

Gene Pool Recombination in Genetic Algorithms*

Heinz Mühlenbein
GMD 53754 St. Augustin
Germany
muehlenbein@gmd.de

Hans-Michael Voigt
T.U. Berlin 13355 Berlin
Germany
voigt@fb10.tu-berlin.de

Abstract:

A new recombination operator, called Gene Pool Recombination (*GPR*) is introduced. In *GPR*, the genes are randomly picked from the gene pool defined by the selected parents. The mathematical analysis of *GPR* is easier than for two-parent recombination (*TPR*) normally used in genetic algorithms. There are n difference equations for the marginal gene frequencies that describe the evolution of a population for a fitness function of size n . For simple fitness functions *TPR* and *GPR* perform similarly, with a slight advantage for *GPR*. Furthermore the mathematical analysis shows that a genetic algorithm with only selection and recombination is not a global optimization method, in contrast to popular belief.

Keywords: Difference equations, genetic algorithms, Hardy-Weinberg equilibrium, recombination.

1. Introduction

Genetic algorithms (GAs) use at least three different components for guiding the search to an optimum — selection, mutation and recombination. Understanding the evolution of genetic populations is still an important problem for biology and for scientific breeding. Mühlenbein and Schlierkamp-Voosen (1993, 1994) have introduced classical approaches from population genetics, the science of breeding, and statistics to analyze genetic algorithms. They describes the evolution of genetic populations as a dynamical system by difference or differential equations. Analyzing GAs that use both recombination and selection turns out to be especially difficult. The problem is that the mating of two genotypes creates a complex linkage between genes at different loci. This linkage is very hard to model and represents the major problem in population genetics (Naglyaki 1992).

For simple linear fitness functions we have found approximate solutions to the equations that describe the evolution of a genetic population through selection and recombination. Looking carefully at the assumptions leading to the approximation, we found that the equations obtained would be exact if a different recombination scheme were used. This recombination scheme we call *gene pool recombination (GPR)*. In *GPR*, for each locus the two alleles to be recombined are chosen *independently* from the gene

* To be published Proceedings of the Metaheuristics Inter. Conf., eds. Osman, I. H. and Kelly, J. P., Kluwer Academic Publishers, Norwell, 1995

pool defined by the selected parent population. The biologically inspired idea of restricting the recombination to the alleles of two parents for each offspring is abandoned. The latter recombination we will call *two-parent recombination (TPR)*.

The idea of using more than two parents for recombination is not new. Already Mühlenbein (1989) used eight parents; the offspring allele was obtained by a majority vote. Multi-parent recombination has also been investigated recently by Eiben, Raue and Ruttkey (1994) though their results are somewhat inconclusive. For binary functions the bit-based simulated crossover (BSC) of Syswerda (1993) is similar to GPR. However, his implementation merged selection and recombination. An implementation of BSC which separates selection and recombination was empirically investigated by Eshelman and Schaffer (1993). GPR is an extension of BSC, it can be used for any representation — discrete or continuous.

In this paper we will investigate TPR and GPR for discrete binary functions. It will be shown that GPR is easier to analyze than TPR. Furthermore, it converges faster. Nevertheless, in many cases TPR can be considered as an approximation to GPR.

2. Response to selection

In this section we summarize the theory presented in Mühlenbein and Schlierkamp-voosen (1993, 1994). Let $\bar{f}(t)$ be the average fitness of the population at generation t . The response to selection is defined as

$$R(t) = \bar{f}(t+1) - \bar{f}(t). \quad (1)$$

The amount of selection is measured by the selection differential

$$S(t) = \bar{f}_s(t) - \bar{f}(t), \quad (2)$$

where $\bar{f}_s(t)$ is the average fitness of the selected parents. The equation for the response to selection relates R and S :

$$R(t) = b(t) \cdot S(t). \quad (3)$$

The value $b(t)$ is called the *realized heritability*. For many fitness functions and selection schemes, the selection differential can be expressed as a function of the standard deviation σ_p of the fitness of the population. For *truncation selection* (selecting the $T \cdot N$ best individuals) and for normally distributed fitness, the selection differential is proportional to the standard deviation (Falconer 1981):

$$\frac{S(t)}{\sigma_p(t)} = I.$$

The value I is called the *selection intensity*. For arbitrary distributions one can show the following estimate (Nagaraja 1982):

$$\frac{S(t)}{\sigma_p(t)} \leq \sqrt{\frac{1-T}{T}}.$$

For normally distributed fitness the famous equation for the response to selection is obtained (Falconer 1981):

$$R(t) = I \cdot b(t) \cdot \sigma_p(t). \quad (4)$$

The above equation is valid for a large range of distributions, not just for a normal distribution. The response depends on the selection intensity, the realized heritability, and the standard deviation of the fitness distribution. In order to use the above equation for prediction, one has to estimate $b(t)$ and $\sigma_p(t)$. The equation also gives a design goal for genetic operators — to *maximize the product of heritability and standard deviation*. In other words, if two recombination operators have the same heritability, the operator creating an offspring population with larger standard deviation is to be preferred.

The equation defines also a design goal for a selection method — to *maximize the product of selection intensity and standard deviation*. In simpler terms, if two selection methods have the same selection intensity, the method giving the higher standard deviation of the selected parents is to be preferred. For proportionate selection as used by the simple genetic algorithm (Goldberg 1989) it was shown by Mühlenbein and Schlierkamp-Voosen (1993) that

$$\frac{S(t)}{\sigma_p(t)} = \frac{\sigma_p(t)}{\bar{f}(t)}.$$

The equation shows that the selection intensity of proportionate selection goes to zero for $t \rightarrow \infty$, whereas truncation selection has a constant selection intensity. Proportionate selection selects too weak in the final stages of the search.

Before showing that the evolution of a genetic population depends indeed on the selection intensity, the heritability and the standard deviation, we will derive exact difference equations for two and three loci directly.

3. Exact analysis of TPR for two loci

For simplicity we restrict the discussion to two loci and proportionate selection. In this case there are four possible genotypes: $(0, 0)$, $(0, 1)$, $(1, 0)$, and $(1, 1)$ which we index by $i = (0, 1, 2, 3)$. We denote their fitness values m_0, m_1, m_2 , and m_3 respectively. Let $q_i(t)$ be the frequency of genotype i at generation t . We assume an infinite population and *uniform crossover*. For proportionate selection the exact equations describing the evolution of the frequencies q_i can be derived easily. These equations — known for diploid chromosomes in population genetics (Crow and Kimura 1970) — assume an infinite population.

$$q_i(t+1) = \frac{m_i}{\bar{f}(t)} q_i(t) + \epsilon_i \frac{D(t)}{2\bar{f}(t)^2} \quad i = 0, 1, 2, 3 \quad (5)$$

with $\epsilon = (-1, 1, 1, -1)$. $\bar{f}(t) = \sum_{i=0}^3 m_i q_i(t)$ is the average fitness of the population. $D(t)$ defines the deviation from linkage equilibrium

$$D(t) = m_0 m_3 q_0(t) q_3(t) - m_1 m_2 q_1(t) q_2(t). \quad (6)$$

Note that $D(t) = 0$ if $q_0(t) q_3(t) = q_1(t) q_2(t)$ and $m_0 m_3 = m_1 m_2$. The first condition is fulfilled if the genotypes are binomially distributed. This assumption is called the *Hardy-Weinberg equilibrium* in population genetics. The general nonlinear difference equations have not yet been solved analytically (see the discussion by Naglyaki (1992)), but it is possible to derive an exact expression for the realized heritability. By summation we obtain

$$R(t) = \bar{f}(t+1) - \bar{f}(t) = \frac{V(t)}{\bar{f}(t)} - (m_0 + m_3 - m_1 - m_2) \frac{D(t)}{2\bar{f}(t)^2}, \quad (7)$$

where $V(t) = \sigma^2(t) = \sum q_i(t)(m_i - \bar{f}(t))^2$ denotes the variance of the population. Using $S(t) = V(t)/\bar{f}(t)$ we obtain the exact equation for the heritability,

$$b(t) = 1 - (m_0 - m_1 - m_2 + m_3) \frac{D(t)}{2\bar{f}(t)V(t)}. \quad (8)$$

In general, $b(t)$ depends on the genotype frequencies. Note that $b(t) = 1$ if $D(t) = 0$ or $m_0 + m_3 = m_1 + m_2$. The second assumption is fulfilled for the function ONEMAX(2) which has the fitness values $m_0 = 0, m_1 = m_2 = 1, m_3 = 2$. From (5) we obtain

$$\begin{aligned} \bar{f}(t+1) &= q_1(t+1) + q_2(t+1) + 2q_3(t+1) \\ &= \frac{q_1(t) + q_2(t) + 4q_3(t)}{\bar{f}(t)} = 1 + \frac{2q_3(t)}{\bar{f}(t)}. \end{aligned}$$

Let $p(t)$ denote the frequency of allele 1. Then by definition $\bar{f}(t) = 2p(t)$. Therefore we obtain

$$R(t) = 1 - p(t) + \frac{B_3(t)}{p(t)}, \quad (9)$$

where $B_3(t)$ denotes how far $q_3(t)$ deviates from the frequency given by the binomial distribution: $B_3(t) = q_3(t) - p^2(t)$.

The exact difference equation for $p(t)$ can be written as

$$p(t+1) = p(t) + \frac{1}{2} (1 - p(t)) + \frac{B_3(t)}{2p(t)}. \quad (10)$$

This equation has two unknown variables, $p(t)$ and $q_3(t)$. Therefore $p(t)$ cannot be directly computed. Selection leads the population away from the binomial distribution, and TPR is not able to recreate a binomial distribution for the offspring population.

We now discuss a function where $D(t) = 0$ if the population starts in a Hardy-Weinberg equilibrium. An example is MULT(2) with fitness values $m_0 = 1, m_1 = m_2 = 2, m_3 = 4$. In this case the difference equation for $p(t)$ is given by

$$p(t+1) = p(t) \frac{2}{1 + p(t)}, \quad (11)$$

which is solved easily.

In summary, even linear fitness functions lead to difficult systems of difference equations. The genetic population moves away from Hardy-Weinberg equilibrium. A class of multiplicative fitness functions with $m_0 m_3 = m_1 m_2$ leads to simpler equations, because the population stays in Hardy-Weinberg equilibrium.

4. Gene Pool Recombination

The exact analysis of recombination together with selection leads to difficult nonlinear differential equations. Recombination of two genotypes creates a linkage between the genes at different loci. This linkage is very hard to describe mathematically. Therefore we decided to look for a recombination operator that leads to simpler equations, like those we used as an approximation. This operator must maintain the population in a Hardy-Weinberg equilibrium. More general for n loci, it must create a multinomial distribution of the genotypes. Fortunately, there is a simple recombination scheme that fulfills this condition; we call it *gene pool recombination (GPR)*.

Definition: In gene pool recombination the two “parent” alleles of an offspring are randomly chosen for each locus with replacement from the gene pool given by the parent population **selected** before. Then the offspring allele is computed using any of the standard recombination schemes for TPR.

For binary functions GPR is obviously a Bernoulli process. Let $p_i^s(t)$ be the frequency of allele 1 at locus i in the **selected** parent population. Then GPR creates offspring with allele frequency $p_i(t+1) = p_i^s(t)$ and variance $p_i(t+1)(1-p_i(t+1))$ at locus i .

In order to analyze GPR we will derive difference equations for the gene frequencies, valid for arbitrary fitness functions and infinite populations. As before, we restrict the analysis to the case of two loci and proportionate selection.

Let $q_i(t)$ be the frequency of genotype i at generation t . For $n = 2$ loci, the marginal gene frequencies $p_1(t)$ and $p_2(t)$ can be obtained from

$$p_1(t) = q_2(t) + q_3(t) \quad p_2(t) = q_1(t) + q_3(t).$$

We assume that the initial population has a binomial distribution. This means that

$$\begin{aligned} q_0(0) &= (1-p_1(0))(1-p_2(0)) \\ q_1(0) &= (1-p_1(0))p_2(0) \\ q_2(0) &= p_1(0)(1-p_2(0)) \\ q_3(0) &= p_1(0)p_2(0). \end{aligned}$$

Then the following theorem holds:

Theorem 1 *Let the initial population have a binomial distribution. For an infinite population with GPR and proportionate selection, the marginal frequencies $p_1(t)$ and $p_2(t)$ can be obtained from*

$$p_1(t+1) = p_1(t) \frac{m_2(1-p_2(t)) + m_3p_2(t)}{\bar{f}(t)} \quad (12)$$

$$p_2(t+1) = p_2(t) \frac{m_1(1-p_1(t)) + m_3p_1(t)}{\bar{f}(t)}. \quad (13)$$

The realized heritability, $b(t)$, is given by

$$b(t) = 1 - (m_0m_3 - m_1m_2)(m_0 - m_1 - m_2 + m_3) \frac{p_1p_2(1-p_1)(1-p_2)}{V\bar{f}} \quad (14)$$

where p_1, p_2, V , and \bar{f} depend on t .

Proof: Proportionate selection selects the genotypes for the parents of population $t+1$ according to

$$q_i^s(t) = \frac{m_i}{\bar{f}(t)} q_i(t).$$

From $q_i^s(t)$ the marginal frequencies $p_1^s(t)$ and $p_2^s(t)$ can be obtained from the two equations $p_1^s(t) = q_2^s(t) + q_3^s(t)$ and $p_2^s(t) = q_1^s(t) + q_3^s(t)$. For each locus, GPR is a Bernoulli process; therefore, the marginal gene frequencies of parents and offspring remain constant

$$p_1(t+1) = p_1^s(t), \quad p_2(t+1) = p_2^s(t).$$

Combining these equations gives equations (12) and (13). The expression for the realized heritability can be obtained after some manipulations.

Remark: Theorem 1 can be extended to arbitrary functions of size n , or genetically speaking to n loci. This means that the evolution of an infinite genetic population with GPR and proportionate selection is fully described by n equations for the marginal gene frequencies. In contrast, for TPR one needs 2^n equations for the genotypic frequencies. For GPR one can in principle solve the difference equations for the marginal gene frequencies instead of running a genetic algorithm.

Note that $b(t) = 1$ if $m_0m_3 = m_1m_2$ or if $m_1 + m_2 = m_0 + m_3$. Let us first consider ONEMAX(2). The average fitness is given by $\bar{f}(t) = p_1(t) + p_2(t)$. If $p_1(0) = p_2(0)$, we have $p_1(t) = p_2(t) = p(t)$ for all t . From $\bar{f}(t) = 2p(t)$ we obtain

$$R(t) = 1 - p(t) \quad (15)$$

and

$$p(t+1) = p(t) + \frac{1}{2}(1 - p(t)). \quad (16)$$

This equation is similar to the equation obtained for TPR. It can be solved easily. Both equations become equal if $B_3(t) = 0$. This shows that for linear fitness functions, GPR and TPR give similar results — with a slight advantage for GPR, which converges faster.

Let us now turn to the function MULT(2). Combining $\bar{f}(t) = (1+p(t))^2$ with equation (12), we obtain equation (11). For MULT(2) TPR and GPR lead to the same difference equation. One can show that in general for multiplicative functions ($m_0m_3 = m_1m_2$) TPR and GPR are equal.

For many loci the above analysis can easily be extended to fitness functions which are called “unitation” functions. For these functions the fitness values depend only on the number of 1’s in the genotype. Again for simplicity we consider three loci only. Let u_i denote the fitness of a genotype with i 1’s. Under the assumption that $p_1(0) = p_2(0) = p_3(0)$, all marginal frequencies have the same value, which we denote as $p(t)$. Then we obtain for the marginal frequency $p(t+1) = c \cdot p(t)$

$$c = \frac{u_1(1-p)^2 + 2u_2p(1-p) + u_3p^2}{u_0(1-p)^3 + 3u_1p(1-p)^2 + 3u_2p^2(1-p) + u_3p^3}, \quad (17)$$

where $p = p(t)$. If $c > 1$ the marginal frequency $p(t)$ increases, if $c < 1$ it decreases. As a specific example we analyze a “deceptive” function of 3 loci (see Goldberg 1989). Let the fitness values of this function be $u_0 = 28, u_1 = 26, u_3 = 30$. The global optimum is at 111, the local optimum at 000. The fitness value for u_2 will be varied. Depending on u_2 and the initial population, the genetic population will converge to 000 or 111. In figure 1 c is shown for $u_2 = 25$ and $u_2 = 0$. For $u_2 = 0$, the area where the marginal frequency p is decreasing is larger than the area where the marginal frequency is increasing. This means that the population will converge to the second optimum if $p(0) = 0.5$. For $u_2 = 25$, the population will converge to 111 if $p(0) = 0.5$.

Remark: The analysis of unitation functions of three or more loci shows that a genetic algorithm using selection and recombination only is **not a global optimization** method. Depending on the frequency distribution of the genotypes and the fitness values, a genetic algorithm with infinite population size will deterministically converge to

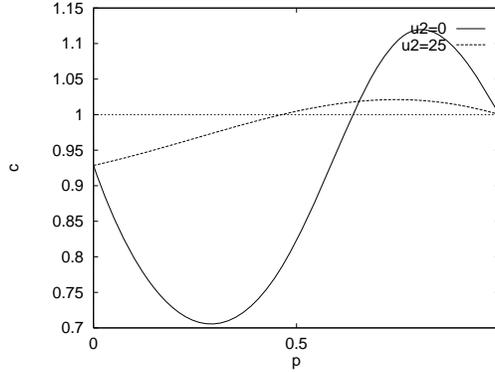


Figure 1: The line $c = 1$ divides the attractor regions 000 and 111.

one of the local optima. The equations derived in this chapter can be used to determine the optima to which the genetic algorithm will converge.

As a last example we will analyze a fitness function where the results of TPR and GPR are very different. For simplicity we take two loci. The fitness values are defined as $m_0 = 0.99, m_1 = 0, m_2 = 0$, and $m_3 = 1$.

Table 1: Results for TPR and GPR for a bimodal function

t	REC	q_0	q_1	q_3	\bar{f}	Var
0	TPR	0.250	0.250	0.250	0.4975	0.2475
1	TPR	0.372	0.125	0.378	0.7463	0.1857
2	TPR	0.369	0.125	0.381	0.7463	0.1857
3	TPR	0.365	0.125	0.385	0.7464	0.1856
9	TPR	0.287	0.121	0.471	0.7556	0.1819
0	GPR	0.250	0.250	0.250	0.4975	0.2475
1	GPR	0.247	0.250	0.252	0.4975	0.2475
2	GPR	0.242	0.250	0.257	0.4977	0.2476
3	GPR	0.233	0.250	0.268	0.4983	0.2477
6	GPR	0.120	0.226	0.427	0.5457	0.2467
9	GPR	0.000	0.006	0.988	0.9883	0.0115

Table 1 shows that GPR very slowly changes the gene frequencies at the beginning. In fact, if $m_0 = 1$ the population would stay in equilibrium. After three generations GPR changes the gene frequencies very quickly. In contrast, TPR dramatically changes the frequencies in the first generation. The population immediately goes to the equilibrium points for the symmetric fitness function $m_0 = m_3 = 1, m_1 = m_2 = 0$. It takes TPR a long time to leave this equilibrium point and march to the optimum.

Proportionate selection should not be used for a genetic algorithm, its selection intensity is far too low (Mühlenbein and Schlierkamp-Voosen 1994). Unfortunately, for other selection methods the equations for the marginal gene frequencies are difficult to obtain. For truncation selection an approximate analysis can be done by using the equation for the response to selection. The following theorem was proven by Mühlenbein and Schlierkamp-Voosen (1993) in a different context.

Theorem 2 Let $p(t)$ be the frequency of allele 1 in the population at generation t . Let the fitness have a binomial distribution, i.e. $\sigma_p(t) = \sqrt{n \cdot p(t) \cdot (1 - p(t))}$.

If the population is large enough to converge to the optimum, and if the selection intensity I is greater than 0, then the number of generations needed to reach equilibrium is approximately

$$GEN_e = \left(\frac{\pi}{2} - \arcsin(2p_0 - 1) \right) \cdot \frac{\sqrt{n}}{I}, \quad (18)$$

where $p_0 = p(0)$ denotes the probability of allele 1 in the initial population. The value $p(t)$ can be approximated as

$$p(t) \approx 0.5 \left(1 + \sin\left(\frac{I}{\sqrt{n}}t + \arcsin(2p_0 - 1)\right) \right). \quad (19)$$

Remark: The assumptions of the theorem are fulfilled for the ONEMAX(n) function. In this case the theorem is also approximately valid for TPR (Mühlenbein and Schlierkamp-Voosen 1994). For ONEMAX(n) GPR converges about 25% faster than TPR.

5. Genetic drift

The analysis of the previous sections is valid for very large populations. This leads to deterministic equations for averages. In finite populations the chance introduced by finite sampling has to be modelled. The mathematical analysis can be done in principle with Markov chains, but, unfortunately, the number of transitions scales exponentially with the number of loci and the size of the population. The analysis gets simpler if we assume no selection. This case is called *genetic drift*. The finiteness of the population causes convergence to a single genotype, even without selection. It is a result of sampling with replacement. Genetic drift is an important factor for genetic algorithms, because if the selection is too weak, then genetic drift is causing the “convergence” of the population.

Asoh and Mühlenbein (1994) have analyzed genetic drift for TPR. They showed that a genetic algorithm with TPR but without selection converges surprisingly quickly. This means that just by sampling with replacement, the variance of the population is continuously reduced. At the equilibrium the population consists of a single genotype only. For TPR the following proposition was obtained by fitting numerically obtained values:

Proposition 1 Let the number of loci be n . Let each gene have two alleles. Using TPR the mean convergence time $\tau_n(N)$ of a population of size N is approximately

$$\tau_n(N) \approx 1.4N (0.5 \ln(n) + 1.0) \quad \text{for } p(0) = 1/2. \quad (20)$$

Genetic drift for GPR is similar to a standard statistical process. Given n independent sampling processes with replacement, each process having a population N , what is the expected time for *all* n processes to converge, i.e. to consist of copies of one member only? This is a classical statistical question which can be answered for certain distributions. We state the following theorem without proof.

Table 2: Genetic drift for GPR (n loci, N popsize).

n/N	5	11	51	101	401
1	5.7	13.9	68.9	137.9	553.1
2	8.0	19.2	94.8	189.5	759.0
4	10.6	25.4	124.4	248.4	993.5
8	13.4	32.0	156.3	311.9	1247.0
16	16.4	38.9	189.5	378.0	1510.0

Theorem 3 *Let the expected mean convergence time of the process be exponentially distributed with mean $\tau = 1/\lambda$. Then the expected mean time for all n processes to converge is given by*

$$\tau_n = \frac{1}{\lambda} \sum_{\nu=1}^n \frac{1}{\nu}. \quad (21)$$

The sum can be approximated by $\ln(n) + \gamma$ with $\gamma = 0.577\dots$. By setting $1/\lambda = 1.4N$ equations 20 and 21 have the same asymptotic order. We have not been able to verify that the expected mean convergence time of GPR is exponentially distributed. Therefore we present numerical simulations obtained by Markov chains in table 2.

For $n = 1$ GPR and TPR are equal. For large N the mean time to convergence is approximately $\tau_1 = 1.39N$, as predicted by the proposition. The increase of τ with n is smaller than given by equation 21. A comparison of the numerical values for TPR presented by Asoh and Mühlenbein (1994) and table 2 shows that τ_n is only slightly larger for GPR than for TPR. This demonstrates that TPR and GPR give on the average very similar results, despite the fact that the underlying statistical processes are different.

Genetic drift is an important factor for genetic algorithms, especially in small populations. Whereas large populations converge deterministically to an equilibrium, can small populations converge to any of the many optima.

6. Conclusion

Gene pool recombination more closely resembles standard statistical process than does two-parent mating. For genetic algorithms GPR clearly separates the two stages of a search (which can be identified in any mathematical search algorithm): selecting promising areas to be searched and searching these areas in a reasonable manner. GPR searches within the selected area more reasonably than does TPR, which is more biased to the specific genotypes of the population.

GPR is mathematically more tractable than TPR. Nevertheless, the behavior is often surprisingly similar. In terms of population genetics: GPR keeps the population in a Hardy-Weinberg equilibrium, i.e., the genotypes always have a binomial distribution.

The equations obtained for GPR with infinite population clearly show that a genetic algorithm using only recombination is *not a global optimizer*. The genetic algorithm will not converge to a small isolated peak, even if it is the global optimum. Depending on the initial distribution of the genotypes, the population will deterministically converge to an area that has a large attractor region with many good fitness values. For a small population the genetic algorithm may converge to any good fitness value, just by chance.

The GPR idea can be easily extended to other problem domains. For continuous functions it directly leads to multi-parent search as described by Glover (1994). Vari-

ants of GPR have been successfully used for the traveling salesman problem. GPR liberates genetic algorithms from two-parent mating and all the difficult problems connected with it. Our theoretical analysis confirms the observation of Eshelman and Schaffer (1993): "Our hunch that the unique niche for pair-wise mating is much smaller than most GA researchers believe, and that the unique niche for two-point crossover is even smaller."

Acknowledgment: The theorem concerning genetic drift was proven by J. Bendisch. The simulations for the genetic drift were done by G. Paaß. This work is part of the SIFOGA project supported by the Real World Computing Partnership.

7. References

- Asoh H. and Mühlenbein H., On the mean convergence time of evolutionary algorithms without selection and mutation, in: *Parallel Problem Solving from Nature*, eds. Davidor Y., Schwefel H.-P. and Männer R., Lecture Notes in Computer Science 866, (Springer-Verlag, 1994) pp. 88–97.
- Crow J. F. and Kimura M., *An Introduction to Population Genetics Theory*, (Harper and Row, New York, 1970).
- Eiben A., Raue P.-E. and Ruttkay Z., Genetic algorithms with multi-parent recombination, in: *Parallel Problem Solving from Nature*, eds. Davidor Y., Schwefel H.-P. and Männer R., Lecture Notes in Computer Science 866, (Springer-Verlag, 1994) pp. 78–87.
- Eshelman L. J. and Schaffer J. D., Crossover's niche, in: *Fifth Int. Conf. on Genetic Algorithms*, ed. Forrest S., (Morgan Kaufmann, San Mateo, 1993) pp. 9–14.
- Falconer D. S., *Introduction to Quantitative Genetics*, (Longman, London, 1981).
- Glover F., Genetic algorithms and scatter search: unsuspected potentials, *Statistics and Computing* 4(1994) 131–140.
- Goldberg D., *Genetic Algorithms in Search, Optimization and Machine Learning*, (Addison-Wesley, Reading, 1989).
- Mühlenbein H., Parallel genetic algorithm, population dynamics and combinatorial optimization, in: *3rd Int. Conf. on Genetic Algorithms*, ed. Schaffer H., (Morgan Kaufmann, San Mateo, 1989) pp. 416–421.
- Mühlenbein H. and Schlierkamp-Voosen D., Predictive Models for the Breeder Genetic Algorithm I. Continuous Parameter Optimization, *Evolutionary Computation* 1(1993) 25–49.
- Mühlenbein H. and Schlierkamp-Voosen D., The science of breeding and its application to the breeder genetic algorithm, *Evolutionary Computation* 1(1994) 335–360.
- Nagaraja H., Selection differentials, in: *Encyclopedia of Statistical Science*, eds. Kotz E., Johnson N. and Read C., (Wiley, New York, 1982) pp. 334–336.
- Nagylaki T., *Introduction to Theoretical Population Genetics*, (Springer, Berlin, 1992).

Syswerda G., Simulated crossover in genetic algorithms, in: *Foundations of Genetic Algorithms 2*, ed. Whitley L. D., (Morgan Kaufmann, San Mateo, 1993) pp. 239–255.