INDIAN INSTITUTE OF TECHNOLOGY PALAKKAD

Institute Timetable Automation

Mayank Singla
Rudr Tiwari

Indian Institute of Technology
Palakkad, Kerala

May 18, 2023

Agenda

INDIAN INSTITUTE OF TECHNOLOGY PALAKKAD

1. About the Project
2. Project Challenges
3. Synopsis of Prior Works
4. Genetic Algorithm
5. Simulated Annealing Algorithm
6. Greedy Initialization for Simulated Annealing
7. Results
8. Demo

About the Project

This project aims to develop an automated solution where faculty and academic section can enter the data about the courses offered in a semester, the list of faculties offering courses, the slot system, and the details of available rooms.

It should also provide a workflow that allows the academic section to create an optimal timetable (room-slot-course allocation) by considering the various constraints.

Project Challenges

1. Slot system allocation
2. Room allocation
3. Minimal clashes
4. Common courses allocation
5. Multiple campuses
6. Credit-based system compatibility

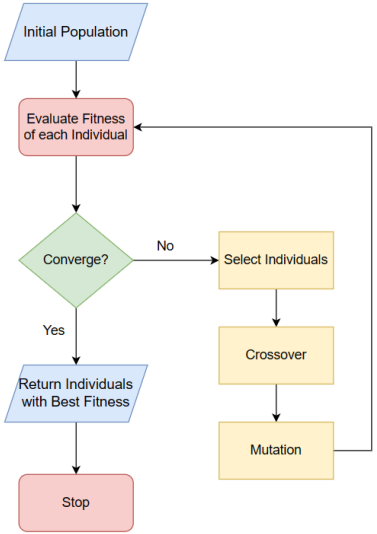
Synopsis of Prior Works

INDIAN INSTITUTE OF TECHNOLOGY PALAKKAD

- Our first attempt was to create a slot system as well along with the timetable to minimize the clashes
- For this we opted for a combination of brute force and greedy approaches and successfully implemented this approach
- We also constructed a basic web interface for quicker and faster visualization of the results of our algorithm
- But based on several reviews, it was concluded that having a fixed slot system is more convenient for both the students and the faculty
- Also, our approach was a complete brute force approach, so we need some better algorithms to solve all the problem constraints in an efficient manner

Genetic Algorithm

Algorithm Flowchart



Genetic Algorithm (Contd.)

Using Genetic Algorithm in Timetabling

INDIAN INSTITUTE OF TECHNOLOGY PALAKKAD

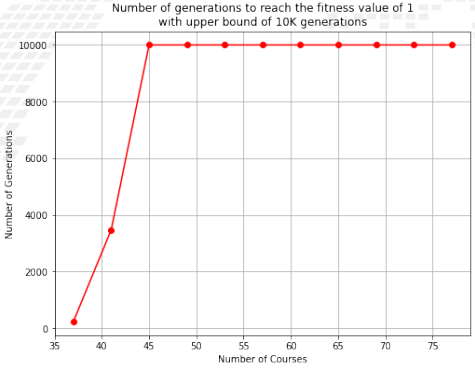
Randomized Approach: Perform each step randomly with some associated probabilities

- **Individual Representation:** List of classes also called a **Schedule**
- **Class:** Contains a course and randomly allocated slots and rooms
- **Fitness Function:** Inversely proportional to the number of conflicts
- **Selection Procedure:** Using tournament selection with some elite schedules
- **Crossover:** Randomly mixing the classes from two schedules to create a new schedule
- **Mutation:** Randomly pick classes for an existing schedule from a newly created schedule

Greedy and Maximal Approach: Perform each step greedily to always have a maximal allocation of the list of classes

- **Individual Representation:** List of maximally allocated classes and unallocated classes, also called a **Schedule**
- **Class:** Contains a course and greedily allocated slots and rooms
- **Fitness Function:** Ratio of the number of allocated courses and the total number of courses
- **Selection Procedure:** Using tournament selection with some elite schedules
- **Crossover:** Take different permutations of the sequence of classes from the two parents, create their maximal allocation and pick the one with the maximum fitness
- **Mutation:** Unallocating an allocated class and trying to allocate an unallocated class such that the allocation is always maximal

Observations



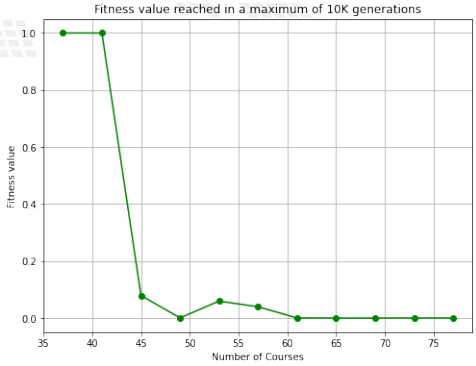
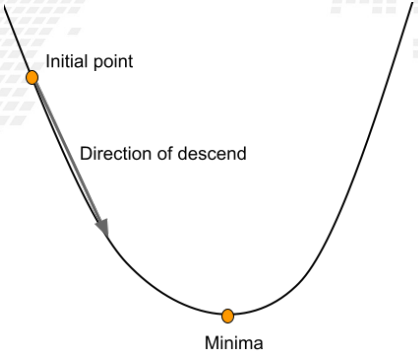


Figure: Results for 10K Generations

Simulated Annealing

Hill Climbing



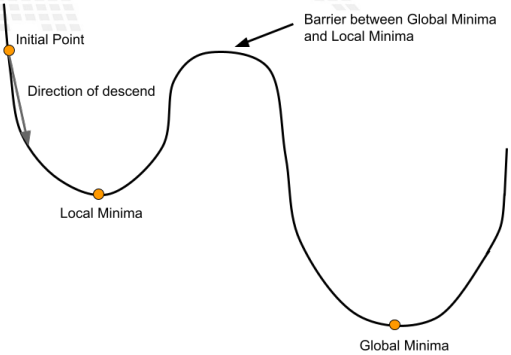
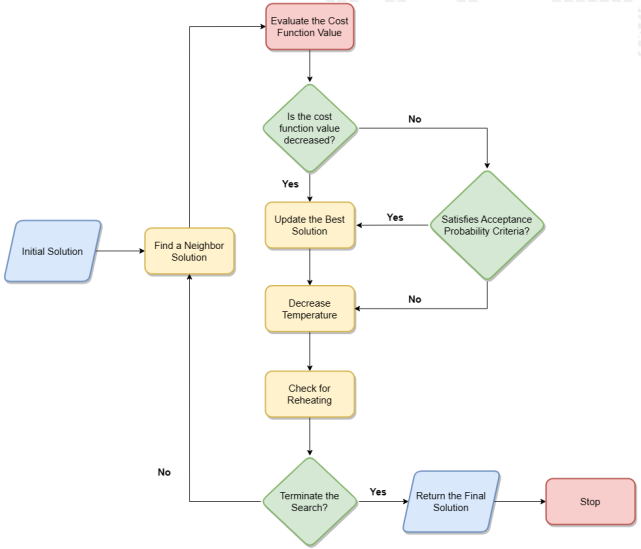


Figure: Successful and Unsuccessful Hill Climbing

Simulated Annealing (Contd.)

Algorithm Flowchart



Simulated Annealing (Contd.)

Using SA in Timetabling

INDIAN INSTITUTE OF TECHNOLOGY PALAKKAD

- **Cost Function** - The cost function is the sum of all the conflicts. Hard constraints are multiplied with a larger multiplier as compared to soft constraints

- **Neighbor Function** - Swapping Neighbourhood - one of the most used neighborhood searching algorithms in our implementation
  Other neighbor functions include simple searching and a mix of simple searching and swapping neighborhood algorithm

- **Temperature Function** - The temperature function which we choose in our implementation is

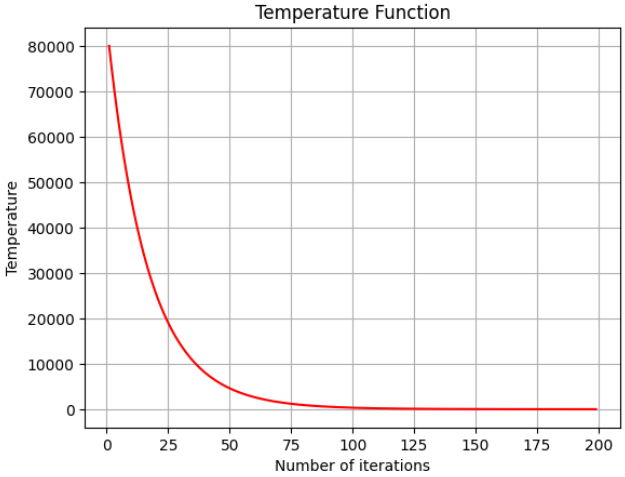
$$t = t \cdot (1 - \frac{\ln t - \ln FT}{NMOVE})$$

- **Acceptance Probability** - The acceptance probability we used is widely used in most implementations of simulated annealing -

$$P(oldState,\ newState,\ temp) = \begin{cases} 1, & \text{if } newState < oldState \\ e^{\frac{oldState - newState}{temp}}, & \text{otherwise} \end{cases}$$

Simulated Annealing (Contd.)

Temperature Function



Greedy Initialization

- **Course Selection Heuristic** - Choose a course with the smallest value of $\frac{apd_i(\widetilde{M})}{\sqrt{nl_i(\widetilde{M})}}$

  $apd_i(\widetilde{M})$ - Number of available slots in the timetable $\widetilde{M}$ for course c_i
  $nl_i(\widetilde{M})$ - Number of lectures of course c_i yet to be assigned in $\widetilde{M}$

- **Slot-Room Selection Heuristic** - After choosing a course we pick a slot-room pair for it. The pair with the smallest value of the below function is chosen -

$$k1 \cdot uac_{i,j}(\widetilde{M}) + k1 \cdot \gamma + k2 \cdot \Delta h$$

Results

INDIAN INSTITUTE OF TECHNOLOGY PALAKKAD

No.of courses	Hard Conflicts Genetic Algorithm	Hard Conflicts Simulated Annealing	Soft Conflicts Simulated Annealing
37 (random data)	0	0	0
45 (random data)	13	0	0
57 (random data)	43	0	2
69 (random data)	90	1	3
97 (random data)	141	3	16
94 (real data)	222	11	62

Table: Comparison between Genetic Algorithm and Simulated Annealing used on timetable initialized by Genetic Algorithm

Results (Contd.)

INDIAN INSTITUTE OF TECHNOLOGY PALAKKAD

No.of courses	Hard Conflicts Greedy Algo.	Soft Conflicts Greedy Algo.	Hard Conflicts SA	Soft Conflicts SA
37 (random data)	0	0	0	0
45 (random data)	0	0	0	0
57 (random data)	0	0	0	0
69 (random data)	0	0	0	0
97 (random data)	6	33	0	39
94 (real data)	18	34	8	33

Table: Comparison between Greedy Algorithm and Simulated Annealing used on timetable initialized by Greedy Algorithm

Demo Time

Thank You!