International Journal of Bio-Science and Bio-Technology
Vol.8, No.6 (2016), pp. 129-138
http://dx.doi.org/10.14257/ijbsbt.2016.8.6.13

Performance Comparison between Genetic Algorithm and Population-Based Incremental Learning Algorithm using Student Ranking Application

Ch. Viswanathsarma[1], Debnath Bhattacharyya[1] and Hye-jin Kim[2]

[1]*Department of Computer Science and Engineering,*
Vignan's Institute of Information technology
Visakhapatnam-530049, AP, India
vissu.viit@gmail.com, debnathb@gmail.com
[2]*Sungshin Women's University,*
2, Bomun-ro 34da-gil,
Seongbuk-gu, Seoul, Korea
hyejinaa@daum.net
(Corresponding Author)

Abstract

This paper compares the performance of Genetic Algorithm and Population-based Incremental Learning algorithm on a student ranking application. The proposed student ranking application focuses on proving the students' achievements in fields like attendance, achievements and backlogs apart from academics. This application produces the output according to the prioritized weights inputted by the user. Genetic Algorithm and Population-based Incremental Learning is evolutionary search based optimization algorithm which can arrive at near-optimal solution faster by using randomized techniques. These are stochastic search based algorithms that include randomness to escape local optima. The two algorithms are compared based on the accuracy of optimal value, number of generations and time taken to reach optimal value.

Keywords*: Evolutionary computing, Genetic Algorithm, Optimization, Performance comparison, Population-based Incremental Learning, Ranking*

1. Introduction

Lot of work has been done on Genetic Algorithm and Population-based Incremental Learning individually but there are very few resources that discusses about the two algorithms together. This work analyses and compares two evolutionary optimization algorithms - Genetic Algorithm (GA) and Population-based Incremental Learning (PBIL) and determines which of them is better.

GA is derived from Darwinian's theory of "survival of the fittest". It is efficient, adaptive and robust search process. It is guided by principles of evolution and natural Genetic to perform randomized search and optimization. The Population-based Incremental Learning Algorithm (PBIL) is an Estimation of Distributed Algorithms, as proposed in [5]. Population-based Incremental Learning supposes that all the variables are independent. At each step of the algorithm a probability vector is maintained. Next generation is generated based on this probability vector.

Genetic algorithms generate solutions to optimization problems using techniques inspired by natural evolution, such as mutation, selection, and crossover whereas Population-based Incremental Learning is an algorithm where the genotype of an entire population (probability vector) is evolved rather than individual members.

ISSN: 2233-7849 IJBSBT

The performance analysis and comparison of the two algorithms is done by taking different parameters into consideration like number of iterations, inputs given, time complexity on student ranking application.

1.1. Student Ranking Application

The student ranking application deals with ranking of the performance of the students in a particular institution or organization. Generally, the student ranking analysis is made on the basis of academic by using normal sorting techniques which involve the simple database with limited fields and low level complexity in the analysis. In such an analysis the main priority would be the academics, where the achievements of students in other activities like extra-curricular, co-curricular and sports are given less importance in analyzing the student's abilities. This lessens the chance of making better decisions about student's capabilities. Such a ranking system is biased towards academic achievement.

The student ranking application proposed considers the performance of student in both academic as well as non-academic fields to select top all-rounder's rather than only top academicians. This application rank the students based on multiple parameters rather than only academics. It considers criteria like the achievements of students in extra-curricular, co-curricular and academics. The application takes priorities for different parameters (attendance, achievements, sports and backlogs) as input and then optimizes this multi-objective function using two different algorithms – Genetic Algorithm and Population-based Incremental Learning as shown in Figure 1. The performance of the two algorithms is compared on a student ranking application.

The remaining part of the study is divided as follows. Section 2 discusses about optimization problem. Section 3 reviews Genetic algorithm and is further subdivided into 3 subsections. Section 3.1 discusses selection operator, section 3.2 explains working of crossover operator and section 3.3 explains working of mutation operator. Section 4 is a brief discussion on PBIL. Section 5 compares the performance of the two algorithms and finally section 6 concludes the paper.

2. Optimization

Optimization is a subject that deals with the problem of minimizing or maximizing a certain function in a finite dimensional Euclidean space over a subset of that space. In general, it is a technique to select the best solution (with regard to some criteria) from a set of feasible solutions Evolutionary algorithms can handle large search spaces, multi objective functions, multimodal functions. They use randomized techniques to arrive at solution faster which may otherwise take lifetime. They perform well even when there is little or no domain specific knowledge about the search space. They are preferred for problems with huge solution space, little or no domain specific knowledge and where little bit of inaccuracy, imprecision and uncertainty is an acceptable trade-off for faster, low cost, practical and robust algorithm.

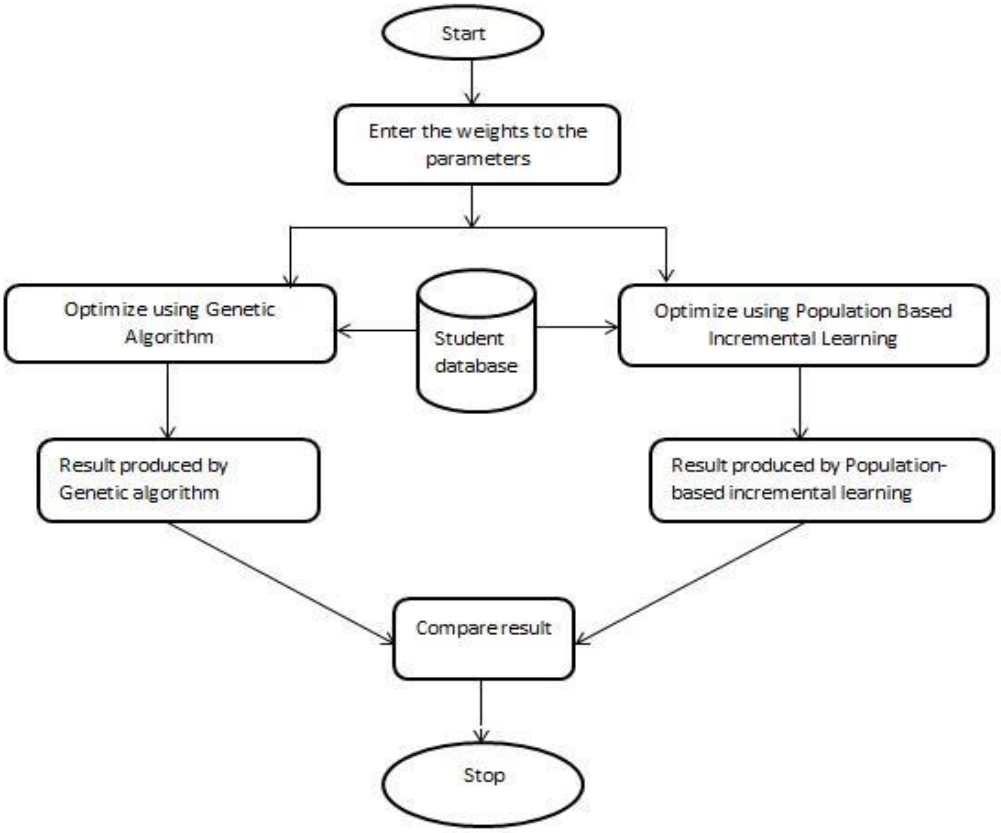


Figure 1. Steps for Processing SRA Application

Though randomized these algorithms use information from previous iterations and evolves the solution space to guide it towards the optimal solution. Evolutionary algorithms are stochastic search based technique which deliberately adds randomness to search process to avoid being caught at local optima.

3. Genetic Algorithm

Genetic Algorithm is an evolutionary search-based optimization technique which is based on Darwinian's theory of „Survival of The Fittest". Genetic algorithms are often reviewed as function optimizer [10], although the range of problems to which Genetic algorithms have been applied are quite broad [3],[4]. It performs search from a population of points rather than from a single point thus accelerating the search process. This is the implicit parallelism in GA. Though randomized it moves towards optimal solution by using results from previous iterations. The best solution from previous generation plays a vital role in determining next population.

Genetic algorithm involves application of three operators namely

- Selection (Reproduction)
- Crossover (Recombination)
- Mutation

3.1. Encoding

This application uses binary encoding. Different kinds of encoding techniques are available [2, 9]. Binary encoding is a mapping from integer to binary system. Length of chromosome is equal to the number of records in the database. Each record is represented

by a unique bit in the chromosome and position of the bit is related to student id *i.e.*; the student with id 1 is represented by first bit of the chromosome string.

Genetic Algorithm maintains a population of such chromosome. Bit 1 at a particular position in the chromosome indicates that the particular student is among top 10. Thus the implicit constraint on the chromosome is that no more than 10 bits can be 1 in any chromosome.

1	0	0	0	1	1	0	1	0	1	0	0	0	0	0

Figure 3. Chromosome Representation for SRA

For example, above chromosome represents a database with 15 records of which student with sid 1, 5, 6, 8 and 10 are among the top 5 students.

3.2. Selection

In this step parent chromosome are randomly selected from the present population to generate next generation of chromosome. Chromosome with better fitness has higher probability of getting selected as parent chromosome for reproduction.

3.3. Crossover (Recombination)

This step involves selecting a split point and exchanging the genes of parent chromosome about the split point to produce child chromosome for next generation. To meet the equality constraint on volume of 1s in the chromosome a special crossover technique is used as proposed in [5]. Crossover allows GA to exploit a particular point in search space deeper.

3.3.1. Working of Crossover: First of all a split point is selected for both the chromosomes, let them be p1 and p2. Each child gets half of the chromosome from 1 parent and the other half from the other parent.

0	0	0	1	0	0	1	1	0	0	0	0	0	0	0	1	0	1	0	0
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20

Chromosome 1

1	*0*	*0*	*1*	*0*	*0*	*1*	*0*	*0*	*1*	*0*	*0*	*0*	*0*	*0*	*0*	*0*	*0*	*1*	*0*
1	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*	*10*	*11*	*12*	*13*	*14*	*15*	*16*	*17*	*18*	*19*	*20*

Chromosome 2

Let us assume p1= p2=1.

i.e.. child1 is formed as: par1 [1-10]:par2 [11-20] and
child2 is formed as: par2 [1-10]:par2 [11-20]

But volume of 1s in par [1-10] is not equal to volume of 1s in par2 [1-10]. So shift p2 by 1 point. So p1=1, p2=2

0	0	0	1	0	0	1	1	0	0	0	0	0	0	0	1	0	1	0	0
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20

↑
P1 chromosome 1

1	0	0	1	0	0	1	0	0	1	0	0	0	0	0	0	0	0	1	0
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20

↑P2 chromosome 2

Child 1 is

0	0	0	1	0	0	1	1	0	0	0	0	0	0	0	0	0	1	0	1
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20

And child 2 is:

0	*0*	*1*	*0*	*0*	*1*	*0*	*0*	*1*	*0*	*0*	*0*	*0*	*0*	*0*	*1*	*0*	*1*	*0*	*0*
1	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*	*10*	*11*	*12*	*13*	*14*	*15*	*16*	*17*	*18*	*19*	*20*

And the volume of 1s in each chromosome is equal *i.e.*. 5 in this case.

The algorithm to determine split point that doesn't violate equality constraint on number of 1s is:

Let

ones_in_par1 be number of 1s from p1 to p1+geneLength/2
ones_in_par2 be number of 1s from p2 to p2+geneLength/2
for p1 = 0 to geneLength

for p2 = 0 to geneLength
if(ones_in_par1 is equal to ones_in_par2)
break

3.4. Mutation

In this step bits are flipped from 0's to 1's and 1's to 0's with a low probability. This step ensures that the algorithm does not get trapped at local minima. It allows GA to explore search space far rather than sticking to one area in search space.

To satisfy equality constraint on number of 1s in a chromosome a special mutation operator is used as proposed in [5].

3.4.1. Steps for Mutation

Step 1: Randomly select 2 bits in the chromosome.
Step 2: If they are equal go to step 1 else flip both the bits.

3.5. Decoding

After multiple iterations the algorithm selects one of the encoded chromosomes as best solution. The students with id corresponding to 1s in the chromosome are the top 10 students among all records in the database. The top 10 records are sorted and displayed.

4. Population-based Incremental Learning

Population-based Incremental Learning combines genetic algorithm with competitive learning [4, 7]. It is a variant of GA which is simpler than GA and often outperforms GA. Unlike GA it doesn't involves application of complex operators (selection and recombination) instead it generates the next generation by updating a prototype vector (probability vector) based on its learning from the present population. The probability of finding a 1 at i^{th} of solution vector is proportional to the i^{th} value in probability vector [6].

PBIL algorithm works as follows:

Step 1: A population of solution and a probability vector is generated randomly.
Step 2: The fitness of each individual is evaluated based on the function to be optimized and the individuals are ranked.
Step 3: The probability vector is then updated based on learning (either positive or negative learning) from current population.
Step 4: Mutation is performed with a very low probability.
Step 5: Create next generation using probability vector.

Steps 2 to 5 are repeated until the population converges to an optimal solution.

It can use either positive learning or negative learning to update probability vector [7]. In positive learning the probability vector is moved towards the best solution vector and in negative learning the probability vector moves away from the worst solution vector. Typical value of learning rate lies between 0.1 and 0.4.In this study learning rate is taken as 0.2, mutation probability as 0.05 and mutation shift as 0.05. Positive learning is used. The mutation probability is low as too high mutation rate prevents population to converge to any optimum solution.

6. Experimentation and Results

This work is done by using Java platform (version - jdk 1.8). The dataset is taken from Vignan‟s Institute of Information Technology, Visakhapatnam which contains student's records in particular attendance, results and extra-curricular achievements. The dataset contains 918 student records.

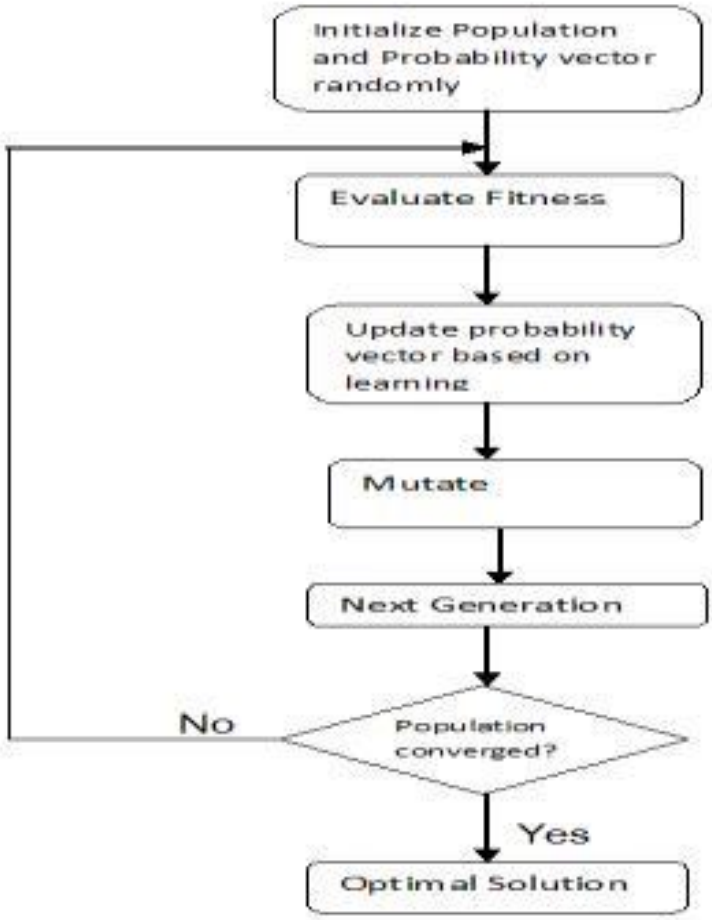
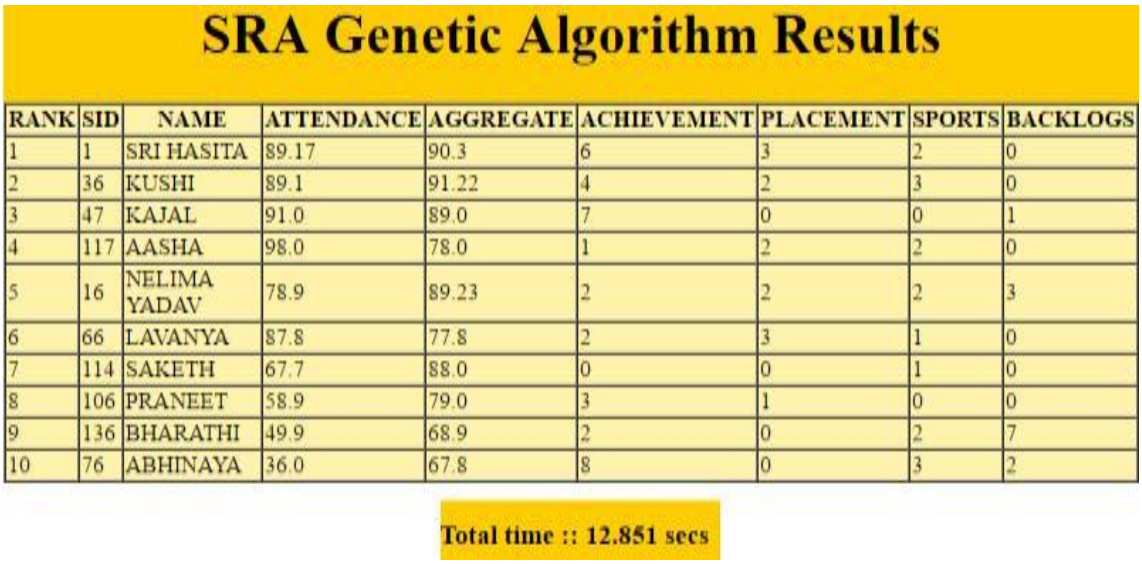


Figure 4. Population-Based Incremental Learning Flowchart

6.1. Comparison of Genetic Algorithm and Population-based Incremental Learning Algorithm

Population-based Incremental Learning performs better than Genetic Algorithm for the problem compared in this paper. PBIL produces more accurate results and in lesser time than GA. It is fast both in terms of number of generations evaluated and time taken to perform those evaluations. At the same time PBIL implementation is much more concise and simple than that of GA.

For this study population size is taken as 10 and each chromosome is 918 bits long (as there are 918 records in database). The objective is to maximize aggregate, achievement and for GA crossover rate is taken as 0.5 and mutation rate as 0.05. For PBIL learning rate is taken as 0.2, mutation probability as 0.05 and mutation shift as 0.05.

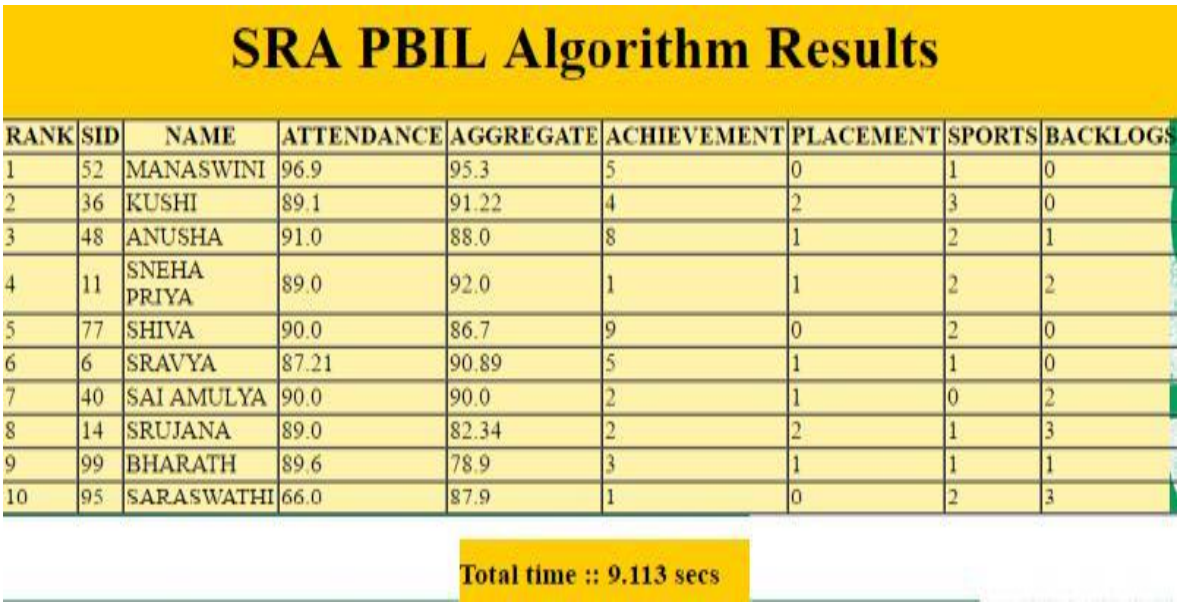


SRA Genetic Algorithm Results

RANK	SID	NAME	ATTENDANCE	AGGREGATE	ACHIEVEMENT	PLACEMENT	SPORTS	BACKLOGS
1	1	SRI HASITA	89.17	90.3	6	3	2	0
2	36	KUSHI	89.1	91.22	4	2	3	0
3	47	KAJAL	91.0	89.0	7	0	0	1
4	117	AASHA	98.0	78.0	1	2	2	0
5	16	NELIMA YADAV	78.9	89.23	2	2	2	3
6	66	LAVANYA	87.8	77.8	2	3	1	0
7	114	SAKETH	67.7	88.0	0	0	1	0
8	106	PRANEET	58.9	79.0	3	1	0	0
9	136	BHARATHI	49.9	68.9	2	0	2	7
10	76	ABHINAYA	36.0	67.8	8	0	3	2

Total time :: 12.851 secs

Figure 5. Genetic Algorithm Results



SRA PBIL Algorithm Results

RANK	SID	NAME	ATTENDANCE	AGGREGATE	ACHIEVEMENT	PLACEMENT	SPORTS	BACKLOGS
1	52	MANASWINI	96.9	95.3	5	0	1	0
2	36	KUSHI	89.1	91.22	4	2	3	0
3	48	ANUSHA	91.0	88.0	8	1	2	1
4	11	SNEHA PRIYA	89.0	92.0	1	1	2	2
5	77	SHIVA	90.0	86.7	9	0	2	0
6	6	SRAVYA	87.21	90.89	5	1	1	0
7	40	SAI AMULYA	90.0	90.0	2	1	0	2
8	14	SRUJANA	89.0	82.34	2	2	1	3
9	99	BHARATH	89.6	78.9	3	1	1	1
10	95	SARASWATHI	66.0	87.9	1	0	2	3

Total time :: 9.113 secs

Figure 6. Population-Based Incremental Algorithm Results

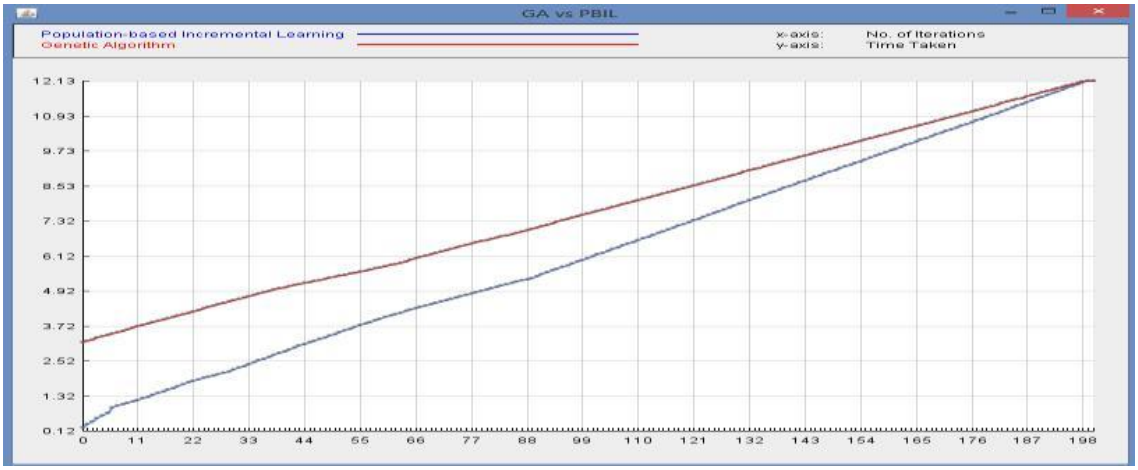


Figure 7. Comparison of GA and PBIL using Student Ranking Application

Figure. 7 show that PBIL produces more accurate results in comparatively lesser time than GA.

References

[1] A. Rexhepi, A. Maxhuni and A. Dika, "Analysis of the impact of parameters values on the Genetic Algorithm for TSP", IJCSI International Journal of Computer Science Issues, vol. 10, iss. 1, **(2013)**.
[2] R. Malhotra, N. Singh and Y. Singh, "Genetic Algorithms: Concepts, Design for Optimization of Process Controllers", Published by Canadian Center of Science and Education, vol. 4, no. 2, **(2011)**, pp. 39-54.
[3] M. R. Murty, J. V. R Murthy and P. V. G. D. P. Reddy, "Dimensionality Reduction Text data Computing (IJARITAC) ,indexed Copernicks, Indian Science,NA-digest,Ulrich-PD, vol. 2, iss. 2, (print: 0975-8070, Online 0975-8089),DOI 10.5958/j.0975-8070.2.2.010,Research Impact indicator 0.021, **(2011)**, pp. 41-49.
[4] S. Y. Yang, S. L.Ho, G. Z. Ni1, J. M. Machado and K. F. Wong, "A New Implementation of Population Based Incremental Learning Method for Optimizations in Electromagnetics", IEEE TRANSACTIONS ON MAGNETICS, vol. 43, no. 4, **(2007)**, pp. 1601-1604.
[5] J. F. A. Madeira, H. Rodrigues and H. Pina, "Genetic Methods in Multi- objective Optimization of Structures with an Equality Constraint on Volume", Research Gate publication 221228627.
[6] J. Bekkerand and Y. Olivier, "Using the Population-based Incremental Learning Algorithm with Computer Simulation: Some Applications", (South African Journal of Industrial Engineering, vol. 19, no. 1, **(2008)**, pp. 53-71.
[7] S. Baluja, "Population Based Incremental Learning – A Method for Integrating Genetic Search Based Function Optimization and Competitive Learning", (Tech. Rep. No. CMUCS-94-163). Pittsburgh, PA: Carnegie Mellon University, **(1994)**.
[8] S. Baluja and R. Caruana, "Removing the Genetics from the Standard Genetic Algorithm", CMU-CS-95-141, Pittsburgh, Carnegie Mellon University, **(1995)**.
[9] Practical Genetic Algorithms by Randy L. Haupt, John Wiley & Sons Inc, Chapter 1 and 2, **(2004)**, pp. 1-47.
[10] T. V. Mathew, "Genetic algorithm", Indian Institute of technology Bombay.
[11] M. R. Murty, J. V. R. Murthy and P. V. G. D. P. Reddy, "Homogeneity Separateness: A New Validity Measure for Clustering Problems", International conference and published the proceedings in clustering with prediction of optimal number of clusters, International journal of Applied research on Information Technology AISC and computing, Springer, (indexed by COPUS, ISI proceeding DBLP *etc*), vol. 248, **(2014)**, ISBN 978-3-319-03106, pp. 1-10.
[12] K. Lahari and M. R. Murty, "Partition based clustering using Genetic Algorithms and Teaching Learning Based Optimization: Performance Analysis", International Conference and published the proceedings in AISC, Springer DOI: 10.1007/978-3-319-13731-5_22, vol. 2, pp. 191-200.