ARTIFICIAL INTELLIGENCE LAB

OFFLINE 5

ON

IMPLEMENTATION OF K-SAT BY GENETIC ALGORITHM

NAME:ISHITA HAQUE

STUDENT ID:1205069

SECTION:A2

L-4T-1

Here ,

k: number of literals per clause;

v: number of variables;

c: number of clauses in conjunctive normal form

m: number of literals to be exactly true in Exactly-m k-SAT problem.

Number of maximum iterations is 20.

Table of Simulations:

k	v	c	N (initial popsize)	m	No. Of Generations		Best Fitness	
					k-SAT	Exactly-m k-SAT	k-SAT	Exactly-m k-SAT
3	5	4	4	1	1	20	4	2
4	4	5	11	1	1	20	5	3
3	5	4	6	2	1	20	4	3
4	5	8	15	2	1	20	8	5
3	3	7	6	2	1	20	7	4

In solving both k-SAT and Exactly-m k-SAT problem the initial population size has been chosen randomly. The upper limit of this range is (2^v), since all the individuals of the initial population has to be unique. So number of individuals cannot be greater than 2^v.

All individuals are chosen randomly. Also, to make sure that the same individual is not inserted again, hashing has been performed. For this I have calculated binary representation of the individual originally in Boolean form, assigning true=1 and false=0,and calculated its decimal

value. Hash values returned is obtained by dividing this value by the population size(popsize). I have then checked previous inserted individuals to see if any of them has this hash value. If a match is found, the process s repeated again until a unique individual yet not inserted has been found. As different binary representations have different decimal values, no two different individual can have same hash value.

Both k-SAT problem and Exactly-m k-SAT problem have been provided with the same initial population. Fitness functions are different for each of them. In case of k-SAT problem, if one of the literals becomes true for a particular individual, that clause is considered true and count is incremented. But in exactly-m k-SAT, a clause will be considered true only when exactly m literals become true. In that case a counter(temp) has been kept, when the value of temp exceeds m, no more literal is checked and the clause is considered false for this particular individual.

In case of one point crossover, random number has been chosen in interval [0,L-1]. Values of indices of two candidate individuals including and greater than this number are then swapped. In case of two point crossover, random number c and d has been chosen in interval [0,L-1].

Studying the table we can see that in case of k-SAT problem best fitness is found after just 1 iteration, because in most input clauses at least 1 literal is true and as a result the whole clause becomes true and the fitness is incremented. But in case of Exactly-m k-SAT problem highest number of iteration is needed and best fitness found is less than ideal case in most cases. By iterating more maybe we can achieve better results.