An Efficient Grouping Genetic Algorithm

R.Sivaraj
Research Scholar & Assistant Professor (Sr. Grade)
Department of CSE
Velalar College of Engineering and Technology
Erode, Tamil Nadu, India

Dr.T.Ravichandran
Principal & Research Supervisor
Hindusthan Institute of Technology
Coimbatore, Tamil Nadu, India

ABSTRACT
Genetic algorithm is an intelligent way for solving combinatorial, NP hard problems and many other problems which cannot be easily solved by applying traditional mathematical formula. The proposed method gives a new variant of the Standard Genetic algorithm which is very simple and will easily find the solution even for complex problems. It implements the concept of grouping to reach the optimal solution. The selection pressure which is a crucial factor that determines the efficiency of the algorithm is very much reduced in the proposed algorithm. The convergence velocity of the algorithm is greatly improved thereby reducing the time taken for the algorithm to reach the solution.

General terms
Grouping, Combinatorial NP hard problems, efficiency

Keywords
Genetic algorithms, Selection Pressure, Convergence Velocity

1. INTRODUCTION
Genetic Algorithms (GAs) are stochastic search algorithm based on the evolutionary ideas of natural selection and genetics. They follow the principles of Charles Darwin of "Survival of the Fittest" where the fittest individuals or the individuals which overtakes other individuals in the competition acquire the resources available. It is employed to direct the search process from a randomized initialization to more prospective specialized direction in the search space which is usually large. Many genetic operators are used to aid the process of exploration.

Genetic algorithms are considered as an optimization strategies where points in the problem space are analogous to organisms which are involved in the process of natural selection. The problem space is represented as population and the points as individuals where each one corresponds to possible solution to a given problem. Each point in the problem space is represented by a bit vector, is comparable to a chromosome with each bit position analogous to gene and each bit value analogous to an allele.

A fitness score or objective function value is calculated for each solution representing the abilities of an individual to `compete'. The individual with highest fitness score is selected to take part in the next generation. High fit individuals are given more chance to take part in the next generation by replacing the low fit individuals in the pool of population. The process is repeated until a best solution is found or a termination criterion specified by the user is reached.

Genetic algorithms are special, when compared to other randomized search procedures like Monte Carlo algorithms, because parent genomes are stored in memory and the offsprings are generated inheriting the characteristics from the parents. It is a robust method which utilizes only little information to search effectively in a large or complex search space.GA is nowadays widely used in multiple scientific domains and there arises a necessity for high performance which mainly focuses on speed and reliability. The proposed paper is of this kind where it introduces a new grouping genetic algorithm that can improve the performance of the standard genetic algorithm.

1.1 Pseudocode of the Standard Genetic Algorithm:
1. **[Start]** Generate random population of n Chromosomes (Suitable solutions for the problem)
2. **[Fitness]** Evaluate the fitness f(x) of each chromosome x in the population
3. **[New population]** Create a new population by repeating following steps until the new population is complete
   (i) **[Selection]** Select two parent chromosomes from a population according to their fitness (the better fitness, the bigger chance to be selected)
   (ii) **[Crossover]** With a crossover probability cross over the parents to form a new offspring (children). If no crossover was performed, offspring is an exact copy of parents.
   (iii) **[Mutation]** With a mutation probability mutate new offspring at each locus (position in Chromosome).
   (iv) **[Accepting]** Place new offspring in a new population
4. **[Replace]** Use new generated population for a further run of algorithm
5. **[Test]** If the end condition is satisfied, stop , and return the best solution in current population
6. **[Loop]** Go to step 2

The standard genetic algorithm first initializes the population and then selects parents for mating. Selection deserves a crucial part in Genetic algorithm since it is the one which mainly determines the evolutionary search spaces. It is used to improve the chances of the survival of the fittest individuals. There are many traditional selection mechanisms like Roulette Wheel selection, Rank Selection, Tournament Selection and many user specified selection mechanisms specific to the problem definition. Selecting the parents in the whole population will definitely increase the selection pressure. When other genetic operators like crossover, mutation are applied, it takes a long run to complete the iteration and to converge to an optimal solution. The proposed algorithmic approach tries to overcome this problem by reducing the selection pressure and thereby improving the convergence velocity.

2. PROPOSED METHODOLOGY

GAs performs well when the population is divided into sub populations [2]. A variant of the standard approach is used in this paper. The proposed algorithm will divide the genes within the chromosomes into groups and will apply the genetic operators to attain the solution. The chromosome is divided into 'n' number of groups such that the first group will consist of first 'i' genes, the second group will consist of next i genes and so on. Rather than applying the genetic operators to the entire chromosome, the proposed methodology applies the operators to the groups of genes and tries to find out the best solution in the group and will finally combine the solutions. Since only the subsets of the genes within the chromosomes are used in each group and they are selected for crossover and mutation, the selection pressure is greatly reduced. The algorithm also will reach the optimal solution space quickly compared to the standard genetic algorithm. The optimal solution for each group can also be easily reached. Since the genes are divided into groups, we call the proposed methodology as Gene Grouping Genetic algorithm (GGGA).

The performance of Gene Grouping Genetic algorithm is evaluated with different selection mechanisms and the genetic operators as the choice of these parameters are very crucial to the success of GA [3, 4].The selection schemes that are tried for this implementation are Roulette Wheel Selection, Tournament Selection, Rank Selection combined with elitism [7]. In the Roulette wheel selection, the first step is to calculate the cumulative fitness of the whole population through the sum of the fitness of all individuals. After that, the probability of selection is calculated for each individual as being pseli = fi/Pfi. Then, an array is built containing cumulative probabilities of the individuals. The expected count is calculated as fi/f where f is the average of the fitness functions. The integer part of the expected count is taken as the actual count probability of the chromosome and the chromosomes with low count are replaced with chromosomes with high count.

Therefore, individuals are selected according to their probabilities of selection. Tournament selection [5] involves running several "tournaments" among a few individuals chosen at random from the population. The winner of each tournament (the one with the best fitness) is selected for crossover. Rank Selection [5] is the process of assigning the ranks to all the individuals based on the fitness values and the individuals with highest rank are chosen for selection. Elitism is the process of selecting the few best parents and replacing always the worst. The few best individuals will be retained over the generations and will not take part in the crossover or mutation. This means all the individuals will be more or less the same in the population and the population will converge quicker improving the convergence velocity of the algorithm. In mutation, the randomly chosen bits are inverted ie., the chromosome bits 1s are changed to 0s and the 0s are changed to 1s. This is done in order to introduce diversity of values in the chromosomes.It has been shown in [6] that the GA's that maintain the best discovered solution either before or after the selection operator asymptotically converge to the global optimum.

Crossover is the process of combining or mating two chromosomes to produce new offsprings in order to combine the best genes in different chromosomes to reach an optimal solution [7]. The commonly used crossover operators are One point crossover, Two point Crossover and Uniform Crossover. In one point Crossover, one random position in the chromosomes is chosen. The genes in both chromosomes are interchanged after that bit position such that child 1 is head of chromosome of parent 1 with tail of chromosome of parent 2 and child 2 is head of 2 with tail of 1. In two point crossover, two random positions in the chromosomes are chosen. The genes in both parents are exchanged after the two bit positions to form two offsprings. In uniform Crossover [8], a random mask is generated. If the random mask bit is 1, the bit is chosen from the first parent and if it is 0, then the bit is chosen from the second parent. ***Mutation*** is the process of randomly flipping the bits with very low probability.

In the algorithms where the chromosomes are represented in a way that promotes easy crossover, the fitness evaluation is very expensive as in [9]. In the algorithms where the fitness evaluation is simple, either the crossover operation is complicated or it needs to be repeatedly applied on chromosomes to get legal strings [10], [11]. In this sense, selection and crossover are complementary to each other in terms of computational complexity. So whenever Genetic algorithm is applied, appropriate crossover and mutation operators have to be chosen.

The parallel execution of the gene groups will execute faster and will reach the solution quickly. The selection process and the genetic operators are simultaneously applied to all the groups and finally the solutions are combined which tends to decrease the selection pressure. Since the genes are applied genetic operators only in groups rather than to the entire chromosome, the overhead is greatly reduced in the proposed Gene Grouping Genetic Algorithm. The skeleton of the proposed algorithm given in figure 1 shows the steps that are involved in it.

3. IMPLEMENTATION

To analyze the performance of GGGA, the 0/1 knapsack problem [12] is chosen. The problem can be stated as follows: Given a set of n objects, each object Oi is associated with a weight Wi and profit Pi and a knapsack capacity C. The problem is to find a binary vector X=<x1, x2……, xn>, in order to maximize the profit, subject to the constraint $Wixi \leq C$, where xi = 0 or 1 for i= 1….n. The encoding scheme that is chosen for this problem is binary encoding where the chromosome representation is a bit vector consisting of zeroes and ones. If the object i is present in the knapsack, its corresponding gene in the chromosome will take the value of 1 and if it is not included in the knapsack, its corresponding gene will take the value of 0.

The genetic parameters chosen for this implementation are:

No of bits in each Chromosome: 200
Elitism %: 10
Crossover Probability Pc: 0.85
Mutation Probability Pm: 0.05

No of Groups in GGGA: 4 (4 groups *50 genes)

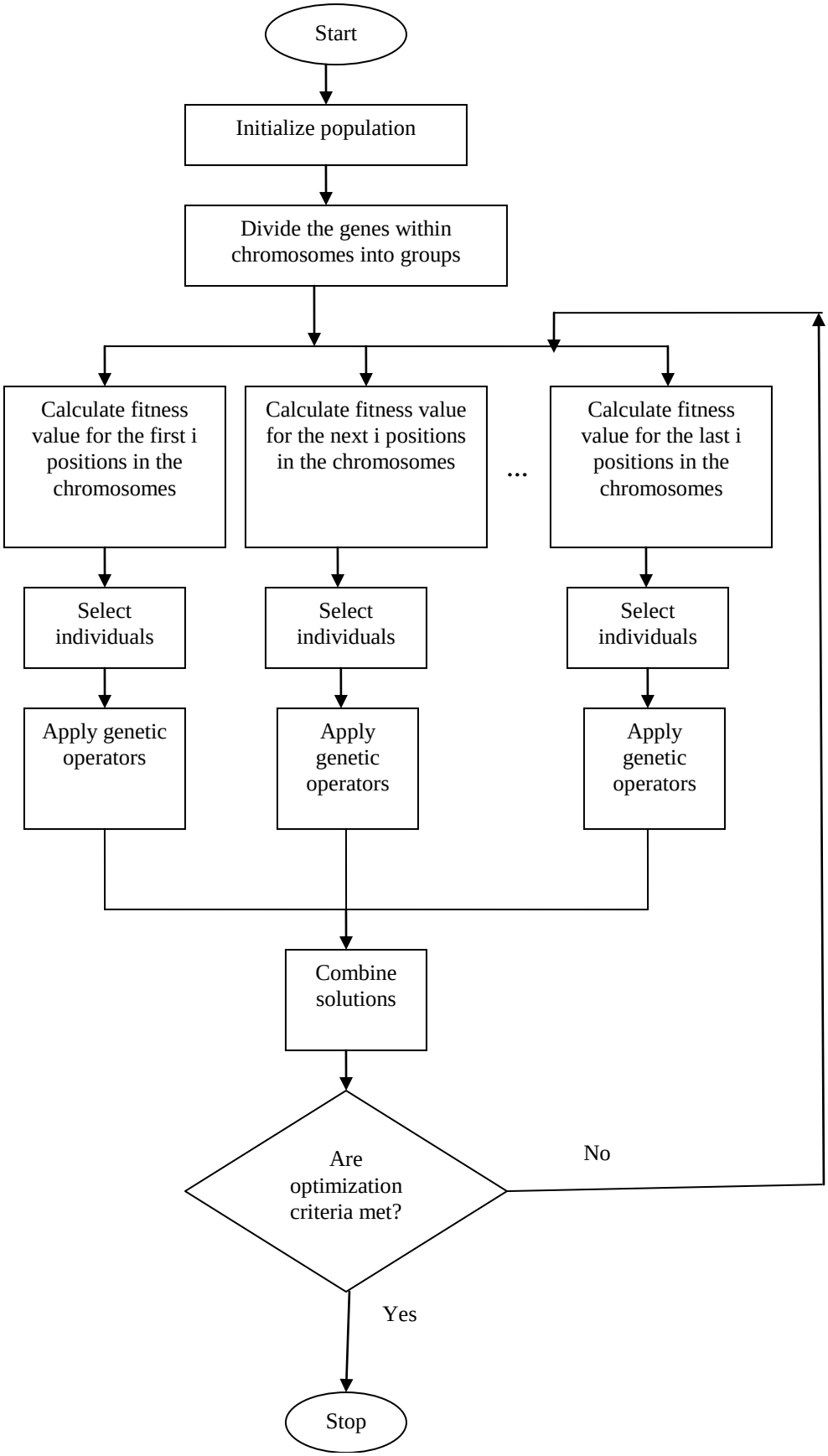


Figure 1.Gene Grouping Genetic algorithm

4. EMPIRICAL RESULT ANALYSIS

The experiments were carried out many times with varying number of objects with different weights and profits are taken. The algorithm efficiency is analyzed by varying the genetic operators and other parameters.

4.1 Effect of varying the Selection schemes:

The GGGA and SGA is run with commonly used selection schemes like Roulette Wheel Selection, Tournament Selection, Rank Selection all combined with elitism and the results are compared. The results clearly show that the GGGA performs better with all selection mechanisms. As Elitism always retains the best individuals over the iterations, its solution will be optimal or sub optimal. Hence in all our experiments, elitism is used to carry the few best individuals over the generations until it satisfies the stopping criterion and only the remaining individuals are used for selecting the parents for crossover. Figure 2 shows the comparison of SGA and GGGA when Roulette Wheel Selection is used.

Figure 2.GGGA Vs SGA with Roulette Wheel Selection

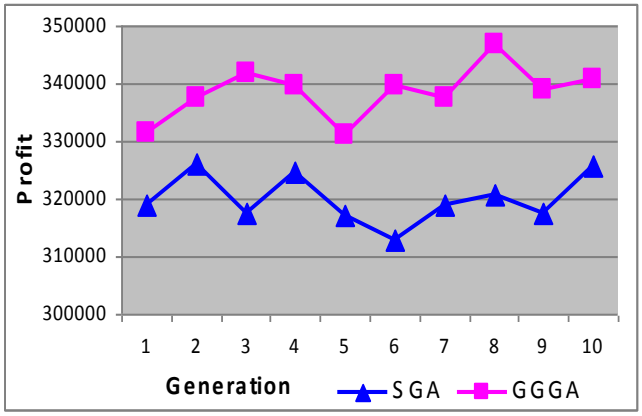


Figure 3 and Figure 4 shows the comparison of profits obtained from SGA and GGGAwith Rank Selection and Tournament Selection respectively

Figure 3.GGGA Vs SGA with Rank Selection

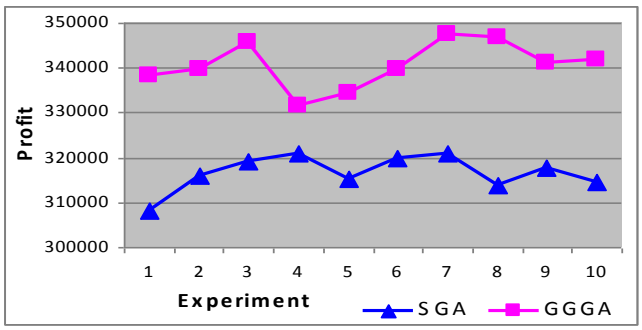
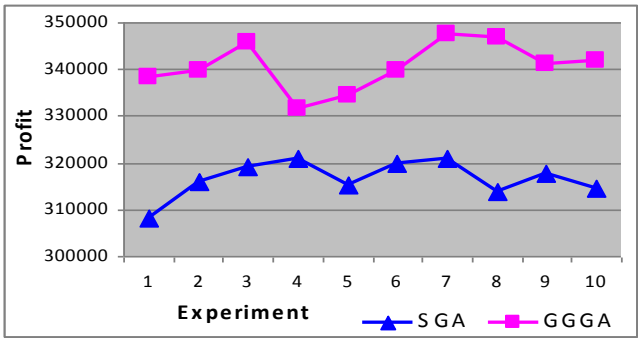


Figure 4.GGGA Vs SGA with Tournament Selection



4.2 Effect of varying the crossover operators:

The crossover is the vital process in Genetic algorithm which greatly impacts the final solution. The chromosomes for mating will be chosen according to the given crossover probability Pc.

Figure 5.GGGA Vs SGA for One point crossover

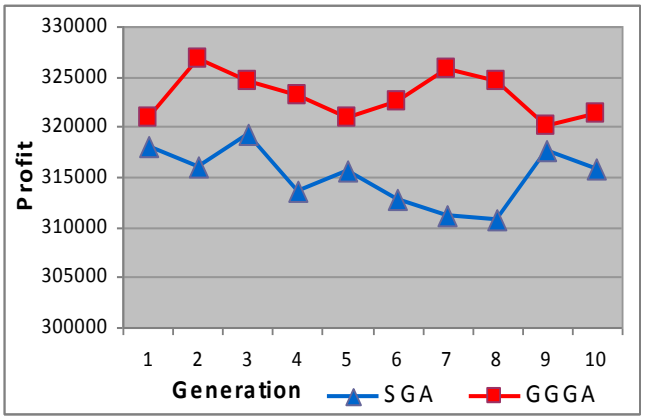


Figure 6. GGGA Vs SGA for Two point crossover

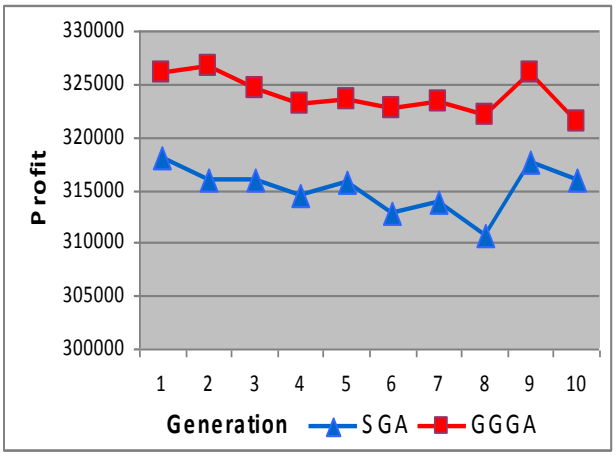
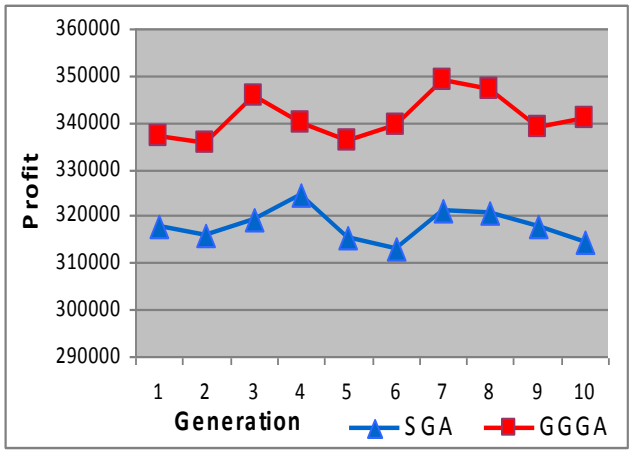


Figure 7.GGGA Vs SGA for Uniform crossover

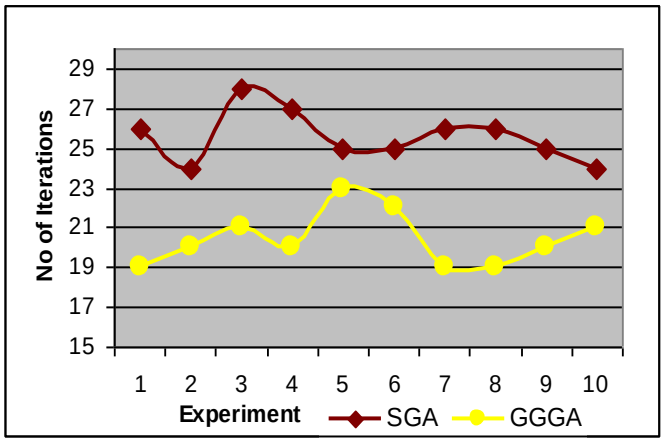


Figures 5, 6 and 7 show the profits obtained when GGGA is applied with One point Crossover, Two point Crossover and Uniform Crossover respectively and the results are compared with Standard Genetic Algorithm. The results clearly show that for all the crossover types, GGGA tends to perform better and produce maximum profit. Among the three types of crossover that are analyzed here, GGGA results in high profit when uniform crossover is used.

4.3 Convergence velocity:

Convergence velocity is the speed with which the algorithm reaches the solution. As the genes in the chromosomes are grouped and they are run simultaneously, this will reduce the time taken for processing. The performance of GGGA with SGA is compared and the results are given below in the figure 8.GGGA takes less number of iterations than SGA to reach the optimal solution.

Figure 8.Convergence Velocity of SGA Vs GGGA



5. CONCLUSION

The selection pressure is an important factor in the genetic algorithm which determines the chromosomes to be selected for reaching the selection. The proposed Gene Grouping Genetic algorithm reduces the selection pressure and hence gains its superior performance over standard genetic algorithm. But GGGA has a limitation that it is applicable only for complex problems where they can be divided into non overlapping sub-problems and whose solutions of the sub problems can again be combined into the solution of the original problem. Thus GGGA produces better and efficient results than SGA when applied to complex problems.

6. REFERENCES

[1] Holland, J.H., Adaptation in Natural and Artificial Systems, 2nd Edition, MIT Press, 1992.

[2] Tanese, R, 'Distributed Genetic algorithm', Proceedings of the Third International Conference on Genetic Algorithms, 1989, pp. 434-439.

[3] Grosso, P.B., 'Computer Simulations of Genetic Adaptation: Parallel Subcomponent Interaction in a Multilocus Model', Doctoral Dissertation, the University of Michigan, 1985.

[4] Pettey, C, B., Leuze, M. R., & Grefenstette, J. J., ' A Parallel Genetic Algorithm', In Proceedings of the Second International Conference on Genetic Algorithms, 1987, pp.155-161

[5] Back, T, 'Evolutionary Algorithms in Theory and Practice', Oxford University Press, 1996.

[6] G. Rudolph, "Convergence analysis of canonical genetic algorithms, "IEEE Trans. Neural Networks, vol. 5, no. 1, pp. 96–101, 1994.

[7] Goldberg, D.E. 'Genetic Algorithms in Search, Optimization and Machine Learning', Addison-Wesley, 2003.

[8] M. Gen, R. Cheng, "Genetic Algorithms and Engineering Optimization," John Wiley & Sons, Inc., New York, 2000.

[9] J. N. Bhuyan, V. V. Raghavan, and V. K. Elayavalli, "Genetic Algorithm for clustering with an ordered representation," in Proc. 4th Int. Conf. Genetic Algorithms. San Mateo, CA: Morgan Kaufman, 1991.

[10]D. R. Jones and M. A. Beltramo, "Solving partitioning problems with genetic algorithms," in Proc. 4th Int. Conf. Genetic Algorithms. San Mateo, CA: Morgan Kaufman, 1991.

[11]G. P. Babu, "Connectionist and evolutionary approaches for pattern clustering," Ph.D. dissertation, Dept. Comput. Sci. Automat., Indian Inst.Sci., Bangalore, Apr. 1994.

[12]Martello, S, Toth, P: ' Knapsack problems: Algorithms and Computer Implementations', J.Wiley & Sons, 1990