CHAPTER 6

CONCLUSIONS AND FUTURE SCOPE

6.1 Conclusions

In this thesis, Greedy Genetic Algorithm GGA to solve NP-Complete and NP-Hard problems, is proposed. The proposed GGA algorithm applies greedy approach in various operators of genetic algorithm such as greedy initial population generation, greedy cross over, and greedy mutation. The results of GGA algorithm confirms that the greedy approach is useful and provide better results for solving NP-Complete and NP-Hard problems.

The GGA algorithm is applied on Travelling Salesman Problem TSP, N-Queen Problem and 0/1 Multiple Knapsack Problem. While solving TSP problem the greedy approach is applied in initial population generation and cross over operation of genetic algorithm. The nearest neighbor first greedy approach is used in initial population generation and cross over operation to solve TSP problem. The GGA algorithm is applied on TSP instance EIL51. Eil76, Att48 and A280, and results are 6% to 19% better than the existing algorithms.

The proposed GGA algorithm is applied on some instances of N-Queen problem (with number of queens varies from 8 to 100). Greedy mutation operator is proposed to solve N-Queen problem using GGA. The greedy mutation operator mutates those two queens from a chromosome which are attacking with each other. The results are calculated in terms of execution time in seconds. The proposed GGA algorithm (with greedy mutation) solves 13 queen instances in 34.2 ms whereas the other algorithms solve it in more than 6 seconds.

The proposed GGA algorithm is also applied on 0/1 Multiple Knapsack Problem. The GGA algorithm generates initial population of chromosomes using greedy approach. In initial population generation, the GGA algorithm first arranges all the items in decreasing order of their profit/weight ratio. This greedy approach selects those items first which have high value of profit/weight ratio. Thus the chromosomes generated in initial population generation are generated using greedy approach and these chromosomes have high value of profit. The GGA algorithm is applied on *p01* instance of MKP problem which have 10 items and two knapsacks. The results ensure that the GGA algorithm perform 15.94% better than the standard genetic algorithm while solving MKP problem.

Following are the outcomes of the presented research:

1. The performance of the genetic algorithm is enhanced by applying hybrid genetic operators to solve NP-Complete and NP-Hard problems. The percentage improvements in the results are 6% to 19%.
2. Greedy approach is very useful in hybridization of genetic operators specially cross over (and mutation).
3. Hybrid genetic algorithm can be applied to solve many NP-Hard and NP-Complete problems. In this work it is applied on TSP, N-Queen and Knapsack problems.

6.2 Future Scope

This work improves the performance of Genetic Algorithm to solve NP-Complete and NP-Hard problems. Following are the future scope of this work.

1. The proposed genetic algorithm can be applied on other NP-Complete/NP-Hard problems such as Graph Coloring, Set Cover etc.
2. The GGA algorithm can be applied on larger instances of the problems such as TSP with thousands of cities, N-Queen with thousands of queens or MKP problem with thousands of items and hundreds of knapsacks.

3. Greedy techniques can be used in other genetic operators such as selection, encoding etc.
4. The performance of Genetic Algorithm can be tested on different values of genetic parameters such as cross over rate, population size, mutation rate etc.
5. The greedy approach can also be applied on other heuristic techniques such as Ant Colony Optimization, Particle Swarm Optimization PSO etc.