Computer Engineering and Intelligent Systems
ISSN 2222-1719 (Paper) ISSN 2222-2863 (Online)
Vol 3, No.1, 2012
www.iiste.org

Analyzing the Impact of Genetic Parameters on Gene Grouping Genetic Algorithm and Clustering Genetic Algorithm

R.Sivaraj (Corresponding author)
Research Scholar and Assistant Professor (Senior Grade)
Department Of Computer Science and Engineering
Velalar College of Engineering and Technology
Erode, Tamil Nadu, India
E-mail: rsivarajcse@gmail.com

Dr.T.Ravichandran
Principal and Research Supervisor
Hindusthan Institute of Technology
Coimbatore-641032, Tamil Nadu, India
E-mail: dr.t.ravichandran@gmail.com

Abstract

Genetic Algorithms are stochastic randomized procedures used to solve search and optimization problems. Many multi-population and multi-objective Genetic Algorithms are introduced by researchers to achieve improved performance. Gene Grouping Genetic Algorithm (GGGA) and Clustering Genetic Algorithm (CGA) are of such kinds which are proved to perform better than Standard Genetic Algorithm (SGA). This paper compares the performance of both these algorithms by varying the genetic parameters. The results show that GGGA provides good solutions, even to large-sized problems in reasonable computation time compared to CGA and SGA.

Keywords: Stochastic, randomized, multi-population, Gene Grouping Genetic Algorithm, Clustering Genetic Algorithm.

1. Introduction

Evolution is the process which enables individuals or species in one generation to modify or improve in the next generation. The nature helps individuals to adapt to the changing environment through the process of evolution. As species evolve over time, they become more complex and hold better characteristics. This process helps more fit individuals to retain in the environment and those which cannot retain the environment die and run out of species. Genetic Algorithm is of such kind which follows the “Principle of Natural Evolution and Genetics”.

The basic principles of Genetic Algorithms (GA) were first laid down rigorously by Holland [1975]. They simulate those processes in natural populations which are essential to evolution. The general outline of the Standard Genetic Algorithm (Goldberg [1989]) process is given below in figure 1.Exactly which processes are essential for evolution and which processes have little or no role to play is still a matter of research, but the foundations are clear.

```
BEGIN /* genetic algorithm */
    Generate initial population
    Compute fitness of each individual
    WHILE NOT finished DO
              BEGIN /* produce new generation */

          FOR population size / DO
                  BEGIN /* reproductive cycle */
                          Select two individuals from old generation for mating (crossover)
                                 /* biased in favors of the fitter ones */
                          Recombine the two individuals to give two offsprings
                          Compute fitness of the two offsprings
                          Insert offsprings in new generation
                 END
    IF population has converged THEN
    Finished: = TRUE
    END
END
```

Figure 1. Outline of Genetic Algorithm

To implement Genetic Algorithm for solving a problem, the parameters need to be encoded in a form (chromosome) that the algorithm can manipulate to generate new solutions that inherit traits from the parents used (Davis(Ed) [1991], Michalewicz [1992]). There are many methods for encoding the parameters in a GA like Binary Encoding, Real Encoding, and Permutation Encoding etc. Different encoding techniques work better for different problems. The collection of ‘n’ chromosomes is called as population. The initial population to be used for the algorithm is chosen randomly. The fitness value or objective function value is calculated for each chromosome in the population. Formulating fitness function is a crucial part in Genetic Algorithm process because it is the one upon which individuals to be carried over to the next generation is decided. After the calculation of fitness values, the parents to be used for crossover are selected using appropriate selection mechanisms.

The chromosomes with high fitness values are selected for crossover and those with low fitness values are discarded in the current generation itself and will not contribute any more in the GA process. The resulting new chromosomes (Offsprings) are then mutated and this forms the next generation of chromosomes. Fitness values of the new generation are calculated and the remaining steps are continued until the termination criterion is satisfied. The termination criterion can be either fixed number of generations or the process is allowed to run until fitness values of chromosomes in the population converges to a single optimum value which may be taken as the solution.

Genetic Algorithms are used in variety of problems in almost all domains. Though GA is used for solving optimization problems, it is not directly suitable to solve constrained optimization problems. Many researchers attempt to solve this problem by introducing new genetic operators that can eventually improve the performance of the genetic algorithm. It has been found that it is usually beneficial to run methods that are simpler, and execute them several times, rather than using methods that are very complex but need to be run only once. Gene Grouping Genetic Algorithm (GGGA) and Clustering Genetic Algorithm (CGA) are of such kinds which were introduced in the idea of providing simple and efficient approaches for solving large and complex optimization problems.

2. Implementation

To analyze the performance of the algorithms, the 0/1 knapsack problem (Martello [1990]) is chosen. The problem can be stated as follows:

Given a set of n objects, each object O_i is associated with a weight w_i and profit p_i and a knapsack capacity C. Let x_i be a variable, the value of which is either zero or one, The variable xi has the value one when the ith item is carried in the knapsack and zero otherwise.

Given {w1, w2,…. w_n} and { p_1,p_2,..p_n}, our objective is to

$$\text{Maximize } f(x1,x2,,..xn)=\sum_{j=1}^{n} pjxj$$

$$\text{Subject to the constraint } \sum_{j=1}^{n} wjxj \leq C, \qquad xj=0 \text{ or } 1,\ j=1,2\ldots n$$

It is called as 0/1 knapsack problem because only one copy of the object is placed in the knapsack. It is a typical example of NP complete problem whose time and space efficiency cannot be expressed in terms of its input size. The difficulty here is that when number of items is added, the computational power required to solve this problem will grow exponentially.

2.1 Gene Grouping Genetic Algorithms

R. Sivaraj, T. Ravichandran [2011] proposed GGGA which divides the entire chromosome length into ‘r’ groups and performs selection, crossover and mutation individually within all groups and then finally combines the solution. Figure 2 shows the chromosome format and its grouping.

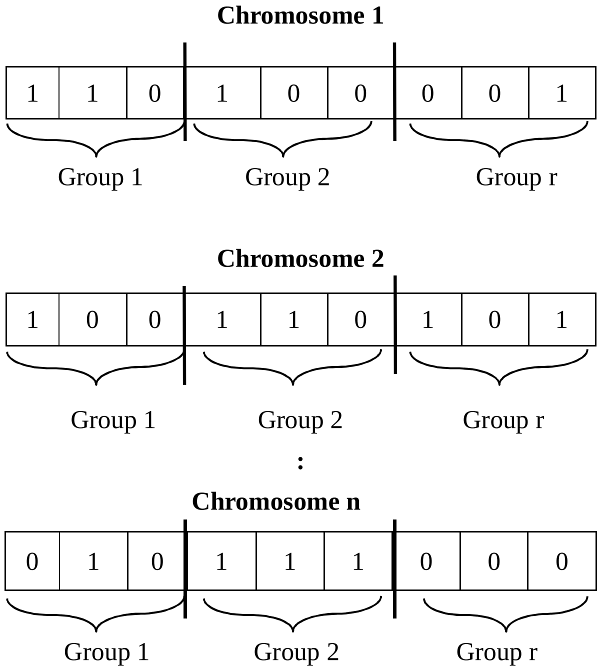
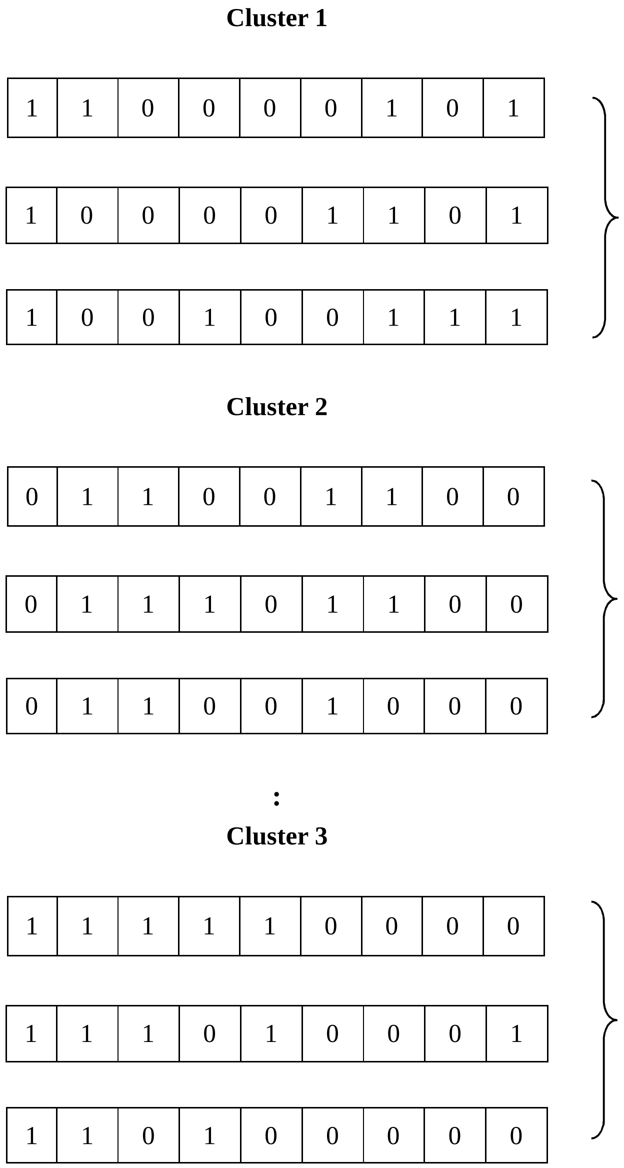


Figure 2. Chromosome format of Gene Grouping Genetic Algorithm

2.2 Clustering Genetic Algorithm (CGA)

Rather than GGGA which splits the genes in each chromosome into r groups, CGA (R. Sivaraj, T. Ravichandran [2011]) divides the entire set of chromosomes in the population into clusters. If there are n chromosomes, it is divided into c clusters of size n/c each which tends to have similar fitness values. This is a new approach where K-means clustering algorithm is used to cluster the chromosomes rather than Genetic Algorithm used to improve clustering (Rui et al [2010]). The genetic parameters are applied to the chromosomes in each cluster and finally the results are combined. Figure 3 shows the chromosome format of CGA and how 'c' clusters are formed in the population.



Cluster 1

1 1 0 0 0 0 1 0 1

1 0 0 0 0 1 1 0 1

1 0 0 1 0 0 1 1 1

Cluster 2

0 1 1 0 0 1 1 0 0

0 1 1 1 0 1 1 0 0

0 1 1 0 0 1 0 0 0

:

Cluster 3

1 1 1 1 1 0 0 0 0

1 1 1 0 1 0 0 0 1

1 1 0 1 0 0 0 0 0

Figure 3. Chromosome format of Clustering Genetic Algorithm

The important feature of Genetic Algorithm is that it performs better or worse depending on the genetic parameters chosen. If they are not chosen properly, it will lead to the undesired result. Hence to compare the overall performance of both these approaches, the genetic parameters are varied and the empirical results are studied. The results are first studied for 200 objects.

Binary Encoding where binary values 0 and 1 are allowed is used to represent the chromosomes. If the binary value 1 is present as i^{th} gene in the chromosome, it indicates that i^{th} object is chosen in the knapsack. If it is 0, then it indicates that i^{th} object is not chosen in the knapsack. The mutation is performed at the rate of 0.01%. The remaining parameters are varied for different experiments.

2.3 Impact of varying the population size

The population size is a major factor contributing to the optimality of the solution obtained by the algorithm. Higher the population size, higher is the accuracy. But if the population size is too high, it will take a long time to converge.

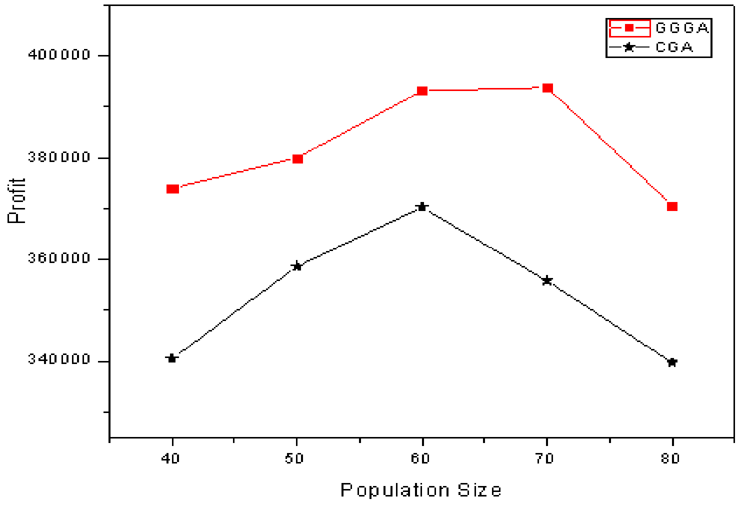


Figure 4. Impact of varying the population size

Hence,a moderate amount of chromosomes have to be maintained in the population to achieve a tradeoff between optimality and time efficiency. The algorithms are run with various population sizes and the results studied are shown in figure 4.

2.4 Impact of varying Selection Mechanisms

Selection method chosen for the problem also impacts much on the final solution as it is the prime factor which selects and carries over the best chromosomes to the next generations. The commonly used selection mechanisms are Roulette Wheel selection, Tournament Selection, Rank Selection, Range Selection etc. Recent results (Rudolph [1999], Zitzler et al [2000]) show clearly that elitism can speed up the performance of the GA significantly and also it helps to prevent the loss of good solutions once they have been found. Hence elitism of 10% is included in all the selection mechanisms implemented here.

2.4.1. Roulette Wheel Selection

Roulette Wheel selection is the process where the wheel with all the chromosomes are spun ‘n’ times and the probability which each one takes on the wheel is studied. The chromosomes with high probabilities are selected as parents for the next generation. The results obtained by implementing the Roulette Wheel mechanism for the given inputs are given in figure 5.

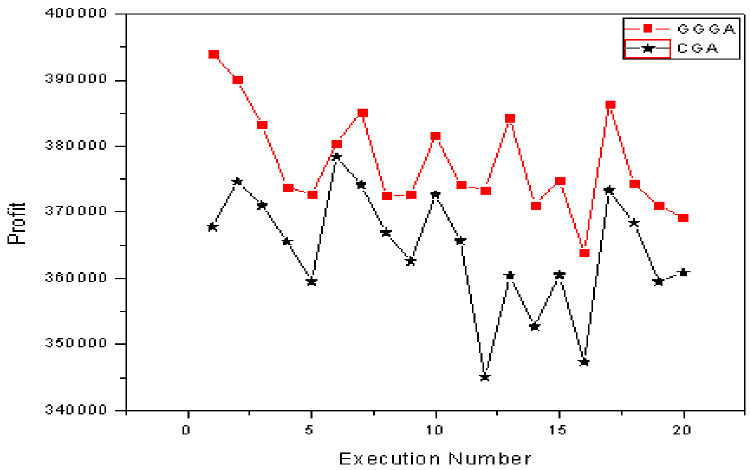


Figure 5. Impact of Roulette Wheel Selection

2.4.2. Tournament Selection

Tournament Selection is the process by which ‘n’ tournaments are being conducted for‘s’ random samples and the winner of each tournament is carried to the next generation. There are many types of Tournament Selection mechanisms like Binary Tournament Selection, Larger Tournament Selection, Boltzmann tournament selection and Correlative Tournament Selection.

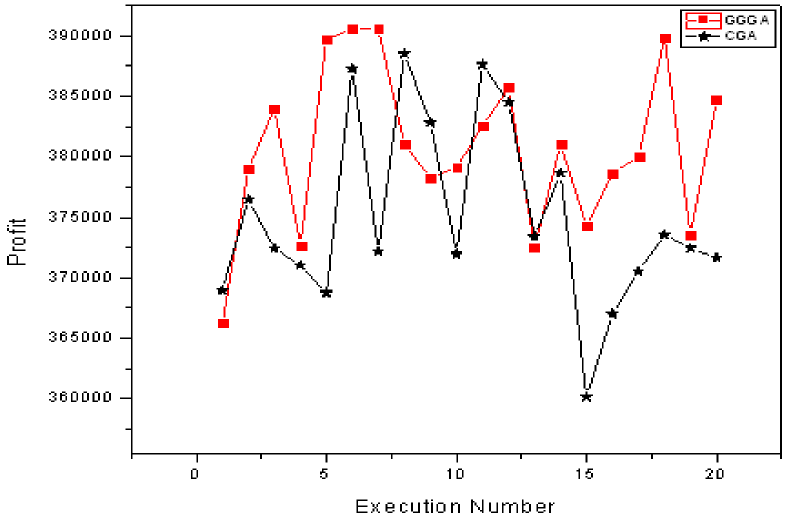


Figure 6. Impact of Binary Tournament Selection

Binary Tournament is one where two individuals are chosen at random and the better of the two individuals is selected with fixed probability p, $0.5 < p \sim 1$. The results obtained by implementing the Binary Tournament selection mechanism are given in figure 6.

2.5 Impact of Crossover types

Crossover operation is performed to combine two chromosomes and to obtain two new offsprings which inherit characteristics from both parents. The crossover types are One point crossover, Two point crossover, Uniform crossover etc. The probability of the number of chromosomes to be crossed over (Pc) is determined by the user depending upon the problem which also have its impact in the final solution.

One point crossover is the type where one random point is chosen in the length of the chromosome and the chromosomes are interchanged. The head of first chromosome and the tail of second chromosome are combined to form first offspring. Similarly, the head of second chromosome and the tail of first chromosome are combined to form second offspring. Two point crossover is one where two random points are chosen and the genes in the parents are interchanged alternatively to form new parents.

Uniform crossover is the one where a random mask is introduced which contains random number of 0s and 1s.If a bit in the mask is 1 then the gene for the first offspring is taken from the first parent and is taken from the second parent otherwise. Uniform Crossover is studied here and the crossover probabilities chosen are 80 % and 90 %. The results obtained are given in the figure 7.

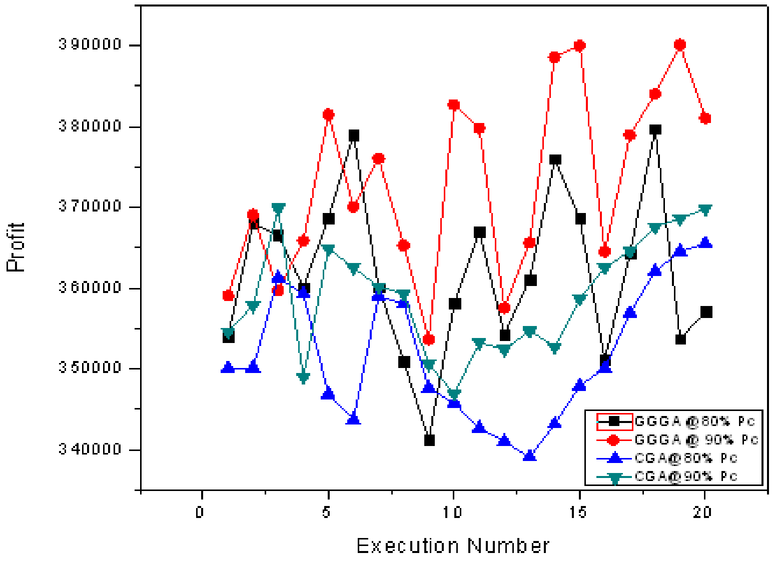


Figure 7. Impact of Uniform crossover

2.6 Convergence Velocity

All the previous experiments were conducted for fixed number of iterations (Termination Criteria) and the chromosome which has maximum occurrence in the final iteration is taken as the solution.

Now the algorithm is allowed to run until any one of the chromosomes is repeated many times in the population and the values are recorded. The speed with which the algorithm reaches the final optimal solution is called as convergence velocity. GGGA seems to converge faster than CGA. Figure 8 shows the comparison results.

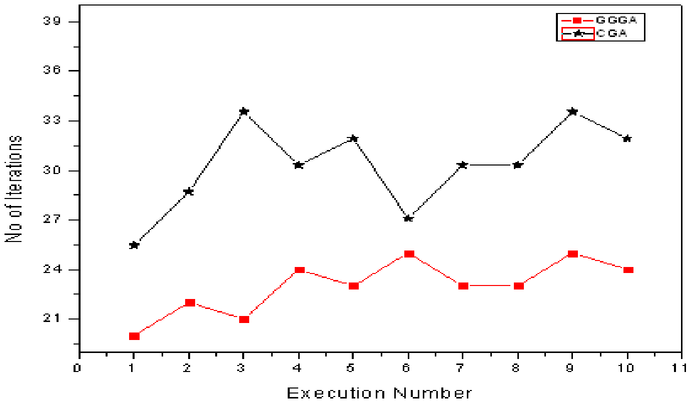


Figure 8. Convergence Velocity of GGGA Vs CGA

2.7 For varying number of objects

The experiments are repeated with best genetic operators obtained from the above results and the profits obtained are studied for different number of objects in the knapsack problem. The parameters chosen for the experiments are

- Encoding : Binary encoding,
- Selection : Tournament Selection with 10 % elitism
- Crossover : Uniform Crossover at Crossover Probability of 90%
- Population size : 60.

The number of objects or items included in the problem is chosen as 100 and 300 and the empirical results obtained are shown in figure 9 and figure 10 respectively.

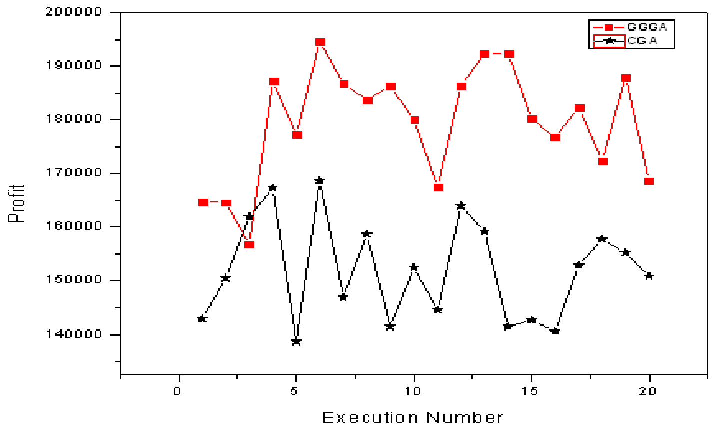


Figure 9. Comparison of profits obtained by GGGA Vs CGA for 100 objects

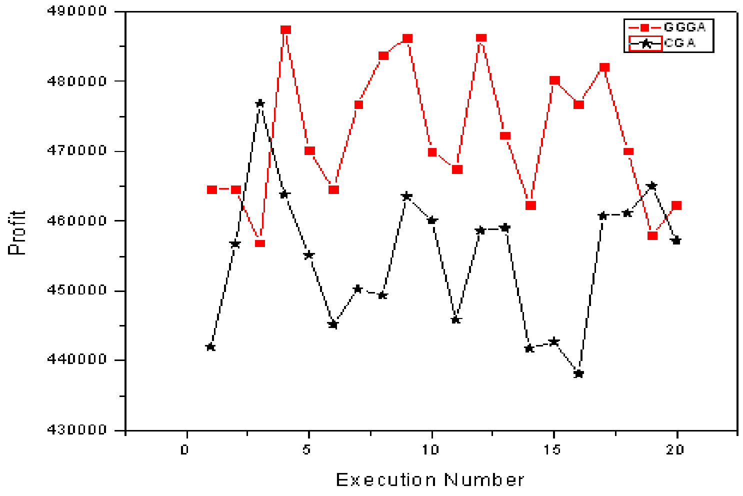


Figure 10. Comparison of profits obtained by GGGA Vs CGA for 300 objects

3. Conclusions

From the above experiments and their results, comparison of both algorithms is done with different genetic operators. The results clearly show that Gene Grouping Genetic Algorithm works better compared to Clustering Genetic Algorithm which works better than Standard Genetic Algorithm. As a future work, mathematical modeling has to be included in both algorithms to improve the performance. Implementation of these methods for other types of problems can also be thought of as a future work

References

Davis (Ed.), L. (1991), “Handbook of Genetic Algorithms”, Van Nostrand Reinhold, New York

Goldberg D.E., (1989), “Genetic Algorithms in Search, Optimization and Machine Learning”, Addison-Wesley, New York

Holland, J. H (1975), “Adaptation in natural and artificial systems.” Ann Arbor, MI: University of Michigan Press

Martello, S & Toth, P. (1990), “Knapsack problems: Algorithms and Computer Implementations”, J.Wiley & Sons

Michalewicz, Z. (1992), “Genetic Algorithms + Data Structures= Evolution Programs”, Springer, New York

Rudolph, G. (1999), “Evolutionary search under partially ordered sets”. Technical Report No.CI-67/99, Dortmund: Department of Computer Science/LS11, University of Dortmund, Germany

Rui Chen, Shurong Zou, Hongwei Zhang & Zhongtian Feng,(2010), “Improvement of the Genetic Algorithm and its Application on Clustering”. ICMTMA '10 Proceedings of the International Conference on Measuring Technology and Mechatronics Automation - Volume 02

Sivaraj R & Ravichandran, T(2011),” An Improved Clustering Based Genetic Algorithm for Solving Complex NP Problems, Journal of Computer Science, Vol No: 7, Issue No: 7, pp: 1033-1037

Sivaraj, R & Ravichandran, T.(2011), “An Efficient Grouping Genetic Algorithm”, International Journal of Computer Applications, Vol 21, No: 7 (8)

Zitzler, E., Deb, K. & Thiele, L. (2000), “Comparison of multiobjective evolutionary algorithms: Empirical results”. Evolutionary Computation 8(2): 173-195

This academic article was published by The International Institute for Science, Technology and Education (IISTE). The IISTE is a pioneer in the Open Access Publishing service based in the U.S. and Europe. The aim of the institute is Accelerating Global Knowledge Sharing.

More information about the publisher can be found in the IISTE's homepage:
http://www.iiste.org

The IISTE is currently hosting more than 30 peer-reviewed academic journals and collaborating with academic institutions around the world. **Prospective authors of IISTE journals can find the submission instruction on the following page:** http://www.iiste.org/Journals/

The IISTE editorial team promises to the review and publish all the qualified submissions in a fast manner. All the journals articles are available online to the readers all over the world without financial, legal, or technical barriers other than those inseparable from gaining access to the internet itself. Printed version of the journals is also available upon request of readers and authors.

IISTE Knowledge Sharing Partners

EBSCO, Index Copernicus, Ulrich's Periodicals Directory, JournalTOCS, PKP Open Archives Harvester, Bielefeld Academic Search Engine, Elektronische Zeitschriftenbibliothek EZB, Open J-Gate, OCLC WorldCat, Universe Digtial Library , NewJour, Google Scholar