

2013 AASRI Conference on Intelligent Systems and Control

# APOGA: An Adaptive Population Pool Size Based Genetic Algorithm

B.R.Rajakumar<sup>a\*</sup>, Aloysius George<sup>b</sup><sup>a,b</sup> *Aloy Labs, Bengaluru, India.*

---

## Abstract

In Genetic Algorithm, the population size is an important parameter which directly influences the ability to search an optimum solution in the search space. Many researchers have revealed that having a large number of population leads to the accuracy of getting an optimal solution. But having a large population size will not be a good idea in case where the search space is small. Hence, optimal size for population pool has been determined, but the size is kept fixed. Despite the pool size is optimal, the fixed size population leads to time complexity and make the search more complex by increasing the number of generation to converge. So, the population pool size needs to be dynamically varying through the entire GA evolution of new solutions. This paper proposes an adaptive population pool based genetic algorithm, termed as APOGA, in which the population pool size either grown or shrunk at every iteration based on the performance status of the algorithm. The proposed algorithm is implemented and the performance is compared with standard genetic algorithm while solving benchmark test function with varying size of solution space. The experimental results show that APOGA outperforms standard GA for all the solution spaces.

© 2013 The Authors. Published by Elsevier B.V. Open access under [CC BY-NC-ND license](#).  
Selection and/or peer review under responsibility of American Applied Science Research Institute

*Keywords:* Genetic Algorithm (GA), APOGA, Remaining Life Time (RLT), chromosome, population pool

---

## 1. Introduction

GA is a stochastic search technique [1] [9] [3] being applied in a many real world optimization problems to determine the optimal or near optimal solutions [2] [10] [13] [8]. The GA process is initiated by arbitrary generation of population of chromosomes through which possible solutions of a given problem are considered

---

\* Corresponding author  
E-mail address: [rajakumar@ieee.org](mailto:rajakumar@ieee.org)

[6] [7]. The goodness of each chromosome is calculated using an evaluation function, termed fitness function [11]. The GA operators, crossover and mutation simulate the process of natural reproduction and mutation of species respectively [4]. As these operators hybrid the existing solutions, the resultant solutions may be far better than the existing [5]. In traditional GA, the user fixes a constant population size and it remains the same throughout the entire iterative process. If the initial population size is too small, GA may fail to reach high quality solution. In contrast, large population size makes the GA computationally complex [12]. The effect of selecting good initial population for GA performance has been studied by Pedro *et al.* [16]. They proposed to maintain diversity in the initial population.

JanneKoljonen *et al.* [14] have showed that decreasing population size increases the optimization to a certain extent. However increasing the population size increases the reliability of the optimization. The same effect by increasing the population size also been experimentally evaluated by Stanley Gotshall *et al.* [17] with three different kind of problems. Zhanshan *et al.* [15] have pioneered the effects of using dynamic size population for the performance of genetic algorithm over the fixed size population. Lobo *et al.* [18] have demonstrated the advantage of varying population size. Yuri R.Tsoy [19] reported that, in most cases large population size with less number of generations are better than smaller population size which requires large search time.

Valko *et al.* [20] proposed criteria to increase or decrease population pool size based on fitness, current iteration number and total number of iterations. According to them, population pool size can be increased either if the best fitness of a population increases or if there is no improvement in the best fitness for certain number of iterations. If the population pool does not meet the criteria to increase, then the size will be decrease at a rate of 1-5%. Just decreasing 1-5% of the population pool does not make stability with the increasing criteria. In other words, the population pool will be heavily increased when the corresponding criteria are met, however shrinking has been performed here with less significance. As a result, the probability of increasing the population pool size becomes more that leads to computational inefficiency. In 2000, Vaart *et al.* [21] calculated a life time measure for each chromosome called as Remaining Life Time (RLT) and was decreased by one in every iteration. When an RLT reaches zero, the corresponding chromosome is eliminated in the pool. Despite, RLT is based on various fitness values of population pool, the method does not discuss about increasing need of population pool size.

## 2. The APOGA

The outline of APOGA is to validate the size of population pool and to decide on whether to increase or decrease the size. With the reference of addressed issues in [20] and [21], APOGA intends to resolve them by hybridizing both the algorithms. The flow chart of APOGA is given in given in fig 1. The initialization of the population pool remains same as been done in standard GA.

### 2.1. Initialization and Fitness Calculation

Let  $P$  be the population pool with size  $N_p$ ,  $C_i : i = 1, 2, \dots, N_p$ ,  $x_j : j = 1, 2, \dots, N_l$  be the gene in chromosome and  $N_l$  be the length of each chromosome. The representations for population pool and the chromosome can be given as  $P = [C_1, C_2, \dots, C_{N_p}]$  and  $C_i = [x_1, x_2, \dots, x_{N_l}]$ , respectively. Once the initial population is set up, the next step in GA is to evaluate the fitness of each individual, (i.e. chromosome) in the population. Since the genetic algorithm is based on the Darwinian's principle in which only the fittest can survive, the fitness value of the individuals must be calculated. In our case, we define the fitness function as a minimization function, which can be represented as follows

$$C^{optimal} = \arg \min_{C_i \in P} F(C_i) \quad (1)$$

## 2.2. RLT Calculation

We use RLT calculation, which was proposed in [21], to shrink the population pool. The mathematical formulation is given below

$$RLT(i) = \begin{cases} LT^{\min} + \eta \cdot \frac{F^{worst} - F(i)}{F^{worst} - F^{avg}} & \text{if } F(i) > F^{avg} \\ \frac{1}{2}(LT^{\min} + LT^{\max}) + \eta \cdot \frac{F^{avg} - fitness(j)}{F^{avg} - F^{best}} & \text{if } F(i) < F^{avg} \end{cases} \quad (2)$$

where  $\eta = \frac{1}{2}(LT^{\max} - LT^{\min})$ . Here  $LT^{\max}$  and  $LT^{\min}$  are the maximum and minimum lifetimes of the population, respectively that can be set according to the size of search space. RLT implicitly represents the number of iterations up to which iteration, the particular chromosome is able to perform [2] [3]. RLT is continuously monitored at the end of every iteration. Once the RLT of a chromosome attains zero, the chromosome is removed from the population pool.

## 2.3. Genetic Operations

Like standard GA, APOGA firstly performs selection followed by crossover and mutation. In the selection process,  $N_p/2$  is selected based on the best fitness. It is to be noted that at every iteration  $N_p$  varies. The selected chromosomes are paired as parents and single point crossover is performed. For each pair of parents, an offspring is generated and hence for  $N_p/2$  pair of chromosomes,  $N_p/2$  offspring are generated. The obtained offspring are subjected to Gaussian random mutation, which is a popular mutation technique, to produce  $N_p/2$  new offspring.  $N_p/2$  new offspring and  $N_p/2$  parent chromosomes constitute new population pool and subjected population resizing process.

## 2.4. Population Pool Resizing

For the new offspring, RLT is calculated using Eq. (4) along with their fitness. Based on the RLT of entire population pool and the fitness values, the resizing takes place. For resizing, we exploit the criteria used in [20]. As like in [20], the population size is grown at a rate as follows

$$G = \alpha \times (I^{\max} - I) \times \frac{F^{newbest} - F^{oldbest}}{F^{initialbest}} \quad (3)$$

where,  $\alpha$  is any random value between the range (0,1),  $G$ ,  $F^{newbest}$ ,  $F^{oldbest}$ ,  $F^{initialbest}$ ,  $I^{\max}$  and  $I$  are growth size, best fitness value of current iteration, best fitness value of previous iteration, initial best fitness, maximum number of generations and current iteration number, respectively.

The growth size is considered if the best fitness value of the current population is improved, i.e. higher than the best fitness value of previous population. If improved, then the population size is increased by introducing new  $G$  chromosomes into the population. The population pool is also grown, if the best fitness does not improve for  $T$  number of iterations. If neither of the above criteria is satisfied,  $D\%$  of the chromosomes, which has least RLT values, are removed from the population pool. The RLT value of remaining chromosomes in the population pool is decreased by 1, except the best chromosome. Then the RLT of each chromosome is verified. If the RLT value of any chromosome is less than or equal to zero, then that chromosome is considered as not fit to continue in the search and hence it is removed from the population. Now, the new population enters into the next iteration and the similar process continues till the termination criterion is met.

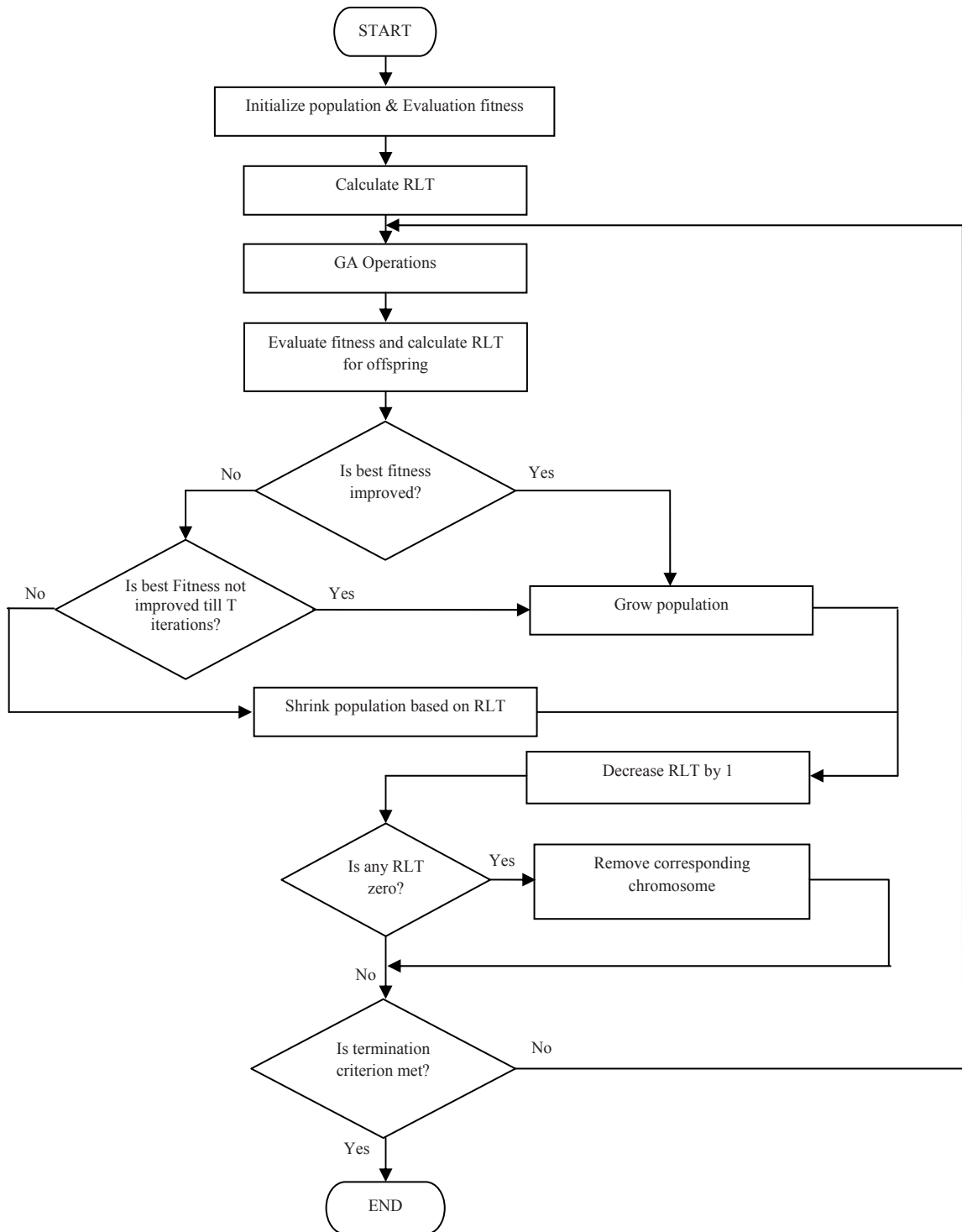


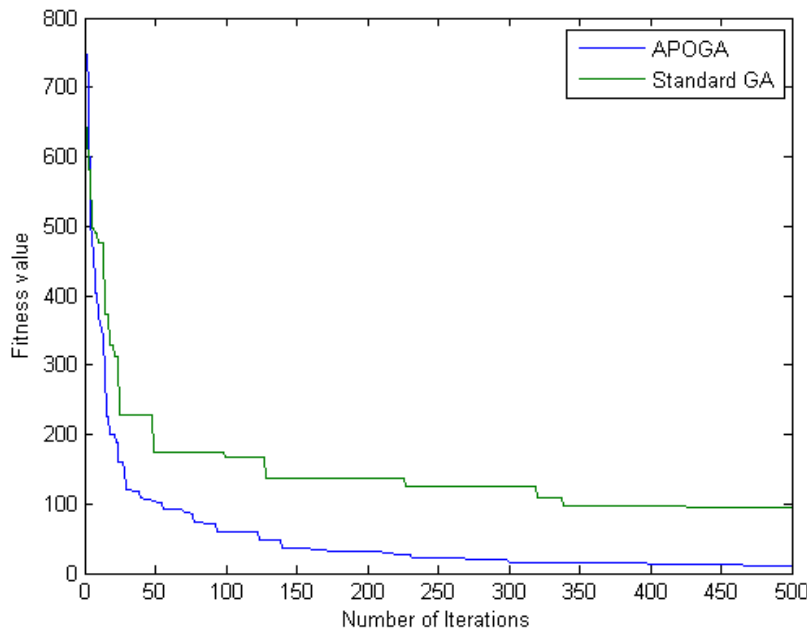
Fig 1.Flowchart for APOGA

### 3. Experimental Results

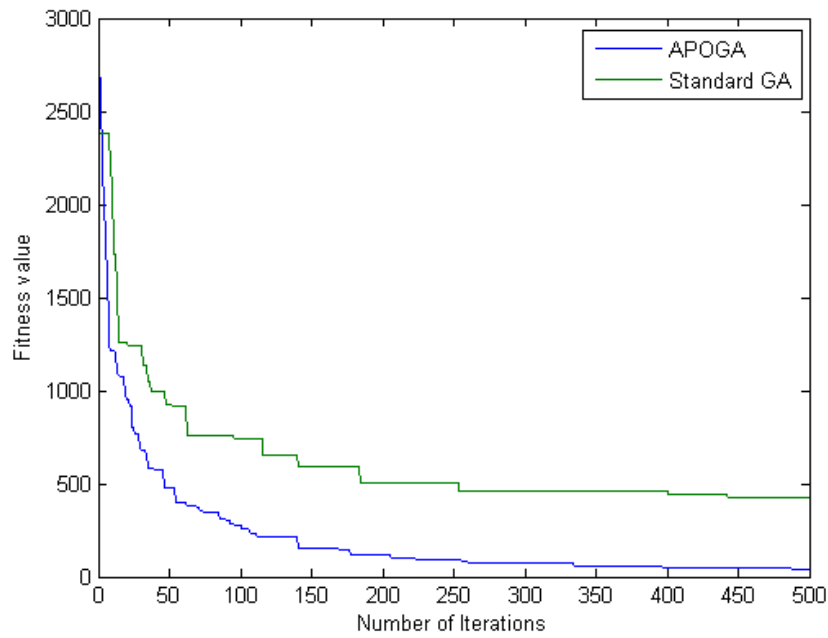
APOGA is simulated in MATLAB 7.12 and experimented on popular unimodal, continuous benchmark test function called as Dejong type I function [23]. Despite the test function is with the nominal solution space of  $(-5.12, 5.12)$ , in our work we experiment the algorithm by varying the solution space as  $(-10.24, 10.24)$ ,  $(-20.48, 20.48)$ ,  $(-100, 100)$  and  $(-300, 300)$ . In simulation, the initial population pool size, chromosome length, crossover probability, mutation probability,  $LT^{\max}$ ,  $LT^{\min}$ ,  $D\%$ ,  $\alpha$  and  $I^{\max}$  are fixed as 10, 30, 0.5, 0.15, 10, 1, 0.2, 1% and 500 respectively.

#### 3.1. APOGA versus standard GA

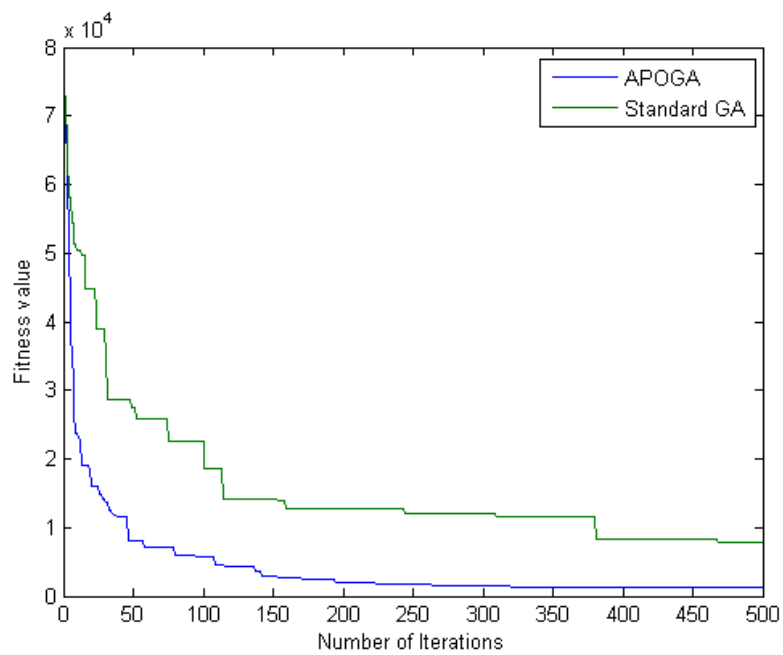
APOGA and GA are executed for finding optimal solutions of Dejong type I function and the convergence graphs are obtained for varying solution spaces. Fig. 2 demonstrates us that the convergence deviation is notably high between APOGA and GA for all the solution spaces. We can also observe that in the initial set of iterations, the best fitness values of both APOGA and GA are almost similar but when it crosses around 50<sup>th</sup> iteration (approximately) APOGA shows a drastic minimization. This in turn it explains us that the population pool size is becoming adaptive to the fitness function after a defined number of iterations. The results are encouraging as APOGA outperforms standard GA even the size of solution space is widely varied.



(ii)



(iii)



(iv)

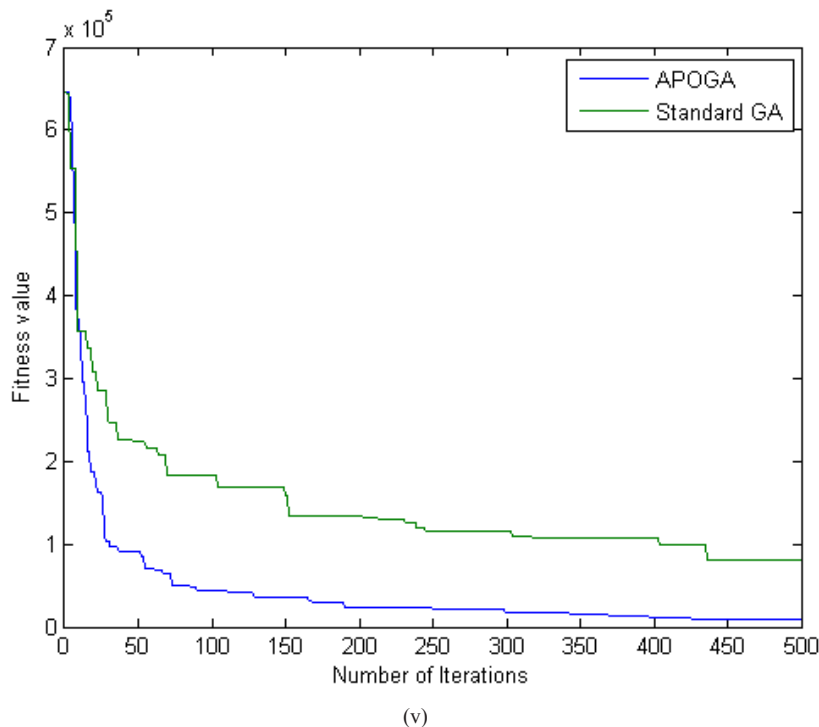


Fig. 2. Convergence graph between APOGA and GA when the solution space is (i)  $(-5.12, 5.12)$ , (ii)  $(-10.24, 10.24)$ , (iii)  $(-20.48, 20.48)$ , (iv)  $(-100, 100)$  and (v)  $(-300, 300)$

#### 4. Conclusion and Future work

This paper presented an adaptive population pool size based GA, which were called as APOGA. APOGA exploited RLT formulation for each chromosome by hybridizing with population pool growth and shrink criteria. APOGA was experimented with standard GA to visualize the convergence performance on Dejong Type I function with widening solution spaces. For five different varying solution spaces, the results were obtained. They showed that APOGA outperforms standard GA for all the test cases. As an interesting finding, APOGA and GA were similar in performance at first little iteration. After certain number of iterations, APOGA showed rapid convergence over the standard GA. Firstly we would state that both GA and APOGA are initialized with common population pool. As a result, they exhibited similar performance but in the later stages, due to increased and shrunk population pool, APOGA produced chromosome with minimum fitness. The results were encouraging and hence, APOGA will be tested with a collection of benchmark test functions and the performance will be studied over other GA variants.

#### References

- [1] Lei Wang and Dun-bing Tang. An improved adaptive genetic algorithm based on hormone modulation mechanism for job-shop scheduling problem. *Expert Systems with Applications*. 2011;38 7243-7250.

- [2] R. P. Abeysooriya and T. G. I Fernando. Canonical Genetic Algorithm To Optimize Cut Order Plan Solutions in Apparel Manufacturing. *Journal of Emerging Trends in Computing and Information Sciences*. 2012;3:2.
- [3] Abdolreza. AsadiGhanbari, Ali Broumandnia, HamidrezaNavidi and Ali.Ahmadi. Brain Computer Interface with Genetic Algorithm. *International Journal of Information and Communication Technology Research*. 2012; 2:1.
- [4] Mohammad SazzadulHoque, Md. Abdul Mukit and Md. Abu NaserBikas. An Implementation Of Intrusion Detection System Using Genetic Algorithm. *International Journal of Network Security & Its Applications*. 2012;4:2.
- [5] Rahul Kala, Harsh Vazirani, AnupamShukla and RituTiwari. Offline Handwriting Recognition using Genetic Algorithm. *International Journal of Computer Science Issues*, 2010;7:1.
- [6] Nashat Mansour, MohamadAwad and Khaled El-Fakih. Incremental Genetic Algorithm. *The International Arab Journal of Information Technology*. 2006;3:1.
- [7] Jin-Lee Kim. Integrated Genetic Algorithm and Its Applications for Construction Resource Optimization. In *Proceedings of the Winter Simulation Conference, USA*. 2010.
- [8] DawoodTalebiKhanmiri, NasibehNasiri, and TaherAbedinzadeh. Optimal Reactive Power Dispatch Using an Improved Genetic Algorithm. *International Journal of Computer and Electrical Engineering*. 2012;4:4.
- [9] EzgiDenizÜlker and SadıkÜlker. Comparison Study for Clonal Selection Algorithm and Genetic Algorithm. *International Journal of Computer Science & Information Technology*. 2012;4:4.
- [10] Vosough. Amir. Optimization the Rankine Cycle with Genetic Algorithm. *2nd International Conference on Mechanical, Production and Automobile Engineering, Singapore*. 2012; 28-29.
- [11] M. Srinivas and L. M. Patnaik. Adaptive Probabilities of Crossover and Mutation in Genetic Algorithms. *IEEE Transactions On Systems, Man And Cybernetics*. 1994;24:4.
- [12] F.G. Lobo and C.F. Lima. Adaptive Population Sizing Schemes in Genetic Algorithms. *Studies in Computational Intelligence*. 2007; 54 185–204.
- [13] AkhileshVerma and Archana. A Survey on Image Contrast Enhancement Using Genetic Algorithm. *International Journal of Scientific and Research Publications*. 2012; 2:7.
- [14] JanneKoljonen and Jarmo T. Alander. Effects of population size and relative elitism on optimization speed and reliability of genetic algorithms. In *Proceedings of the 12th Finnish Artificial Intelligence Conference STeP, Finland*. 2006; 26-27.
- [15] Zhanshan (Sam) Ma and Axel W. Krings. Dynamic Populations in Genetic Algorithms. In *Proceedings of the ACM symposium on Applied Computing, Brazil*. 2008; 16-20.
- [16] Pedro A. Diaz-Gomez and Dean F. Hougen. Initial Population for Genetic Algorithms:A Metric Approach. In *proceeding of the International Conference on Genetic and Evolutionary Methods, Las Vegas, USA*. 2007; 25-28.
- [17] Stanley Gotshall and BartyRylander. Optimal Population Size and the Genetic Algorithm. *School Of Engineering University of Portland, USA*. 2003.
- [18] F.G. Lobo and C.F. Lima. A Review of Adaptive Population Sizing Schemes in Genetic Algorithms. In *Proceedings of the Workshops on Genetic and evolutionary computation, USA*. 2005; 25-26.
- [19] Yuri R. Tsoy. The Influence of Population Size And Search Time Limit on Genetic Algorithm. In *Proceedings of the 7<sup>th</sup> Korea-Russia International Symposium on Science and Technology, Republic of Korea*. 2003.
- [20] A. E. Eiben, E. Marchiori, and V. A. Valko. Evolutionary algorithms with on the fly population size adjustment. In X. Yao et al., editors, *Parallel Problem Solving from Nature PPSN VIII, LNCS 3242*, pages 2004; 41–50.
- [21] Thomas Back, A. E. Eiben, and N. A. L. van der Vaart. An empirical study on GAs without parameters. In *Parallel Problem Solving from Nature, PPSN VI*. 2000; 315–324.
- [22] J. Arabas, Z. Michalewicz, and J. Mulawka. GAVaPS – a genetic algorithm with varying population size. In *Proc. of the First IEEE Conf. on Evolutionary Computation, Piscataway, NJ*, 1994. IEEE Press 73–78.



- [23] Molga, M., Smutnicki, C., 2005. “Test functions for optimization needs,” available at <http://www.zsd.ict.pwr.wroc.pl/files/docs/functions.pdf> (accessed on 19 February 2013)