{ } 2024 { }

Solving The Examination Timetabling Problem: A Nash Equilibrium Approach With Genetic Algorithm And Tabu Search Comparisons

Instructor: Lương Trung Kiên

Presented by: Nguyễn Duy Tùng

Bùi Văn Quyến

Bùi Tuấn Đạt

{ } AIP490_G3 { }

Table of contents

01 Introduction

02 Methodology

03 Experiment

04 Result

05 Conclusion

06 Demo

1

Introduction

Problem and Motivation

- ❖ The Examination Timetabling Problem (ETP) is a critical scheduling issue faced by educational institutions globally.
- ❖ Challenges include handling the increasing number of courses and students with limited resources.
- ❖ Traditional manual timetabling methods are labor-intensive and error-prone.

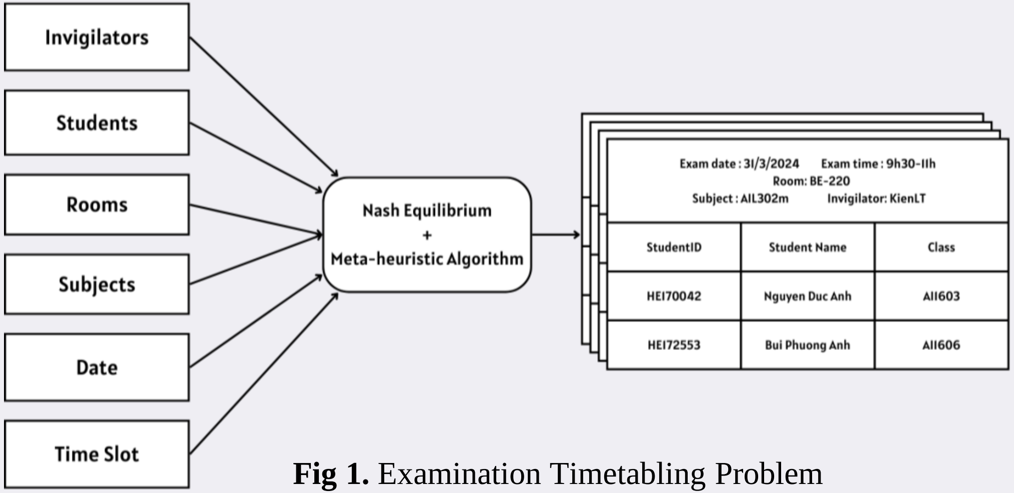


Fig 1. Examination Timetabling Problem

Three main group of stakeholders:

1. The Academic Department wants an even distribution of rooms between timeslots
2. Students wish their exams to be evenly distributed throughout the entire period.
3. Invigilators wants compress their invigilation schedule into as few days as possible while keeping the number of timeslots as close to the required number as they can.

Problem and Motivation

- Nash Equilibrium is a concept in game theory, in which the problem to be solved is described as a game and involves the participation of players.
- Each player has their strategy to receive certain rewards and penalties.
- Nash Equilibrium point describes a situation where no participant can gain their expected payoff by unilaterally changing their strategy if the strategies of the others remain unchanged.
- Utilizing Nash Equilibrium Approach to address fairness between students, invigilators and academic department in exam scheduling.

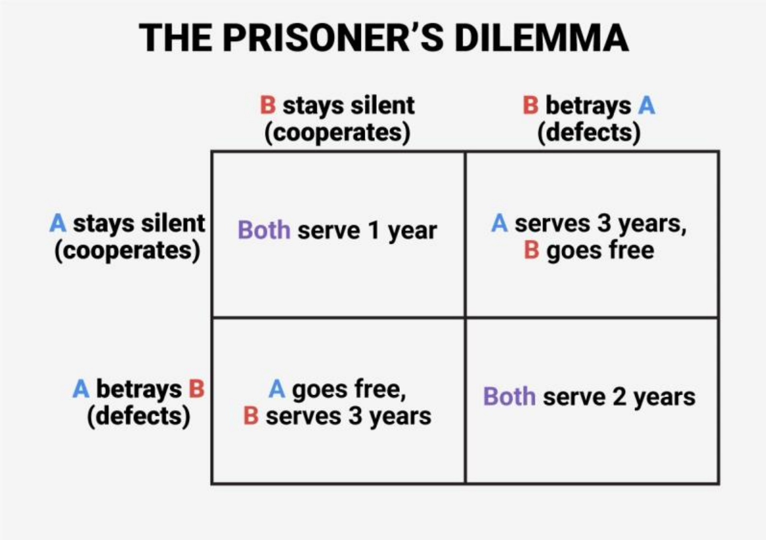


Fig 2. The Prisoner's Dilemma

Problem and Motivation

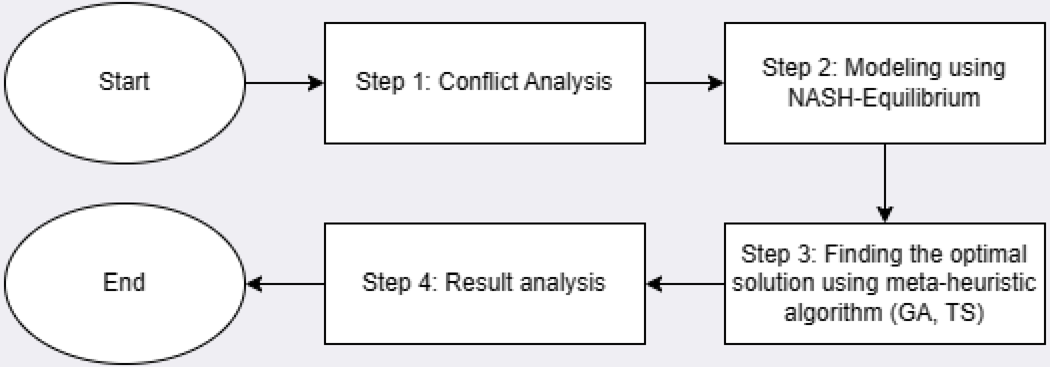


Fig 3. The proposed process of ETP

Related Work

Paper	Author	Year
University Examination Timetable Scheduling Using Constructive Heuristic Compared to Genetic Algorithm	Salem	2023
Examinations Timetabling System Based on A Genetic Algorithm	Mohammed et al.	2022
Hybrid intelligent water Drops algorithm for examination timetabling problem	Aldeeb et al.	2022
Genetic Algorithm for Solving Multi-Objective Optimization in Examination Timetabling Problem	Ngo et al.	2021
A fast simulated annealing algorithm for the examination timetabling problem	Leite et al.	2019
Solving examination timetabling problem in UniSZA using ant colony optimization	Khair et al.	2018
Graph coloring heuristics for solving examination timetabling problem at Universiti Utara Malaysia	Abdul-Rahman et al.	2014
A tabu search hyper-heuristic approach to the examination timetabling problem at the MARA university of technology	Kendall et al.	2005

Related Work

Nash Equilibrium Approach Application

Teaching Assignment Based on NASH Equilibrium and Genetic Algorithm	Chi et al. 2023
Nash-equilibrium algorithm and incentive protocol for a decentralized decision and scheduling problem in sustainable electroplating plants	Yan et al. 2021
On Distributed Scheduling of Flexible Demand and Nash Equilibria in the Electricity Market	Paola et al. 2018
Efficient nash equilibrium resource allocation based on game theory mechanism in cloud computing by using auction	Nezarat et al. 2015
A Nash-equilibrium based algorithm for scheduling jobs on a grid cluster	Spata 2007

2

Methodology

Input Data

- N_M: The number of Students
- N_I: The number of Invigilators
- N_R: The number of Rooms
- N_S: The number of Subjects
- N_D: The number of Days
- α: Maximum number of students in each room.
- β: The number of slots a day

- $N_T = \beta * N_D$: denotes number of timeslots.
- Matrix A = $\{a_{m,s}, 1 \le m \le N_M, 1 \le s \le N_S\}$, $a_{m,s}$ = 1 if student m will take subject s, otherwise 0.
- Matrix C = $\{c_{s,i}, 1 \le s \le N_S, 1 \le i \le N_I\}$, $c_{s,i}$ = 1 if subject s can be supervised by invigilator i.
- Vector L = $\{l_s, 1 \le s \le N_S\}$, l_s denotes the length of subject s.
- Vector Q = $\{q_i, 1 \le i \le N_I\}$, $q_i \in \mathrm{N}$, denotes number of slots invigilator i need to supervise.

Some denotations can be inferred from input data:

- Vector E = $\{e_m \mid e_m = \sum_{s=1}^{N_S} a_{m,s}, 1 \le m \le N_M\}$, e_m denotes the number of subjects that each student need to take part in.
- Vector F = $\{f_s \mid f_s = \sum_{m=1}^{N_M} a_{m,s}, 1 \le s \le N_S\}$, f_s denotes number of students that take subject s.
- Vector G = $\left\lceil \frac{F}{\alpha} \right\rceil = \{g_s, 1 \le s \le N_S\}$, g_s denotes number of rooms needed to hold for subject s.

Decision Variable

The decision variable is a matrix D = $\{d_{s,t,i}, 1 \le s \le N_S, 1 \le t \le N_T, 1 \le i \le N_I\}$, Where $d_{s,t,i} = 1$ if subject s is held at slot t and supervised by invigilator i, $d_{s,t,i} = 0$ if subject s is not held at slot t and not supervised by invigilator i.

Some denotations can be inferred from decision variable:

- Matrix H = $\{h_{s,t} \mid h_{s,t} = \begin{cases} 1 \text{ if } \sum_{i=1}^{N_i} d_{s,i,t} > 0 \\ 0 \quad \text{otherwise} \end{cases}, 1 \le s \le N_S, 1 \le t \le N_T\}$, $h_{s,t}$ denotes if subject s is held at timeslot t.
- Vector X = $\{x_s \mid x_s = \begin{cases} t \text{ if } h_{s,t-1} = 0 \text{ and } h_{s,t} = 1 \\ 0 \quad \text{otherwise} \end{cases}, 1 \le s \le N_S\}$, x_s denotes the slot start of subject s.
- Vector Y = X + L – 1 = $\{y_s, 1 \le s \le N_s\}$, $y_s \epsilon [1, N_T]$ denotes the slot end of subject s.
- Matrix Z = A * X = $\{z_{m,s} \mid z_{m,s} \in [1, N_T], 1 \le m \le N_M, 1 \le s \le N_s\}$, $z_{m,s}$ denotes the slot start for subject s of student m.
- Matrix K = $\{k_{i,d} \mid k_{i,d} = \begin{cases} 1 \text{ if } \sum_{t=\beta(d-1)+1}^{\beta d} \sum_{s=1}^{N_s} d_{s,t,i} > 0 \\ 0 \quad \text{otherwise} \end{cases}, 1 \le i \le N_I, 1 \le d \le N_D\}$, $k_{i,d}$ denotes the number of slots invigilator i supervises in day d.

Nash Equilibrium Approach

We model the examination timetabling problem into a game with the participation of 3 groups of players:

- Special player P_0 represents the interest of academic department.
- Player B_m $(1 \leq \mathrm{m} \leq N_M)$, represents the m-th student.
- Player V_i $(1 \leq \mathrm{i} \leq N_I)$, represents the i-th invigilator.

Nash Equilibrium Approach

Special player P_0 represents the interest of academic department.

The benefit of player P_0 is the uniformity in the number of rooms between each time slot. This benefit is represented by the following payoff function, denoted as $Payoff_{P0}$.

Let μ is the mean of number of rooms in each slot:

$$\mu = \frac{\sum_{t=1}^{N_T} \sum_{s=1}^{N_S} \sum_{i=1}^{N_I} d_{s,t,i}}{N_T} \qquad [1]$$

Then, we utilize the standard deviation equation to derive the payoff function of player P_0:

$$Payoff_{P0} = w_1 * \sqrt{\frac{\sum_{t=1}^{N_T} \left(\sum_{s=1}^{N_S} \sum_{i=1}^{N_I} d_{s,t,i} - \mu\right)^2}{N_T - 1}} \qquad [2]$$

Nash Equilibrium Approach

Player $\boldsymbol{B_m}$ (1 ≤ m ≤ $\mathbf{N_M}$), represents the m-th student.

B = $\{b_1, b_2, \ldots, b_{N_M}\}$ is a set of players who are students. Each student wishes that his/her exam subjects be distributed evenly throughout the entire period.

Let $\frac{N_T}{e_{b_m}}$ be the time slot gap student b_m wants between two exams and let U = sort(Z) in descending order be the slot start for each exam, from the last exam to the first one that each student joined. We have payoff of a student b_m:

$$Payoff_{b_m} = \ln\left(\frac{1}{e_{b_m} - 1} * \sum_{i=1}^{e_{b_m}-1} e^{\left|u_i - u_{i+1} - \frac{N_T}{e_{b_m}}\right|}\right) \qquad [3]$$

Payoff of all players $\boldsymbol{B_m}$:

$$Payoff_{AllB} = w_2 * \frac{1}{N'_M} * \sum_{m=1}^{N'_M} Payoff_{b_m} \qquad [4]$$

N'_M: The number of students who have more than 1 exam.

Nash Equilibrium Approach

Player V_i ($1 \le i \le N_I$), represents the i-th invigilator.

V = $\{v_1, v_2, \ldots, v_{N_I}\}$ is a set of players who are invigilators. Each invigilator v_i seeks to compress their invigilation schedule into as few days as possible:

$$g_{v_i} = \sum_{d=1}^{N_D} k_{v_i,d} \qquad [5]$$

The actual number of exam slots for each invigilator v_i should be as similar as possible to the number of slots required.

$$h_{v_i} = \left| \sum_{s=1}^{N_s} \sum_{t=1}^{N_t} d_{s,t,v_i} - q_{v_i} \right| \qquad [6]$$

We deduce the payoff function of a player v_i:

$$Payoff_{v_i} = w_4 * g_{v_i} + w_5 * \ h_{v_i} \qquad [7]$$

Payoff of all players v_i

$$Payoff_{AllV} = w_3 * \frac{1}{N_I} * \sum_{i=1}^{N_I} Payoff_{v_i} = w_3 * \frac{1}{N_I} * \sum_{i=1}^{N_I} \left(w_4 * g_{v_i} + w_5 * \ h_{v_i} \right) \qquad [8]$$

Nash Equilibrium Approach

The Fitness Function:

$$F = Payoff_{P0} + Payoff_{AllB} + Payoff_{AllV} \qquad [9]$$

$$= w_1 * \sqrt{\frac{\sum_{t=1}^{N_T}\left(\sum_{s=1}^{N_S}\sum_{i=1}^{N_I} d_{s,t,i} - \mu\right)^2}{N_T - 1}} + w_2 * \frac{1}{N'_M} * \sum_{m=1}^{N'_M}\left(\ln\left(\frac{1}{e_{b_m} - 1} * \sum_{i=1}^{e_{b_m}-1} e^{\left|u_i - u_{i+1} - \frac{N_T}{e_{b_m}}\right|}\right)\right)$$

$$+ w_3 * \frac{1}{N_I} * \sum_{i=1}^{N_I}\left(w_4 * \sum_{d=1}^{N_D} k_{v_i,d} + w_5 * \left|\sum_{s=1}^{N_s}\sum_{t=1}^{N_t} d_{s,t,v_i} - q_{v_i}\right|\right) \qquad [10]$$

- H. Nikaido and K. Isoda presented a concept of Nash equilibrium for non-cooperative games in 1955 and demonstrated that this equilibrium point occurs when the overall payoff value for all players is reaching the extreme point.

Hard Constraints

With students:

- No student should be required to sit more than one examination simultaneously.

$$\sum_{s=1}^{N_S} h_{s,t} * a_{m,s} \leq 1 \qquad \forall m \in [1, N_M], t \in [1, N_T] \qquad [11]$$

With invigilators:

- The number of supervisors must be equal to the number of rooms needed to organize each subject at each slot.

$$\sum_{i=1}^{N_I} d_{s,t,i} = g_s \;\; or\; 0 \qquad \forall s \in [1, N_S], t \in [1, N_t] \qquad [12]$$

- No invigilator should be required to sit more than one examinations simultaneously.

$$\sum_{s=1}^{N_S} d_{s,t,i} \leq 1 \qquad \forall i \in [1, N_i], t \in [1, N_t] \qquad [13]$$

- Examiners only supervise subjects they are capable of supervising.

$$\sum_{i=1}^{N_I} \sum_{t=1}^{N_T} d_{s,t,i} * c_{s,i} = g_s * l_s \qquad \forall s \in [1, N_s] \qquad [14]$$

Hard Constraints

With academic department:

- The number of rooms used in a slot must not exceed the allowed.

$$\sum_{s=1}^{N_S}\sum_{i=1}^{N_I} d_{s,t,i} \le N_R \qquad \forall t \in [1, N_t] \qquad [15]$$

- Each subject must be held once.

$$\sum_{t=1}^{N_T} h_{s,t} = \ l_s \qquad \forall s \in [1, N_s] \qquad [16]$$

- For each exam, the assigned invigilator needs to monitor all consecutive slots in which that exam takes place.

$$\sum_{t=x_s}^{y_s} d_{s,i,t} = l_s \ or\ 0 \qquad \forall s \in [1, N_s], i \in [1, N_i] \qquad [17]$$

- For exams with L time slot, it cannot end later than the end of morning or afternoon.

$$\left\lfloor \frac{X}{\beta/2} \right\rfloor = \left\lfloor \frac{Y}{\beta/2} \right\rfloor \qquad [18]$$

Algorithms - Genetic Algorithm

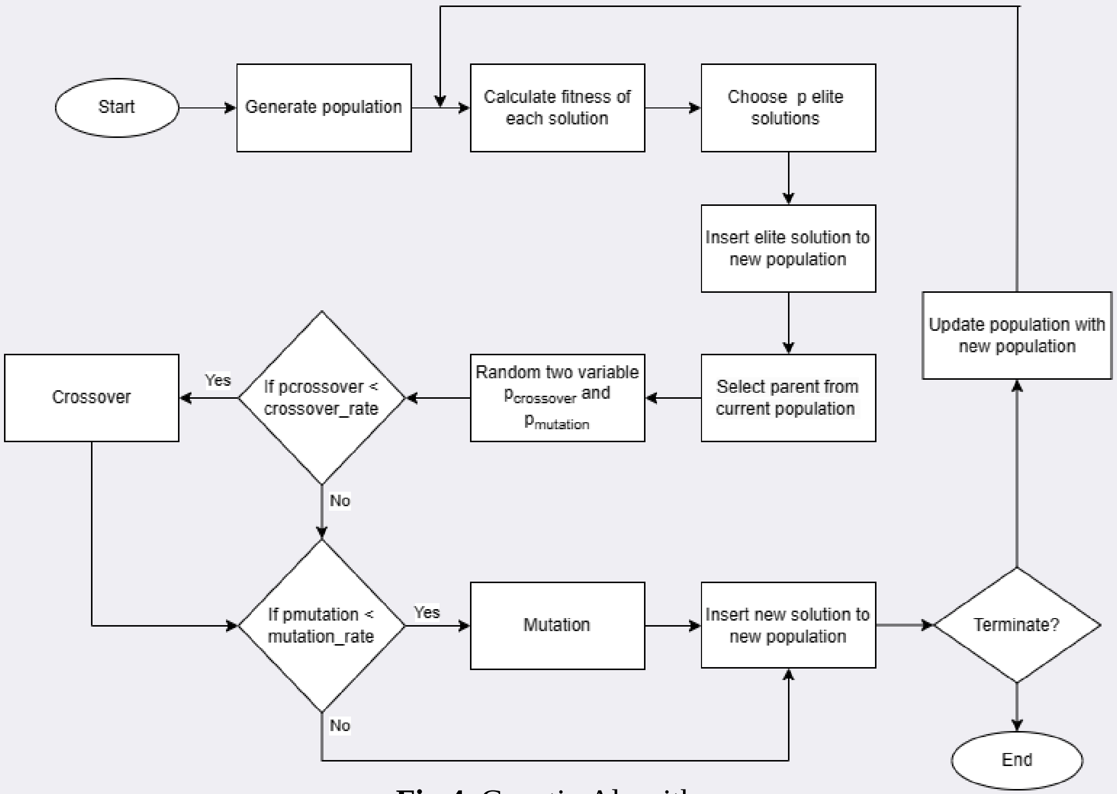


Fig 4. Genetic Algorithm

Algorithms - Genetic Algorithm

{ }

A solution is a decision variable D

A population is a list of solution

Calculate fitness

For every solution s in population has the fitness as:

$$\text{s.fitness} = \begin{cases} F(s) & if\ C(s) = 0 \\ F(s) * 100 & if\ C(s) = 1 \end{cases}$$

Where C(s) = 1 if s violates any constraints and 0 otherwise

Elite solutions is top solution with best fitness in a population

Crossover

Choose a random subject.

Create p_{child} by swapping slot and invigilators of that subject in $p_{father}\ and\ p_{mother}$

Mutation

Choose a random subject from solution s

Replace the timeslot of chosen subject in p_{father} with another random timeslot

For chosen subject:

Choose a random invigilator who supervises the subject.

Replace the chosen invigilator with another randomly chosen invigilator.

{ }

Algorithms - Tabu Search

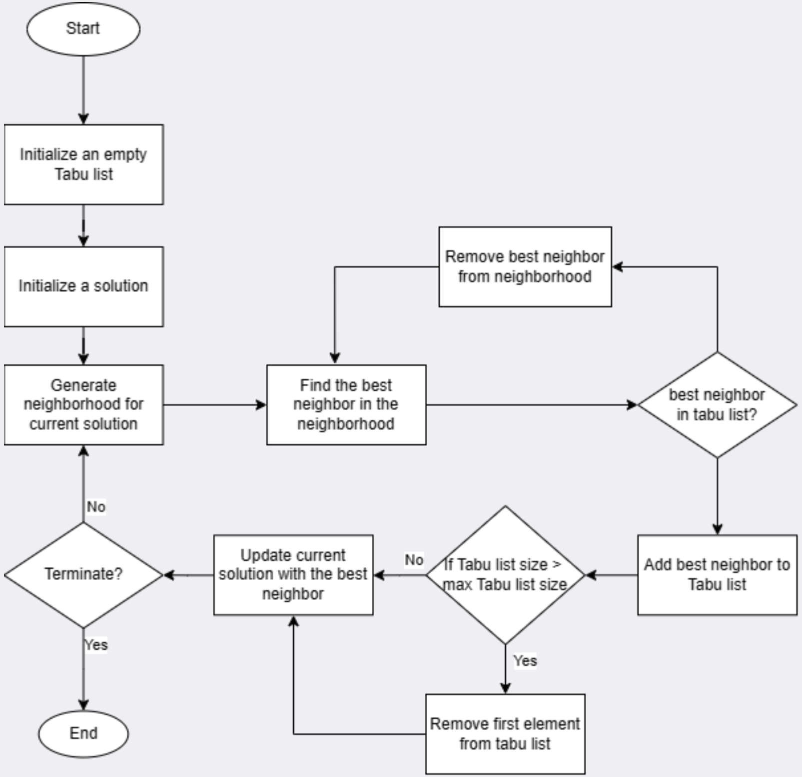


Fig 5. Tabu Search Algorithm

Algorithms - Tabu Search

Generate neighborhood for a solution:

Choose two random subjects (s_1, s_2).

Swap timeslot of $s_1$and s_2

Change slot of s_1 to a new random slot.

Choose another two random subjects (s_1', s_2')

Swap invigilators of s_1' and s_2'

Choose a random invigilator that supervise subject s_1'

Replace the chosen invigilator with a new random one.

Add new solution to the neighborhood.

Best neighbor is the solution with the best fitness in the neighborhood

Calculate fitness

For every solution s in population has the fitness as:

$$\text{s.fitness} = \begin{cases} F(s) & if\ C(s) = 0 \\ F(s) * 100 & if\ C(s) = 1 \end{cases}$$

Where C(s) = 1 if s violates any constraints and 0 otherwise

3

Experiment

Dataset

FPT 2023 Spring Final Examination

- The number of Students: N_M = 11,509
- The number of Invigilators: N_I = 275
- The number of Rooms: N_R = 100
- The number of Subjects: N_S = 156
- The number of Days: N_D = 9
- Maximum number of students in each room: α = 22
- The number of slots a day: β = 6

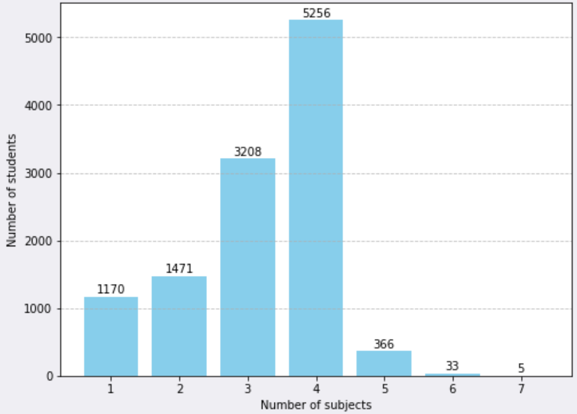


Fig 6. Distribution of Students by Number of Subjects

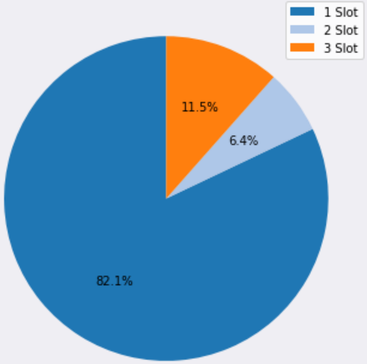


Fig 7. Proportion of Exam Timeslots

Parameters Setting

The Fitness Function

$$F = w_1 * \sqrt{\frac{\sum_{t=1}^{N_T}\left(\sum_{s=1}^{N_s}\sum_{i=1}^{N_I} d_{s,t,i} - \mu\right)^2}{N_T - 1}} + w_2 * \frac{1}{N_M'} * \sum_{m=1}^{N_M'}\left(\ln\left(\frac{1}{e_{b_m} - 1} * \sum_{i=1}^{e_{b_m}-1} e^{\left|u_i - u_{i+1} - \frac{N_T}{e_{bm}}\right|}\right)\right)$$

$$+ w_3 * \frac{1}{N_I} * \sum_{i=1}^{N_I}\left(w_4 * \sum_{d=1}^{N_D} k_{v_{i,d}} + w_5 * \left|\sum_{s=1}^{N_s}\sum_{t=1}^{N_t} d_{s,t,v_i} - q_{v_i}\right|\right)$$

Parameters for the Fitness Function:

$$w_1 = \frac{1}{20}, w_2 = \frac{6}{20}, w_3 = \frac{13}{20}, w_4 = \frac{1}{2}, w_5 = \frac{1}{2}$$

Algorithm Configuration

- Genetic Algorithm:
  - Number of generations: 50,000
  - Population size: 200
  - Elite rate: 0.1
  - Crossover rate: 0.8
  - Mutation rate: 0.4

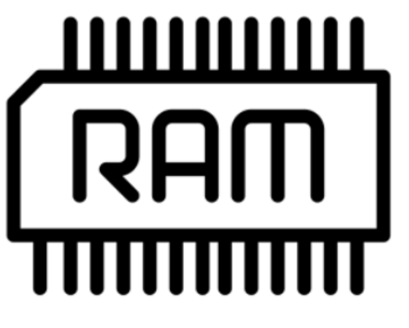
- Tabu Search:
  - Number of iterations: 50,000
  - Tabu list size: 500
  - Neighbor size: 100 and increase 50 after 5,000 iterations

Both the GA and TS were executed over a duration of 12 hours.

Environment Training

Ubuntu 22.04

Intel Core i9 processor, 3.6 GHz.



64GB of DDR4 memory

$\boxed{4}$

Result

Genetic Algorithm and Tabu Search Comparison

{ }

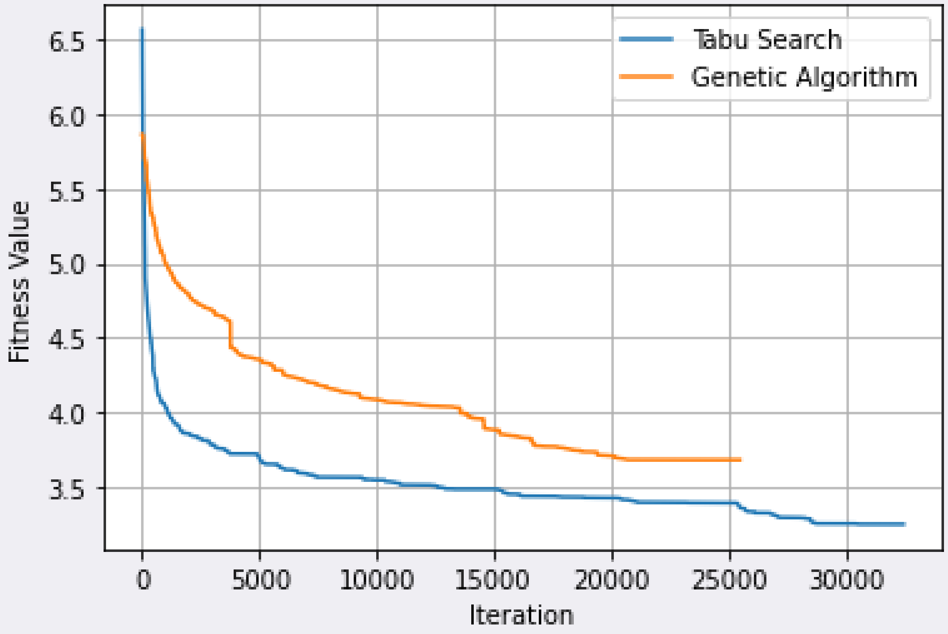


Fig 8. The optimal fitness value for each iteration of two algorithms.

{ }

Genetic Algorithm and Tabu Search Comparison

{ }

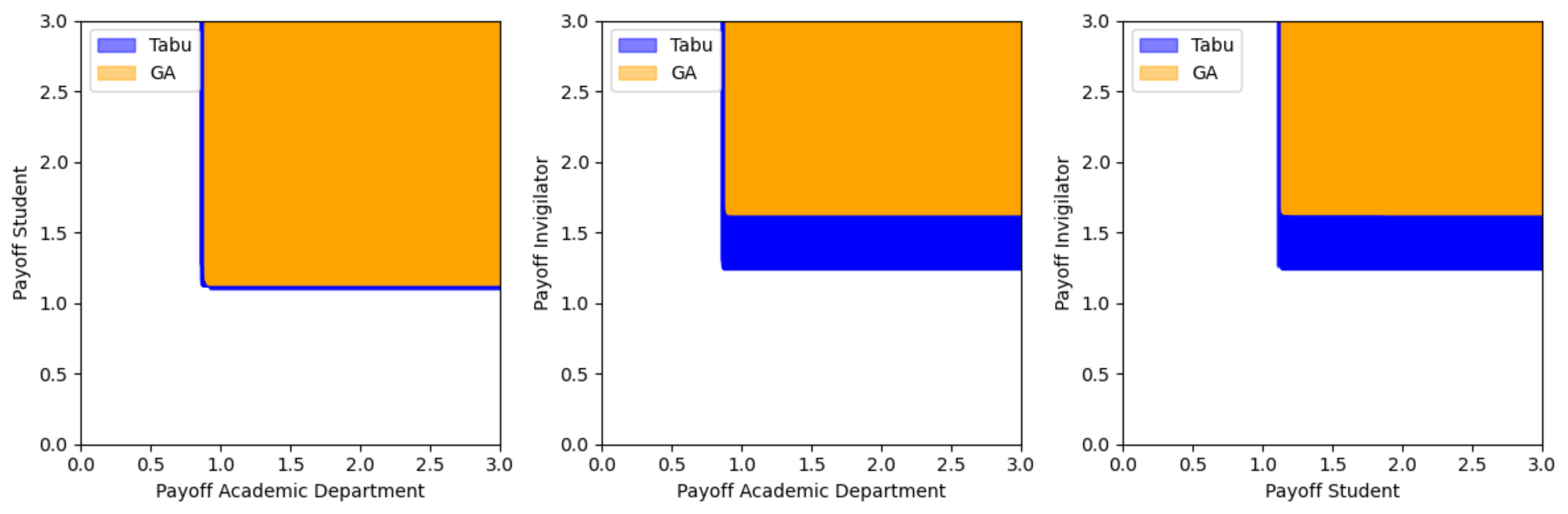


Fig 9. Hypervolume comparison of GA (orange) and Tabu Search (blue)

{ }

Evaluation Method

{ }

Academic department satisfaction calculation:

- Number of room available for each slot is r rooms.
- Number of exams will be schedule is e exams.
- Number of slots for examination is t .
- After assignment, number of rooms scheduled for each slot is $r_1, r_2, \ldots\ldots, r_t$.
- Ideal number of rooms assignment for each slot is:

$$\bar{r} = \left\lfloor \frac{e}{t} \right\rfloor$$

=> Satisfaction rate of academic department is:

$$\frac{1}{t} \sum_{i=1}^{t} \left(1 - \frac{r_i - \bar{r}}{r - \bar{r}}\right) * 100$$

{ }

Evaluation Method

Student satisfaction calculation:

- A student s takes x exams ($x \geq 2$) within t slots.
- Each exam is scheduled in slot $t_1, t_2, \ldots, t_x$ in ascending order.
- An ideal distance between two exams is calculated :

$$dist = \left\lfloor \frac{t}{x} \right\rfloor$$

- Formula to calculate satisfaction rate of this student s above is:

$$\frac{1}{x-1} * \sum_{i=1}^{x-1} dist_i * 100, \qquad dist_i = \begin{cases} \dfrac{t_{i+1} - t_i}{dist} & if\ t_{i+1} - t_i < dist \\ \dfrac{dist - \min\left((t_{i+1} - t_i) mod\ dist, dist\right)}{dist} & , otherwise \end{cases}$$

Evaluation Method

Invigilator satisfaction calculation:

- An invigilator i have to supervise q_i slots within d examination days.
- After assignment, invigilator i have to supervise h_i slots in g_i days.
- Each examination day have β slots.
- Ideal expectation for number of rest day is:

$$day = \mathrm{d} - \left\lceil \frac{q_i}{\beta} \right\rceil$$

- Satisfaction rate of number of day of invigilator i is calculated: $satis_1 = \min(\frac{d-g_i}{day} * 100, 100)$
- Satisfaction rate of task requirement is calculated: $satis_2 = \max((1 - \frac{|h_i-q_i|}{q_i}) * 100, 0)$
- Satisfaction rate of invigilator is: $\frac{satis_1+satis_2}{2}$

Best Result Analysis

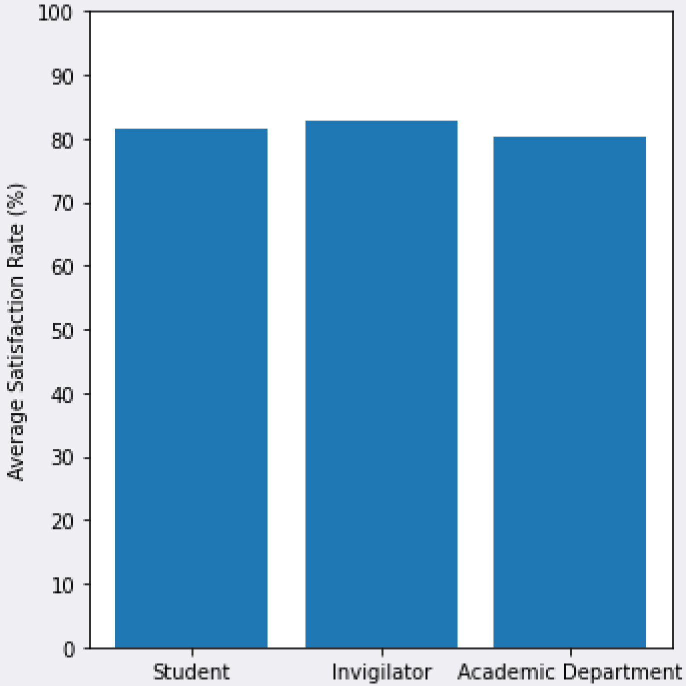


Fig 10. The average satisfaction rate of all stakeholders

Best Result Analysis

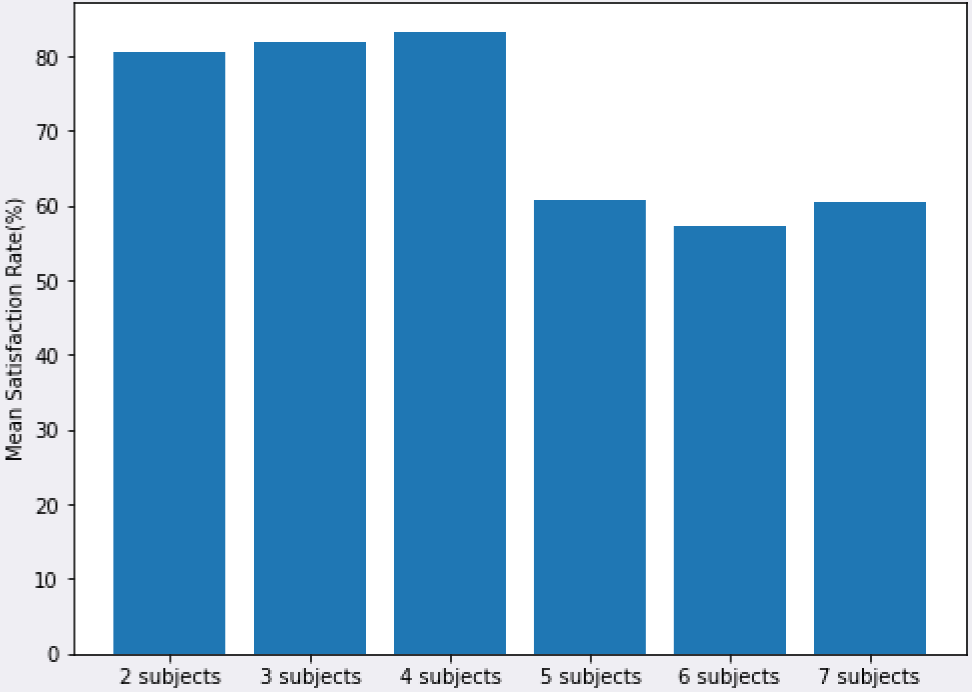


Fig 11. The average satisfaction rate of students with different number of subjects

Best Result Analysis

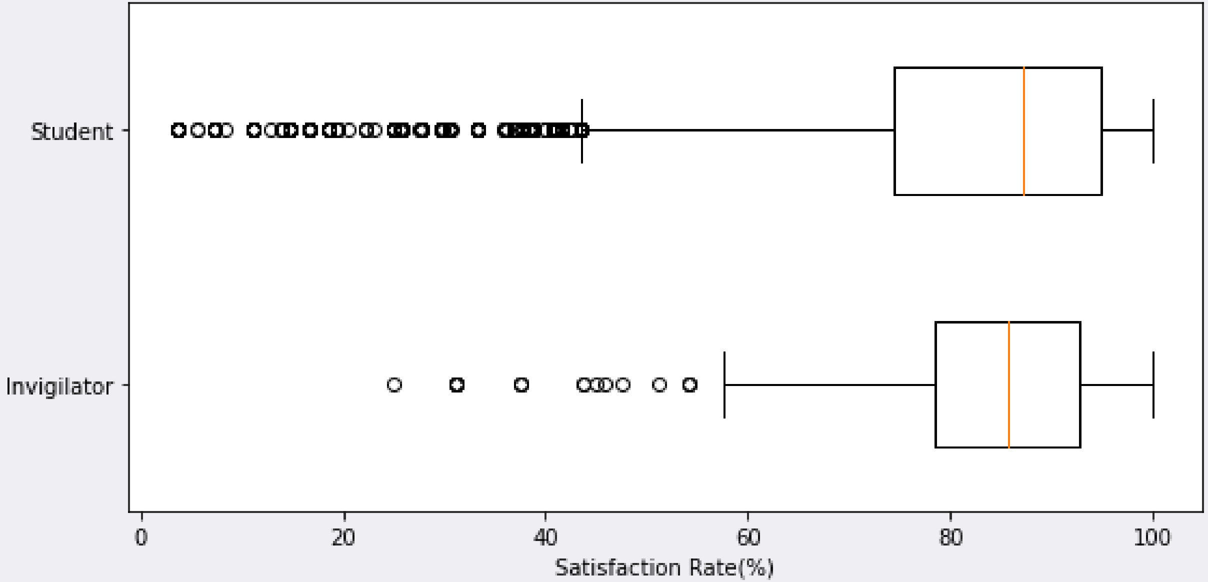


Fig 12. The distribution of student satisfaction and invigilator satisfaction.

Best Result Analysis

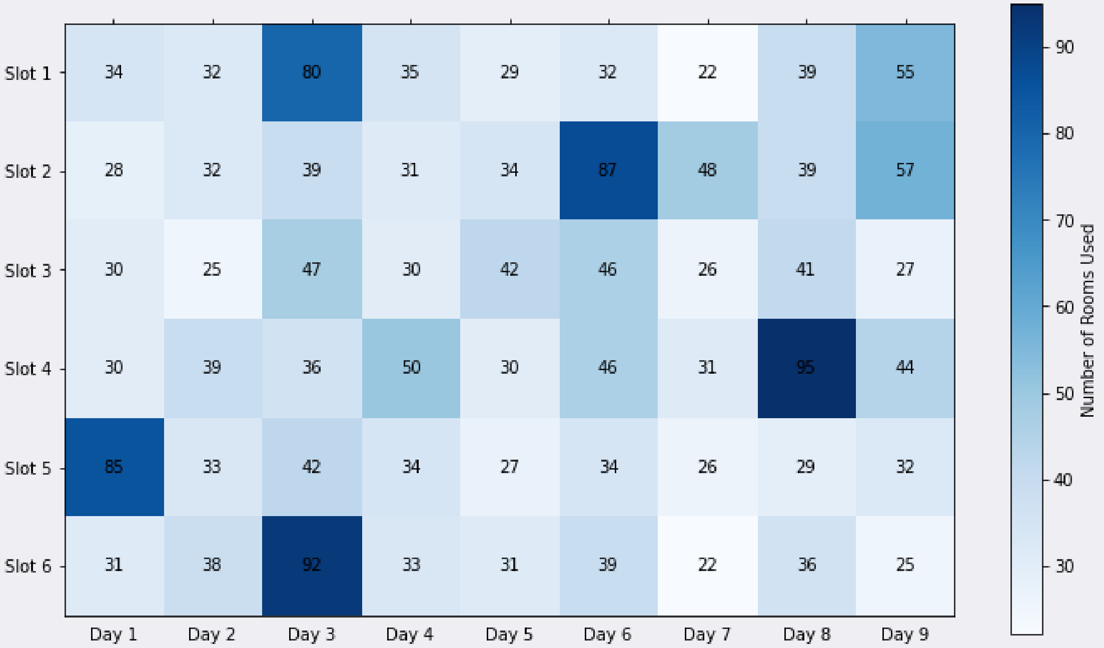


Fig 13. The number of rooms in each time slot.

5

Conclusion

Conclusion

- Addressing the scheduling challenges of exams by integrating game theory – Nash Equilibrium with a Genetic Algorithm and Tabu Search framework.
- Tabu Search has better performance than Genetic Algorithm.
- Future research could explore a hybrid model that combine the strengths of GA and Tabu Search
- Incorporating a wider range of constraints and objectives, such as: invigilator preferences

{ }

6

Demo

{ }

{ }

Thanks for Listening

{ }