Self-regulated Population Size in Evolutionary Algorithms

Carlos Fernandes and Agostinho Rosa

LaSEEB-ISR-IST, Technical Univ. of Lisbon (IST)
{cfernandes, acrosa}@laseeb.org

Abstract. In this paper we analyze a new method for an adaptive variation of Evolutionary Algorithms (EAs) population size: the Self-Regulated Population size EA (SRP-EA). An empirical evaluation of the method is provided by comparing the new proposal with the CHC algorithm and other well known EAs with varying population. A fitness landscape generator was chosen to test and compare the algorithms: the Spear's multimodal function generator. The performance of the algorithms was measured in terms of success rate, quality of the solutions and evaluations needed to attain them over a wide range of problem instances. We will show that SRP-EA performs well on these tests and appears to overcome some recurrent drawbacks of traditional EAs which lead them to local optima premature convergence. Also, unlike other methods, SRP-EA seems to self-regulate its population size according to the state of the search.

1 Introduction

Although varying the population size of EAs during the run seems to be a rather natural and rewarding approach when implementing this type of algorithms, that particular parameter has not been widely studied as far as variation is concerned. Unlike other operators' parameters - like mutation rate for instance -, population size, with few exceptions, remained away from major efforts in finding parameter control methods. GAVaPS [1] (see next section) introduced some interesting concepts that gave rise to an optimist expectation about the performance of EAs with varying population size. But some aspects of the algorithm, namely population size self-regulation, could not be reproduced in other tests [5] [7]. The authors of GAVaPS suggested that the algorithm could adapt its population size according to the state of the search, balancing exploration and exploitation by increasing the population size on a first stage and then reducing the number of chromosomes on later stages. But, in further studies, a different behavior was observed. In [7], the authors noted that GAVaPS, when applied to a Royal Road problem, either grew its population size up to several thousand individuals or decreased it until extinction. These features were observed with different parameter values, that is, no combination of parameters was found to improve the stability of the population. In [5], GAVaPS also evolved into large populations when applied to Spears' multimodal problems, giving rise to a poor performance when compared to other EAs. Despite these disappointing general results of GAVaPS, some other studies indicate that varying the population size of EAs may

T.P. Runarsson et al. (Eds.): PPSN IX, LNCS 4193, pp. 920 – 929, 2006.

increase their performance on some problems (see [5], for instance). Also, although GAVaPS did not attain the expected impact, some of its concepts are very interesting and worth further exploring. With these issues in mind, we intended to develop a decentralized variation process that may lead to a self-regulated behavior of the population size, at least within a small subset of the parameters values, thus exploring more conveniently the search space and making use of the resources in a more rational way. The proposed process relies on the genetic diversity of the population during the run. Our results indicate that this may be a promising path to follow when developing EAs with varying population size.

2 Previous Research

According to Eiben and al. [4] parameter control mechanisms of EAs may be divided into three categories:

- Deterministic methods: parameter values are changed by some deterministic rule.
- Adaptive methods: values vary during the EA run depending on its behavior.
- Self-Adaptive methods: the values are codified within the chromosome and evolve together with the problem solutions.

In this paper we focus our attention on the variation of the population size of EAs during the run. Some techniques described below fall into the adaptive methods categories, while others, like RVPS [3] and PRoFIGA [5] are deterministic methods. Our proposal may also be classified as an adaptive method. However, the variation process in SRP-EA may also be viewed as a result of a varying crossover rate, which is indirectly controlled by the genetic diversity of the population.

The Genetic Algorithm with Varying Population Size (GAVaPS) [1] does not have an explicit selection mechanism. As in natural systems, population size is defined by the birth and death of individuals occurring at each iteration. A parameter called *lifetime* is introduced. It defines the number of generations in which each individual is allowed to remain *alive*, that is, a part of the population and the evolutionary process. After its creation, the chromosome is assigned to a specific lifetime, according to its fitness. Three lifetime calculation methods are proposed. The algorithm proceeds in a generational manner, at each time step increasing each individual's *age*. When an individual's age exceeds its lifetime, the chromosome is removed from the population. Since fittest individuals remain in the population for more generations, thus having a higher probability to be engaged in a reproduction process and generate offspring, GAVaPS' chromosomes have equal probability to be selected to reproduce, independently of their fitness value. This concept of lifetime/age provides the algorithm with the necessary selection pressure, which reduces the need for selection strategies: GAVaPS randomly pairs the chromosomes for crossover operations. The intensity of the pressure is controlled by two parameters, *minLT* and *maxLT*, that define, respectively, the minimum and maximum lifetime allowed for each chromosome. Higher difference between the two values leads to a more selective algorithm. However, this process may have a serious drawback since increasing the *maxLT* parameter will result in larger populations and, as stated above, an increasingly high population size is a characteristic of GAVaPS. The algorithm also introduces another parameter: reproduction rate (ρ). Its value defines the number of new chromosomes created in each generation t, depending on the size of the current population.

The Adaptive Population size Genetic Algorithm (APGA) [2] is very similar to GAVaPS. The only difference resides in reproduction rate, which in APGA has a fixed value of two individuals. This technique follows the reproduction strategy of the Steady-State GA and prevents the population from growing out of control has it often happens with GAVaPS. On the other hand, such a low reproduction rate results in populations with few individuals unless a high value for *maxLT* is used. But, even in the last case, the population size is very stable and apparently does not react to the evolution process and different search stages (see section 4). However, the algorithm performs well on some problems and clearly outperformed GAVaPS when applied to the Spears' multimodal problems [5]. Besides a low reproduction rate, APGA also uses an elitist strategy by keeping unchanged the age of the best individual.

The Population Resizing on Fitness Improvement GA (PRoFIGA) was proposed in [5] by Eiben, Marchiori and Valkó. The variation process of PRoFIGA is based on the improvement of the best fitness in the population. The process intends to balance exploration and exploitation by growing the population in earlier and exploratory stages and gradually decrease it in later stages of the search. When the population gets trapped in local optima, the process is supposed to generate another growing phase of the population, thus increasing diversity and escaping the local optima. The authors present a heuristic for size variation during the run that increases or decreases the population size according to whether or not the best fitness of the population has been improved and, if the later case is observed, for how long it has remained unchanged.

In the Random Variation of Population Size GA (RVPS) [3] the population size is randomly changed during the run. The authors concluded that in some cases the performance of RVPS is equivalent to the standard GA. So, when there are no hints about the optimal population size for some problem, it may be appropriate to randomly set and vary the population size of the GA.

Like PRoFIGA and RVPS, the Saw-Tooth Genetic Algorithm [8] is an example of a deterministic method used in the variation of the population size. In this algorithm the population size varies according to a predefined function with a saw-tooth shape. The authors concluded that the Saw-Tooth GA performed well on some particular test functions. However, besides a variable population size, the Saw-Tooth GA also uses a reinitialization mechanism to introduce genetic diversity in the population.

3 Our Proposal

The SRP-EA combines features of CHC [6] and GAVaPS and introduces a dynamic reproduction rate which is indirectly controlled by the genetic diversity of the population. CHC, which stands for *Cross generational elitist selection, Heterogeneous Recombination and Cataclysmic Mutation*, is a variation of the standard GA. It uses no mutation in the classical sense of the concept, but instead it goes through a process of macro-mutation when the best fitness of the population doesn't change after a certain number of generations. The genetic diversity is assured by a highly disruptive crossover operator (HUX) and a reproduction restriction which assures that selected pairs of chromosomes won't generate offspring unless their Hamming Distance is above a certain threshold. Then, each generation, *p/2* pairs of chromosomes are randomly selected from the population with size *p*. All pairs are submitted to the reproduction process. First, their Hamming Distance is computed. If the value is found to be above the threshold then the chromosomes generate two children with the HUX operator. When the process is concluded, the newly generated population of *p'* offspring replaces the worst *p'* chromosomes in the main population, therefore maintaining the size of the population. The threshold is usually set in the beginning of the runs to ¼ of the chromosome length, and decremented when no offspring is generated. When the algorithm gets stuck in local optima, a cataclysmic mutation is applied by replacing the entire population, except the best chromosome, with mutated copies of that individual. Usually, the mutation rate at this point is set to 0.35.

SRP-EA adapts the Hamming Distance restriction of CHC. Remember that the process leads to a changing reproduction rate meaning that in each generation the number of offspring is not necessarily the same. The difference is that in SRP-EA the new chromosomes do not replace the parents' population. Instead, offspring are added to the population, therefore increasing its size, while other individuals are removed via an age/lifetime process similar to the one found in GAVaPS and APGA. The process conduces to a variation in the size of the population and works as follows (SRP-EA pseudo-code is given in figure 1). First SRP-EA assigns a lifetime to each chromosome created (the three lifetime computation strategies of GAVaPS were adopted). Then, in each generation, the age (initially set to 0) of each chromosome is incremented. The chance of survival decreases with the age of the chromosome - the survival probability is set to (*lifetime-age)/maxLT* and when the age of a chromosome reaches its lifetime, the probability of survival reaches zero. There is a difference between the SRP-EA and GAVaPS, since in GAVaPS the individuals remain in the population during its lifetime, while in SRP-EA an individual may die before the age reaches its limit.

The *create new individuals* procedure increases the population size by generating offspring with a restriction based on the Hamming Distance between the parents. When two parents are selected and their Hamming Distance is above the threshold, the children are generated. If the Hamming Distance is below or equal to the threshold then the parents do not cross and the attempt is classified as *failure*. After the *p/2* mating attempts are concluded (where *p* is the size of the population), all newborn children are introduced in the population and the threshold is set to a new value according to the heuristic described in figure 1 (the process repeats until at least one mating attempt succeeds). Also, in the *kill older individuals* procedure, the threshold is increased by a predefined amount (*Inc*) if the number of newborn is higher than number of individuals that died in the present generation. This strategy, along with proper set of the *Dec* and *Inc* parameters, creates a self-regulated population, which increases in the beginning of the search, decreases with convergence, and sometimes reacts to local optima convergence. Notice that this emergent behavior is similar to the one that PRoFIGA intends to simulate by means of a set of deterministic rules. However, the correct way to set the parameter values necessary to attain the desired population behavior and consequent algorithm performance is still unclear, although the tests described in the next section have brought some light into the subject.

```
Procedure SRP-EA
    initialize and evaluate population          /*compute fitness and lifetime */
    while (not termination condition) {
        increase the age of each individual by 1
        create new individuals
        kill older individuals
        evaluate new individuals
        set lifetime of new individuals }   /*Using any kind of strategy*/

Procedure create new individuals
    do {
        mating_events = population_size/2
        for (i = 1 to mating_events)  do{
            select two individuals        /*Any method may be used here*/
            if (hamming distance > threshold)  crossover and mutate    /*Successful mating*/
        }
        if (failed matings> successful matings)   threshold = threshold-Dec
        else                                      threshold = threshold+Inc
    } while (successful matings = 0)

Procedure kill older individuals
        for all individuals except the best do {
            survival probability = (lifetime-age)/maxLT
            if (random [0, 1] > survival probability)   kill individual   }
        if (newborn > dead)    threshold = threshold+Inc
```

Fig. 1. SRP-EA pseudo-code

4 Test Bed Set and Results

To test the efficiency of the proposed method, a Genetic Algorithm with the reproduction procedure described above was tested on several Spears' multimodal problems [9]. In [5], the authors chose that function generator to study different EAs with varying population size.

For that reason, the Spears' problem may be a good benchmark to test the SRP-EA. Also, the generator creates problems with different sizes and degrees of multimodality making it a good tool to test some of the algorithms' characteristics. In the experiences described below we tried to follow the procedures described in [5].

Table 1. Algorithms' setup

Chromosome length L	100
Initial population size N	25, 50, 100, 200
Mutation rate p_m (in APGA and SRP-EA)	0.0025, 0.005, 0.01, 0.02
Crossover rate p_c (in APGA)	0.9
Selection	Random and 4-size tournament
Maximum number of evaluations in each run	10000
Initial threshold (in CHC and SRP-EA)	L/4
Inc, *Dec* (in SRP-EA)	Inc = Dec = 3
minLT (in APGA and SRP-EA)	1
maxLT (in APGA and SRP-EA)	7, 11, 20
Lifetime calculation (in APGA and SRP-EA)	Bilinear (see [1] for details)

The SRP-EA was tested and compared with the CHC and the APGA. In [5] the authors compared the APGA with a Simple Genetic Algorithm (SGA) and other algorithms with varying population size, like GAVaPS and PRoFIGA, and concluded that APGA outperformed those methods through a wide range of Spears' problem instances. For that reason we simplified our analysis and eliminated the results attained with other EAs from the figures below. Furthermore, we are mainly interested in adaptive control methods of the population size, so deterministic methods like the ones used in RVPS and PRoFIGA somehow fall off this paper's subject.

We ran the algorithms on 10 different types of landscapes, with the number of peaks (NP) ranging through 1, 2, 5, 10, 25, 50, 100, 250, 500 and 1000. The distribution of the peaks is linear and the lowest peak height was set to 0.5. Global optimum fitness is 1 in all instances of the problem. All configurations of the EAs created and evaluated no more than 10000 chromosomes in each run. The results were averaged over 100 runs. The initial population size (fixed in CHC) ranged through 25, 50, 100 and 200. Four different mutation rates were tested. The crossover rate of APGA was set to 0.9, following the test setups in [5], and a two point crossover operator was used, except in CHC where we used the HUX operator associated with the method. The value of *minLT* was set to 1 in APGA and SRP-EA, while *maxLT* varied through 7, 11 and 20. All algorithms use elitism. Table 1 resumes the setup.

Before we proceed to a more accurate study some general remarks must be stated.

- The APGA results shown in [5] were properly reproduced in our tests. Also, the configuration used by the authors revealed to be appropriate and, in general, other configurations didn't increase significantly the performance.
- While the tests with APGA and CHC revealed no clear improvement when using tournament instead of random selection, SRP-EA seems to perform better with a tournament selection strategy.
- As expected, CHC performed better with small populations (the algorithm is known to be more able to deal with problems that require small populations).
- Neither APGA nor SRP-EA had significant changes in the performance over the range of *maxLT* values.
- The values of *Inc* and *Dec* parameters were not achieved by means of an exhaustive search and optimization. However, a general inspection revealed that values between 1% and 10% of the chromosome length may lead to good results. Also, results indicate that setting *Inc* = *Dec* appears to be an adequate strategy.

The performance of the algorithms was analyzed under three criteria: the success rate of the algorithm (SR%), that is, the percentage of runs in which the global optimum is achieved; the average number of evaluations (AE) necessary to reach global optimum (considering successful runs); and the average of the best chromosome's fitness (AF) found in each run. Since one of the hypotheses about SRP-EA is its ability to balance exploration and exploitation by adapting the size of the population to the state of the search, therefore increasing the probability to reach optimum, we will focus our attention on the SR% criteria.

Figure 2 illustrates some of the above observations. The graphics depict the success rates achieved by some configurations of CHC and APGA compared with two configurations of SRP-EA which differ in the mutation rate. Success rates of the

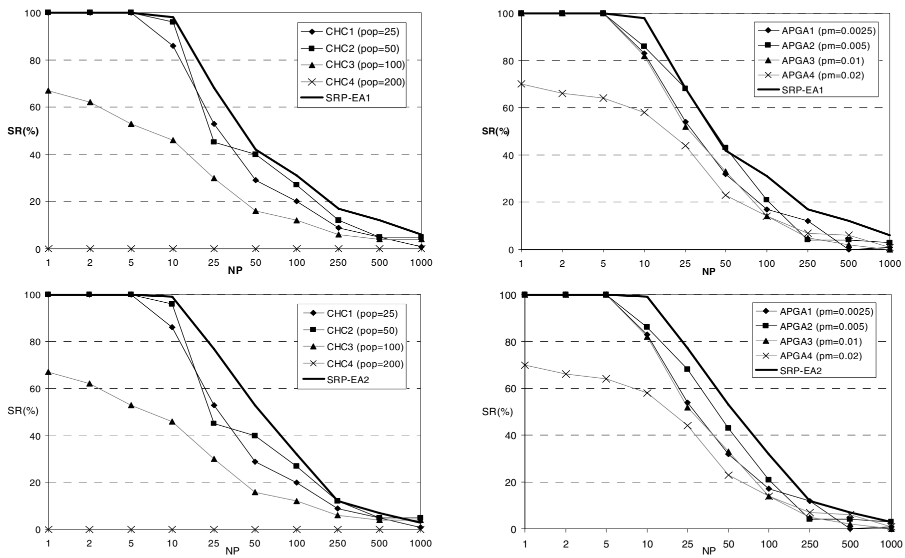


Fig. 2. CHC, APGA and SRP-EA success rates. SRP-EA parameters: N = 100, p_m= 0.005 (SRP-EA1), p_m = 0.0025 (SRP-EA2), $maxLT$ = 11, size 4 tournament; APGA: N = 100, p_c = 0.9 and $maxLT$ = 11.

algorithms are shown over the problem dimension range (number of peaks – NP). The graphics suggest that SRP-EA is more able to reach global optimum than APGA and CHC. Also, p_m = 0.0025 seems to favor SRP-EA performance in Spears' landscapes with higher number of peaks, while p_m = 0.005 works well on medium range problem dimension. Notice also, that CHC with a population of 200 individuals clearly fails in finding the optimal solutions and the same happens for APGA with p_m = 0.02.

When comparing the algorithms in terms of the best chromosome's fitness (AF), the results show that SRP-EA also attains, in general, higher values (see figure 3). However, the performance of SRP-EA pays a price in terms of number of evaluations to reach optima (AE). In figure 3 it is clear that SRP-EA performance comes with an increase in the number of evaluations. These results are not surprising since the population size variation process inherent to SRP-EA conduces to a large exploratory stage in the beginning of the search which increases the probability to reach global optimum but creates a large amount of new individuals, with obvious effects in the number of evaluations necessary to reach that optimum.

Choosing, for each NP, the best results of the algorithms over the complete space of parameter values of table 1 we obtain the curves represented in figure 4a. These results clearly illustrate the SRP-EA potential and its ability to find the global optima of Spears' landscapes.

One last test was conducted to examine the real influence of population variation in SRP-EA. As stated above, the population size variation of SRP-EA relies on a reproduction restriction that in nature is called assortative mating and tends to preserve genetic diversity. In some problems, that may be sufficient to increase convergence rate to global optimum. To try to quantify and distinguish the effects of genetic

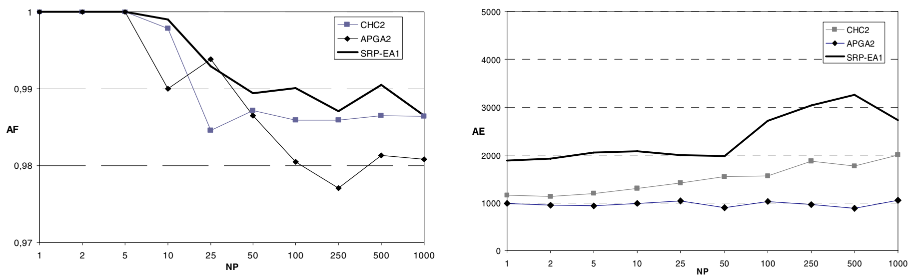
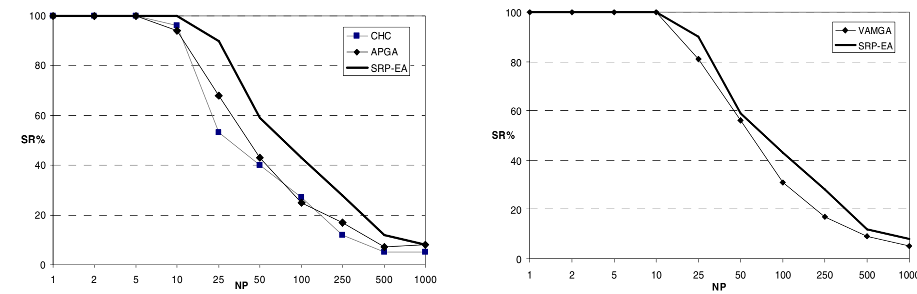


Fig. 3. Fitness of the best individual found and evaluations needed to reach the optima (results averaged over 100 runs) in CHC2, APGA2 and SRP-EA1

diversity maintenance and population size variation in SRP-EA, we created the Varying Assortative Mating EA (VAMEA), in which the procedure *kill older individuals* is replaced by a delete worst generational replacement as in CHC: like SRP-EA, *p'* individuals are created from *p/2* mating attempts (where *p* is the size of the population); then, the *p'* worst elements of the population are replaced by the offspring. This way, we remove the influence of the variation of population size and isolate the effects of the assortative mating found in the SRP-EA reproduction process. VAMEA was tested through the parameters' values range of table 1. Results are shown in figure 4b, where the curves represent the best results found for each NP (covering the complete set of parameter values shown in table 1). The differences found in the curves shape illustrate the role of the population variation mechanism. Although the assortative mating improves the success rates of the other genetic algorithms (as we can see by comparing the VAMEA curve in figure 4b with CHC and APGA curves in figure 4a), those rates experience even further improvement when the population variation process is introduced.

Although we tested SRP-EA with random selection of parents, following GAVaPS and APGA method, best results were achieved with tournament selection. APGA, on the other hand, didn't improve its results when changing the selection method. This outcome is not surprising for two reasons: 1) the way the chromosomes are eliminated from the population is different in SRP-EA, so the same *maxLT* value in SRP-EA and APGA conduces to a lower selection pressure in the first algorithm; 2) to amplify selection pressure in the algorithms, one must raise *maxLT* value; however, in SRP-EA, the increase in *maxLT* may lead to an excessive population growth and the consequent effort in terms of function evaluations (the population of APGA, with its "Steady-State like" reproduction, is almost immune to demographic explosion, even with large values of *maxLT*).

Before we conclude this section, a brief analysis of the population growth of the algorithms is required. Due to its fixed and low reproduction rate, the variation in the population size of APGA is very predictable and consists of small oscillations around an average value. Besides that, the population size seems to evolve without any feedback from the state of the search. Every APGA run over every instance of the problem showed the same behavior. The population size of SRP-EA evolves in a quite diverse manner. As we can see in figure 5, which represents the population growth and the evolution of the best fitness in two independent successful runs of the



a) b)

Fig. 4. Best success rates thorough the complete parameter space

algorithm on a NP=100 landscape, the population size clearly oscillates, sometimes even in severe way. There is a consistent demographic explosion in the beginning of the search which is quickly appeased. Then, the population stagnates in lower values but experience from time to time sudden increases in its size. Inspecting closely the curves below it can be seen that the sudden demographic growth is usually associated with stabilized or slowly growing phases of the best fitness value.

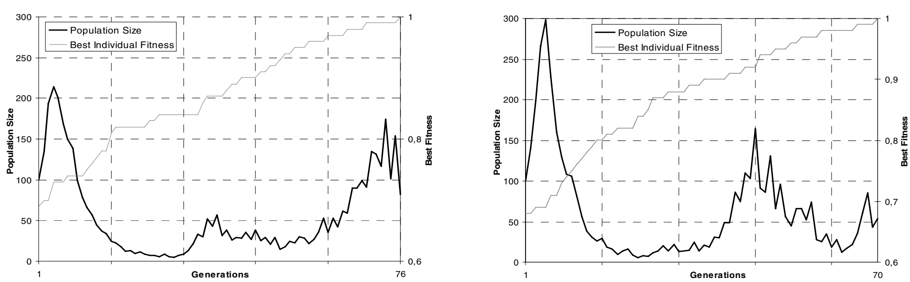


Fig. 5. Population growth and best fitness of SRP-EA in two independent runs. N = 100, p_m = 0.005 and *maxLT* =11. NP = 100.

5 Conclusions and Future Work

The results illustrated SRP-EA superior ability to find the global optima of Spears' landscapes when compared to CHC and APGA. That ability comes not only from the reproduction restriction based on the Hamming Distance between parents (which contributes with genetic diversity maintenance) but also from the population size variation itself. The dynamics of the population size seem to reflect the state of the search and the evolution of the quality of the solutions, in opposition to a more stable growth curve observed in APGA runs.

An in-depth analysis of the new parameters is needed in order to establish some rules that might reduce the complexity of the algorithm and also optimize its performance. Other distance criteria must also be inspected in order to reflect more properly the distribution of the population in the search space, avoiding overcrowded areas which do not contribute to maintain the genetic diversity, and redirecting the search to unexplored areas in an adaptive and non centralized manner. Finally, the application of SRP-EA to dynamic problems with on-line moving optima may be a proper field to evaluate the algorithm' potentialities and test its adaptive characteristics. Some preliminary tests already indicated that SRP-EA may be a useful tool to deal with dynamic problems.

Acknowledgement. The first author wishes to thank FCT, *Ministério da Ciência e Tecnologia*, his Research Fellowship SFRH/BD/18868/2004.

References

1. Arabas, J., Michalewicz, Z., Mulawka, J.: GAVaPS – A Genetic Algorithm with Varying Population Size. In: Proceedings of Evolutionary Computation Conference, IEEE Press, 1994.
2. Bäck, T., Eiben, A.E., van der Sart, N.A.L.: An empirical Study on Gas "without parameters". In Proceedings of the 6th Conference on Parallel Problem Solving from Nature, LNCS, pages 315-324, Springer, Berlin, 2000.
3. Costa, J., Tavares, R., Rosa, A.C.: An Experimental Study on Dynamic Random Variation of Population Size. In Proceedings of IEEE Systems, Man and Cybernetics Conference, Volume, pages 607-612, Tokyo, 1999. IEEE Press.
4. Eiben, A.E., Hinterding, R., Michalewicz, Z.: Parameter Control in Evolutionary Algorithms. IEEE Transaction on Evolutionary Computation, 3(2): 124-141, 1999.
5. Eiben, A.E., Marchiori E., Valkó, V.A.: Evolutionary Algorithms with On-the-Fly Population Size Adjustment. 8th Conference on Parallel Problem Solving from Nature, LNCS, pages 315-324, Springer, Birmingham, 2004.
6. Eschelman, L.J. The CHC Algorithm: How to Have Safe Search When Engaging in Non-traditional Genetic Recombination. In Proceedings of Foundations of Genetic Algorithms-1, pages 70-79, 1990.
7. Fernandes, C., Rosa, A.C.: A Study on Non-random Mating and Varying Population Size in Genetic Algorithms Using Royal Road Functions. In Proceedings of the IEEE Conference on Evolutionary Computation, pages 60-66, 2001.
8. Koumosis, V.K., Katsaras, C.P.: A Saw-Tooth Genetic Algorithm Combining the Effects of Variable Population Size and Reinitialization to Enhance Performance. In IEEE Transactions on Evolutionary Computation, 10(1), pages 19-28, 2006.
9. Spears, W.M.: Evolutionary Algorithms: the role of mutation and recombination. Springer, 2000.