National University of Computer and Emerging Sciences

Artificial Intelligence

"Project 1: Genetic Algorithm"

STUDENTS NAME and REGISTRATION NUMBERS:

Maria Yasir	i18-0454
Maaz Asad	i18-0474

DEGREE PROGRAM:	BS(CS)
Section:	B
SUBJECT NAME:	Artificial Intelligence
DATE OF SUBMISSION:	May 6, 2021
SUBMITTED TO:	Ma'am Labiba Fahad

Data Set Used

We are using the data set provided by the instructors.

We have made use of dictionaries to compensate for repeated values in the data set there by eliminating any repetitions that would plague the results. We have stored the binary values of the data as keys in the respective dictionaries

Working of the Genetic Algorithm

Initializing the starting Population

After loading data from the data set provided, we create a population of randomized day, timeslot, course, teacher, and rooms. The population size is also random between 50 to 100. Each randomized schedule represents a single chromosome in the population and its fitness is stored with in the Structure which is calculated immediately after its creation. Storing the fitness enables us to reduce the overhead of runtime calculations. However, after crossover and mutation, as new chromosomes are created, fitness is re-evaluated.

Crossover

We tried Single point, 2-point and Uniform Crossover. For our implementation 2-point Crossover was the most optimal function as it was giving better results than Single Point Crossover and had shorter execution time than Uniform Crossover. The probability for the Crossover function to be executed is 0.3

Mutation

We have made use of Random Resetting, where we chose a random day and perform random resetting on it. The probability for the Mutation function to be executed is 0.3

Roulette Selection

We are calculation probabilities for each chromosome by using its Total fitness. The probability of a chromosome is given by: fitness/ sum of fitness. Cumulative probability is also calculated and used to help randomly get a chromosome from the population for further processing in the crossover and mutation

Genetic Algorithm

We start with a randomly generated initial population. For the next generation, we first take top 10 fittest chromosomes from the previous generation and insert them in to the new generation. After that we apply Roulette selection on the previous generation to get 2 random chromosomes, C1 and C2. We then apply crossover and mutation based on the probability of those functions to get 2 child chromosomes, C3 and C4. For C3 and C4, their fitness is calculated and the fittest of these 2 chromosomes is passed onto the new generation. C1 and C2 are randomly selected again, and process is repeated until the size of the new generation is equal to the size of previous generation.

We create 200 generations and select the chromosome with the highest fitness from the 200th generation. As we are moving the top 10 fittest chromosomes from previous generation to the next generation, it enables us to keep track of the fittest chromosome even if it occurred in some previous generation prior to the 200th generation.

Time taken to execute for 200 generations is approximately 22 minutes

Evaluating Fitness of Chromosome

We have 6 hard constraints and 4 soft constraints. We have divided 100 fitness points such that each Hard constraint has 14 points and each Soft constraint has 4 points. 14*6 + 4*4 = 100. Each constraint is evaluated separately and adds up to contribute to Total Fitness of the Chromosome. Total Fitness is given as a percentage, higher is better. Hard Constraints total fitness is give out of 84 and Soft Constraints Total fitness is given out of 14.

Each constraint is evaluated separately and we get a fitness of the constraint scaled to the points of the respective type. Closer the value to 0 the better it is. In the end, these values are subtracted from 100 to get the overall fitness

Results

Over the 200 generations, the highest fulfillment of hard constraints we have achieved over numerous attempts is 83.74/84, and the highest Total fitness is 98.7/100