtimetable.com/)

**4-AltexSoft**

**Description:** Scheduling is a crucial, yet highly complicated process in any industry. There’s a huge number of varied factors that have to be taken into consideration when creating a schedule and then there are unexpected disruptions that require timely reaction. Doing it manually is time-consuming and way too ineffective. So, numerous techniques, including mathematical optimization, constraint programming, and machine learning (ML), are used to address this issue.

In this article, we’ll talk about schedule optimization, describe the most typical technology approaches to its automation, and look at some use cases across industries

**Link:** [Schedule Optimization Approaches and Use Cases | AltexSoft](https://www.altexsoft.com/blog/schedule-optimization/)

4) A Literature Review Of Academic (papers) And Details The Algorithm Used :

**Differential Evolution :**

DE is a basic algorithm of the population that employed crossover, mutation and selection operators as in genetic algorithms. The main difference in obtaining better solutions is that genetic algorithms rely on the crossover operation, while in the DE algorithm it is based on the mutation operation.

DE estimates the difference of the parameter vectors to explore the objective function solution space.

DE perturbs current generation vectors with the scaled difference of two randomly selected population vectors. To produce a trial vector in its simplest form DE adds the scaled, random vector difference to a third randomly selected population vector. In the selection stage, the trial vector competes against the population vector of the same index. Once the last trial vector has been tested the survivors of all the pairwise competitions become permanent for the next generation in the evolutionary cycle.

DE is a simple evolutionary algorithm. It works through a simple cycle which

consists of Initialization (initialize population with random numbers) will be done based on the number of variables in the problem; mutation (calculate difference vector) will be done usually

following the scheme DE/rand/1. Crossover/recombination (multipoint crossover) is the feature in

DE unlike GA, where single-point crossover is preferred. Selection (elitist replacement) is the choice

Diagram

Description automatically generated

# Diagram Description automatically generated

Diagram

Description automatically generated

Timetable scheduling using differential evolution problem

Classes are defined in the following fashion:

- Subject that is being taught

- Type of class (lectures or practical)

- The professor conducting the class

- All of the student groups listening to the class together at the same time

- Classroom type allowed (i.e. some classrooms require students to have access to computers,

some do not)

- Length (1 to 4 hours) we assume that all classrooms are of the same size and have the required capacity. Valid hours for holding classes are from 9am to 9pm. The task at hand is to assign a time and classroom for each of the classes given in the mentioned format.

Constraints

1. Resources must not overlap in time:

o No professor can hold two classes at the same time

o No student group can attend two classes at the same time

o No classroom can host to classes at the same time

**Note:** the term "same time" does not only mean the starting time of a class, but what also must be taken into account is the length of a class. If a resources is taken at time T1 and the class lasts for t, then the resource can be taken again only at time T2.

2. The class must be held in one of the allowed classrooms for it (most practical require computers)

3. If a subject has multiple forms of classes, such as lectures, practical and labs, the preferred order is: lecture, practical, lab.

Constraints 1 and 2 must be met (hard constraints), while constraint 3 is a "soft" constraint and can

be violated. The main objective is to minimise the violation of the soft constraints in a feasible

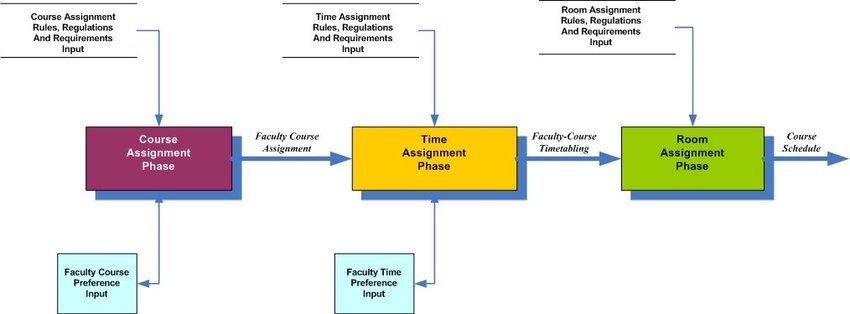
solution that later represents the quality of the obtained solution.

Additional possible criteria for the assessment of the solution:

- Minimize total idleness for each group (pauses between classes)

- Minimize total idleness for each professor (pauses between classes)

- Provide one free hour a week with no classes, for a professors union meeting

- Minimize daily load for professors and groups (less than 6 hours of class a day) A picture containing text, receipt, screenshot

Description automatically generated

Diagram

Description automatically generated

Figure 1 Flow chart of differential evolution algorithms

5) Development Platform

Tools: Pycharm

Programming language: Python

Libraries: random , prettytable, NumPy, tkinder

* Resources:

Link:[**https://www.academia.edu/15338525/A\_Differential\_Evolution\_Algorithm\_for\_the\_University\_course\_timetabling\_problem**](https://www.academia.edu/15338525/A_Differential_Evolution_Algorithm_for_the_University_course_timetabling_problem)

Link: <https://ieeexplore.ieee.org/abstract/document/6329805>

Link:[**https://www.academia.edu/15338525/A\_Differential\_Evolution\_Algorithm\_for\_the\_Un iversity\_course\_timetabling\_problem**](https://www.academia.edu/15338525/A_Differential_Evolution_Algorithm_for_the_Un%20%20%20iversity_course_timetabling_problem)

Link: <https://ieeexplore.ieee.org/abstract/document/6329805>

Link: <https://ieeexplore.ieee.org/abstract/document/5397988>

Link:<https://www.researchgate.net/publication/261487808_Solving_department's_course-scheduling_Problem_using_differential_evolution>

Link:<https://scholar.google.com.eg/scholar?q=plots+differential+evolution&hl=ar&as_sdt=0&as_vis=1&oi=scholart>

Link: <https://www.researchgate.net/figure/Flow-chart-of-differential-evolution-algorithms_fig1_310811940>

Link: <https://www.researchgate.net/figure/Block-diagram-of-Population-Rescaled-Differential-Evolution-with-Weighted-Boosting_fig1_336488064>

Link:

Link For Sheared Folder: <https://github.com/Ayate10671/Ai_project_Time_Table->