Genetic Algorithms using Populations based on Multisets

António Manso[1], Luís Correia[1]

[1] LabMAg - Laboratório de Modelação de Agentes
Faculdade de Ciências da Universidade de Lisboa
Edifício C6, Piso 3, Campo Grande,
1749-016 Lisboa - Portugal
manso@ipt.pt , Luis.Correia@di.fc.ul.pt

Abstract. The traditional representation of the populations used in evolutionary algorithms raises two types of problems: the loss of genetic diversity during the evolutionary process and evaluation of redundant individuals. In [11, 12] the authors propose a new formal model (PLATO) for multiset representation of individuals and their populations which applied to heuristic algorithms, minimizes the problems identified above. This paper presents a computational representation of populations based in multisets, and the adaptation of the genetic algorithm to deal with this type of representation, the Multiset Genetic Algorithm (MGA). A new operator called rescaling is developed as well as a metric to measure genetic diversity. The standard genetic algorithm is applied to some types of problems using the standard and the new type of populations and empirical results shows the genetic diversity is increased and the number of individuals evaluated is decreased as expected.

Keywords: Multiset, Genetic Algorithms, Genetic Diversity, Population-Based Algorithms

1 Introduction

Evolutionary algorithms (EA) are stochastic methods that mimic the process of biological evolution and have become popular tools for search, optimization, machine learning and design problems [1]. When applying these algorithms to complex problems where the search space is complex the most frequent difficulty is the premature convergence of the algorithm due to the lack of genetic variety. Individuals with better fitness propagate their genes in successive generations leading to a premature convergence [2]. Genetic diversity is essential to the evolutionary process, and without it the EA stops at least good solutions especially in problems that have more than one solution (multi-modal optimization) or involve simultaneous objectives, multi-objective optimization.

Several strategies have been proposed to maintain genetic diversity at different stages in the evolutionary process: Selection with weak pressure to fitness like uniform or tournament selection methods; Reproduction promoting the attraction between elements of different individuals or using other reproductive intelligent operators; Mutation with several adaptive operators; Substitution: promoting the

replacement of individuals with similar genotypes but better fitness [2, 3, 4, 5, 6, 7, 21]. EA have revealed potential to reach good balances in order to find a good set of solutions with limited computational power. The maintenance of many good solutions in parallel is desirable and good algorithms have been developed for the clearing, clustering, crowding, fitness sharing and speciation [19].

The efficacy of the EA is directly related to the size of the population. However, in some applications, the evaluation of a large number of individuals is computationally expensive and delays the evolutionary process. To minimize this problem we find in literature some solutions: saving the objective value of individuals in the databases, estimating the ability of individuals using similar systems and prediction of fitness [9,10].

All strategies above are improvements to the original algorithm, which require the redefinition of the operators and all have an increased computational cost. In [11, 12], the authors introduce a new formal model using the concept of multiset for the representation of populations. This model allows the resolution of two problems listed above: maintaining the genetic diversity and avoiding superfluous evaluations. In [15] the authors show a software prototype, based in these ideas and applied to Genetic Algorithms (GA). This paper describes the adaptation of the GA to a multiset based population, also introducing a new operator called rescaling. This variant is called Multiset Genetic Algorithm (MGA). Empirical results show that the MGA produces higher genetic diversity and smaller number of evaluations than the same GA using simple population (SP). The diversity was calculated by two measures of genetic diversity.

2 Multisets and Multipopulations

A multiset is a collection of elements which may appear repeated. The number of times an element occurs in a multiset is called its multiplicity. The cardinality of a multiset is the sum of the multiplicities of its elements [13]. We can define a multiset as a set of ordered pairs <n,e> where n is the cardinality of the element e. In this definition the set {a,a,a,b,b,c} has an equivalent representation in multiset {<3,a>,<2,b>,<1,c>}.

EAs are based on populations of individuals in the form of collections. In the traditional representation it is common to have repeated individuals within the population (Table 1). This can be efficiently represented by a multiset (Table 2). Multipopulations (MP) are populations where the individuals are represented by ordered pairs <n,g> where n is the number of copies of the genome g. To this ordered pair we call multiindividual (MI), and a MP is a set of MIs with number of copies greater than zero. The set of g in a MP is called the support of the MP. In case of Table 2 we have a support set with four elements (notice that all of them have different genotypes).

Populations are dynamic collections where individuals are inserted, removed and searched. To implement MP we must redefine these three operations in the traditional data structure to support MI.

- **Search** - an MP is a set of individuals, grouped in MI, which can be indexed. The index of an individual is defined as a range from the sum of the multiplicity of all previous MI to this value added of the own multiplicity (Table 3). Searching an individual is searching one index in the population. This index is very important to maintain the equivalence between the MPs and the normal populations (SP) necessary for performance comparisons.

Table 1. Population with 8 individuals of the problem MaxOnes.

Genotype	Fitness
11111110	7
11111110	7
11111110	7
11110000	4
11110000	4
00001110	3
00001110	3
00000010	1

Table 2. Multi-Population equivalent of the population represented in Table 1.

Copies	Genotype	Fitness
3	11111110	7
2	11110000	4
2	00001110	3
1	00000010	1

Table 3. Indexing Multiindividuals in the Multipopulation of Table 2.

Copies	Genotype	Fitness	Indexes
3	11111110	7	0,1,2
2	11110000	4	3,4
2	00001110	3	5,6
1	00000010	1	7

- **Insert** - when an individual is appended to the MP, the first operation is to check if there is already a MI with the same genotype; if true this operation increases the number of copies of the MI to incorporate the new individual, if false the individual is inserted in the population and the number of copies is one (Table 4).

Table 4. Append the individual “11111110” and “00000000” in the MP of the Table 3.

Copies	Genotype	Fitness	Indexes
4	11111110	7	0,1,2,3
2	11110000	4	4,5
2	00001110	3	6,7
1	00000010	1	8
1	00000000	0	9

- **Remove** – the elimination of an individual of one MP is done decrementing the number of copies of the MI. If the number of copies decays to zero, the MI is removed from the data structure (Table 5).

Table 5. Remove the individual at the index 7 ("00001110") and the individual "00000010" (index 8) in the MP of the table 4.

Copies	Genotype	Fitness	Indexes
4	11111110	7	0,1,2,3
2	11110000	4	4,5
1	00001110	3	6
1	00000000	0	7

Appending and removing individuals in the MP produces one search for the individual genotype into the MP, incrementing the computational complexity to the evolutionary process. To minimize this issue the data structures to support MP must be efficient in search, not only to select individuals, but also to add and remove them from the MP. Random access is another important aspect for the implementation of some genetic operators.

Genetic Diversity

Genetic diversity is based on the Hamming distance between individual genotypes. This measure between two bit strings returns the number of bits that are different in the genotype representations. By applying the same principle to the entire population we can calculate the Hamming distance of an individual in relation to the population as the sum of Hamming distances between it and all the other elements of the population. The genetic diversity of the population is given by the sum of Hamming distances of the individuals that compose it. We define a measure called the genetic diversity consisting on a normalized form of the Hamming distance for binary strings. It can be efficiently computed as

$$genetic\_diversity = \frac{4}{n^2 l} \sum_{i=0}^{l} k_i (n - k_i), \tag{1}$$

where l is the number of bits of each individual, k_i the number of ones of the allele i in the population and n the population size. The maximum diversity is 1.0 when the percentage of alleles for each gene is 50%, and the minimum diversity is 0.0 when all genes have the same value. This measure produces similar results to *pop_diversity* [20] but is computationally more efficient and is normalized to interval [0, 1].

Different alleles

The number of different alleles in the population is another diversity measure, which calculates the amount of genes that are not fixed in the population [14]. This metric

can measure the ability of a population to explore the search space through the recombination operator.

The maximum diversity is obtained when the population has different alleles in all the genes and the minimum value is obtained when all the genes have the same value overall the population, which means all individuals are identical.

Equation (2) shows the normalized form of the measure

$$different\_allels = \frac{\sum_{i=0}^{l} a_i}{l}, \tag{2}$$

where l is the number of bits of each individual and a_i is zero if all the alleles of the gene i have the same value and one otherwise.

3 Adaptation of the Genetic Algorithm to Multi-Populations

The simple GA (SGA), when using the MP and the traditional operators, benefits automatically of the reduced number of evaluations and increased genetic diversity. The number of evaluations is decreased because they can evaluate many individuals at once (one multiindividual), and the genetic diversity of the population is increased because all the MI in the support set have different genotypes.

The individuals with best fitness increase their number of copies in successive generations. If this number of copies is not controlled the benefits of the MP may be decreased due to the huge number of copies of the best individuals. We can adapt the common EA operators to this new representation, but a lighter intervention can be considered by just introducing a new operator. This will allow us to compare results of the SGA with the Multiset Genetic Algorithm (MGA).

Rescaling

To control the number of copies of the individuals we adapted the schema defined in [11, 12] and introduced a new operator in the genetic process: rescaling (Fig. 1). Rescaling is a population operator that changes the number of copies of a MI preventing it from taking over population. We divide the number of copies of each individual by a factor r and assign the num of copies to the smallest integer that is greater than the division. Fig. 2 shows the effect of the operator in a population of 128 individuals of the problem Maxones over 250 generations.

With the factor r equal to 1.0 the number of individuals grows in successive iteration. This is due to keep constant the number of MI in the population where the fittest individuals accumulate copies in the evolutionary process. Higher values of r stabilize the number of copies of good individuals. Experimental results show that the value of 1.5 to factor r it's a good compromise between number of individuals and number of MI in the population.

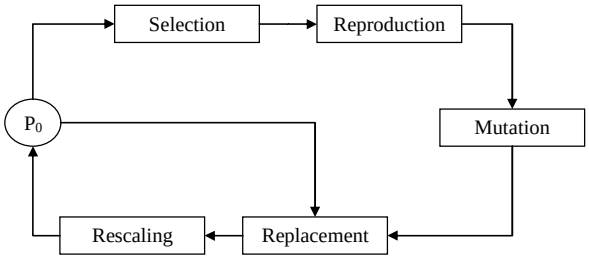

Fig. 1 . Optimization Algorithms Operators

In the first steps of the evolution the number of individuals increases more slowly because there is no individual with greater fitness than the others, and the number of copies is small.

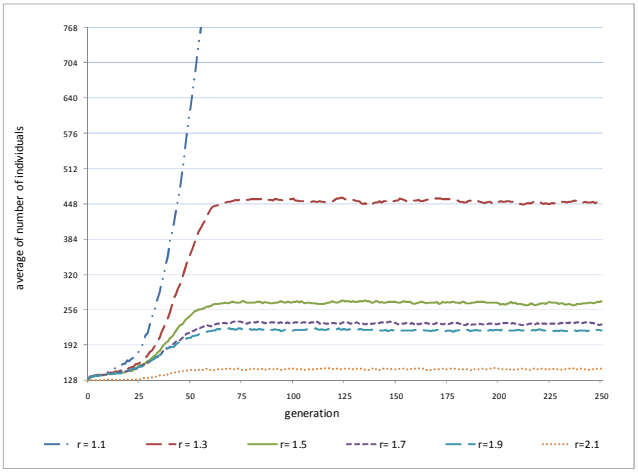

Fig. 2 Effect of rescaling in the number of individuals in the population**.**

4 Empirical study

In our empirical results we use the MUGA simulator [15] to show the effect of MP in the evolutionary process in comparison to SP. To measure the effect of the MP in the evolutionary process we use standard GA operators to evolve the two types of populations: SGA – GA that evolve simple population and MGA – GA that evolve multipopulations. The parameters of the GA are equal for the two types of models and are displayed in Table 8.

Table 8. Parameters used for SGA and MGA except rescaling (only in MGA).

Population size	128 individuals or multiindividuals
Mate population size	128 individuals
Selection	Binary tournament
Combination	Crossover 1-cut point, probability =75%
Mutation	Bitwise mutation with probability =1%
Replacement	Binary tournament with no reposition
Rescaling	Factor = 1.5
Iterations	2500 generations

We use three functions to investigate the effect of MP: Royal Road Function R1 [16], Knapsack [20] and MZ1 [18]. We perform 64 runs for each problem. In each one a new random population is generated and assigned to both MGA and SGA. The results presented below are the arithmetic means of the runs performed.

Royal Road Function R1

This function is designed to investigate, in detail, schema processing and recombination [16]. It uses a 64 bit string. It is unimodal and the search space is organized in steps with constant size.

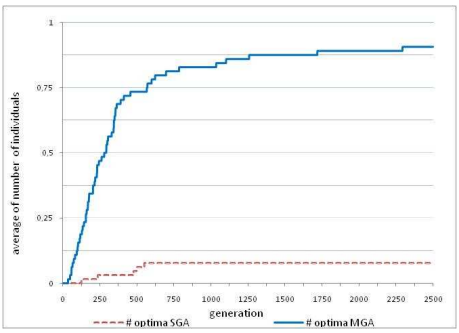

Fig. 3 - Average of number of optima found by generation.

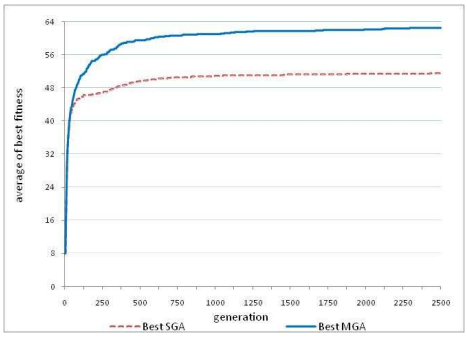

Fig. 4 - Average of value of best individual found by generation.

This function has one maximum and it is hard to discover with no genetic diversity. MGA found the solution 90.6% of the runs and SGA found the optimal solution 7.8% of the runs (Fig. 3, Fig. 4). Fig. 5 and Fig. 6 show the population diversity (different alleles and genetic diversity). The diversity of the SGA is good because the population does not converge and there are many local maxima. MGA has a higher diversity, explained by the constant size of the support set. This genetic diversity in SP eventually generates the best solution but more iterations are necessary. If we take all tests up to 2500 generations we notice the number of evaluations in MGA (22700) is lower than in SGA (29500) (Fig. 7). The number of individuals in MGA stabilizes

in 166 after a few generations (Fig. 8). In the beginning of evolution the number of individuals increases because there are many local maxima with the same fitness. In each higher fitness plateau the number of local maxima decreases which leads to the stabilization of the number of individuals in a smaller value.

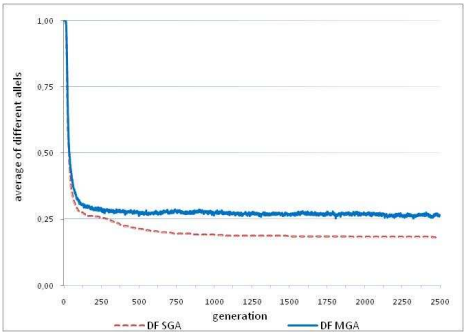

Fig. 5 - Average of number of different alells of the population by generation.

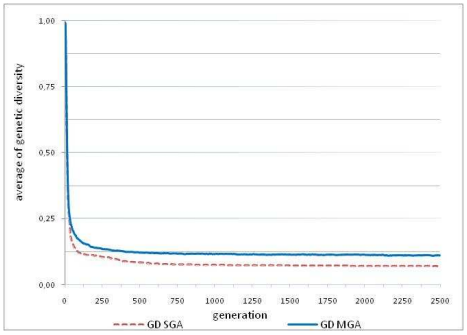

Fig. 6 - Average of genetic diversity by generation.

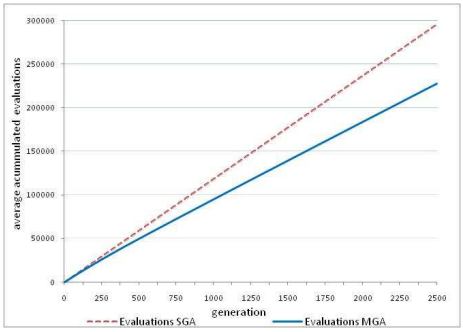

Fig. 7 - Average of accumulated evaluations per generation.

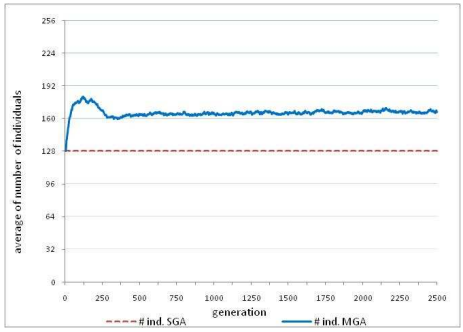

Fig. 8 - Average number of individuals by generation.

Knapsack

Knapsack is a well know NP-complete combinatorial optimization and we use it to show the effects in combinatorial optimization problems. We implement the problem presented in [20]. The maximum capacity is 50% of the total height and penalization is done by the linear function of [18].

Experimental results show that the problem has at least four maxima with the best value known of 1920:

000001111101001011001001011110100110101011111110111

00001111110100101100100101111010001010101111110111
00001110110100101100100101111111011101000111110111
00001111110100101100100101111010111010101110110111

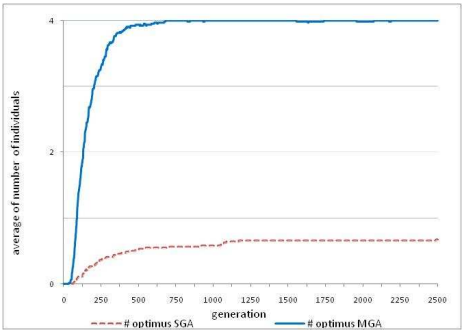

Fig. 9 - Average of number of optima found by generation.

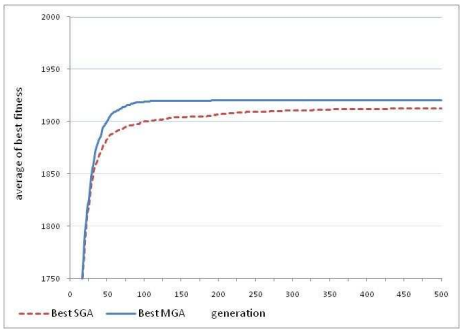

Fig. 10 - Average of value of best individual found by generation.

The MGA always found four optimum solutions (Fig. 9) and the SGA found one optimum 67.1% of the runs. In both simulations a good solution is found. The mean of the best values for SGA is 1915.4 and for MGA is 1920 (the best known) (Fig. 10).

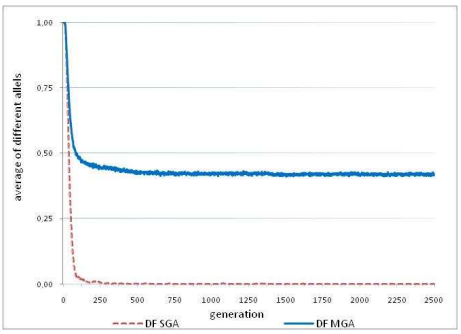

Fig. 11 - Average of number of different alleles of the population by generation.

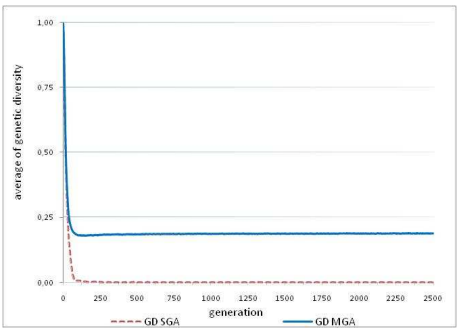

Fig. 12 - Average of genetic diversity by generation.

The number of different alleles (Fig. 11) and genetic diversity (Fig. 12) in SGA converge to zero while in MGA they converge to a good ratio: 41.9% of different alleles and 18.7% of genetic diversity. This lack of genetic diversity explains the difficulty of SGA to find optimum solutions. The number of evaluations is 10.8% higher in SGA and the number of individuals in MGA stabilizes in 226. The higher number of individuals is explained by the number of the optima found by the MGA.

MZ1

MZ1 [18, pp. 36] is a bidimensional function defined in a continuous space and we use it to show the effect of MP in numerical optimization based in real numbers coded in binary strings. The fitness function is described by equation 3. Variable x_1 is defined in the interval $-3.0 \leq x_1 \leq 12.1$ and coded by 18 bits and variable x_2 is defined in the interval $4.1 \leq x_2 \leq 5.8$ and coded by 15 bits.

$$MZ1(x_1, x_2) = 21.5 + x_1 \sin(4\pi x_2) + x_2 \sin(20\pi x_2). \quad (3)$$

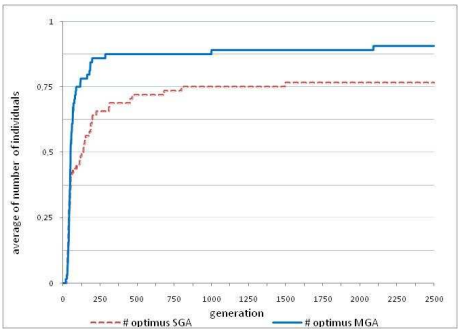

Fig. 13 - Average of number of optima found by generation.

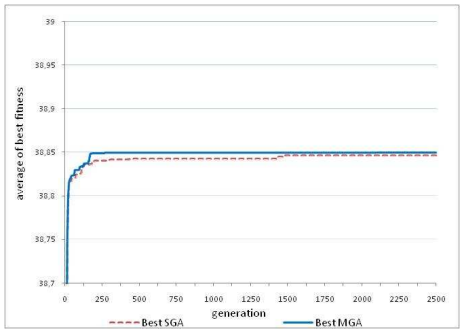

Fig. 14 - Average of value of best individual found by generation.

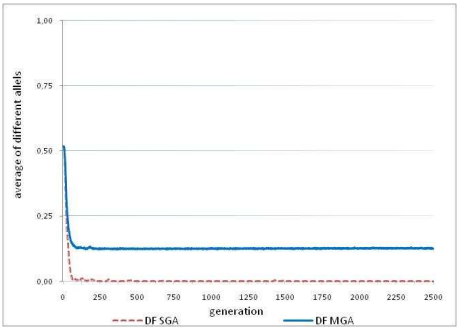

Fig. 15 - Average of number of different alleles of the population by generation

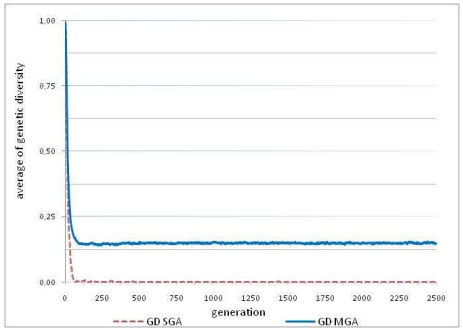

Fig. 16 - Average of genetic diversity by generation

We notice that the results of MGA in this problem are not as good as in the previous ones. Optimization of complex functions is difficult for SGA due to premature convergence of the population. In MGA we also notice premature convergence (Fig. 13 and Fig. 14), in spite of the constant population diversity (Fig. 15 and Fig. 16). This is due to the fact that MGA converges to a local maximum and uses the genetic diversity to form a cluster around it. The number of individuals in the

MP stabilizes in 338, which is a high number resulting from many MI with similar high fitness.

5 Conclusion

In this paper we present an implementation of GA with populations represented by multisets, named Multiset Genetic Algorithm (MGA). This model introduces a new operator called Rescaling, to allow a comparison between MGA and the Simple Genetic Algorithm (SGA). We performed experiments to determine an adequate value for the division factor parameter of the new operator. A metric called genetic diversity, implementing in an efficient way a normalized Hamming distance was proposed to evaluate the performance of the different models.

We conducted a series of tests in different problems, to determine the behavior of the MGA during the evolutionary process and to compare the results with the SGA. The number of evaluations in MGA is lower than in SGA in all the tests. MGA has the largest genetic diversity in all situations. This helped to reach the optimal solution in Royal Road Function R1 and to find and maintain different optimal solutions in the knapsack problem. MGA in MZ1 problem did not get results as significant as in the other problems, but showed some directions to follow in the future. The cluster around a local maximum takes the entire genetic diversity through bits with little significance to the global optimum. The concept of multiindividual (MI) can be redefined with the incorporation of metrics that allow a single MI to represents similar genotypes instead of a single one. Some metrics are being study and will be subject to investigation in the near future.

This work presents the impact of MGA in a optimization process using conventional genetic operators. The standard genetic operators could be redefined, and new ones can be developed take to advantage of the number of copies of the MI. Some operators have already been adapted and are present in MUGA[15]. In the future they will be subjected to analytical treatment to determine their efficiency.

The use of multipopulations involves a computational effort grouping individuals in a MI. Efficient data structures that support efficient search and random access to the individuals will be researched to minimize the computational effort.

The results of the experiences reinforce our conviction that a multiset representation of the population is a powerful way to preserve genetic diversity, to avoid superfluous evaluations and, therefore, to significantly improve the performance of GA.

References

1. Whitley, D.:An Overview of Evolutionary Algorithms: Pratical Issues and Common Pitfalls. Journal of Information and Software , 43:817-831. (2001)
2. Theoretical analysis of diversity mechanisms for global exploration. Proceedings of the 10th annual conference on Genetic and evolutionary computation, pp 945-952 (2008)

3. Amor, B.H.,Rettinger, A.:Intelligent Exploration for genetic Algorithms: using self-organizing maps in evolutionary computation. Proceedings of the 2005 conference on Genetic and evolutionary computation, pp1531-1538. (2005)
4. Cervantes, J., Stephens,C.R. : Rank based variation operators for genetic algorithms. Proceedings of the 10th annual conference on Genetic and evolutionary computation, pp 905-912 (2008)
5. Cherba, M.D., Punch, W.: Crossover gene selection by spatial location. Proceedings of the 8th annual conference on Genetic and evolutionary computation, pp 1111 - 1116 (2006)
6. Hutter, M.:Fitness Uniform Selection to Preserve Genetic Diversity. Proceedings of the 2002 Congress on Evolutionary Computation, pp 783-788 (2002)
7. Lima, C.F., Sastry K., Goldberg, D.E., Lobo, F.G.: Combining competent crossover and mutation operators: A probabilistic model building approach. Proceedings of the 2005 conference on Genetic and evolutionary computation, pp 735 – 742 (2005)
8. Deb, K.: Current trends in evolutionary multi-objective optimization. International Journal for Simulation and Multidisciplinary Design Optimization, pp1-8 (2007)
9. Schmidt, M.D., Lipson H.:Co-evolution of Fitness Maximizers and Fitness Predictors. GECCO Late Breaking Paper, (2005)
10. Kim,E., Cho, S. , An efficient genetic algorithm with less fitness evaluation by clustering.Proceedings of the 2001 Congress on Evolutionary Computation, volume 2, pp 887–894 (2001)
11. Aparício J.N., Correia, L., Moura-Pires, F.: Populations are Multisets-PLATO. GECCO 1999, pp 1845-1850 (1999)
12. Aparício J.N., Correia, L., Moura-Pires, F.: Expressing Population Based OptimizationHeuristics Using PLATO. EPIA 1999, pp 367-383 (1999)
13. Blizard, W. Multiset Theory. Notre Dame Journal of Formal LogicVolume 30, Number 1, Winter (1989).
14. Wagner,S.,Affenzeller,M:The Allele Meta-Model - Developing a Common Language for Genetic Algorithms. Artificial Intelligence and Knowledge Engineering Applications: A Bioinspired Approach, Lecture Notes in Computer Science 3562, pp. 202-211. Springer-Verlag (2005)
15. Manso,A. ,Correia,L.: MUGA (MultiPopulations Genetic Algorithm) - Intelligent Systems Demonstrations Event, EPIA (2005).
16. Mitchell, M.: Introduction to Genetic Algorithms, 5ª ed. MIT Press, pp 94-98 (1998)
17. Jong, Keneth A. de: Evolutionary Computation - A unified approach, The MIT Press, pp 163 (2006)
18 Michalewicz, Z.: Genetic Algorithms + Data Structures = Evolution Programs. 2rd extend Edition Springer-Verlag pp. 36 ; pp 80 (1996)
19 Singh, G, Deb, K.: Comparison of multi-modal optimization algorithms based on evolutionary algorithms, Proceedings of the 8th annual conference on Genetic and evolutionary computation, pp 1305 – 1312 (2006)
20 Evolutionary Computation Benchmark Repository- Fifty item Knapsack Problem, http://www.cs.bham.ac.uk/research/projects/ecb/ (July , 2009)
21 Ochoa, Gabriela. Error Threashold in Genetic Algorithms, Evolutionary Computation Journal, MIT Press; pp. 157-182 (2006)