Artificial Intelligence

Assignment 3 Report

CS-E
17L-4010 Uzair Ahmed , 17L-6326 Ali Haseeb

Introduction And Methodology

- The input data class reads data from registration, room capacity, general.
- Gene represents one block of timetable(list of courses in one slot).
- The chromosome class is for creation of chromosomes and random numbers.Chromosome is represented as a 2D Matrix with row as no of days x column as slots(A Complete Timetable).The class contains a CreateChromosome function which randomly initialise the chromosome.
- Fitness function, crossovers and mutation is also implemented here.Chromosome Comparator just compares two chromosomes by taking the difference of their fitnesses.
- In our main program, firstly input is taken from files followed by the creation of chromosomes.Population of 100 chromosomes is created in which each chromosome is randomly generated.
- Next Step includes calculation of each chromosomes fitness and on the basis of fitness,population is sorted
- Target is set and then the Genetic Algorithm is applied including creating new generations and doing mutations and crossovers.
- Selection criteria is elitism meaning 10% of fittest chromosomes in a population are automatically added to the next generation while from the remaining 90% ,we select the parents for the crossover to generate new offspring for the next generation.Thus,mutation is introduced to maintain randomness in population and crossover is achieved through the combination of two parent chromosomes genes.
- Finally the new population is created and for this fitness is calculated for each chromosome and the process is repeated until we get one such chromosome(TimeTable) which has the targeted fitness.

- Our fitness function goes by all numbers of days for every slot and assigns points on the basis of the following conditions.
    1. Timetable must have all courses.
    2. No of students with 2 exams in the same slot should be minimum.
    3. No student has more than two exams in one slot.
    4. No of students giving exams in one slot should not be more than total capacity.
    5. No of students with 2 exams in consecutive slots should be minimum
    6. No student has no exams in more than 2 consecutive slots.

Our code is successfully creating generations along with changing fitness along with successful crossover,mutation and selection. However after approximately 200 generations the value of fitness functions becomes constant.The fitness value increases with the increasing generations.Once fitness becomes constant,we take the final generations top fitness chromosome and then apply local search.In local search we simply create new childs from this chromosome and continue to create generations after generations.For each child we calculate fitness and add it to the current generation.Every child is created by randomly changing the slots of the Chromosome supplied by GA.Moreover,as fitness becomes constant then again in local search we return the top Chromosome closer to our target fitness.

We have made two fitness functions and attached are the experiment results of the fitness function one. The generation and fitness are changing but due to the crossover of two parents slots become empty and the schedule could not be found.

In the other fitness function, we abide by the condition that total courses in slots equals the total number of courses in the data and that no course is repeated. However, as a result our fitness does not show significant change but on the other side we are successful in getting a schedule.

Along with this we have also implemented the local search.

We have applied GA and local search but somehow due to time constraint,we feel that the fitness function needs to be reevaluated along with local search code.However,we have tried to deliver our best.

As fitness becomes closer to target,we can achieve optimal solutions.

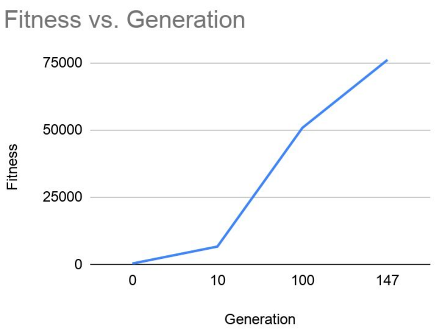
Experiment 1   (with fitness function one)

Generations	Fitness
0	245
10	6568
100	50785
147	76078

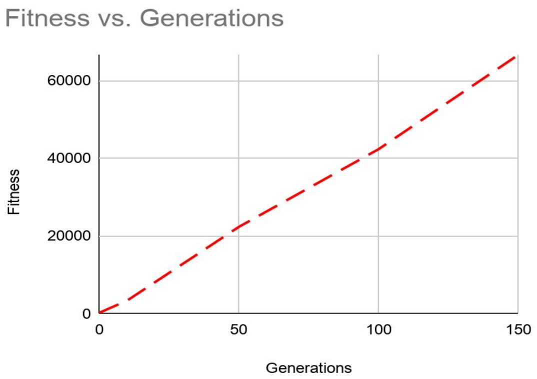
Experiment 2 (with fitness function 1)

Generations	Fitness
0	243
10	3399
50	22367
100	42356
150	66619

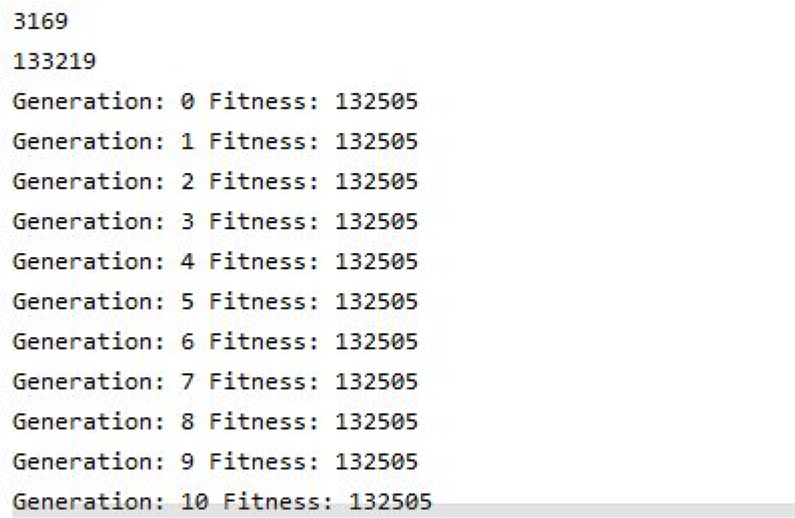
Experiment 1



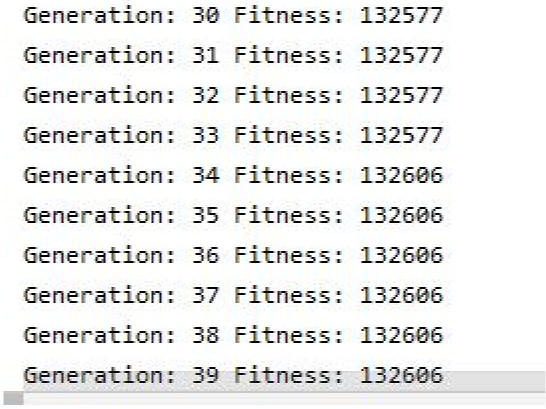
Experiment 2



With fitness function 2

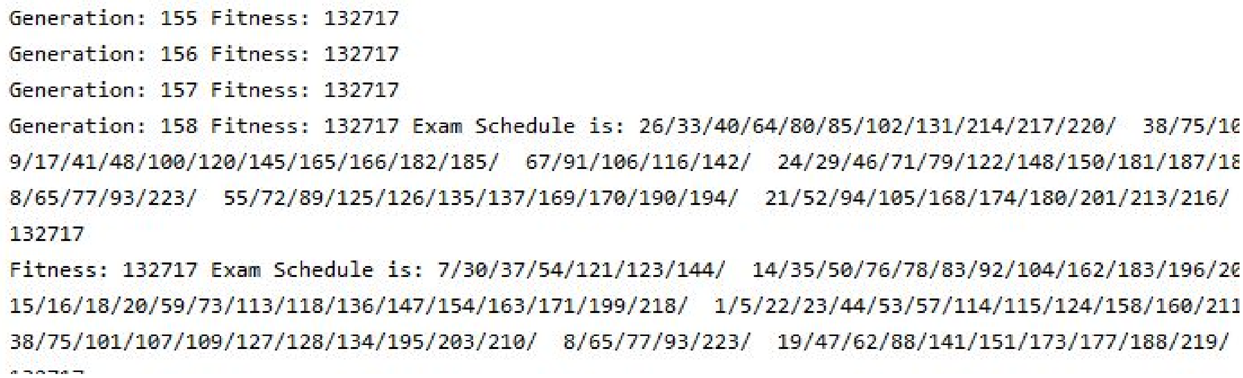


```
3169
133219
Generation: 0 Fitness: 132505
Generation: 1 Fitness: 132505
Generation: 2 Fitness: 132505
Generation: 3 Fitness: 132505
Generation: 4 Fitness: 132505
Generation: 5 Fitness: 132505
Generation: 6 Fitness: 132505
Generation: 7 Fitness: 132505
Generation: 8 Fitness: 132505
Generation: 9 Fitness: 132505
Generation: 10 Fitness: 132505
```



```
Generation: 30 Fitness: 132577
Generation: 31 Fitness: 132577
Generation: 32 Fitness: 132577
Generation: 33 Fitness: 132577
Generation: 34 Fitness: 132606
Generation: 35 Fitness: 132606
Generation: 36 Fitness: 132606
Generation: 37 Fitness: 132606
Generation: 38 Fitness: 132606
Generation: 39 Fitness: 132606
```

With fitness function 2



```
Generation: 155 Fitness: 132717
Generation: 156 Fitness: 132717
Generation: 157 Fitness: 132717
Generation: 158 Fitness: 132717 Exam Schedule is: 26/33/40/64/80/85/102/131/214/217/220/  38/75/10
9/17/41/48/100/120/145/165/166/182/185/  67/91/106/116/142/  24/29/46/71/79/122/148/150/181/187/18
8/65/77/93/223/  55/72/89/125/126/135/137/169/170/190/194/  21/52/94/105/168/174/180/201/213/216/
132717
Fitness: 132717 Exam Schedule is: 7/30/37/54/121/123/144/  14/35/50/76/78/83/92/104/162/183/196/20
15/16/18/20/59/73/113/118/136/147/154/163/171/199/218/  1/5/22/23/44/53/57/114/115/124/158/160/211
38/75/101/107/109/127/128/134/195/203/210/  8/65/77/93/223/  19/47/62/88/141/151/173/177/188/219/
132717
```

<--->