Journal of Computer Science 7 (7): 1033-1037, 2011
ISSN 1549-3636

An Improved Clustering Based Genetic Algorithm for Solving Complex NP Problems

[1]R. Sivaraj and [2]T. Ravichandran
[1]Department of CSE, Velalar College of Engineering and Technology, Erode, India
[2]Hindusthan Institute of Technology, Coimbatore, India

Abstract: Problem statement: The selection process is a major factor in genetic algorithm which determines the optimality of solution for a complex problem. The selection pressure is the critical step which finds out the best individuals in the entire population for further genetic operators. The proposed algorithm tries to find out the best individuals with reduced selection pressure than standard genetic algorithm which is commonly used. **Approach:** The selection process is refined in the proposed algorithm by using the concept of clustering rather than traditional selection mechanisms like Roulette wheel selection, Rank selection, Tournament selection. **Results:** As the selection process is improved in our approach, the convergence velocity of the genetic algorithm is improved by reaching the optimal solution quickly and the optimality of the solution is also fine-tuned. **Conclusion:** A new variant of the standard genetic algorithm is proposed which reduces the execution time of the algorithm by gearing up the selection process to reach the most efficient solution. The fit individuals are selected for crossover and mutation in all generations thereby reaching the solution without much complex process.

Keywords: Genetic algorithm, clustering algorithm, selection pressure, crossover and mutation

INTRODUCTION

Genetic algorithms are adaptive heuristic search algorithms based on the evolutionary concept of natural selection and genetics. It follows the Darwin's principles of "Survival of the fittest" where the fit best individuals retain their positions overtaking the weaker individuals in a group which they compete for the limited resources. Genetic algorithm is robust when compared to other searching mechanisms. Even though it is generally said to be random process, it is not actually random. Instead, it chooses the best individuals in each iteration thereby moving fast towards the stable optimal solution from the initial random population of individuals chosen.

In genetic algorithm, the potential solution to a problem can be represented by a set of parameters called genes and the genes are combined together to form a structure called chromosome. N chromosomes are collectively called as population. In genetic terms, genes are called as genotype and chromosomes are called as phenotype. Initially, the chromosomes in the population are chosen at random. It then applies recombination genetic operators to these structures so as to proceed towards final solution. By calculating the fitness value for all chromosomes, it evaluates these structures and allocates reproductive opportunities in the next generation in such a way that those chromosomes which provide a better solution to the target problem are given more chances to reproduce than those chromosomes which represent poorer solutions. The goodness of a solution is typically determined with respect to the current population. Genetic algorithms are usually seen as function optimizers although the range of problems and areas to which genetic algorithms have been applied is very broad. They are widely used in solving many complex Non-Deterministic Polynomial (NP) problems in different domains whose time efficiency cannot be specified in polynomial time (Patvichaichod, 2011; Kannaiah *et al*., 2011).

The general outline of the standard genetic algorithm (Goldberg, 1989) is given below:

- Choose the initial population of individuals at random
- Evaluate the fitness of each individual in that population using objective function
- Repeat the following steps in each generation until termination criterion (time limit, sufficient fitness achieved, fixed no of iterations) is achieved
- Select the best-fit individuals for crossover
- Breed new individuals through crossover and mutation operations to give birth to offspring

Corresponding Author: R. Sivaraj, Department of CSE, Velalar College of Engineering and Technology, Erode, Tamil Nadu, India

- Evaluate the fitness value of all new individuals
- Replace least-fit individuals with new high fit individuals

The process of representation of chromosomes in terms of genes is called as encoding. There are many types of encoding techniques like binary encoding, value encoding, permutation encoding. They differ in the way the genes are represented and used for processing. The basic steps involved in GA are Initialization, Selection, Reproduction and Termination (Kalyanmoy, 2004).

Fitness function: A fitness function must be formulated for each problem that is to be solved by genetic algorithm. For a particular chromosome, a fitness function or objective function returns a single numeric fitness value or "figure of merit" which is proportional to the "utility" or "ability" of that chromosome in the entire population consisting of n chromosomes. For many problems like functional optimization, objective function value is enough to attain a solution. But for complex combinatorial problems, a combination of performance measures relating to that specific problem will only drive towards the optimal solution.

Selection: Parents for this reproduction (mating or crossover) are selected using some mechanisms (Goldberg, 1989) like Roulette Wheel selection, Rank selection, Truncate selection, Tournament selection, Boltzmann selection. Although all selection mechanisms (Sivaraj and Ravichandran, 2011) have the final target of choosing the best chromosomes for the next crossover phase, they differ in the way they use to evaluate them. They retain the best individuals over the iterations by replacing the worst individuals which have the low probability of being carried over to the next generations.

Genetic recombination operators: Although many genetic recombination operators are available, the commonly used ones are crossover and mutation Fig. 1 and 2.

Reproduction (Crossover): During Reproductive phase, the chromosomes with high fitness values are recombined to form new chromosomes which constitute the individuals for the next generation. Crossover takes two individuals, cuts them at random positions to gets head and tail segments. The tail segments are then swapped between two parents to form two new full length chromosomes or offsprings. The offsprings thus produced inherit the genes and their characteristic from both parents which may eventually turn out to be the optimal solution when they are subjected to the same process in further generations.

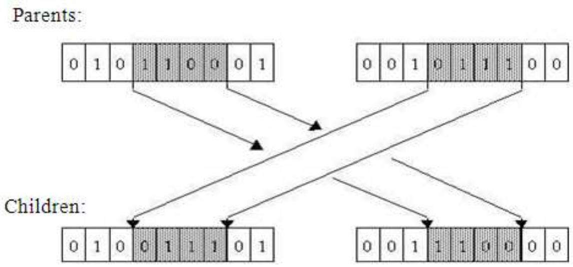


Fig. 1: One point crossover

Mutation is the process of randomly flipping a bit in the entire chromosome with some fixed small probability. It is done to introduce some form of diversity among the chromosomes without sacrificing the characteristics of the parents.

In the process of genetic algorithm, sub-optimal solutions may be attributed to selection pressure in traditional selection mechanisms which ignore the weaker individuals to a larger extent. If they are given some better chance of survival in the next generation, there will be improved chance for them to converge to an optimal solution. The proposed approach is a new variant of the standard genetic algorithm which incorporates the advantageous feature of clustering in selection process of genetic algorithm.

MATERIALS AND METHODS

In many studies, to cluster similar objects using k-means clustering algorithm, genetic algorithm is used (Maulik and Bandyopadhyay, 2000; Tiwari *et al*., 2010). But our approach is proposed in a new direction which in order to improve the efficiency of the final global optimal solution of genetic algorithm k-means clustering is incorporated within.

K means clustering: Clustering is an unsupervised learning mechanism used to group similar objects into clusters. Although different clustering techniques are available, there is no general strategy that works equally well in different problem domains. However, it is better to use some simpler clustering mechanism which runs more number of times rather than complex mechanisms which needs to be run only once. K-means clustering has been a very popular technique for partitioning large data sets with numerical attributes. It is classified as a partitional or non-hierarchical clustering method. It is defined as follows: Given a set D = {X1, . . . , Xn} of n numerical data objects, a user defined natural number $k \leq n$ and a distance measure d, the k-means algorithm is aimed at finding a partition C of D into k non-empty disjoint clusters $C_1, . . . ,C_k$

where $C_i \cap C_j = \Phi$ and $U_{i=1}^{k} Ci = D$ such that the overall sum of the squared distances between data objects and their cluster centers is minimized.

The basic step of k-means clustering is simple. In the beginning, the number of clusters K is defined and the centroid or center of these clusters. Any random objects can be taken as the initial centroids or the first K objects in sequence can also serve as the initial centroids.

Then the K means algorithm will do the three steps below until convergence:

- Iterate the following steps until the clusters become stable (no objects move among groups): Determine the centroid coordinate
- Determine the distance of each object to the centroids
- Group the object based on minimum distance

The distance between objects can be calculated by using Hamming distance, Manhattan distance or Minkowski distance.

To show the performance of the proposed clustering genetic algorithm, a complex NP hard 0/1 knapsack problem is chosen. The Knapsack problem is an example of a combinatorial optimization problem, which seeks for a best solution from among many solutions. It is concerned with a knapsack that has positive integer volume (or capacity) V. There are 'n' distinct items that may potentially be placed in the knapsack. Item i has a positive integer volume (or capacity) Vi and positive integer benefit (or profit) Bi. Let X_i denotes how many copies of item i are to be placed into the knapsack. In 0/1 knapsack problem, only one copy of an item can be placed in the knapsack. Hence Xi is always equal to 1 for all the items selected.

The primary goal is to:
Maximize:

$$\sum_{i=1}^{N} B_i X_i$$

Table 1: Example of knapsack problem solution

Item A	Item B	Item C	Volume of the set	Benefit of the set
0	0	0	0	0
0	0	1	9	5
0	1	0	8	8
0	1	1	17	-
1	0	0	6	4
1	0	1	15	-
1	1	0	14	10
1	1	1	23	-

Subject to the constraints:

$$\sum_{i=1}^{N} V_i X_i \leq V$$

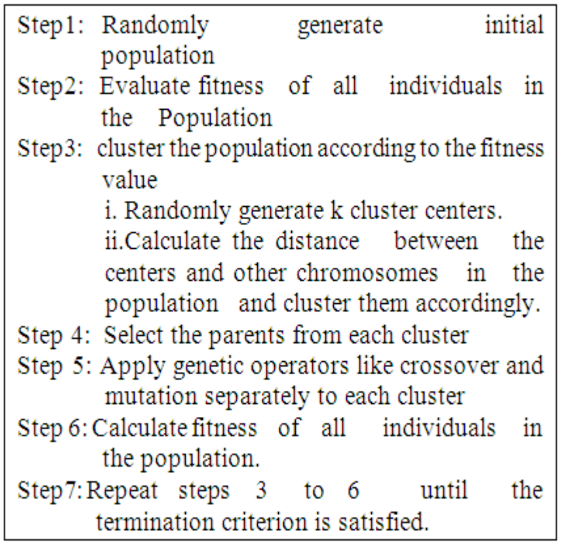
Example of 0-1 knapsack problem: Suppose there is a knapsack that has a capacity of 14 cubic inches and several items of different volumes and different benefits. The primary objective is to include in the knapsack only those items that will have the greatest total benefit that fit within the knapsack's capacity. There are three potential items (labeled 'A,' 'B,' 'C'). Their volumes and benefits are as follows:

Item #	A	B	C
Benefit	4	6	5
Volume	6	8	9

For this problem there are 2^3 =8 possible subsets of items. In order to find the best solution, a subset that meets the constraint and has the maximum total benefit has to be identified.

Table 1 clearly shows that in this example, only 7th row (110) satisfies the constraint. Hence, the optimal benefit for the given constraint (V = 14) can only be obtained with one quantity of A, one quantity of B and zero quantity of C and it is 10.If number of items is less, solution can be easily found. If it is too large to apply simple algorithms to find the best solution, some optimization techniques like Genetic algorithm has to be applied.

Proposed methodology: The proposed Clustering Genetic Algorithm (CGA) tries to reduce the selection pressure of the genetic algorithm by combining the features of k means clustering. The individuals in the initial population are taken at random. The fitness values of the individuals are calculated. Then k means clustering is incorporated. The initial centroids are taken at random. The distance between the individuals are calculated where the fitness value is taken to calculate the similarity measure. The similar individuals are grouped into a cluster. Similarly k clusters are formed with each cluster having individuals with similar fitness values. Traditional selection mechanism is used within each cluster to select parents for crossover and mutation. After the application of these genetic recombination operators, the individuals for the next iteration are produced. In the next iteration, again k means clustering is used to change the cluster centers and the genetic algorithm steps are continued until termination criterion is satisfied. The individuals migrate to other clusters which has minimum distance to cluster center if its fitness value changes (improves or degrades).



```
Step1: Randomly generate initial population
Step2: Evaluate fitness of all individuals in the Population
Step3: cluster the population according to the fitness value
       i. Randomly generate k cluster centers.
       ii.Calculate the distance between the centers and other chromosomes in the population and cluster them accordingly.
Step 4: Select the parents from each cluster
Step 5: Apply genetic operators like crossover and mutation separately to each cluster
Step 6: Calculate fitness of all individuals in the population.
Step7: Repeat steps 3 to 6 until the termination criterion is satisfied.
```

Fig. 2: Outline of the proposed clustering genetic algorithm

RESULTS

The CGA is run with different genetic parameters and the results are analyzed for 200 items. The selection mechanism chosen for implementation is Tournament selection with 10% elitism and one point crossover is chosen for mating. Tournament selection is the process of conducting tournaments for ‘n’ individuals and the winner of each tournament is chosen as parent. Results in the literature (Zitzler *et al*., 2000) show clearly that elitism can speed up the performance of the GA significantly; also it helps to prevent the loss of good solutions once they have been found. Hence elitism is combined with selection mechanism to prevent the loss of good solutions for the next generations once they have been found. The Crossover probability P_c and Mutation probability P_m are chosen as 80% and 1% respectively.

Impact of varying number of clusters: In k means clustering, the value of k is user defined and the value chosen for k will have its heavy impact on the result of clustering. If the number of clusters is very low, the diversity of individuals cannot be achieved. If it is very large, it will create an unnecessary overhead. So it should be chosen accordingly for the specific problem. In this implementation, the number of clusters (k) is varied from 3 to 6 and the results are compared to show the performance of CGA. As shown in Fig. 3, CGA results in high profit when compared to implementation of SGA with different number of clusters.

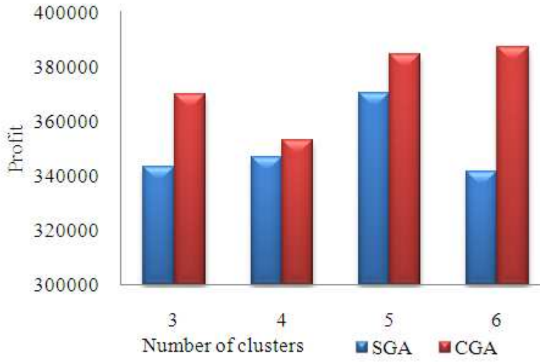


Fig. 3: Impact of varying the number of clusters

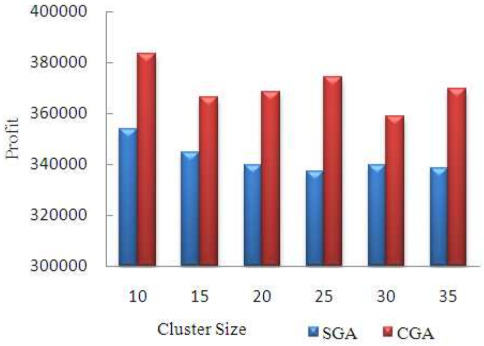


Fig. 4: Impact of varying cluster size

Impact of varying cluster size: The number of chromosomes or individuals for each cluster is varied and the results are reported. The individuals within a cluster should have high cohesion such that their similarity should be more. If number of individuals grouped in a cluster increases, their similarity decreases and this will eventually affect the performance of the final solution. The performance improvement of CGA comparing to SGA is shown in Fig. 4 which shows that if cluster size is low, CGA tends to perform better.

Impact of varying population size: The population size chosen will also directly impact the optimality of the solution. If the population size is set too low, it will soon converge to a local optimum. If it is set too high, it will unnecessarily process all the individuals which take more time to execute. So, the population size should be chosen carefully. As shown in Fig. 5, the results clearly show the superior performance of CGA when the population size is varied. In CGA, the population size will have little impact on the efficiency of the final solution compared to SGA and it has overall better performance than SGA.

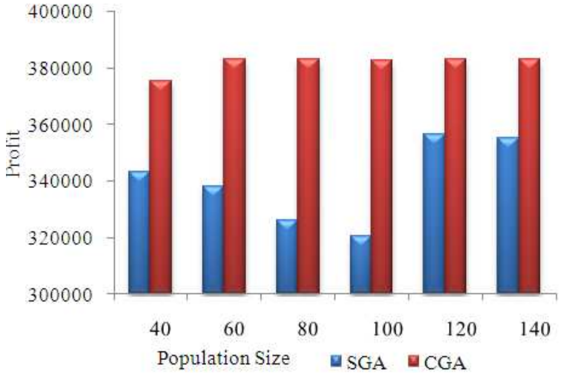


Fig. 5: Impact of varying population size

DISCUSSION

The results clearly shows that the proposed Clustering Genetic Algorithm has reached better profit by selecting optimal subset of items satisfying the constraints . The final solution space of the Genetic Algorithm critically depends upon the genetic parameters chosen. In K-means clustering also the value of k to be chosen is a critical factor to be determined. Hence CGA is run by changing these operators. Figure 3-5 depicts that CGA shows better performance with high profit than SGA with varying genetic parameters like number of clusters, cluster size population size.

CONCLUSION

CGA shows much greater performance than standard genetic algorithm with different genetic parameters chosen. It introduces a checkpoint prior to selection mechanism in each generation by the usage of k-means clustering algorithm. It thus helps in selecting only the best chromosomes to be carried over to further generations and also helps in reducing the pressure for selection process. However, to gain better performance a compromise should be made for the time it takes to complete a generation because of the inclusion of clustering. When the final optimality of the solution is considered, this is negligible compared to the SGA performance which operates on the entire population all the times. Hence with improved performance of CGA, many complex NP problems in different domains and with different functionalities can be easily solved.

REFERENCES

Goldberg, D.E., 1989. Genetic Algorithms in Search, Optimization and Machine Learning. 1st Edn., Addison-Wesley, Reading, Massachusetts, ISBN-10: 0201157675, pp: 432.

Kalyanmoy, D., 2004. Optimization for Engineering Design: Algorithms and Examples. 1st Edn., Prentice-Hall of India, New Delhi, ISBN-10:812030943X, pp: 396.

Kannaiah, S.K., J. Thangavel and D.P. Kothari, 2011. A genetic algorithm based multi objective service restoration in distribution systems. J. Comput. Sci., 7: 448-453. DOI: 10.3844/jcssp.2011.448.453

Maulik, U. and S. Bandyopadhyay, 2000. Genetic algorithm based clustering technique. Patt. Recog., 33: 1455-1465. http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.109.2138&rep=rep1&type=pdf

Patvichaichod, S., 2011. An improved genetic algorithm for the traveling salesman problem with multi-relations. J. Comput. Sci., 7: 70-74. DOI: 10.3844/jcssp.2011.70.74

Sivaraj, R. and T. Ravichandran, 2011. A review of selection methods in genetic algorithms. Int. J. Eng. Sci. Technol., 3: 3792-3797. http://www.ijest.info/docs/IJEST11-03-05-190.pdf

Tiwari, A.K., L.K. Sharma and G.R. Krishna, 2010. Entropy weighting genetic k-means algorithm for subspace clustering. Int. J. Comput. Appli., 7: 27-30. DOI: 10.5120/1263-1628

Zitzler, E., K. Deb and L. Thiele, 2000. Comparison of multiobjective evolutionary algorithms: Empirical results. Evolut. Comput., 8: 173-195. DOI: 10.1162/106365600568202