**AI Programming Assignment 3**

**-Suraj Pandey**

**MT18025**

**1. In Question 1 :**

**Assumptions:**

1. Steps are as 1,2,3,4,5,6,7,8,9
2. User will enter first

**Minimax Algorithm**

Computer is using Minimax Algorithm.

In Minimax algorithm, MAX tries to maximize its own win and MIN tries to minimize its own win.

By tracing the all possible paths for MAX and then MIN, draw the game tree and each leave will either represent these three states:

Win (+1)

Loss (-1)

Draw (0), assume it for Computer .

Then, tries to feed objective function from leave to root of tree (Bottom Up Approach) and find the objective function of root (+1,0,-1).

As MAX will maximize among its child objective functions whereas MIN will minimize among child objective functions.

In this, User never win .

**Algorithm:**

Minimax (X, board, depth)

If board is winning state for X

If X is MAX return 1

Else return -1

If Its draw return 0

If X is MAX

Find maximum objective value path

If X is MIN

Find minimum objective value path

**Minimax with Alpha Beta Algorithm**

It is same as Minimax algorithm, only difference is to tracing of number of nodes to find the best path.

In alpha beta, it pruned less number of nodes in finding the best path.

So, here there is two values [alpha,beta] to each of node.

Alpha gives value of minimum and beta gives value of maximum.MIN wil try to update beta and MAX will try to update alpha.

**Algorithm:**

**For root [alpha,beta] = [-inf,+inf]**

Minimax (X, board, depth ,alpha ,beta)

If board is winning state for X

If X is MAX return 1

Else return -1

If Its draw return 0

If X is MAX

Find maximum objective value path

and if value of that path ,VAL>=beta ,then return

if VAL>alpha:

alpha=VAL

If X is MIN

Find minimum objective value path

And if value of that path,VAL <=alpha, then return

if VAL<beta:

beta=VAL

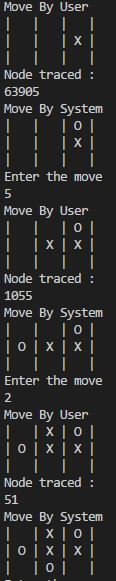
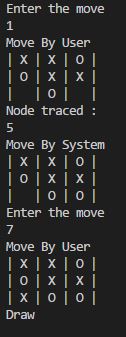
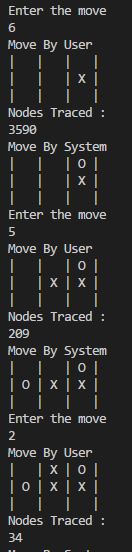
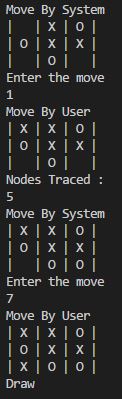
Prune if alpha >= beta(cut the branches).

**Comparison between Minimax and Minimax with alpha beta**

Even though both will give the same result but minimax with Alpha beta is computationally faster since it traces the less number of nodes than minimax .

Both take steps towards the maximum possibility of winning.

Minimax with alpha beta traced less nodes as in output of program than minimax.

**Minimax Minimax with Alpha Beta**

So, here clearly noticed that ,Minimax will take more time to compute the step ,whereas Minimax with alpha beta will take less time to compute the next possible best move.

**1. In Question 2:**

**Assumptions:**

* Time table is made considering the same course will never come collide in same slot.
* Time table is made considering the same professor will never teach in same slot.
* Time Table has constraint as a Professor can teach only atmost two courses.
* Time table has all courses requirement fulfilled slots.
* Time table has constrained as no one course will be teach on same day.
* Time Table is made considering some aspects as similar to IIITD system.
* Evaluation Function take 10 best performing chromosomes from population.
* Fitness function is used as sum of collision in chromosomes i.e, constraints mismatch count.
* For better chromosomes , did 2 pt point and single point crossover with some probability of both.
* [0,0] denote the empty lecture hall with no professor and course.
* Gene has professor with course assigned.
* In each generation [0,0] is inserted with some probability in set of professor and course set such that improve the diversity of chromosomes.

**Genetic Algorithm:**

1. Initialize Population
2. Evaluate population on basis of Evaluation Function
3. Repeat while not converged
4. Crossover
5. Mutation
6. Again Evaluate to select top k chromosomes.

**Chromosome Structure**

Chromosome contains 2D like structure .

In which gene has [course,professor] details from sets of professor and course combinations.

Chromosome contains 5\*8\*L (days\*slots\*lecture halls) genes where L is number of lecture halls present.

**Fitness Function:**

Fitness Function return Fitness value which is computed by uding violating constraints count.

Constraints are :

1. Professor can teach atmost 2 courses.
2. Time table is made considering the same course will never come collide in same slot.
3. Time table is made considering the same professor will never teach in same slot.
4. Time table has all courses requirement fulfilled slots.
5. Time table has constrained as no one course will be teach on same day.
6. Professor can teach only those course which assigned .

If any of constraints is not met ,penalty is made and then calculate the value as negative( Since penalities are there).

More the value of fitness ,more is better chromosome.

**Recombination:**

**Crossover**

Crossover is done as single point and 2pt crossover with probabilities to diversify the population.

**Mutation**

Mutation is done on each chromosome of population with some probabiltiy and replace gene with random generated gene from [course,professor] element from sets of combination of course and professor value.

**Observations:**

1. Performance is good in respect to good output of time table as according to constraints given .
2. After some time, there is stuck in generation of chromosomes with same fitness value.
3. Regeneration are more sometimes.
4. Difficulty to converge with these constraints but result is proper when come.
5. It takes time converge but it will definitely converge.

**Memetic Algorithm**

MA=GA + Local Search

**Algorithm**

1. Initialize population
2. Local Search
3. Evaluation Function to find best k chromosomes
4. Repeat while not converged
5. Crossover
6. Mutation
7. Local Search
8. Evaluation

MA and GA are basically same ,only difference is MA will improve its population by local search algorithm such that take more improved chromosomes from its neighbours.

For this, here ,we do for each chromosome ,find its neighbour as change in one gene ,if it improve the chromosome performance ,replace it with previous one.

Recombination is also done between pair of chromosome in population and then best two will be placed in population and previous one is removed.

**Local Search:**

1. Recombine a pair of parents and replace them if child is better than parents.
2. Pick a chromosome of population and change its gene with other gene and if that chromosome is better than previous ,then replace it in population with previous.

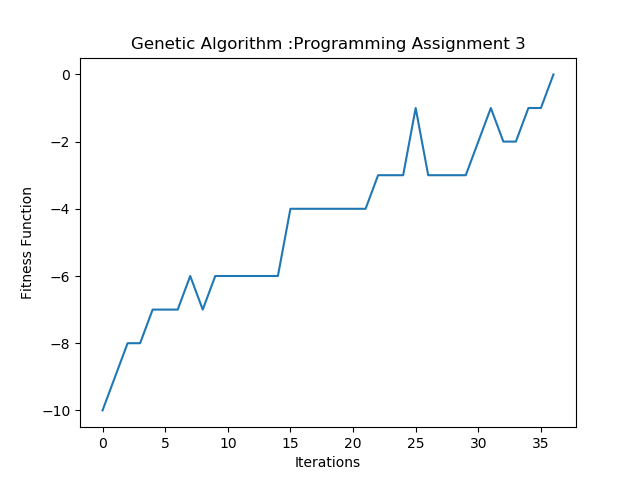
**Performance Evaluation of GA and MA:**

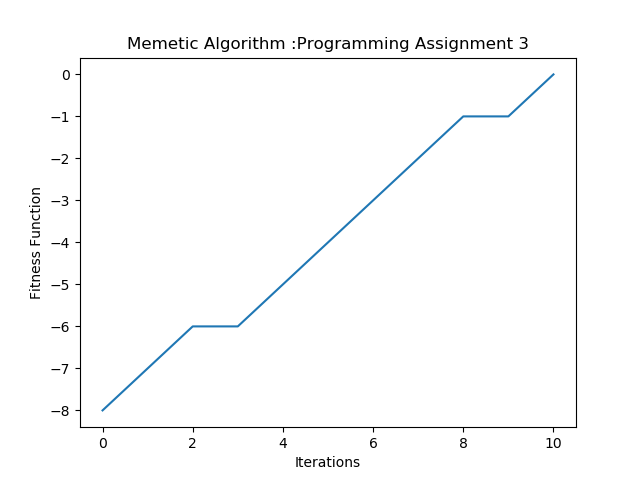
GA and MA are both have fitness value ,crossover and mutation and evaluation function.

But Local Search make MA more powerful than GA in reference to less regeneration for convergence.

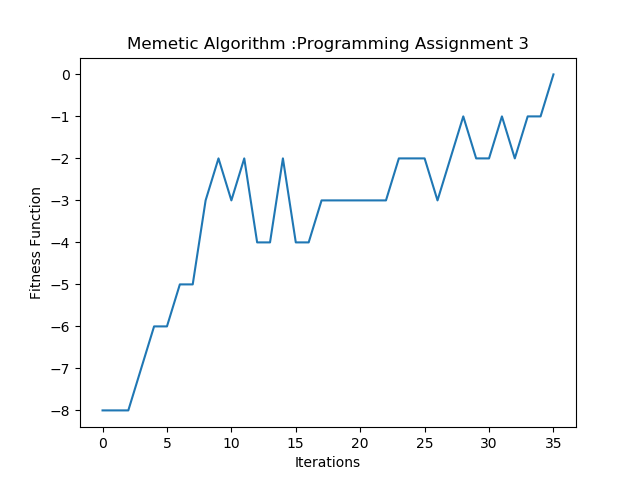
**Observations**

1. MA take less regeneration or iterations than GA
2. There is high probalility of convergence to optimal result in MA since Local search is used to improve the population
3. Local Search is in MA make complex to execute than GA
4. There is less probability that MA got stuck because Local search is there.

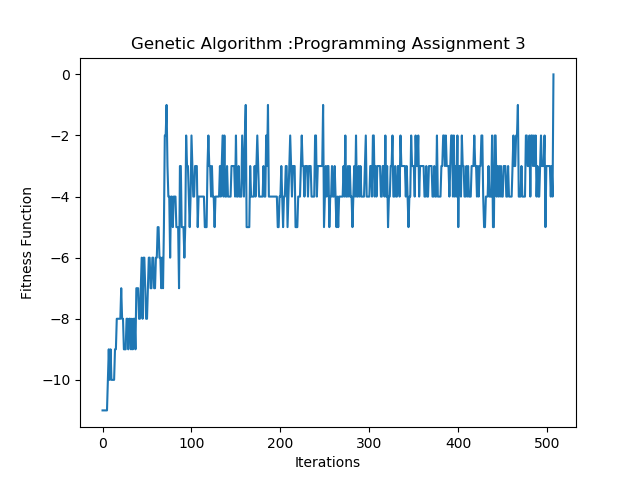
GA when 3 Course 2 Lecture hall 2 professor with each course having 2 slots



MA when 3 Course 2 Lecture hall 2 professor with each course having 2 slots



MA when 5 Course 2 Lecture hall 3 professor with each course having 2 slots



GA when 10 Course 3 Lecture hall 5 professor with each course having 2 slots

**Constraints Satisfaction Problem:**

It satisfies all constraints with each state and then trace further ,if at any instant state is not satisfying it traces back.

It works like DFS manner

**Assumptions:**

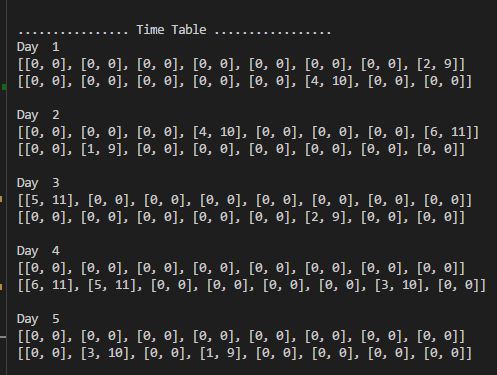
Here ,used [-1,-1] to every lecture hall in beginning.

Constraints are as:

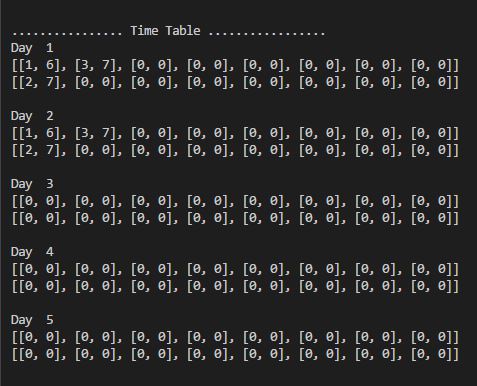
* Professor can teach atmost 2 courses.
* Time table is made considering the same course will never come collide in same slot.
* Time table is made considering the same professor will never teach in same slot.
* Time table has all courses requirement fulfilled slots.
* Time table has constrained as no one course will be teach on same day.

**Observations:**

1. Computationally Low, but some day there is no class and some days have all slots full.
2. Time table prepared very fast.

****

**From GA and MA**

****

**From CSP**

**Results/Inferences:**

1. MA is better than GA in reference to less number of regeneration.
2. CSP is faster than GA and MA but less effective .
3. Minimax traces more number of nodes than minimax with alpha beta pruning so, minimax with alpha beta is fast.
4. GA and MA will get stuck into some value as local optima ,but it will surely converges at last.