Dynamic Population for Genetic Algorithms

S.Usama BUKHARI, Remus BRAD, Ioan BONDREA

***Abstract*—In the field of Genetic Algorithms (GA), finding the best population size for a specific problem is very important issue. Normally, the search space is huge or sometimes contains nearly infinite possibilities. Choosing the right population and its size that can lead to a good solution is always difficult. In this paper, a new concept of dynamic population is presented, which allow GAs to utilize the full domain of possible inputs of the search space while keeping the fix population size.**

Genetic Algorithm, evolutionary computing, dynamic population, Elitism

I. INTRODUCTION

Defining the correct population size is one of the important issues in the field of evolutionary computing [1], [13]. It is argued that a small population size could lead to a poor solution [12], [13], [7] while a large population size could result in more computational time required to find a solution [10], [9], [6], [7]. This leads to a trade-off situation, where the approximation of the correct population size is needed, in order to obtain a good solution. In [12], [14] and [6], it is agreed that, population size is in direct relation to the difficulty "*n*" of problem, i.e bigger the "*n*", bigger the population size needed. In [12] Pelikan uses the Bayesian Optimization Algorithm (BOA) for population sizing. Goldberg gives the idea of building blocks (BBs) [5] that should be supplied to GAs in order to obtain good results. Harik and Lobo [6] concluded that the population size is directly proportional to the number of building blocks of the problem to facilitate a good solution. The algorithm may not find a correct solution, if the BBs are not enough in the initial population. Harik and Lobo also addressed the problem of population sizing using self-adaptation, where two approaches were used: 1) population size remains constant during all iterations; 2) having different population size depending on the parameters, like fitness values (self-adaptation). The most common method to initialize and size the initial population is the empirical method [3]; different population size is feed to the algorithm and the one that gives the best result is reported. However, it is argued that the method should be used only in cases where the estimation of the problem is difficult [9]. Nevertheless, it is possible that in the case where the variables are dependent, the algorithm would gives bad results while population size is increased. In general, the problem to find a good initial population and optimal size is a difficult task [3]. A generic rule to approximate the population size can't be applied in case of different problems [11]. In [2], [15], [16] it is suggested that, diversity in population helps to reach to a good solution. Diversity is also useful in initial population to avoid premature convergence [4], [8].

In this paper, a concept of Dynamic Population (DP) is introduced. DP finds new values from the search space in each generation and discards the same amount of elements from the working population. This technique leads to a more diverse population, addressing the issues of bottleneck for the case of small population size. Section 2 introduces the proposed technique, while section 3 presents the framework, followed by the experiments and results in section 4. Conclusion and future work are shown in section 5.

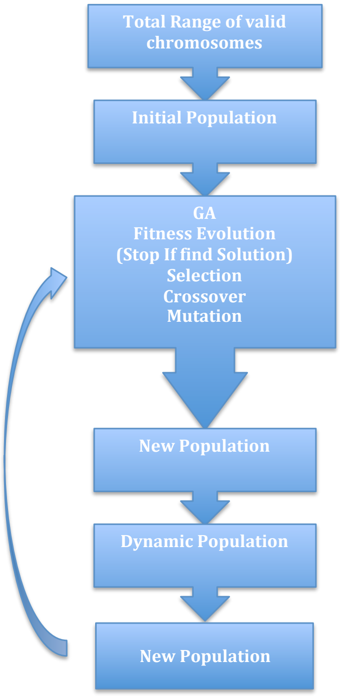
II. THE PROPOSED TECHNIQUE

In any given problem, the range of valid search space for choosing the initial population is very big and in many cases infinite. To choose a small subset of that range, which can lead to a good solution, is always a challenge in terms of diversity and size. Genetic Algorithms are also dependent on mutation and crossover to avoid falling in local minima. Finding the optimum parameters for mutation and crossover for different problems is another difficulty. The concept of Dynamic Population could be the option to overcome the difficulty of choosing the initial population and falling in local minima.

For the proposed method, in every generation, a small amount of chromosomes from the working population is replaced with new chromosomes from the search space. For example, if the size of working population is 100 and the range of search space is [-10000, 10000], the population size will represent very small portion of the range. After the selection of initial population, the results will depend upon mutation and crossover for generating the diversity in next generations. In this regard, the DP can be used in aid with mutation and crossover.

III. THE FRAMEWORK

The use of DP has one parameter, rate of change (Ŕ) with the objective to define how many chromosomes of working population should be replace with the new chromosomes from the search space. The replacement process uses the fitness function to identify the week members of population to be replaced.



```
Pseudo Code:
1. [Initialization] Generate the initial population by random selection.

2. [Fitness Evaluation] calculate the fitness value of each individual in the population.

3. [Breeding]

    • Select the parents from the population using selection mechanisms
    • Mate the parents to produce new offspring
    • Mutate the new offspring
    • Calculate the fitness of offspring
    • Replace the offspring

4. [Dynamic population] Select the weak individuals and replace them with new individuals from the search space.

5. [Termination] Repeat the process from step 2 till the termination condition is reached.
```

IV. EXPERIMENTS AND RESULTS

The comparison of standard Genetic Algorithm (SGA), Genetic Algorithm with Elite (GAE) and DP method will examine the performance and the effect of DP method on the population. Comparison was done on the function in equation 1.

$$F1 = \sum_{i=0}^{n} x_i^2 \qquad (1)$$

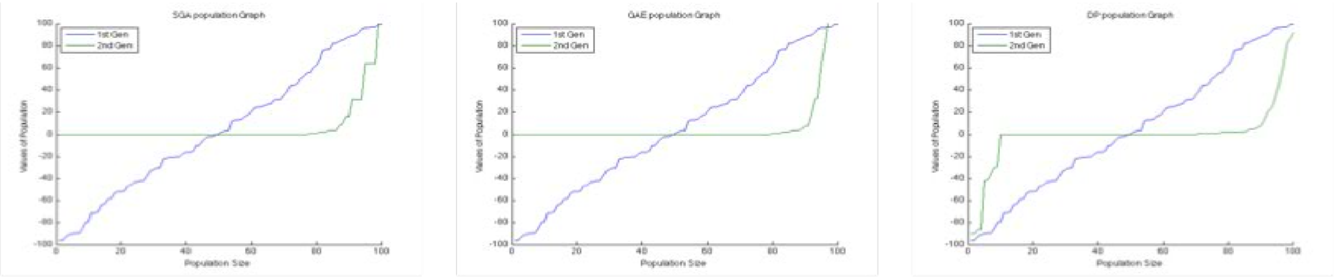
where the domain is $[-100,100]^n$ and n =10.

The initial population was generated randomly with 100 chromosomes. All the algorithms are allowed to run for 50, 100, 200 and 400 times. The effects of change in population are presented in the figures 1 to 4. The change is shown every 50 generations.

The parameters used for the experiments were:

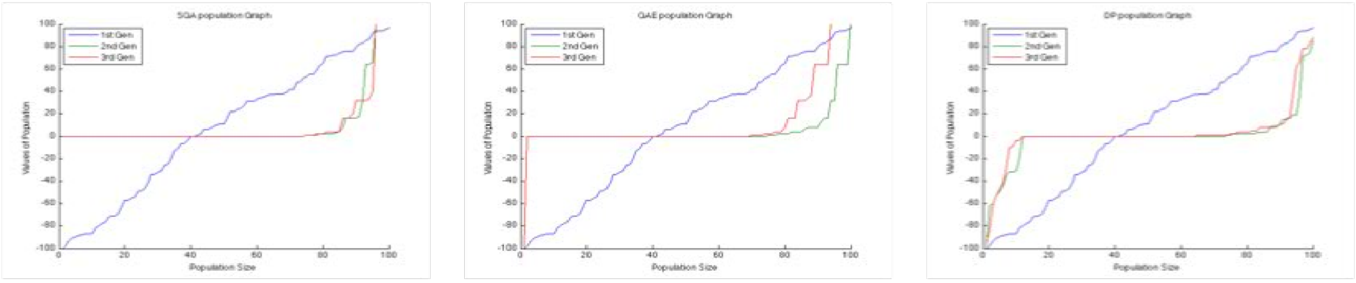
- No. of Individuals: 100
- Selection Mechanism: Tournament Selection
- Crossover Type: Uniform Crossover
- Crossover Rate: 0.20
- Mutation Type: Flap Mutation
- Mutation Rate: 0.3
- Dynamic Population Rate: 0.1

With the above-mentioned parameters, the comparison was made between SGA, GAE and the proposed DP method.



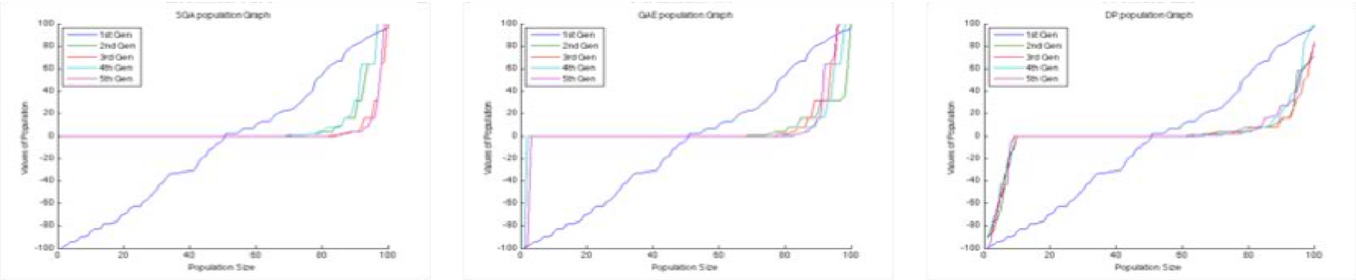
(a) (b) (c)

Figure 1: Comparision of population evaluation with 50 iterations, where (a) SGA (b) GA Elite and (c) Purposed DP method



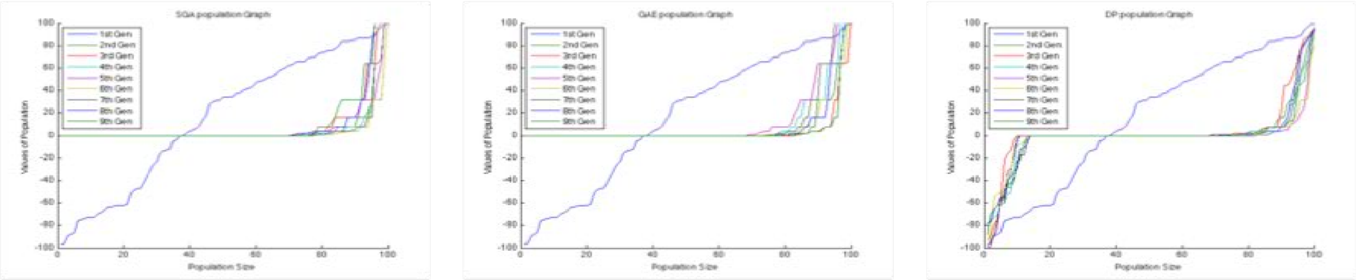
(a) (b) (c)

Figure 2: Comparision of population evaluation with 100 iterations, where (a) SGA (b) GA Elite and (c) Purposed DP method



(a) (b) (c)

Figure 3: Comparision of population evaluation with 200 iterations, where (a) SGA (b) GA Elite and (c) Purposed DP method



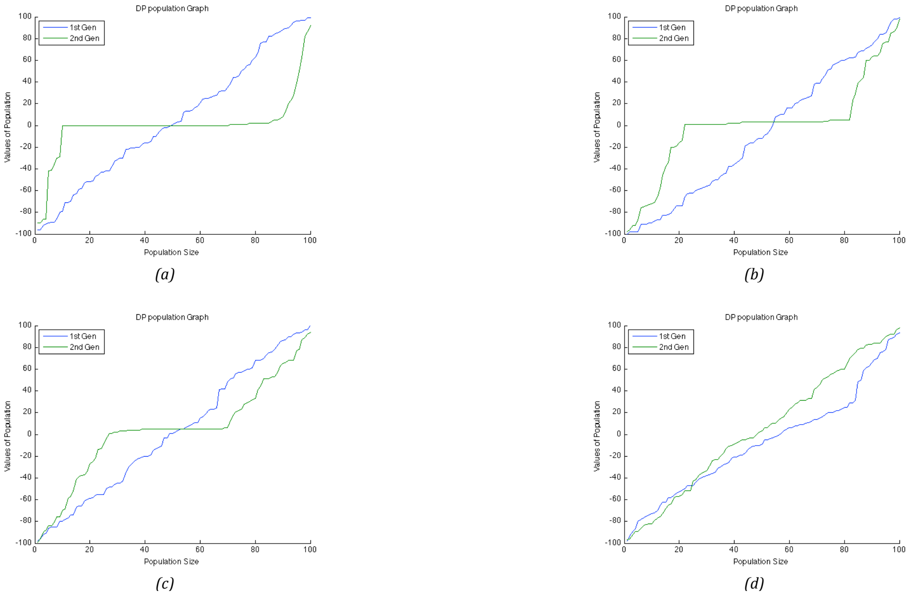
(a) (b) (c)

Figure 4: Comparision of population evaluation with 400 iterations, where (a) SGA (b) GA Elite and (c) Purposed DP method

The graphs in figures 1 to 4 shows that the population of the proposed method is more diverse than the other two algorithms. To compare the effect of the change in Ŕ (from 0.1, 0.2, 0.3 and 0.5), the experiments were completed with the population size of 100, initially generated by random selection. All the algorithms are allowed to run for 50, 100, 200 and 400 times. The change in generation is shown after every 50 generations.

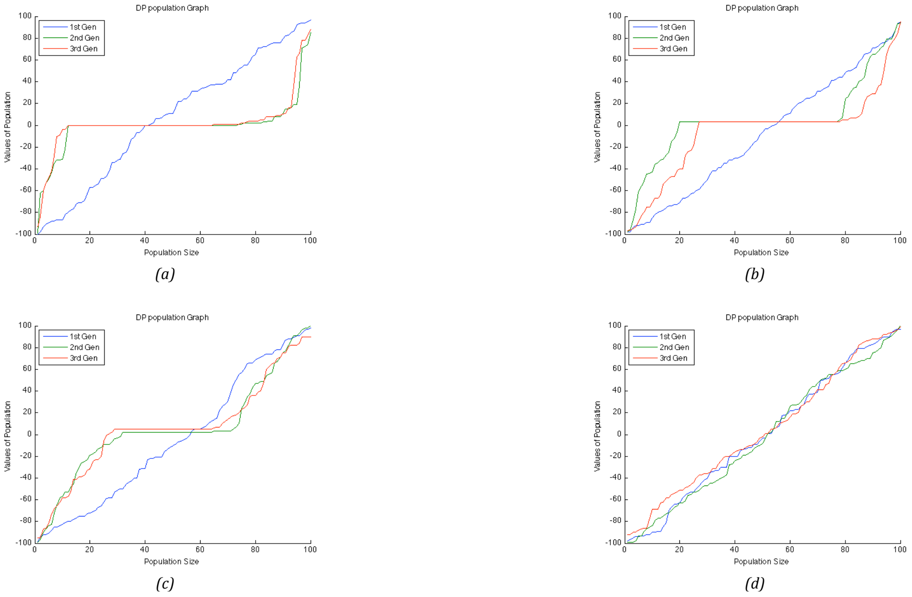
The parameters used for the experiments were:

- No. of Individuals: 100
- Selection Mechanism: Tournament Selection
- Crossover Type: Uniform Crossover
- Crossover Rate: 0.20
- Mutation Type: Flap Mutation
- Mutation Rate: 0.3
- Dynamic Population Rate: 0.1, 0.2, 0.3, 0.5



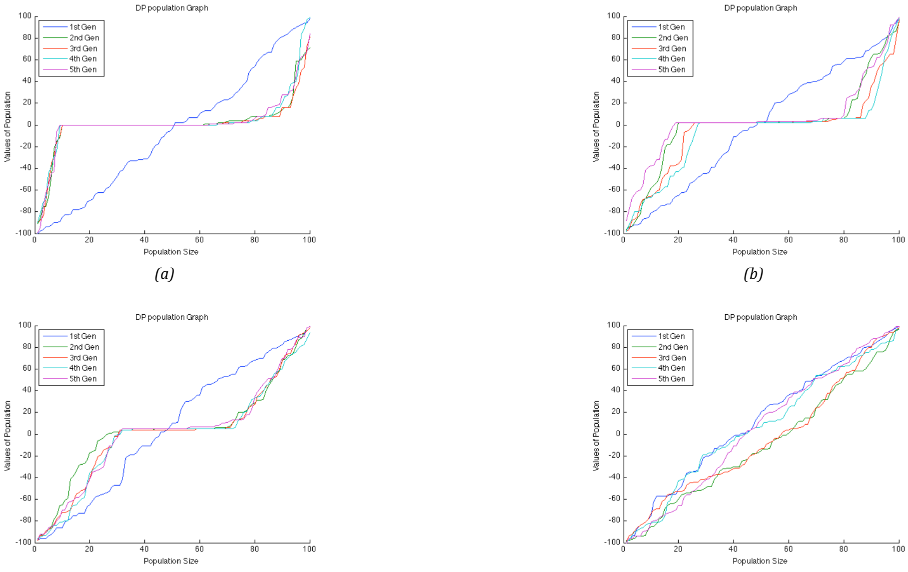
(a) (b) (c) (d)

Figure 5: Comparison of the population evaluation with 50 iterations, with the parameter R of DP is (a) 0.1 (b) 0.2 (c) 0.3 (d) 0.5



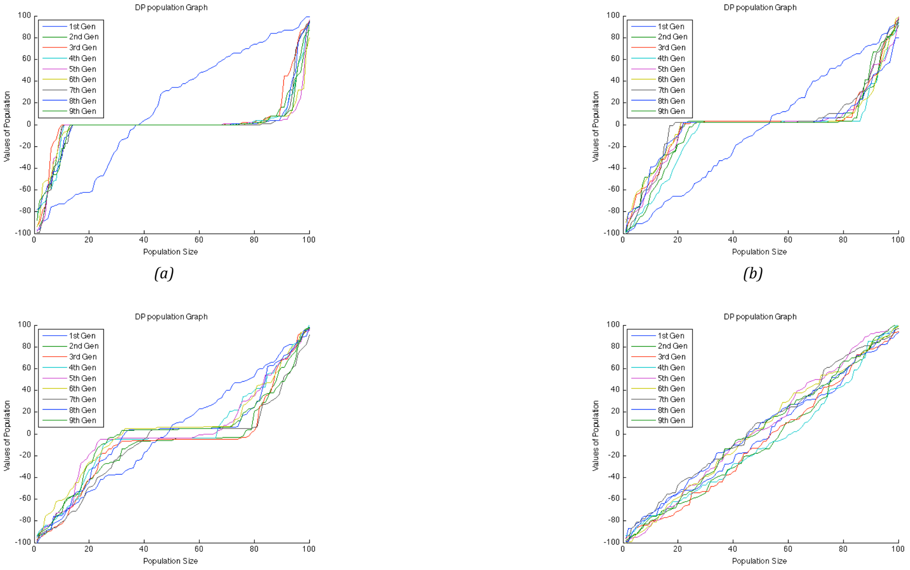
(a) (b) (c) (d)

Figure 6: Comparison of the population evaluation with 100 iterations, with the parameter R of DP is (a) 0.1 (b) 0.2 (c) 0.3 (d) 0.5



(a) (b) (c) (d)

Figure 7: Comparison of the population evaluation with 200 iterations, with the parameter R of DP is (a) 0.1 (b) 0.2 (c) 0.3 (d) 0.5



(a) (b) (c) (d)

Figure 8: Comparison of the population evaluation with 400 iterations, with the parameter R of DP is (a) 0.1 (b) 0.2 (c) 0.3 (d) 0.5

The graphs in figures 5 to 8 are showing that the behaviors of the population for different values of Ŕ are nearly the same for all the iterations. Therefore, we could conclude that the iteration has a minimum role in the way the population behaves. Also, it can be observed that, the bigger the value of Ŕ is, the more the population is diverse.

Detailed experiments have been performed to analyze the results of the DP, SGA and GAE. In this case, the initial population size was set to 100, and the population was generated randomly. The convergence was considered when the algorithm gives the same value for 5, 10, 20, 30, times. The algorithms were run 15 times for all the different iterations numbers (50, 100, 200 and 400) and the results were considered by averaging. The following functions were used to perform the experiments (equations 1 to 3):

$$F1 = \sum_{i=0}^{n} x_i^2$$

where the domain was [-100,100] ^{n} and n =10.

$$F2 = \sum_{i=1}^{n} |x_i| + \prod_{i=1}^{n} |x_i| \qquad (2)$$

where the domain was [-10, 10] ^{n} and n =30.

$$F3 = \sum_{i=0}^{n} \left(\sum_{j=1}^{i} x_j\right)^2 \qquad (3)$$

where the domain was [-100,100] ^{n} and n =10.

Table I

		SGA			GA Elete			Purposed Method(DP)		
		SD	Avg. SD	Cong.	SD	Avg. SD	Cong.	SD	Avg. SD	Cong.
F1	50	17.9	19.6	24	29.1	28.9	15	24.3	28.7	15
	100	25.9	19.7	15	25.3	30.1	17	30.5	30.6	52
	200	16.7	18.5	17	22.3	28.8	87	22.7	28.5	105
	400	30.6	16.5	8	39.7	30.9	37	23.3	28.2	206
F2	50	33.1	26.7	NA	37.5	37.1	NA	33.3	35.1	NA
	100	31.7	29.5	4	34.6	37.5	96	31.7	34.6	75
	200	31.4	33.1	112	38.4	37.1	25	29.7	34.5	154
	400	32.2	31.6	44	31.7	36.3	104	37.2	35.0	258
F3	50	22.9	20.6	7	18.9	22.8	6	29.2	26.7	NA
	100	22.6	19.5	18	26.4	21.9	36	31.6	27.1	40
	200	22.0	19.4	69	20.4	22.5	77	23.9	26.6	91
	400	21.8	19.1	17	17.2	21.4	14	24.9	26.6	241

Showing the Standar Deviation (SD) for the final population, Average SD of all generations and convergence rate (Cong.) of all three algorithm using the population size 100.

Table II

		0.2			0.3			0.4		
		SD	Avg. SD	Cong.	SD	Avg. SD	Cong.	SD	Avg. SD	Cong.
F1	50	29.1	26.9	NA	32.5	32.1	NA	38.0	41.7	NA
	100	24.6	26.9	15	30.9	32.1	41	40.5	41.2	78
	200	26.1	26.6	40	32.9	32.8	109	41.5	40.9	140
	400	30.3	26.5	61	31.1	32.1	167	35.2	40.9	291
F2	50	33.1	34.1	NA	34.5	38.1	NA	49.3	46.2	NA
	100	31.7	33.5	21	34.6	37.5	96	41.1	46.6	75
	200	31.4	33.8	112	35.4	37.1	25	47.6	46.5	154
	400	32.2	34.1	44	33.7	37.3	201	43.3	46.0	258
F3	50	28.7	27.2	7	32.0	31.9	6	39.8	40.8	10
	100	24.6	27.5	38	27.1	31.9	66	38.6	41.1	71
	200	24.0	27.4	61	29.4	32.5	117	34.9	41.6	151
	400	28.8	26.9	100	31.2	32.4	204	39.9	40.9	275

Showing the Standard Deviation (SD) for the final population, Average SD of all generations and convergence rate (Cong.) of different parameters of DP , (0.2, 0.3 and 0.5)using population size 100.

The parameters of above experiments were:

- No. of Individuals: 100
- Selection Mechanism: Tournament Selection
- Crossover Type: Uniform Crossover
- Crossover Rate: 0.20
- Mutation Type: Flap Mutation
- Mutation Rate: 0.3
- Dynamic Population Rate: 0.1,0.2,0.3,0.5

The tables shown very interesting results, as the average Standard Deviation (SD) of the population is very predictable for the proposed method compared to the other algorithms, for all functions. Also, the different values of Ŕ show a predictable behavior of population. The results are revealing that the proposed method takes longer time to converge than others algorithms, which can solve the problem of premature convergence.

V. CONCLUSION

The factor of randomness is an obstacle in analyzing any evolutionary computing algorithm. The results of the present paper shows that Dynamic Population could generate a more predictable and diverse in nature population. Also, the proposed method can be considered as a step forwards towards solving the problem of premature convergence.

Nevertheless, the paper covers only the basic concept of the proposed method. Thus, finding the optimum values of Ŕ for different problems could be the purpose of further research. Due to the much more predictable nature of population diversity, more and better analyzes can be made to predict the convergence velocity and state of population.

REFERENCE

[1] Grefenstette, John J., ed. *Genetic Algorithms and Their Applications: Proceedings of the Second International Conference on Genetic Algorithms*. Psychology Press, 2013.

[2] E. K. Burke, S. Gustafson, and G. Kendall, "Diver- sity in genetic programming: An analysis of measures and correlation with fitness," *IEEE Transactions on Evolutionary Computation*, vol. 8, no. 1, pp. 47–62, 2004.

[3] A. E. Eiben, R. Hinterding, and Z. Michalewicz, "Parameter control in evolutionary algorithms," *IEEE Transactions on Evolutionary Computation*, vol. 3, no. 2, pp. 124–141, 1999.

[4] W. G. Frederick, R. L. Sedlmeyer, and C. M. White, "The hamming metric in genetic algorithms and its application to two network problems," in *Proceedings of the ACM/SIGAPP Symposium on Applied Computing*, 1993, pp. 126–130.

[5] Mathew, Tom V. "Genetic algorithm." *Report submitted at IIT Bombay* (2012).

[6] G. R. Harik and F. G. Lobo, "A parameter-less genetic algorithm," in *Proceedings of the Genetic and Evolutionary Computation Conference*, 1999, pp. 258–265.

[7] V. K. Koumousis and C. P. Katsaras, "A saw- tooth genetic algorithm combining the effects of variable population size and reinitialization to enhance performance," *IEEE Transactions on Evolutionary Computation*, vol. 10, no. 1, pp. 19–28, 2006.

[8] Y. Leung, Y. Gao, and Z. Xu, "Degree of population diversity—a perspective on premature convergence in genetic algorithms and its Markov chain analysis," *IEEE Transactions on Neural Networks*, vol. 8, no. 5, pp. 1165–1176, 1997.

[9] F. G. Lobo and D. E. Goldberg, "The parameter- less genetic algorithm in practice," *Information Sciences—Informatics and Computer Science*, vol. 167, no. 1-4, pp. 217–232, 2004.

[10] F. G. Lobo and C. F. Lima, "A review of adaptive population sizing schemes in genetic algorithms," in *Proceedings of the Genetic and Evolutionary Computation Conference*, 2005, pp. 228–234.

[11] M. Lunacek and D. Whitley, "The dispersion metric and the CMA evolution strategy," in *Proceedings of the Genetic and Evolutionary Computation Conference*, 2006, pp. 447–484.

[12] M. Pelikan, D. E. Goldberg, and E. Cantu´-Paz, "Bayesian optimization algorithm, population sizing, and time to convergence," Illinois Genetic Algorithms Laboratory, University of Illinois, Tech. Rep., 2000.

[13] A. Piszcz and T. Soule, "Genetic programming: Optimal population sizes for varying complexity problems," in *Proceedings of the Genetic and Evolutionary Computation Conference*, 2006, pp. 953–954.

[14] T.-L. Yu, K. Sastry, D. E. Goldberg, and M. Pe- likan, "Population sizing for entropy-based model build- ing in genetic algorithms," Illinois Genetic Algorithms Laboratory, University of Illinois, Tech. Rep., 2006.

[15] Peralta, Richard C., Ali Forghani, and Hala Fayad. "Multiobjective genetic algorithm conjunctive use optimization for production, cost, and energy with dynamic return flow." *Journal of Hydrology* 511 (2014): 776-785.

[16] Auger, Anne, and Benjamin Doerr, eds. *Theory of Randomized Search Heuristics: Foundations and Recent Developments*. Vol. 1. World Scientific, 2011.