

## Performance Analysis of Multi Clustered Parallel Genetic Algorithm with Gray Value

<sup>1</sup>Vishnu Raja, P. and <sup>2</sup>V. Murali Bhaskaran

<sup>1</sup>Department of CSE, Kongu Engineering College, Perundurai, India

<sup>2</sup>Pavaai College of Engineering, Pachal, Namakal, India

---

**Abstract: Problem statement:** Encoding is one of the major factors in genetic algorithms, for a complex problem the optimality for the problem is determined. The encoding mechanism is the initial step which forms the chromosome to form the best fit individuals in the entire population. The proposed method tries to encode the initial population by the binary encoding and by gray encoding. **Approach:** The problem we had identified for the experiment was a Knapsack p. The chromosomes were generated for the knapsack problem with random and the individuals were clustered into in clusters. In each cluster the parent was selected and the selected chromosome was gray coded for further genetic operators. **Results:** By implementing gray encoding mechanism, the experiment result shows the improvement in profit when large population was used. The results were analyzed for the algorithm by implementing it by changes in population size, changes in group size and change in mutation rate. **Conclusion:** The Proposed genetic algorithm reduces the execution time of the algorithm by reducing one step in the genetic operators to reach the optimal solution. The best fit individual is produced in a simple process by applying a mutation reproduction operator to the gray value.

**Key words:** Gray encoding, genetic algorithm, clustering algorithm, encoding mechanism, crossover and mutation operator, Multi Clustered PARALLEL Genetic Algorithm (MCPGA)

---

### INTRODUCTION

Multi Clustered Parallel Genetic Algorithm (MCPGA) is a special kind of genetic algorithm that work under the principle of “Natural evolution and Genetics” (Back, 1996). Based on this principle the process which enables the individuals in one generation to modify or improve in the further generations.

The basic principles of genetic algorithms were first implemented rigorously by Holland (1975). Here they simulate the process of GA in natural populations which is essential to evolution. Generally GAs in search methods are used is very difficult to solve a problem for identifying solutions to a particular problem. GA attempt to mimic the success of natural evolution by emulating its mechanisms. Essentially, a population of possible solution is maintained and the fittest individuals remain in the population and reproduce the best individuals for the next generation are created.

In the traditional genetic algorithm, the representation of the chromosome or individuals is a bit string of fixed length. Each bit position in the chromosome represents a particular feature of the individual and the genuine value of the feature represents whether the particular feature is added in the

chromosome or not. Generally GAs has a number of operators with its related parameters. The operators are set initially for a GA operation and remain the same till the experiment is completed. Many researchers have adopted schemes that have the parameters and/or the operators change during the GA experiment. This study will explore how GA adopts itself to binary and non binary operators. In this study, we have proposed a new genetic algorithm named GMCPGA-Grey encoded multi clustered parallel genetic algorithm for solving the 0/1-knapsack problem and compared the characteristics of the algorithm on the basis of profit and efficiency as parameter.

**Knapsack problem:** In order to obtain the most reliable and better result, the proposed GMCPGA is applied to 0/1 knapsack problem. Because the knapsack problem to implement is easy and results can be analyzed quickly.

The widely read 0/1 knapsack problem is NP-hard in nature and has practical importance. Knapsack problem is defined by a set of items associated with its weight and the profit with each item and the capacity is limited to its maximum of the knapsack. The objective of the problem is to identify a subgroup of items which

has the maximum profit of the group with a constraint, i.e. the total weight does not exceed the given capacity. Generally the knapsack is defined as follows.

Given a set of  $m$  items and a set of  $n$  knapsacks with profit  $P_{ij}$  of the item  $j$  according to knapsack  $i$ ,  $W_{ij}$  the weight of the item  $j$  according to the knapsack  $j$  and capacity  $C_{ij}$  for the knapsack  $i$ :

Vector:

$$x = (x_1, x_2, \dots, x_m) \in \{0,1\}^m$$

Such that:

$$\forall \{1,2,\dots,n\}: \sum_{j=1}^m W_{ij} x_j \leq C_j$$

And:

$$f_i(x) = (f_1(x), f_2(x), \dots, f_n(x))$$

Is maximum, where:

$$f_i(x) = P_{ij} X_j$$

And  $X_j=1$ , if item  $j$  is selected otherwise 0.

### MATERIALS AND METHODS

Here, we discussed about the gray code when applied to the genetic parameters and the proposed GMCPGA is discussed in detail.

**Encoding schemes:** Usually when genetic algorithms are applied to real world problems, the parameters are encoded as a bit of strings using standard encoding schemes namely, binary encoding, value encoding, permutation encoding and gray encoding. In which in most of the genetic algorithms binary encoding, the representation of 0s and 1s is used. The bit string of the each parameter are combined together to form a single bit string called individuals (chromosomes) which represents the entire parameter in a single vector which are included in the problem and each bit value represents a gene value in a chromosome:

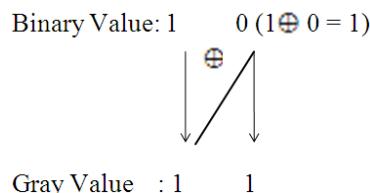
1 0 1 1 0 1

Gray encoding, is a different way of coding mechanism which contains the property with an increase in the parameter value by one step corresponds to a change in the code by a bit. The formula for converting the binary code to a gray code is given by:

$$g_k = \begin{cases} b_1 & \text{if } k=1 \\ b_{k+1} \oplus b_k & \text{if } k > 1 \end{cases}$$

where,  $g_k$  is the  $k^{\text{th}}$  gray code bit and  $k^{\text{th}}$  binary code bit is represented by  $b_k$ . The bits are represented from 1 to  $n$  from the MSB position (left). The symbol  $\oplus$  represents the addition mod 2 for converting binary to gray code.

Consider, 101101 an individual with the binary string representation. The equivalent gray value is 110110:



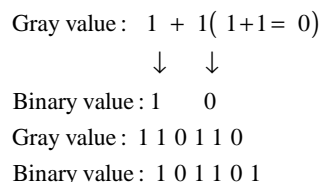
i.e., 101101 and its equivalent gray code is 110110.

By considering the above example, it is noticed that in both the binary and gray value the Most Significant Bit (MSB) is same. With this gray code value when the crossover is applied, it can be seen that then bits of MSB of the binary code determines the MSB  $n$  bits of the gray code. Thus again when new offspring is produced, each offspring can be seen as a resemblance of the parent from which the MSB is obtained. By changing the  $(n+1)$  Gray bit code it affects only the  $(n+1)^{\text{st}}$ ,  $(n+2)^{\text{nd}}$ . Binary code bits, the maximum size of the perturbation is same as in the binary code case. So, when gray code value is used in the genetic algorithm, the crossover is not necessary to produce the new offspring.

The conversion of the gray value to its equivalent binary value is obtained by the following equation:

$$b_k = \sum_{i=1}^k g_k$$

where,  $b_k$  is the binary value and  $g_k$  is the Gray value for the  $k^{\text{th}}$  bit respectively:



**Mutation:** The genetic reproduction operator Mutation is generally used to find the new points in the search space to evaluate. When a particular individual is chosen for mutation (Chen *et al.*, 2004), an allele is

chosen by random in the chromosome and the modification is done on the identified genetic.

In the case of genetic algorithm when binary value is used, the bits to be mutated are changed from 0 - 1 or from 1-0. Let  $R_i$  be the range for the parameter  $\xi_i$ . If binary value is used, then mutation of the  $k^{th}$  bit corresponds to the disruption of  $R_i 2^k$ . If Gray value is used, changing the  $k^{th}$  bit of the gray value can affect all available bits of the corresponding binary value from the  $k^{th}$  bit to the  $n^{th}$  bit. Thus the magnitude of the corresponding disruption can be up to  $R_i 2^{k+1}$  and disruptions of all sizes from  $R_i 2^{-n}$  up to this maximum are possible. The disruption direction is determined by the bit value that is mutated. Under binary encoding, changing the bit from 0 to 1 produces a disruption from MSB to LSB and changing the bit from 1-0 will always produce a change towards MSB.

**Proposed methodology:** The proposed Gray coded Multi Clustered Parallel Genetic Algorithm (GMCPGA) produces better profits in the knapsack compared with the binary coded algorithm. The individual chromosome is selected at random from the initial population. Usually the initial population will be in the binary form. The fitness value for each individual is calculated. Based on the fitness value they like individuals are combined into a single cluster. Similarly alike individuals with same fitness value are grouped to form N numbers of clusters. The selection mechanism (Back, 1996) from any one of the selection mechanism (Goldberg, 1989) is used to select parent in each cluster. The selected individual is then gray coded. The mutation operator is applied to the converted parent to produce the new offspring. New offspring is converted as offspring with binary value to calculate the new fitness value and is used in the next generation. This procedure continues till the termination condition is satisfied. The best individual is obtained based on the fitness value.

The overall structure of the proposed methodology is shown in the pseudocode 1. Gray code Multi Clustered Parallel Genetic Algorithm.

- Step 1: Generate the initial population at random.
- Step 2: Calculate the Fitness Value for Each individual.
- Step 3: Sort the individuals based on the fitness value
- Step 4: Divide the individuals into n clusters based on the fitness value.
- Step 5: For each cluster perform the following

- Using Selection Mechanism, Select the individuals from each cluster.
- Convert the individual to Gray code.
- Mutate the Parent

- Convert the offspring to Binary Value.
- Calculate the fitness Value.

- Step 6: Group the clusters together
- Step 7: Allow the migration of individuals based on fitness.
- Step 8: Until Termination condition is reached repeat from step 5.
- Step 9: Select the best Individual.

Pseudocode 1: Gray coded Multi Clustered Parallel Genetic Algorithm

## RESULTS

The proposed Gray Based Multi Clustered Parallel Genetic Algorithm (GMCPGA) is run with various parameters based on the selection mechanism, population size, mutation rate for the knapsack problem and the obtained score is used to analyze the results. Here Tournament selection mechanism is used to select the individual from the initial population. Tournament selection is the mechanism by which the tournament is conducted for the total number of individuals in each cluster and the parent is chosen for further genetic operators. The initially 100 chromosomes are chosen with an elitism of 10% (Zitzler *et al.*, 2000; Kannaiyah *et al.*, 2011; Maulik and Bandyopadhyay, 2000; Patvichaichod, 2011). The experiment is done many times with varying number of objects and weights. And the result is analyzed and discussed based on the score obtained.

**Impact of population size:** By varying the population size, the optimality of the algorithm can be analyzed. The experiments are carried out with fixed number of objects and the population size varies among the groups, with the number of groups is constant. The Fig. 1 shows the effect of population size over the Score obtained and also we found that the profit increases when the population size is increased.



Fig. 1: Impact of population size

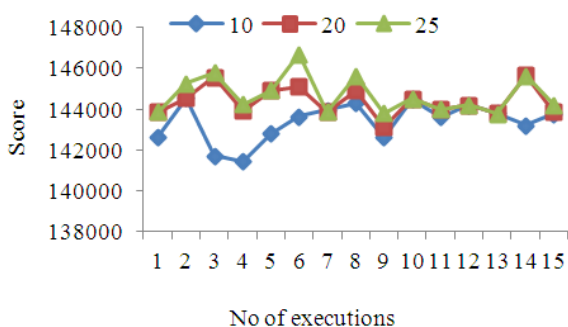


Fig. 2: Impact of group size

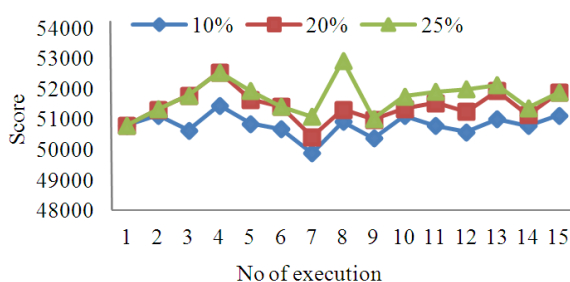


Fig. 3: Impact of mutation rate

**Impact of group size:** By making the other GA parameters constant, the change in the number of groups for a fixed number of objects and the population size. Here we had considered the population size as 100 and by changing the group size to 10 individuals, 20 individuals and 25 individuals in a group. So that the number of group various as 10, 5 and 4 respectively. The Score obtained by changing the group size is shown the Fig. 2. The group size is increased if the population size increases. From the Fig. 2 it is shown that the score is better when the group size increases.

**Impact of mutation rate:** Mutation is the genetic operator which is used to maintain the diversity from one generation to the next generation. Generally mutation alters one or more gene values in the chromosome to make a change, so that the fitness value may be improved and the chromosome may take part in the next generation. This can be done by changing the mutation rate accordingly. The mutation rate is increased in the low order cluster. We had experimented by changing the mutation rate among the clusters and several experiments were carried out. Since we had clustered based on the fitness value, the low order clusters will have lower fitness value. Here we considered the mutation rate as 10,

20 and 30 and applied to the low order clusters. The results are analyzed based on the Score obtained with different mutation rates. From the Fig. 3 it is observed that increased in the mutation rate gives the best performance. Here the mutation is applied only to low order clusters to have active participation of the individual in the GA.

## DISCUSSION

The results obtained clearly show that the proposed Multi Clustered parallel genetic algorithm with gray value had obtained the better profit when using the gray value. The final result of the genetic algorithm depends upon the genetic parameters applied to the algorithm. When using the gray value it is strong enough to apply mutation recombination operator alone to the chromosome. From the result, it is concluded that the algorithm works well with increased group size, increased population size and in high mutation rate.

## CONCLUSION

The experiment results show that Gray coded Multi Clustered parallel genetic algorithm works well in increased population with increase Score. Efficiency is also increased. Improvement in score is achieved when the high mutation rate applies to the low order clusters. It is concluded that GA works well when gray code value is applied. The analysis should be made with the implementation of binary encoding. Hence from the experimental results it is concluded that the proposed algorithm performs well large population size with a high mutation rate.

## REFERENCES

- Back, T., 1996. Evolutionary Algorithms in Theory and Practice. 1st Edn., Oxford University Press, Oxford, ISBN-10: 0195099710, pp: 314.
- Chen, X., W. Gui, L. Chen and Z. Hu, 2004. A multi-population genetic algorithm based on chaotic migration strategy and its application to inventory programming. Proceedings of the 5th World Congress on Intelligent Control and Automation, Jun. 15-19, IEEE Xplore Press, China , pp: 2159-2162. DOI: 10.1109/WCICA.2004.1341968
- Goldberg, D.E., 1989. Genetic Algorithms in Search, Optimization and Machine Learning. 1st Edn., Addison-Wesley, Reading, Massachusetts, ISBN-10: 0201157675, pp: 432.

- Holland, J.H. 1975. Adaptation in natural and artificial systems: An introductory analysis with applications to biology, control and artificial intelligence. University of Michigan Press, Ann Arbor, ISBN-10: 0472084607, pp: 183.
- Zitzler, E., K. Deb and L.Thiele, 2000. Comparison of Multiobjective evolutionary algorithms: Empirical results. *Evolut Comput.*, 8: 173-195. DOI: 10.1162/106365600568202
- 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. DOI: 10.1016/S0031-3203(99)00137-5
- 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