A Study on Non-random Mating and Varying Population Size in Genetic Algorithms Using a Royal Road Function

Carlos Fernandes
Laseeb-Evolutionary
Systems and Biomedical Engineering Lab.,
Av Rovisco Pais, 1, Torre Norte, 6.21,
Lisbon, Portugal
cfernandes@lovelace.isr.ist.utl.pt

Agostinho Rosa
Laseeb-Evolutionary
Systems and Biomedical Engineering Lab.,
Av Rovisco Pais, 1, Torre Norte, 6.21,
Lisbon, Portugal
acrosa@isr.ist.utl.pt

Abstract- In this paper we present a study on the effects of non-random mating and varying population size in Genetic Algorithms (GAs) performance. We tested two algorithms: the non-incest Genetic Algorithm with varying population size (niGAVaPS) and the negative Assortative Mating Genetic Algorithm with varying population size (nAMGAVaPS), on a Royal Road function. These algorithms mimic natural species behavior by selecting parents according to parenthood (niGAVaPS) or phenotype similarity (nAMGAVaPS). We show that both algorithms outperform simple GA in the example shown. The results suggest that this may be due to the fact that genetic diversity is kept at a higher level by niGAVaPS and nAMGAVaPS, preventing the premature convergence of the algorithms to local optima.

1 Introduction

The parent selection techniques used in the Simple Genetic Algorithm (simple GA) [Holland75] [Goldberg89] only take into account the individual fitness, that is, its adaptation to the problem (environment). In Genetics the name of this process is *random mating*. In random mating, an individual mates with another, regardless of its parenthood or likeness [Roughgarden79] [Russel98]. Very often, the mating selection of natural species is also based on the familiar relationship between parents and/or its phenotype similarity. This behavior is called *non-random mating* and it is divided in two general categories: *inbreeding in the broad sense* and *assortative mating*. Inbreeding in the broad sense is the occurrence of mating between familiar related individuals more or less often than expected by chance. If it is more often, then it is called inbreeding in the strict sense or, as it is more commonly known, incest. Mating between relatives less often than in random mating, as it happens with humans, is called outbreeding. Assortative mating refers to mating between individuals with phenotype similarities. It is positive if similar individuals mate more often and negative if dissimilar individuals have more chance of mating with each other.

Another characteristic of the simple GA is the fixed size of the population. In natural environments the population size of a species changes according to a growth rate [Roughgarden79] [Song88] and tends to stabilize around an appropriate value, as long as the environment conditions don't suffer drastic changes. If these conditions do change the population size will reflect those changes.

The purpose of this work is to test the effects of incest prohibition and negative assortativeness in GAs with varying population size. According to population dynamics in biology these two particular cases of non-random mating maintain a higher genetic diversity in natural populations than random mating. We expect to maintain the genetic diversity of a GA and create conditions for reducing the percentage of runs in which the algorithm gets trapped in local optima.

The paper is organized as follows: Section 2 gives an overview of related work on non-random mating and varying population size. Section 3 describes our approach, which was applied to a Royal Road function described in Section 4. Section 5 presents the results and Section 6 concludes the paper.

2 Previous Work

Studies on non-random mating in GAs can be found in [Eshelman91], [Craig95] and [Fernandes00].

In [Eshelman91] the incest prevention is done by avoiding crossovers between parents with a Hamming distance below a certain threshold. This process is similar to assortative mating although the authors call it incest prevention.

[Craig95] proposes a method for incest prevention where recombination between individuals with a certain degree of parenthood is not allowed. The authors define an incest prevention degree, which defines how far back in the family tree the GA must look in order to prevent recombination between two related individuals.

0-7803-6657-3/01/$10.00 ©2001 IEEE

In [Mich94] the GAVaPS – Genetic Algorithm with varying population size – is presented. The GAVaPS is a GA with varying population size where each individual has a lifetime, which is measured in generations. An individual remains in the population during its lifetime. Parents are selected randomly because the selective pressure is assured by the fact that genomes representing better solutions have longer lifetimes.

In [CostaTav99] the authors compare GAs with fixed and varying population sizes and conclude that, in most situations, the varying population size GA's performance is at least equivalent to the simple GA, even if the size of the population is changed randomly during the run.

3 New Approach

The aim of this work is to study the effects of varying population size and non-random mating, used simultaneously, on GAs diversity and performance.
In this Section we describe the characteristics of niGAVaPS and nAMGAVaPS.

3.1 Non-incest Genetic Algorithm with varying population size

The niGAVaPS [Fernandes00] combines some ideas of [Craig95] and [Mich94]. Each individual has an id, an ancestry table and a lifetime, which is computed according to its fitness and population characteristics.

3.1.1 Incest prohibition

In niGAVaPS an incest prohibition degree is defined which controls how far back in the family tree of an individual the algorithm must look, in order to prevent recombination with a related individual. Each individual has an ancestry table, with a specific size that grows with the incest prevention degree. The GA checks this table in order to prevent recombination between related individuals

In degree 1 of prevention the ancestry table of an individual contains the ids of its parents. In degree 2 the table will also have the grandparents' ids and so on with increasing degrees.

Table 1 shows the forbidden relationships in crossover operations for degree 1 and 2. In degree 0, the ancestry table is empty and all mating is allowed.

Degree	Comment
0	All mating allowed (simple GA)
1	No mating with itself, parents and siblings.
2	In addition to degree 1, no mating with grandparents, uncles and cousins.

Table 1 - Incest prevention degrees

A new genome, created by the crossover operator, inherits the ids of its parent's ancestry tables except for the ones referring to the older generation – for example, the grandparents ids of its parents, if the degree of prevention is 2, are not inherited by the child.
The size of the ancestry table is given by the following rule:

$$Table\_size = \sum_{n=0}^{degree} 2^n$$

3.1.2 Lifetime calculation strategy

The lifetime of each genome is computed by the following function [Mich94] at the moment of its creation:

$$lifetime\,[i] = MinLT + \eta \frac{fitness\,[i] - MinFit}{AvgFit - MinFit}$$

if

$$AvgFit \geq fitness\,[i]$$

or

$$lifetime\,[i] = \frac{1}{2}(MinLT + MaxLT) + \eta \frac{fitness\,[i] - AvgFit}{MaxFit - AvgFit}$$

if

$$AvgFit < fitness\,[i]$$

where

$$\eta = \frac{1}{2}(MaxLT - MinLT)$$

The variables MinLT and MaxLT are the minimum and maximum generations in which a genome can be kept in the population. The lifetime is the basic mechanism of the varying population size. The genomes remain in the population for a certain number of generations, depending on the quality of the solutions they represent. Better solutions remain in the population for more generations and will have a higher chance of generating offspring.

3.2 Negative Assortative Mating Genetic Algorithm with varying population size

The negative Assortative Mating GA [Fernandes01] differs from the simple GA in the parents' selection. While in simple GA, two parents are selected by a specific method (Roulette Wheel, for instance), in nAMGAV one parent (1st parent) and a set A of genomes are chosen by that same method. The 2nd parent will be the one, belonging to set A, less similar to the first parent. For the Royal Road function was use the Hamming distance to measure the similarity between two genomes. In this case the genome chosen for second parent will be the one with maximum Hamming distance to the first parent – see Figure 1. The Hamming distance between two binary strings is the number of bits in which the strings differ. If several 2nd parents have the same Hamming distance the one with higher fitness is selected.

1st parent	*2nd parent set*	*Ham. Dist.*
110101	101111	3
	111000	3
	100101	1
	111110	3
	011000	4

Parents: 110101 and 011000

Figure 1 – An example of parent selection in negative Assortative Mating GA. The first parent and a set A of #A genomes are chosen. A genome from set A (#A=5), which maximizes the Hamming distance to the first parent (more dissimilar), is chosen as the second parent.

In nAMGAVaPS the population size varies in the same manner that was described in Section 3.1.

4 Royal Road

The Royal Road functions [Mitchell92] were designed to study building blocks' interactions during the GA process. These functions are made up of a structure of building blocks that allows the monitoring of the GA progress. The GA must follow a path formed by this structure - the "Royal Road" - to achieve the global optimum. In a typical run of a GA it can be seen that there are fitness jumps, which correspond to finding another *level* in the search process. In the lowest level (level 0) the fitness (score) is computed by checking small blocks of bits. Higher levels are sequences of lower level blocks.

In the example shown in this paper, the genome has 240 bits and the highest level is level 4. By trying to maximize the score for a certain level, the GA can get trapped in local optima, never reaching level 4. The complete parameter settings of this Royal Road function can be found in [Holland93].

4.1 Similarity Measure

In order to use Assortative Mating in GAs, it is necessary to design a measure of similarity between two genomes. This measure depends on the problem and on the genetic representation of the genomes. As we discussed in Section 3, the Hamming distance measures the similarity between two genomes for the Royal Road functions. To study the effect of non-random mating and varying population size in the genetic diversity of a population we also need to develop a diversity measure suited for the problem.

4.2 Diversity Measure

To measure the population diversity *d(P)* of a certain generation we used the following function [Jones94]:

$$d(P) = \frac{\sum_{i=1}^{L} \min(F_i, 1 - F_i)}{L/2}$$

where

$$F_i = \frac{\sum_{j=1}^{N} P_i(j)}{N}$$

and

$$P_i(j) = \begin{cases} 1, & \textit{if the } j^{th} \textit{ gene of genome } i \textit{ has allele } 1 \\ 0, & \textit{otherwise} \end{cases}$$

The constant N represents the population size and L the genome dimension in bits.

Figure 2 shows the diversity evolution of a simple GA with four different mutation rates (mean value of 100 runs per configuration) applied to the Royal Road function. As expected from theory, the diversity increases with higher mutation rates, therefore the measure seems appropriate for this problem. The best performance was achieved with a mutation rate of 0.005. Higher mutation tends also to be destructive. So, high genetic diversity does not imply, by itself, better performance.

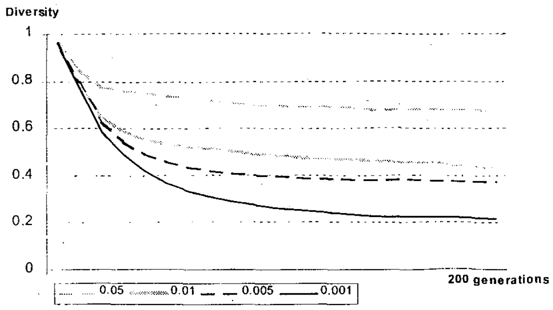

Figure 2 – Genetic diversity of simple GA with different mutation rates (mean values over 100 runs for each configuration).

The simple GA parameter values used for this test are the same described in Section 5. For each configuration we made 100 runs, each evaluating 80000 new genomes corresponding to 200 generations.

5 Results

Using the Royal road function described in Section 4 we compared the performance of the simple GA, niGAVaPS and nAMGAVaPS. The characteristics of the simple GA are the following:

- Population size: 500
- 1-elitism
- 1-point crossover with probability 0.8
- 1-bit mutation with probability 0.005 (mutation always flips a bit)
- Roulette wheel parent selection technique. Fitness is evaluation (no scaling factor or fitness scaling function).

The niGAVaPS uses the same values except for initial population size (400 individuals) and crossover probability (probability of 1.0). The reason for this change in values is the following: in simple GA, two parents that do not pass the crossover test are copied to the next generation (reproduction). In niGAVaPS, it makes no sense to use reproduction as the genomes remain in the population during their lifetime. The initial population size is reduced to 400 so that in each generation both simple GA and niGAVaPS create 400 new genomes. The same applies to nAMGAVaPS.

We also tested particular cases of niGAVaPS and nAMGAVaPS: the niGA and nAMGA, which have fixed population size, and the GAVaPS.

The parameter values of GAs with varying population size were the following:

- Initial population size: 400
- Crossovers per generation: 200
- MinLT = 1
- MaxLT = 6
- Roulette wheel parent selection technique (we also tested random selection of parents as in [Mich94] but the algorithm performed poorer)

The incest prohibition was tested with different degrees and, in assortative mating GAs, tests were made with several sizes of set A (2^{nd} parent set).

5.1 Performance analysis

In this Section we present results for 6 different sets of algorithms: simple GA, GAVaPS, niGA and niGAVaPS (degree 1 to degree 4), nAMGA and nAMGAVaPS (#A=2 to #A= 5, #A=10 and #A=20). For each configuration, we made 100 runs, each evaluating 160000 new genomes corresponding to 400 generations.

In Figure 3 we can see the results obtained by the best configuration of each set.

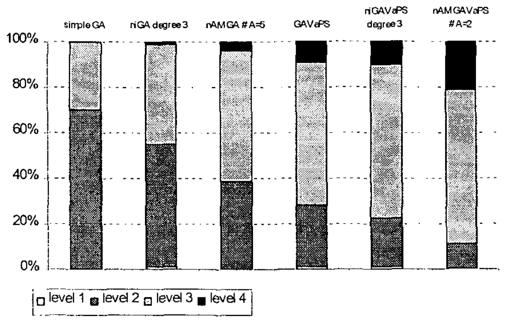

Figure 3 – Levels attained (mean values over 100 runs for each configuration) for the simple GA, niGA degree 3, nAMGA #A=5, GAVaPS, niGAVaPS degree 3 and nAMGAVaPS #A=2.

We can see that the use of incest prevention, negative assortative mating and varying population size outperforms simple GA. The performance difference is particularly noticed between simple GA and nAMGAVaPS. The best performance was obtained by putting together non-random mating and varying population size (niGAVaPS and nAMGAVaPS).

Figure 4 and Figure 5 show the effect of increasing the incest prohibition degree and the assortative mating second parent set's size, respectively. In both figures we can see that there is a direct relation between performance, degree of incest prohibition and set A size. In Figure 4 the results of niGA with degree 4 are not shown because in most of the runs the algorithm stopped. The population size is not large enough to allow this degree of prevention. After a few generations it may be impossible to find parents which are not related.

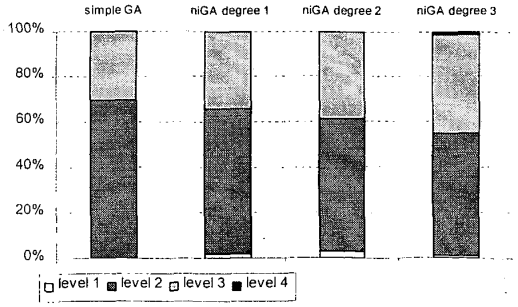

Figure 4 - Levels attained (mean values over 100 runs for each configuration) for the simple GA and niGA (degrees 1, 2 and 3).

We did the same tests with varying population size but the same relation was not evident. The niGAVaPS and nAMGAVaPS outperform, in general, the GAVaPS but there is no monotonic relationship between degree of prohibition or size of set A and algorithm performance.

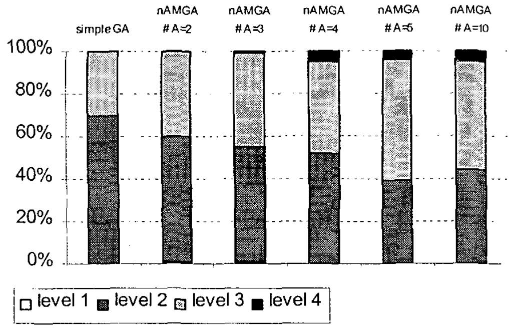

Figure 5 - Levels attained (mean values over 100 runs for each configuration) for the simple GA and nAMGA (#A=2 to #A= 5 and #A=10).

Table 1 shows all the numeric results. The percentage of runs in each algorithm that attained a certain level and the mean number of generations to attain are shown for all configurations. We can see that the simple GA only attained level 3 in 30% of the runs and all the other configurations achieved a higher rate. Simple GA's mean number of generations to attain level 3 is worse than every other configuration with the exception of niGA.

			Lev.2	Lev.3	Lev.4
Simple GA		Gener.	87.7	231.0	-
		%runs	70%	30%	0%
niGA	degree 1	Gener.	58	244.4	-
		%runs	98%	34%	0%
	degree 2	Gener.	70.2	238.7	-
		%runs	97%	39%	0%
	degree 3	Gener.	57.57	235.3	-
		%runs	99%	45%	1%
GAVaPS		Gener.	42.93	185.8	-
		%runs	99%	71%	9%
nAMGA	#A=2	Gener.	41.717	172.5	-
		%runs	99%	40%	0%
	#A =3	Gener.	35.050	139.55	-
		%runs	99%	45%	1%
	#A =4	Gener.	36.6	109.17	-
		%runs	100%	48%	5%
	#A =5	Gener.	38.2	160.33	-
		%runs	100%	61%	4%
	#A =10	Gener.	35.7	146.43	-
		%runs	100%	56%	5%
niGAVaPS	degree 1	Gener.	45.2	172.96	280
		%runs	100%	71%	13%
	degree 2	Gener.	39.49	205.86	258.75
		%runs	98%	70%	8%
	degree 3	Gener.	44:64	171.04	224.00
		%runs	99%	77%	10%
	degree 4	Gener.	47.4	185.29	300
		%runs	100%	70%	11%
nAMGAVaPS	#A =2	Gener.	31.3	150.11	253.81
		%runs	100%	89%	21%
	#A =3	Gener.	35.4	139.18	262.18
		%runs	100%	85%	23%
	#A =4	Gener.	32.2	122.62	256.87
		%runs	100%	89%	16%
	#A =5	Gener.	32.9	136.25	267.78
		%runs	100%	80%	18%
	#A =10	Gener.	32.5	138.18	220.53
		%runs	100%	88%	19%

Table 2 – Results for all configurations of simple GA, niGAVaPS and nAMGAVaPS (mean values over 100 runs). The mean number of generations to attain each level and the percentage of runs that reached each level are shown. The number of generation's calculation for each level only takes into account the successful runs.

5.1.1 Computational time analisys

The computation time analysis shows us that the nAMGAVaPS with #A=5 is about 40% slower than the simple GA. Figure 6 shows the time needed by the simple simple GA, niGAVaPS and nAMGAVaPS to compute one generation.

The tests, averaged over 100 runs for each configuration, were performed on a Pentium II at 233MHz.

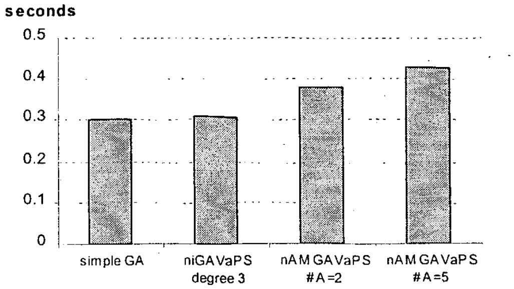

Figure 6 – Computational time of one generation.

To make a more accurate analysis of the algorithms performance, using the results shown in Table 2, we developed the following function, which measures the performance of an algorithm in attaining a level of the Royal Road function:

$$p = (mean\_gen + \frac{MAX\_GEN * 100}{\%level\ L} - MAX\_GEN) * t_{generation}$$

where *%level L* is the percentage of runs which attained level *L*, *mean_gen* is the mean number of generations required to attain level L, *MAX_GEN* is the maximum generations per run and $t_{generation}$ is the running time of one generation.

Each run evaluated 400 generations (*MAX_GEN*=400). The performance measure function *p* is a product of the mean number of generations required to attain a level L and the running time of one generation. Note that all the algorithms evaluate 400 new individuals in each generation.

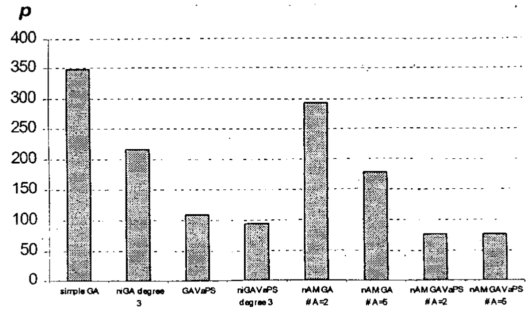

Figure 7 – Performance measure using function *p*.

Figure 7 shows us the performance of the several algorithms in attaining level 3. We can see that the simple GA, although it attains the lowest computational time of one generation, is the algorithm with worst performance. The nAMGAVaPS and nGAVaPS attain the best performance of all configurations.

5.2 Genetic Diversity

In Figure 8 we can see the genetic diversity of the simple GA, niGAVaPS, nAMGA and nAMGAVaPS's populations. In simple GA, the diversity decreases during the runs while in niGAVaPS the diversity decreases in the first 100 generations, it remains stable for approximately 100 generations and then slightly increases. In nAMGA and nAMGAVaPS the genetic diversity remains stable after the first 100 generations. The graphic shows us that a higher diversity does not imply better performance. The nAMGA has the highest diversity but does not achieve the highest performance. The best performance was achieved by nAMGAVaPS, which kept the genetic diversity in values similar to those obtained by the simple GA with a mutation rate of 0.01 – see Figure 2. Although with similar diversity, nAMGAVaPS clearly outperformed the simple GA with that mutation rate.

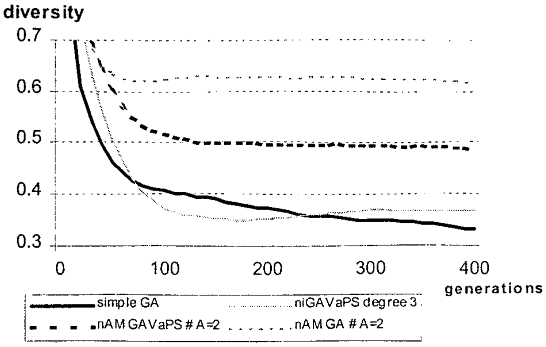

Figure 8 – Genetic diversity observed in 100 runs (mean values) of the simple GA, niGAVaPS degree 3, nAMGA and nAMGAVaPS with #A=2.

6 Conclusions

The niGAVaPS and nAMGAVaPS outperform simple GA with the Royal Road function described in Section 4. While simple GA attains level 3 only in 30% of the runs, niGAVaPS degree 3 and nAMGAVaPS with #A=2 attain that same level in 77% and 89% of the runs, respectively. Level 4 is attained in 10% of runs by niGAVaPS and 21% of the runs by nAMGAVaPS while simple GA never converged, in the tests we made, to the optimal solution.

Although the running time of the varying population size and non-random mating algorithms is higher than simple GA the mean time to attain a level 3 in several runs is lower. The tests also showed us that non-random mating and varying population size better perform when used together.

The higher performance of niGAVaPS and nAMGAVaPS may be due to the fact that these algorithms maintain a higher genetic diversity in the population thus avoiding getting trapped in local optima. We saw that the way the genetic diversity is kept in a higher level is important because high mutation rates, which maintain higher diversity in populations, don't imply better GA performance. The diversity in non-random mating and varying population GAs is maintained by mating dissimilar individuals. In nAMGAVaPS the parents are chosen according to their Hamming distance. Genomes with higher Hamming distance have more chance of mating. By preventing incest, niGAVaPS increases the probability of crossover between dissimilar genomes because related individuals are, in general, more similar than non-related ones. Although mutation has an important role in diversity maintenance, by assuring that no allele permanently disappears from the population, a high mutation rate can be destructive and in the limit (mutation rate = 1) will lead to random search. Usually there is an optimal mutation rate and increasing it will maintain genetic diversity for a longer period but will also decrease GA performance. Non-random mating and varying population size maintain higher genetic diversity than the simple GA with the same mutation rate leading to better performance.

Bibliography

[CostaTav99] Costa, J., Tavares, R., Rosa, A.C. (1999) "An Experimental Study on Dynamic Random Variation of Population Size". Proceedings of the 1999 IEEE International Conference on Systems, Man, and Cybernetics, pp.607-612.

[Craig95] Craighurst, R., Martin, W. (1995) "Enhancing GA Performance through Crossover Prohibitions Based on Ancestry". Proceedings of the Sixth International Conference on Genetic Algorithms, pp.130-135.

[Eshelman91] Eshelman, L.J., Schaffer, J.D. (1991) "Preventing Premature Convergence in Genetic Algorithms by Preventing Incest". Proceedings of the Fourth International Conference on Genetic Algorithms, pp.115-122.

[Fernandes00] Fernandes, C., Tavares, R., Rosa, A.C. (2000) "niGAVaPS – Outbreeding in Genetic Algorithms". Proceedings of the 2000 ACM Symposium on Applied Computing, pp.477-482.

[Fernandes01] Fernandes, C., Tavares, R., Rosa, Munteanu, C., Rosa, A.C. (2001) "Assortative Mating in Genetic Algorithms for Vector Quantization Problems". To be published in the Procedings of the 2001 ACM Symposium of Applied Computing.

[Goldberg89]Goldberg, D.E. (1989) "Genetic Algorithms in Search, Optimization and Machine Learning". Addison-Wesley Publishing Company, Inc.

[Holland75] Holland, J.H. (1975) "Adaptation in Natural and Artificial Systems". MIT Press, Cambridge, Massachusetts.

[Holland93] Holland, J.H. (1993) "Royal Road Functions", Genetic Algorithms (GAList Internet mailing list)Digest, volume 7 issue 22.

[Mich94] Michalewicz Z. (1994) "Genetic Algorithms + Data Structures = Evolution Programs" (second, extended edition). Springer-Verlag.

[Mitchell92] Mitchell, M., Forrest, S., Holland, J. H. (1992) "The Royal Road for Genetic Algorithms: Fitness Landscapes and GA Performance". Proceedings of the First European Conference on Artificial Life, pp.245-254.

[Roughgarden79] Roughgarden, J. (1979) "Theory of Population Genetics and Evolutionary Ecology". Prentice-Hall.

[Russel98] Russel, P.J. (1998) "Genetics". Benjamin/ Cummings.

[Song88] Song, J., Yu, J. (1988) "Population System Control". Springer-Verlag.

[Jones94] Jones T. (1994) "One offspring or two". Genetic Algorithms (GAList Internet mailing list) Digest, volume 8 issue 21.