

# Population Control in Evolutionary Algorithms: Review and Comparison

Yuyang Guan<sup>1</sup>, Ling Yang<sup>1</sup>, and Weiguo Sheng<sup>2</sup>(✉)

<sup>1</sup> College of Computer Science, Zhejiang University of Technology,  
Hangzhou 310023, Zhejiang, People's Republic of China

<sup>2</sup> Department of Computer Science, Hangzhou Normal University,  
Hangzhou 311121, Zhejiang, People's Republic of China

w.sheng@ieee.org

**Abstract.** Population size in evolutionary algorithms (EAs) is critical for their performance. In this paper, we first give a comprehensive review of existing population control methods. Then, a few representative methods are selected and empirically compared on a range of well-known benchmark functions to show their pros and cons.

**Keywords:** Evolutionary algorithms · Population control · Parameter control

## 1 Introduction

Evolutionary algorithms (EAs) are a broad class of stochastic search heuristics that perform optimization or learning tasks. While they have been successfully applied to various science and engineering problems, the performance of EAs depends critically on the setting of various parameters [1], among which the population size is perhaps the most important one. Generally, a too small population size will lead to the premature convergence of EAs, while a too large size will reduce the EAs' efficiency. In traditional EAs, the population size is typically specified by the user to a fixed value and remains constant during the entire evolution. However, it has been well recognized that the optimal size of population is problem-specific [2–6]. Further, it has been demonstrated that different population sizes may be required at different stages of evolution [7–10]. To specify the population size in such a manner therefore may significantly limit the performance of EAs.

To deal with the above issue, dynamic population control has become an active yet important research topic and many different schemes have been proposed in literature [9, 11–14]. Several works of reviewing or comparing existing methods have been presented. In 1999, Costa [2] gave a comparative study of population control methods, in which five dynamic population control schemes are reviewed and compared. More than a decade later, Karafotias [15] provided a survey regarding to the parameter control in EAs. Although a few dynamic

population control methods have been reviewed, this work focuses mainly on parameter setting schemes for variation operators of EAs. Recently, Piotrowski [16] presented another review along with comparison results of several dynamic population control schemes. The scope of this work, however, is limited to the differential evolution (DE).

In this paper, we give a comprehensive review of the population control methods proposed in EA literature. Further, several representative methods have been selected and compared to gain deep understanding of their pros and cons.

## 2 Population Control in Evolutionary Algorithms

Much work has been carried out regarding to the control of population size in EAs. In this section, we first provide an overview of theoretical studies in the field and then review exiting methods, which are divided in three categories, i.e., deterministic, adaptive and self-adaptive methods.

### 2.1 Theoretical Studies

In [17], Holdener recognized that the optimal allocation of trials in genetic algorithm (GA) characterized by the conjunction of substantial complexity and initial uncertainty as well as a requirement of employing new information in the population to rapidly reduce this uncertainty. This area remains untouched for many years until Goldberg studied the population size as a parameter from a theoretical point of view based on the decomposition for designing competent GAs. In this work, Goldberg [18] considers the evolution of GA as the growth of building blocks (BBs) and points out that a successful GA should ensure an adequate supply of BBs in the initial population. In [19], Reeves calculated the minimum population size required for at least one instance of every allele being presented at each locus in binary string as well as n-nary representation. While having all alleles present in the initial population is important, this work suggests that the present of desired building blocks in the initial population is crucial for the GAs' performance. Later, in [20], Goldberg determined the probability of actual building blocks presented in a population and developed models to calculate population size required for the success probability of GA asymptotically reaching 100%. Apart from the building block theory, the population size has also been studied under the statistical decision. In [21], De Jong presented several equations to determine the population size based on the noise theory in statistical decision. In [7], Goldberg extended the equations to calculate the size of population based on the permissible errors between building blocks, variation and complexity of the problem. While, in [22], Harik developed an alternative equation to predict the quality of solution found in a given population size based on the theory of random walks and gamblers ruin. Unfortunately, the above theoretical studies are generally not practical for real applications. However, they do reveal that the population size is crucial for the performance of EAs.

## 2.2 Deterministic Methods

To deal with the issue of population size, many methods seek to dynamically change it during evolution based on deterministic rules. In [12, 23–25], Fernández et al. presented a method based on the phenomena of plague [12], in which a fixed number of individuals (i.e., the worst ones) are removed at every generation of evolution. Instead of removing individuals at each generation, Brest [13] proposed a method, which gradually reduces the population size by half each time when a specific condition is met during the evolution. This reduction mechanism has been successfully used in EAs for solving various optimization problems [26–32]. Extending the above mechanism, in [33], Brest et al. designed a method, which starts with a small population size at the beginning. Then, during the evolution, the population is first increased with a specific size determined by a constant value and then reduced by half. Although it has shown to be promising [33], Neri [34] pointed out that such a method is not working well in DE. Rather than increasing or decreasing a specific number of individual after each specific number of generations during evolution, a few methods have been proposed to automatically adjust the size of population based on predefined functions. For example, Koumouis et al. introduced functions with saw-tooth shape [35] and sinusoidal oscillating [36] for adjusting the population size. Other functions, such as linear functions [30, 37, 38] have also been used for population control and applied in various optimization problems [39–42].

## 2.3 Adaptive Methods

Adaptive methods utilize feedback from the search to determine the direction and magnitude of change of population size. Based on the feedback information used, we divide the existing methods into three categories and review them accordingly.

**Fitness Based Methods.** The methods falling into this category adaptively change the population size based on the fitness of individuals. In [43–47], the size of a population is modified if the fitness of the best individual does not change for a period of time. Specifically, Montiel [43] et al. calculated the amounts of individuals should be inserted or deleted from the population based on two variables, “Cycling” and “Variance”. The “Cycling” is used to account the number of times that the fitness of the best individual does not have a significant change and the “Variance” is used to calculate the variance of the fitness of the best individuals during the “Cycling” period. In [44], new individuals are inserted into the population when the fitness of the best individual does not improve in  $m$  generations. In [45, 48], the population is first ordered based on the fitness of individuals and a predefined percent of individuals is inserted or deleted from the population. In [46], a predefined number of individuals are added if the best individual does not change for  $m$  generations while the population is resized to its initial size if the best individual improves its fitness. In [47], a new population is created by selecting good individuals from an extinguished population when

extinction occurs, and at every generation, individuals will be eliminated when they reach their lifetime. In [49–53], subpopulations are used for competitive evolution and are resized based their fitness. Typically, when the average fitness of a subpopulation falls below a predefined significance level, it will be removed. Specifically, in [50], Smorodkina provided a competition scheme between two parallel subpopulations, while Hinterding [51] and Harik [52] do not restrict the number of subpopulations. Schlierkamp-Voosen [49] employs a migration scheme, in which the best individual is migrated to all other subpopulations. While, Zhan [53] suggested that all the poor subpopulations randomly migrate one individual to the good subpopulations.

Apart from the above methods, some methods employ lifetime and survival probability, which is derived from the fitness, to control the population size. The classical method is the Genetic Algorithm with Varying Population Size (GAVaPS) proposed by Arabas et al. [11], which introduced the concept of age for the individual. In this method, population size is dictated by birth and death of individuals, controlled by their lifetime and measured by fitness. Proceeding in a generational manner, each individual's age is increased at each time step and removed when it exceeds its lifetime. The lifetime mechanism is extended using non-random mating to prevent incest by Fernández et al. [54,55], and applied to multimodal optimization [56], vehicle detection [57] and distribution network reconfiguration [10]. The main drawback of GAVaPS is that, in the worst case, the population size will be doubled in the successive generations thus may lead to population explosion. To address such an issue, Bäck [58] proposed to preserve all the fittest individuals and only two new individuals will be generated and inserted into the population at each generation. This scheme has been applied to co-operative co-evolution by Lorio and Li [59]. In [47], Zhang developed an extinction scheme for population control. In this scheme, individuals will be eliminated if there are no improvements made during the evolution. While, in [60], Vellev introduced the concept of survival probability, which defines a collective probability used to control the size of the population. Along this line, Cook [61] described another formula to calculate the survival probability for population size control.

**Distribution Based Methods.** This kind of methods control the population size based on the distribution information of population. In [7], Goldberg first proposed a micro GA in which the population is reinitialized when a nominal convergence (e.g., when all the individuals have either identical or very similar genotypes) is detected. In [62], a similar re-initialization scheme is introduced with an elitist strategy along with a competition scheme, which chooses the winner individuals to be included into the population. The micro GA has been successfully applied in various applications [63–65], which show a better performance compared to the standard GA. Another way to measure the distribution of population is to calculate the distance between individuals. In [66,67], Khor and Tan suggested the desired population size should be bounded within a limit, measured by the maximum distance between each other in the population.

Liang [68] combined the relative ascending directions and the distance between two individuals to determine whether they are located in the same peak. This information is then used for population control. In [69], Yang calculated the sum Euclidean distances and generated a new population if it does not change for a period of time. While in [70], the individual with the smallest Euclidean distance will be removed at every generation as a mechanism for population control. In [71], Auge introduced a dispersion degree for population control. A population initialization is launched when the calculated dispersion degree meets the stopping criterion and the population size is increased by a factor of 2. In addition, Shi [72] assumed the center of the population moved slowly when the variance of population is large. In this case, the population size is decreased to accelerate the speed of convergence and vice versa.

**Fitness and Distribution Based Methods.** The methods in this category are based on both the fitness and distribution information for population control. Smith [73] presented such a method, in which the mean fitness and variance of population is calculated and compared to a predefined desired value to decide whether assign more (or less) individuals to a population. This method was extended by Harik [22], in which the distance between the best and second best BBs is also considered. In [74], Tirronen provided a method based on fitness diversity measured by the distance between pairs of individuals along with their fitness values to control population size.

**Population Status Based Methods.** These methods change the population size when the current population status meets a certain condition. They have been used in genetic programming (GP), where solutions are represented as trees of variable size and depth, for population control. In [75], Wagner proposed to employ the node information for population control. Two kinds of nodes (i.e., soft and hard) are used and the soft node is employed to control the addition of new individuals to a population while the hard node is used to control which individuals might be added. This method has been used in GP to address forecasting problem in [76, 77]. In [70], Ding presented an artificial bee colony algorithm where the colony size is used to control the reproduction and death rate of the population. If the colony size is larger than the initial size, the reproduction rate threshold will be increased while the death threshold is decreased.

## 2.4 Self-adaptive Methods

This approach encodes the population size to be adapted into the individuals and undergo mutation and recombination. Such a method was first proposed by Teo [78] to adapt the population size in different evolution (DE). The method, however, shows no significant advantages over the DE with fixed population sizes. In [79], Eiben discussed such an approach and proposed a hybrid variant of self-adaptive method for GA. The work shows that it is possible to define the population size at individual level. The results show that the proposed method has better performance than GAs with fixed population sizes.

**Table 1.** Parameter setting of the six algorithms

Methods	Parameter	Values
dynNP-DE	Number of different population sizes ( $P_{max}$ )	4
	Crossover operator	2-point
SAMDE	The sign of trigger (P)	P = 0: increase P = 4: decrease
	Crossover operator	2-point
GAVaPS	MaxLifeTime/MinLifeTime	7/1
	The rate of choosing individuals to reproduce	40%
APAGA	MaxLifeTime/MinLifeTime	11/1
	The rate of choosing individuals to reproduce	Fixed 2
PL-GA	Crossover operator	Uniform
GPS-GA	Crossover operator	Uniform

### 3 Experimental Comparison

In this section, we shall experimentally compare several representative methods for dynamically population control to show their advantages and disadvantages. In the following subsections, after describing the data sets in Subsect. 3.1 and the parameter settings of the algorithms in Subsect. 3.2, we report the comparison results in Subsect. 3.3.

#### 3.1 Test Suit

To access the performance of different population control strategies, a range of well-known benchmark test functions have been considered. For the unimodal case, we select four functions presented in [21] with minimum value zero, the Sphere Model, Griewangk's Function, Step Function and Quartic Function. For the multi-modal case, the Rastrigin's function [29], Six-Hump Camel Back Function, Branin Function and Goldstein-Price's Function have been considered. In addition, the Schwefel's Function [71], which is a deceptive function, and the Rosenbrock's Valley Function [16], which has a very complex search space, has also been considered.

#### 3.2 Algorithms and Parameter Settings

The following six population control algorithms have been selected for comparison. The GAVaPS [11], in which each newly born individual is allocated with a lifetime measured by fitness. Proceeding in a generational manner, individual's age is increased at each generation and removed when it exceeded its lifetime. The APAGA [58], which is similar as the GAVaPS with an exception that the fittest member will always kept in the population. In addition, only two new

**Table 2.** Comparison results of six methods on 10 benchmark functions

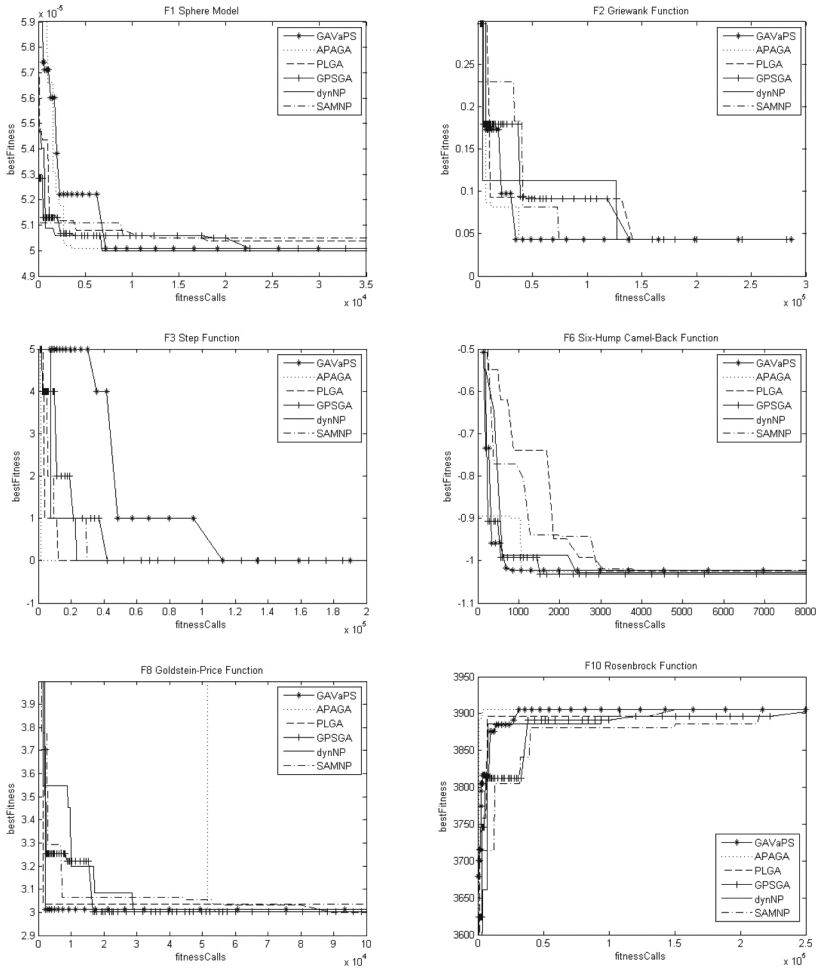
Fun.	No. of Calls	Methods					
		GAVaPS	APAGA	PL-GA	GPS-GA	dynNP-DE	SAMDE
$f_1$	100000	5.02E-05	<b>5.01E-05</b>	5.02E-05	5.04E-05	5.02E-05	5.02E-05
$f_2$	250000	4.94E-02	5.14E-02	5.74E-02	4.91E-02	5.08E-02	<b>3.87E-02</b>
$f_3$	200000	0.11E+00	0.11E+00	<b>0.00E+00</b>	<b>0.00E+00</b>	<b>0.00E+00</b>	<b>0.00E+00</b>
$f_4$	300000	7.43E-03	<b>3.52E-03</b>	1.05E-02	1.56E-02	1.59E-02	1.23E-02
$f_5$	100000	-1.02E+01	-1.02E+00	<b>-1.03E+00</b>	-1.02E+00	<b>-1.03E+00</b>	<b>-1.03E+00</b>
$f_6$	200000	0.40E+00	0.45E+00	<b>0.39E+00</b>	<b>0.39E+00</b>	0.40E+00	0.40E+00
$f_7$	200000	0.31E+01	0.31E+01	0.31E+01	0.39E+01	<b>0.30E+01</b>	0.31E+01
$f_8$	300000	2.86E-02	3.25E-01	2.42E-02	5.73E-02	<b>1.76E-02</b>	2.64E-02
$f_9$	100000	-8.38E+02	-8.38E+02	<b>-8.37E+02</b>	-8.36E+02	<b>-8.37E+02</b>	-8.36E+02
$f_{10}$	300000	3.89E+04	3.89E+04	3.88E+04	3.89E+04	<b>3.90E+04</b>	<b>3.90E+04</b>

individuals will be generated and inserted into the population at each generation. The PL-GA [17], which starts with a small population size and doubling the population size when the GA converges. The GPS-GA [50], in which two sub-populations are evolved in parallel. The dynNP-DE [13], in which the population size is periodically descending during the evolution, and the new population size is equal to half the previous population size. The SAMDE [45], in which the population diversity is considered to increase or decrease the population sizes. These algorithms are based on different rationales for population control and have been widely applied for various applications.

The parameter setting for the six algorithms are taken from the original papers. However, to make the comparison fair, a few parameters, which are common for these algorithms, are set as the same values. Specifically, the initial population size  $N$  of 20, crossover rate of 0.65 and bit mutation operator rate 0.015 are used. All algorithms to be compared adopt the tournament operator for selection. The other parameter settings for the six algorithms are shown in Table 1.

### 3.3 Results and Discussion

The comparison results of the six algorithms are shown in Table 2 in terms of solution quality. Each algorithm is run on each function 10 times. The best results over the ten trials are then averaged. From the results, it can be seen that the dynNP-DE and SAMDE algorithms outperform other four algorithms on  $f_1$ ,  $f_3$ ,  $f_5$ ,  $f_7$ ,  $f_9$  functions. Between the dynNP-DE and SAMDE, the dynNP-DE shows better performances on multi-modal functions, while SAMDE is good at searching the unimodal functions. From the function point of view, for the function  $f_3$ , dynNP-DE, SAMDE, PL-GA and GPS-GA can identify the global optimum, while the GAVaPS and APAGA fail to do so. This result could suggest that frequently removing individuals from the population will decrease the robustness of the algorithm. For the functions  $f_6$ , only PL-GA and GPS-GA is



**Fig. 1.** Comparing the convergence properties of six algorithms on different benchmark

able to find promising solutions. For the function  $f_8$ , which is the most difficult to solve among the ten functions, it can be noticed that all algorithms can be trapped in local optimum. In overall, dynNP-DE performs the best as it could deliver the best results on most of test functions. This is followed by the SAMDE, PL-GA and GPS-GA.

Figure 1 shows the convergence properties of the six strategies, in which the average of best fitness values are plotted versus the actual function evaluations of the six algorithms. It can be observed that GAVaPS and APAGA can converge much faster than the dynNP-DE, SAMDE, PL-GA and GPS-GA. For example, on the test function  $f_{10}$ , GAVaPS and APAGA are able to identify the optimal solution with around  $4 \times 10^3$  function evaluations. Specifically, on unimodal



functions, for instance,  $f_1$ ,  $f_2$ , PL-GA and GPS-GA perform the worst in terms of convergence and can take more than  $1 * 10^4$  function evaluations to converge than other four algorithms. Generally, PL-GA is the slowest method among the six methods, while the GPS-GA is the fastest strategy.

From both the quality and convergence property, it can be generally concluded that the GAVaPS and APAGA perform well on unimodal functions and its extended version APAGA has even better performance. The main fault of GAVaPS is excessive growth of individuals and results in an expensive evolution while APAGA slows down the increasing rate of population and improves its efficiency. The PL-GA and GPS-GA have good performance for multi-model functions especially in terms of convergence speed. While, the dynNP-DE and SAMDE algorithms can perform the best in most of the test functions by average.

## 4 Conclusions

In this paper, we first present an extensive review of population control methods in evolutionary algorithms. Then, we experimentally examine and analyze a few representative methods on a set of well-known benchmark functions. The results show that the dynNP-DE perform the best in most of the test functions by average. This mainly due to the dynNP-DE is able to maintain a proper balance of exploration and exploitation thus achieving a good performance.

**Acknowledgments.** This work was supported by the National Natural Science Foundation of China (Grant No. 61573316).

## References

1. De Jong, K.: Parameter setting in EAs: a 30 year perspective. In: Lobo, F.G., Lima, C.F., Michalewicz, Z. (eds.) *Parameter Setting in Evolutionary Algorithms*. SCI, vol. 54, pp. 1–18. Springer, Heidelberg (2007). [https://doi.org/10.1007/978-3-540-69432-8\\_1](https://doi.org/10.1007/978-3-540-69432-8_1)
2. Costa, J.C., Tavares, R., Rosa, A.: An experimental study on dynamic random variation of population size. In: *1999 IEEE International Conference on Systems, Man, and Cybernetics, IEEE SMC 1999 Conference Proceedings*, vol. 1, pp. 607–612. IEEE (1999)
3. Eiben, Á.E., Hinterding, R., Michalewicz, Z.: Parameter control in evolutionary algorithms. *IEEE Trans. Evol. Comput.* **3**(2), 124–141 (1999)
4. Eiben, A.E., Marchiori, E., Valkó, V.A.: Evolutionary algorithms with on-the-fly population size adjustment. In: Yao, X., et al. (eds.) *PPSN 2004*. LNCS, vol. 3242, pp. 41–50. Springer, Heidelberg (2004). [https://doi.org/10.1007/978-3-540-30217-9\\_5](https://doi.org/10.1007/978-3-540-30217-9_5)
5. Hu, T., Banzhaf, W.: The role of population size in rate of evolution in genetic programming. In: Vanneschi, L., Gustafson, S., Moraglio, A., De Falco, I., Ebner, M. (eds.) *EuroGP 2009*. LNCS, vol. 5481, pp. 85–96. Springer, Heidelberg (2009). [https://doi.org/10.1007/978-3-642-01181-8\\_8](https://doi.org/10.1007/978-3-642-01181-8_8)

6. Romero, G., Mora, A.M., Fernandes, C.: Studying the effect of population size in distributed evolutionary algorithms on heterogeneous clusters. *Appl. Soft. Comput.* **38**(C), 530–547 (2016)
7. Goldberg, D.E.: Sizing populations for serial and parallel genetic algorithms. In: *Proceedings of the 3rd International Conference on Genetic Algorithms*, pp. 70–79 (1989)
8. Schaffer, J.: *A study of control parameters affecting online performance of genetic algorithms for function optimization*, San Mateo, California (1989)
9. Smith, R.E., Smuda, E.: Adaptively resizing populations: algorithm, analysis, and first results. *Complex Syst.* **9**, 47–72 (1995)
10. Weise, T., Wu, Y., Chiong, R.J.: Global versus local search: the impact of population sizes on evolutionary algorithm performance. *J. Global. Optim.* **66**(3), 511–534 (2016)
11. Arabas, J., Michalewicz, Z., Mulawka, J.: GAVaPS—a genetic algorithm with varying population size. In: *Proceedings of the First IEEE Conference on Evolutionary Computation, IEEE World Congress on Computational Intelligence*, pp. 73–78. IEEE (1994)
12. Fernández, F., Tomassini, M., Vanneschi, L.: An empirical study of multipopulation genetic programming. *Genetic Program. Evol. Mach.* **4**(1), 21–51 (2003)
13. Brest, J., Maučec, M.S.: Population size reduction for the differential evolution algorithm. *Appl. Intell.* **29**(3), 228–247 (2008)
14. Ahrari, A., Shariat-Panahi, M.: An improved evolution strategy with adaptive population size. *Optimization* **64**(12), 2567–2586 (2015)
15. Karafotias, G., Hoogendoorn, M., Eiben, Á.E.: Parameter control in evolutionary algorithms: trends and challenges. *IEEE Trans. Evol. Comput.* **19**(2), 167–187 (2015)
16. Piotrowski, A.P.: Review of differential evolution population size. *Swarm Evol. Comput.* **32**, 1–24 (2017)
17. Holdener, E.A.: *The art of parameterless evolutionary algorithms*. Ph.D. thesis, Missouri University of Science and Technology (2008)
18. Goldberg, D.E., Deb, K., Clark, J.H.: Genetic algorithms, noise, and the sizing of populations. *Urbana* **51**, 61801 (1991)
19. Reeves, C.R.: Using genetic algorithms with small populations. In: *ICGA*, vol. 590, p. 92 (1993)
20. Goldberg, D.E., Sastry, K., Latoza, T.: On the supply of building blocks. In: *Proceedings of the 3rd Annual Conference on Genetic and Evolutionary Computation*, pp. 336–342. Morgan Kaufmann Publishers Inc. (2001)
21. De Jong, K.A.: *Analysis of the behavior of a class of genetic adaptive systems* (1975)
22. Harik, G., Cantú-Paz, E., Goldberg, D.E., Miller, B.L.: The Gambler’s ruin problem, genetic algorithms, and the sizing of populations. *Evol. Comput.* **7**(3), 231–253 (1999)
23. Fernandez, F., Vanneschi, L., Tomassini, M.: The effect of plagues in genetic programming: a study of variable-size populations. In: Ryan, C., Soule, T., Keijzer, M., Tsang, E., Poli, R., Costa, E. (eds.) *EuroGP 2003*. LNCS, vol. 2610, pp. 317–326. Springer, Heidelberg (2003). [https://doi.org/10.1007/3-540-36599-0\\_29](https://doi.org/10.1007/3-540-36599-0_29)
24. Fernandez, F., Tomassini, M., Vanneschi, L.: Saving computational effort in genetic programming by means of plagues. In: *The 2003 Congress on Evolutionary Computation, CEC 2003*, vol. 3, pp. 2042–2049. IEEE (2003)

25. de Vega, F.F., Cantú-Paz, E., López, J.I., Manzano, T.: Saving resources with plagues in genetic algorithms. In: Yao, X., et al. (eds.) PPSN 2004. LNCS, vol. 3242, pp. 272–281. Springer, Heidelberg (2004). [https://doi.org/10.1007/978-3-540-30217-9\\_28](https://doi.org/10.1007/978-3-540-30217-9_28)
26. Brest, J., Zamuda, A., Fister, I., Maučec, M.S.: Large scale global optimization using self-adaptive differential evolution algorithm. In: 2010 IEEE Congress on Evolutionary Computation (CEC), pp. 1–8. IEEE (2010)
27. Brest, J., Maučec, M.S.: Self-adaptive differential evolution algorithm using population size reduction and three strategies. *Soft Comput.* **15**(11), 2157–2174 (2011)
28. Zamuda, A., Brest, J., Mezura-Montes, E.: Structured population size reduction differential evolution with multiple mutation strategies on CEC 2013 real parameter optimization. In: 2013 IEEE Congress on Evolutionary Computation (CEC), pp. 1925–1931. IEEE (2013)
29. Yang, M., Cai, Z., Guan, J., Gong, W.: Differential evolution with improved population reduction. In: Proceedings of the 13th Annual Conference Companion on Genetic and Evolutionary Computation, pp. 143–144. ACM (2011)
30. Ali, M.Z., Awad, N.H., Suganthan, P.N., Reynolds, R.G.: An adaptive multipopulation differential evolution with dynamic population reduction. *IEEE Trans. Cybern.* **47**(9), 2768–2779 (2017)
31. Iacca, G., Mallipeddi, R., Mininno, E., Neri, F.: Super-fit and population size reduction in compact differential evolution. In: Memetic Computing, pp. 1–8 (2011)
32. Zamuda, A., Brest, J.: Population reduction differential evolution with multiple mutation strategies in real world industry challenges. In: Rutkowski, L., Korytkowski, M., Scherer, R., Tadeusiewicz, R., Zadeh, L.A., Zurada, J.M. (eds.) EC/SIDE -2012. LNCS, vol. 7269, pp. 154–161. Springer, Heidelberg (2012). [https://doi.org/10.1007/978-3-642-29353-5\\_18](https://doi.org/10.1007/978-3-642-29353-5_18)
33. Brest, J., Zamuda, A., Fister, I., Maučec, M.S., et al.: Self-adaptive differential evolution algorithm with a small and varying population size. In: 2012 IEEE Congress on Evolutionary Computation (CEC), pp. 1–8. IEEE (2012)
34. Neri, F., Tirronen, V.: Recent advances in differential evolution: a survey and experimental analysis. *Artif. Intell. Rev.* **33**(1–2), 61–106 (2010)
35. Koumousis, V.K., Katsaras, C.P.: A saw-tooth genetic algorithm combining the effects of variable population size and reinitialization to enhance performance. *IEEE Trans. Evol. Comput.* **10**(1), 19–28 (2006)
36. Koumousis, V., Dimou, C.: The effect of oscillating population size on the performance of genetic algorithms. In: Proceedings of the 4th GRACM Congress on Computational Mechanics (2002)
37. Tanabe, R., Fukunaga, A.S.: Improving the search performance of shade using linear population size reduction. In: 2014 IEEE Congress on Evolutionary Computation (CEC), pp. 1658–1665. IEEE (2014)
38. Yuan, X., Zhang, B., Wang, P., Liang, J., Yuan, Y., Huang, Y., Lei, X.: Multi-objective optimal power flow based on improved strength pareto evolutionary algorithm. *Energy* **122**, 70–82 (2017)
39. Polakova, R., Tvrdik, J., Bujok, P.: Evaluating the performance of l-shade with competing strategies on CEC 2014 single parameter-operator test suite. In: IEEE Congress on Evolutionary Computation, pp. 1181–1187 (2016)
40. Viktorin, A., Pluhacek, M., Senkerik, R.: Network based linear population size reduction in shade. In: International Conference on Intelligent Networking and Collaborative Systems, pp. 86–93 (2016)

41. Guo, S.M., Tsai, S.H., Yang, C.C., Hsu, P.H.: A self-optimization approach for l-shade incorporated with eigenvector-based crossover and successful-parent-selecting framework on CEC 2015 benchmark set. In: *Evolutionary Computation*, pp. 1003–1010 (2015)
42. Zheng, Y.J., Zhang, B.: A simplified water wave optimization algorithm. In: *Evolutionary Computation*, pp. 807–813 (2015)
43. Montiel, O., Castillo, O., Melin, P., Sepúlveda, R.: Intelligent control of dynamic population size for evolutionary algorithms. In: *IC-AI*, pp. 551–557 (2006)
44. Wang, H., Rahnamayan, S., Wu, Z.: Adaptive differential evolution with variable population size for solving high-dimensional problems. In: *2011 IEEE Congress on Evolutionary Computation (CEC)*, pp. 2626–2632. IEEE (2011)
45. Wang, X., Zhao, S., Jin, Y., Zhang, L.: Differential evolution algorithm based on self-adaptive adjustment mechanism. In: *2013 25th Chinese Control and Decision Conference (CCDC)*, pp. 577–581. IEEE (2013)
46. Elsayed, S.M., Sarker, R.A.: Differential evolution with automatic population injection scheme for constrained problems. In: *2013 IEEE Symposium on Differential Evolution (SDE)*, pp. 112–118. IEEE (2013)
47. Zhang, C., Chen, J., Xin, B., Cai, T., Chen, C.: Differential evolution with adaptive population size combining lifetime and extinction mechanisms. In: *2011 8th Asian Control Conference (ASCC)*, pp. 1221–1226. IEEE (2011)
48. Zhao, S., Wang, X., Chen, L., Zhu, W.: A novel self-adaptive differential evolution algorithm with population size adjustment scheme. *Arab. J. Sci. Eng.* **39**(8), 6149–6174 (2014)
49. Schlierkamp-Voosen, D., Muhlenbein, H.: Adaptation of population sizes by competing subpopulations. In: *Proceedings of IEEE International Conference on Evolutionary Computation*, pp. 330–335. IEEE (1996)
50. Smorodkina, E., Tauritz, D.: Greedy population sizing for evolutionary algorithms. In: *IEEE Congress on Evolutionary Computation, CEC 2007*, pp. 2181–2187. IEEE (2007)
51. Hinterding, R., Michalewicz, Z., Peachey, T.C.: Self-adaptive genetic algorithm for numeric functions. In: Voigt, H.-M., Ebeling, W., Rechenberg, I., Schwefel, H.-P. (eds.) *PPSN 1996. LNCS*, vol. 1141, pp. 420–429. Springer, Heidelberg (1996). [https://doi.org/10.1007/3-540-61723-X\\_1006](https://doi.org/10.1007/3-540-61723-X_1006)
52. Harik, G.R., Lobo, F.G.: A parameter-less genetic algorithm. In: *Proceedings of the 1st Annual Conference on Genetic and Evolutionary Computation*, vol. 1, pp. 258–265. Morgan Kaufmann Publishers Inc. (1999)
53. Zhan, Z.H., Zhang, J.: Co-evolutionary differential evolution with dynamic population size and adaptive migration strategy. In: *Proceedings of the 13th Annual Conference Companion on Genetic and Evolutionary Computation*, pp. 211–212. ACM (2011)
54. Fernández, C., Rosa, A.C., Rosa, A.C.: NiGaVaPS - a outbreeding in genetic algorithms. In: *ACM Symposium on Applied Computing*, pp. 477–482 (2000)
55. Fernández, C., Rosa, A.: Self-regulated population size in evolutionary algorithms. In: *International Conference on Parallel Problem Solving from Nature*, pp. 920–929 (2006)
56. Fernández, C., Rosa, A., Pais, A.R., Norte, T.: A study on non-random mating and varying population size in genetic algorithms using a royal road function. In: *Proceedings of the 2001 Congress on Evolutionary Computation*, vol. 1, pp. 60–66 (2001)
57. Lee, H.S., Lee, J.H., Kim, E.T.: Optimal classifier ensemble design for vehicle detection using GAVaPS. *J. Inst. Control Robot. Syst.* **16**(1), 96–100 (2010)

58. Bäck, T., Eiben, A.E., Van Der Vaart, N.A.L.: An empirical study on gas “Without Parameters”. In: International Conference on Parallel Problem Solving from Nature, pp. 315–324 (2000)
59. Iorio, A., Li, X.: Parameter control within a co-operative co-evolutionary genetic algorithm. In: Guervós, J.J.M., Adamidis, P., Beyer, H.-G., Schwefel, H.-P., Fernández-Villacañas, J.-L. (eds.) PPSN 2002. LNCS, vol. 2439, pp. 247–256. Springer, Heidelberg (2002). [https://doi.org/10.1007/3-540-45712-7\\_24](https://doi.org/10.1007/3-540-45712-7_24)
60. Vellev, S.: An adaptive genetic algorithm with dynamic population size for optimizing join queries. *Adv. Res. Artif. Int.* **82**, 82–88 (2008)
61. Cook, J.E., Tauritz, D.R.: An exploration into dynamic population sizing. In: Conference on Genetic and Evolutionary Computation, pp. 807–814 (2010)
62. Krishnakumar, K.: Micro-genetic algorithms for stationary and non-stationary function optimization. In: 1989 Symposium on Visual Communications, Image Processing, and Intelligent Robotics Systems, pp. 289–296. International Society for Optics and Photonics (1990)
63. Coello, C.A., Pulido, G.T.: Multiobjective optimization using a micro-genetic algorithm. In: Proceedings of the 3rd Annual Conference on Genetic and Evolutionary Computation, pp. 274–282. Morgan Kaufmann Publishers Inc. (2001)
64. Xu, Y., Liu, G.: Detection of flaws in composites from scattered elastic-wave field using an improved  $\mu$ GA and a local optimizer. *Comput. Methods Appl. Mech.* **191**(36), 3929–3946 (2002)
65. Ryoo, J., Hajela, P.: Handling variable string lengths in GA-based structural topology optimization. *Struct. Multidiscip. Optim.* **26**(5), 318–325 (2004)
66. Khor, E.F., Tan, K.C., Wang, M.L., Lee, T.H.: Evolutionary algorithm with dynamic population size for multi-objective optimization. In: Conference of the IEEE Industrial Electronics Society, IECON 2000, vol. 4, pp. 2768–2773 (2000)
67. Tan, K.C., Lee, T.H., Khor, E.F.: Evolutionary algorithms with dynamic population size and local exploration for multiobjective optimization. *IEEE Trans. Evol. Comput.* **5**(6), 565–588 (2001)
68. Liang, Y., Leung, K.S.: Genetic algorithm with adaptive elitist-population strategies for multimodal function optimization. *Appl. Soft. Comput.* **11**(2), 2017–2034 (2011)
69. Yang, M., Cai, Z., Guan, J., Guan, J.: An improved adaptive differential evolution algorithm with population adaptation. In: Conference on Genetic and Evolutionary Computation, pp. 145–152 (2013)
70. Ding, M., Chen, H., Lin, N., Jing, S., Liu, F., Liang, X., Liu, W.: Dynamic population artificial bee colony algorithm for multi-objective optimal power flow. *Saudi. J. Biol. Sci.* **24**(3), 703–710 (2017)
71. Auger, A., Hansen, N.: A restart CMA evolution strategy with increasing population size. In: The 2005 IEEE Congress on Evolutionary Computation, vol. 2, pp. 1769–1776 (2005)
72. Shi, E.C., Leung, F.H.F., Law, B.N.F.: Differential evolution with adaptive population size. In: International Conference on Digital Signal Processing, pp. 876–881 (2014)
73. Smith, R.E., Smuda, E.: Adaptively resizing populations: algorithm, analysis, and first results. *Complex Syst.* (1993)
74. Tirronen, V., Neri, F.: Differential evolution with fitness diversity self-adaptation. In: Chiong, R. (ed.) *Nature-Inspired Algorithms for Optimisation*. SCI, vol. 193, pp. 199–234. Springer, Heidelberg (2009). [https://doi.org/10.1007/978-3-642-00267-0\\_7](https://doi.org/10.1007/978-3-642-00267-0_7)

75. Wagner, N., Michalewicz, Z.: Genetic Programming with Efficient Population Control for Financial Time Series Prediction (2001)
76. Wagner, N., Michalewicz, Z.: Parameter Adaptation for GP Forecasting Applications (2007)
77. Wagner, N., Michalewicz, Z., Khouja, M., McGregor, R.R.: Time series forecasting for dynamic environments: the DyFor genetic program model. *IEEE Trans. Evol. Comput.* **11**(4), 433–452 (2007)
78. Teo, J.: Exploring dynamic self-adaptive populations in differential evolution. *Soft Comput.* **10**(8), 673–686 (2006)
79. Eiben, A.E., Schut, M.C., Wilde, A.R.D.: Is self-adaptation of selection pressure and population size possible?: a case study. In: *International Conference on Parallel Problem Solving from Nature*, pp. 900–909 (2006)