Artificial Intelligence

FIRST SEMESTER 2021-22

Programming Assignment 1

Prepared for:

Professor Sujith Thomas
INSTRUCTOR IN CHARGE
Artificial Intelligence

Prepared By:

Saksham Bansal **2019A7PS0056G**

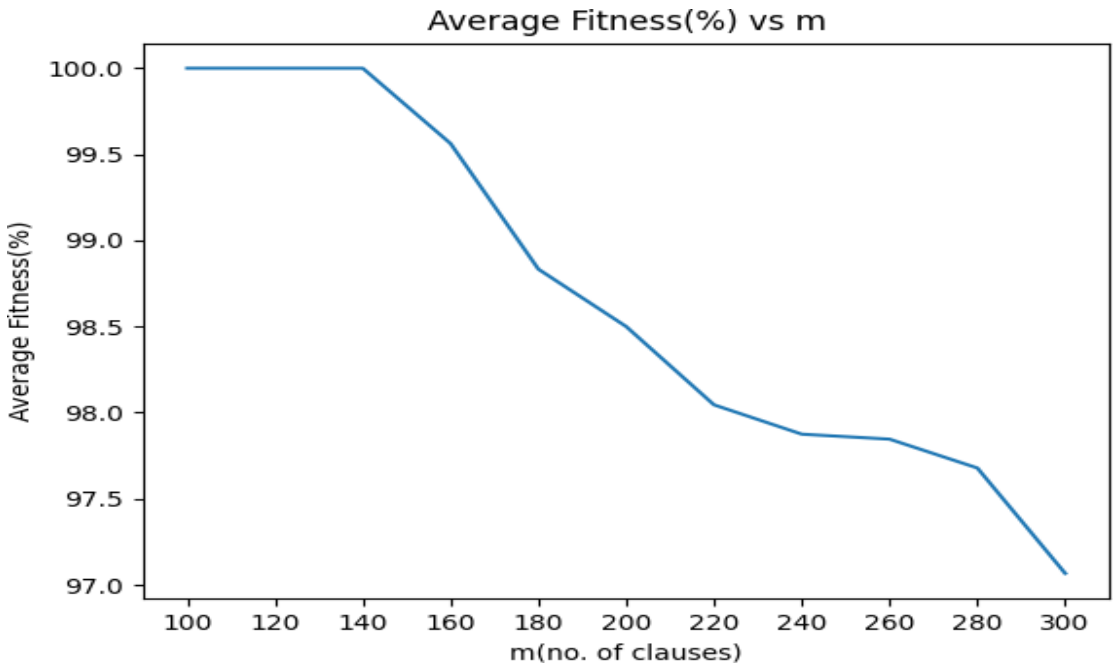
29-09-2021

Final Algorithm-

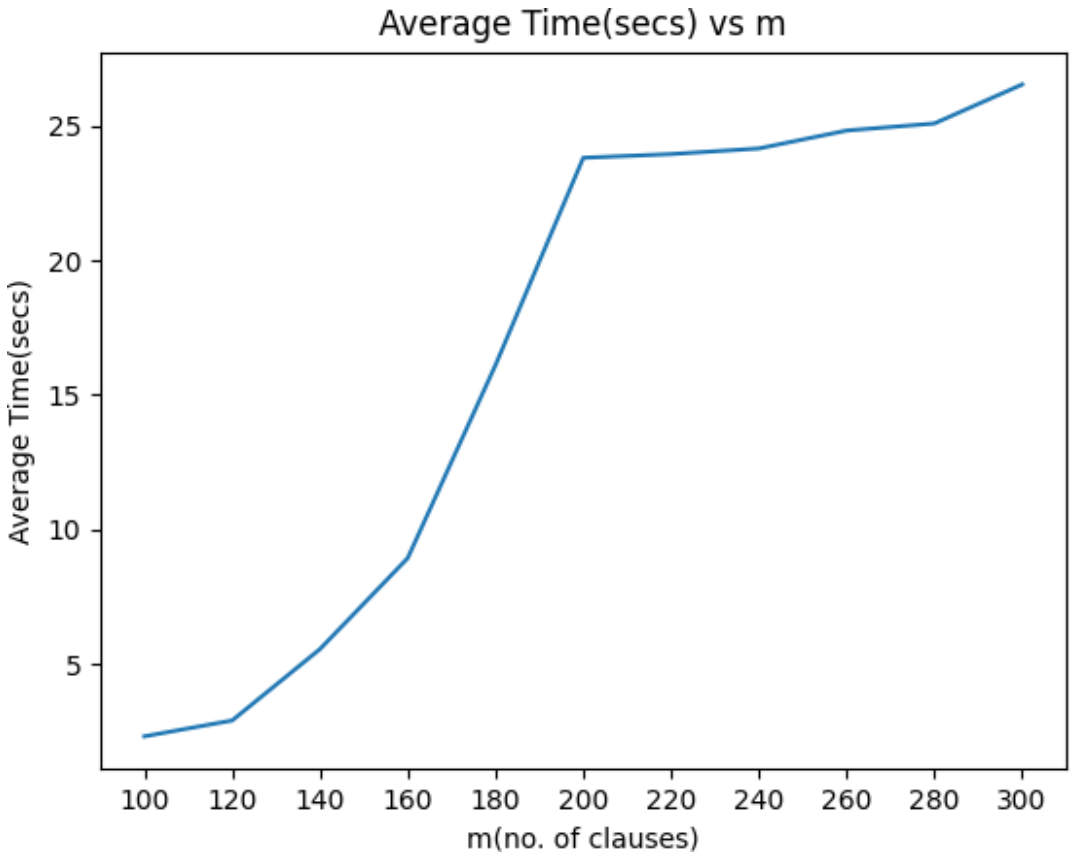
The model is a numpy array of size 50, where each value is either 0(False) or 1(True) and is randomly assigned.
In the first step, a fixed number of such models are generated and then stored in a list in decreasing order of their fitness values(calculated in the same step).
Next, a fixed number of parents with highest fitness values are also stored in a different list.
In the next step, crossover happens, where each selection of the 2 mating parents is done randomly based on probability of their selection, which has a weightage of their respective fitness values.
In the mating stage the models are appended at two randomly selected indexes, and finally a list of these created children and chosen parents are returned.
Next comes the mutation process, which has a probability attached with it, which decreases as the number of generations increases, initially at a slower rate and then at a faster rate.

Finally, the models generated after mutation are returned as the next generation and the process is repeated until the max fitness is reached or max time is reached or a fixed number of generations with equal fitness value appear consecutively.

1. **Average of Fitness Values vs Number of Clauses Graph:**



2. Average of Time Taken vs Number of Clauses Graph:



3. Improvements in the Algorithm:

- Returning a fixed number of parents with best fitness value, along with the children models created, to the next generation.
- Selection of parents for mating, with a probabilistic approach, with weightage given to their respective fitness values.
- Double Crossover in the parent models, where the two indexes are randomly selected.
- Mutation probability is a function of the number of generations created, it is initially high and then decreases initially at a slower pace and then at a faster pace.
  It is given by the following function-

$$prob = prob - (prob - c1)\frac{(1 - e^{ct})}{(1 + e^{ct})}$$

Here t is the number of generations with the same fitness values.

Approaches that failed:

- Creating all possible parent model pairs and sorting them on the basis of summation of their fitness values and giving weightage while selection of parents to this fitness value.
- Selecting an index in the vicinity of the middle for the mating process.
- Selecting parents purely on the basis of their high fitness value, which did not have the possibility for exploration process.

4. Problems Difficult for GA algorithm to find a good Solution(From Graph)

Sentences with few clauses, that is, small values of m are under-constrained and have multiple satisfying models, thus a possible solution can be reached in fewer generations. As the value of m increases the constraints get tighter and the number of possible solutions decrease in number or may not exist, in this case the probability of the solution converging to a local optima increases as with multiple generations several similar models may be repeated. Also, if the CNF is not satisfiable, 100% fitness cannot be achieved and the maximum possible fitness cannot be predicted, making it difficult to terminate the program.

5. Difficulty of satisfying a 3-CNF sentence in n variables

In this problem, we keep the number of symbols(n) constant, and the number of clauses(m) variable. As we increase the value of m, the probability of having contradicting clauses increases. Say for satisfying a clause we need the truth value of symbol 1 to be True, i.e symbol(1) = T, and for satisfying another the requirement is symbol(1) = F, in this case the 3-CNF cannot be satisfied and with increasing value of m probability of having such a case increases as n is constant.

Example:- (1or2or3)and(-1or2or3) where symbol(2) = F and symbol(3) = False(for satisfying other clauses) any other example of this kind where (1orF)and(-1orF).