Replacement Strategies in Steady State Genetic Algorithms: Static Environments

Jim Smith and **Frank Vavak**
Intelligent Computer System Centre
University of the West of England
Bristol England BS16 1QY
jim@ics.uwe.ac.uk

Abstract

This paper investigates the effects of a number of replacement strategies for use in steady state genetic algorithms. Some of these (deleting the oldest, worst or random members, or deletion by "Kill Tournament") are well known from the literature. The last, "conservative" replacement was developed for use in time-varying problems and combines a Replace–Oldest strategy with modified selection tournaments, where one candidate is always the oldest member of the population.

A Markov chain analysis is provided to model the expected time for a single member of the optimal class to take over finite populations. For strategies which replace the oldest member, a linear approximation is developed for the probability that it belongs to the optimal class. It is shown that under certain conditions this approximation yields a transition matrix which is identical to a strategy of deletion by binary Kill Tournament.

Predicted and simulation results confirm that without enforced elitism Replace–Random and Replace–Oldest strategies cannot guarantee takeover, and for a fixed parent selection method the speed of takeover is dramatically altered by the choice of replacement strategy.

1 Introduction

As suggested by the commonly used metaphor of "Natural Selection", the selection mechanism in Genetic Algorithms (GAs) is the driving force behind the hoped for improvement in the fitness distribution of the population. Essentially its role is to take advantage of the fitter individuals produced by the reproductive operators (recombination and mutation) and increase their relative frequency in the population, so that they are more likely to be chosen as parents during the next round of reproduction.

What distinguishes GAs from many biological models is the use of a fixed population size. This enforces a split of the selection mechanism into two phases, namely parental selection and replacement strategy. For Generational GAs (GGAs) the latter is simple: all members of the previous population are deleted, and if good solutions are to be preserved and propagated then this burden falls upon the reproductive operators. This simplifies the task of producing analyses of the expected behaviour of the algorithm using measures such as Takeover Time (e.g. [9]) and Selection Intensity (e.g. [2, 4]).

For Steady State GAs (SSGAs), there are choices of replacement strategy. De Jong and Sarma examined the time taken for a superior individual (initially occupying 10 % of the population) to take it over, when using Replace–Worst and Replace–Random strategies. They attributed the observed differences to variance in the expected lifetimes of individuals [7]. However these deterministic approaches can hide some of the effects of applying stochastic processes to finite sized populations, such as loss of the optimal solution. More recent work by Chakraborty has applied an exact probabalistic approach to takeover times for a number of selection mechanisms in both GGAs and SSGAs, using a Markov Chain analysis [5, 6]. For SSGAs this was done for Replace–Worst (as used in GENITOR [15]), Replace–Random and deletion by exponential ranking. This model is introduced in Section 2.

In Section 3 Chakraborty's results are repeated for Replace–Worst, and Replace–Random, and extended to deletion by "Kill Tournament", plus elitist variants of the last two. In Section 4 linear models are developed for the probability that the oldest member of the population belongs to the fitter class. These allow the Markov Models to be approximated for another commonly used method, namely Replace–Oldest (this will also be referred to as First In First Out = FIFO) in both its elitist and non-elitist forms. Finally an approximate model is presented for a more conservative selection mechanism introduced in [14] for dealing with non-stationary environments.

This "Conservative" mechanism consists of a FIFO replacement strategy used with modified deterministic binary tournaments for parent selection. In these modified tournaments one of the two candidates for parenthood is always the oldest member in the population. Because each member is guaranteed entry in at least one tournament, there is an implicit form of elitism, since when the fittest member of the population is also the oldest it will win both parental selection tournaments. This holds at least under selection and recombination although the best may still be lost through mutation if used.

In Section 5 theoretical and empirical results are compared in order to investigate the quality of the approximate models. All of the replacement strategies are compared on the basis of the probability of takeover, time taken to reach a range of landmarks during takeover, and the rate at which total convergence is reached. These results are then discussed in the context of function optimisation.

2 Analysis of Takeover Times

In this section the population of a Genetic Algorithm is considered to be split into classes according to fitness. Theoretical and simulation results are compared for the time taken for a single member of an optimal (i.e. fittest) class to completely take over finite sized populations of size N. Only selection is used here, i.e. a parent is chosen by tournament selection and a copy is made to put in to the population according to the replacement strategy used. These results demonstrate the ability of the selection and replacement operators to exploit fitter individuals, and also the degree to which they are prone to stochastic effects namely the loss of the fittest class.

The Markov model used here is that of [5, 6] in which the number of members of a given fitness class present in a population of size N at a time $t$is represented by the variable *X(t)*. For a GA the elements *X(0), X(1),... X(t)* form a Markov chain with *N + 1* discrete states. Further, since there is no introduction of variety due to reproduction, the states *X(t) = 0* and *X(t)* = N are absorbing, and all others are transitory. Other authors have applied similar techniques to study genetic drift in the absence of selection (e.g. [1, 10]).

For a SSGA, each iteration comprises of a single generation-replacement step, and so the number of members of any class varies by at most +/- 1. Hence, the corresponding transition matrix, P , for the Markov chain is tridiagonal. The non-zero elements of P are defined by the choice of selection and replacement strategies. The number of members of the class will increase if a member is selected to be copied and a non member replaced, and decrease if a nonmember is copied and a member replaced. Subtracting the probabilities of these two events from unity gives the probability of keeping the same number of members. Once P is thus defined, it is simple to calculate the n-step transition matrix P^n. If a starting configuration of the GA has, say, i members of the optimal class,then X(0) = i . The probability that *X(t)* = N, (i.e. that the class has taken over the population) is then given by the matrix element P^t_{iN}. In this paper the situation is considered where only one member of the optimal class is initially present, so the takeover probabilities are given by P^t_{1N}.

For binary tournament selection, with the fittest always selected, and i members of the class in the population, the probability of selecting a member of that class for copying is given by:

$$p_{select} = p(both\ candidates\ in\ class) + 2p(one\ in\ class\ and\ one\ not).$$

If both the candidates in the tournament are selected from the population at random, with replacement, then each one can be in the class with probability i/N which gives:

$$p_{select} = \frac{i^2}{N^2} + 2\frac{i}{N}\left(\frac{N-i}{N}\right) = \frac{i(2N-i)}{N^2} \tag{1}$$

This deterministic binary tournament selection can be generalised in two ways which increase and decrease the selection pressure respectively. The first is to increase the number of participants in the tournament to some value n . In this case (assuming that the fittest member wins the tournament) the probability of selecting a member of the optimal class becomes:

$$p_{select} = \sum_{j=1}^{n} \left(\frac{i}{N}\right)^j \left(\frac{N-i}{N}\right)^{n-j} \left(\begin{array}{c} n \\ j \end{array}\right) \tag{2}$$

The second generalisation, which can be used to reduce selection pressure, is to select the fittest member of the tournament with a probability $s \leq 1.0$. In this case the probability of selecting a member of the optima class becomes:

$$p_{select} = \frac{i^2}{N^2} + 2s\frac{i}{N}\left(\frac{N-i}{N}\right) \tag{3}$$

For the purposes of this paper we will mostly restrict ourselves to considering binary tournaments of the form (3) as the focus is on the effect of different replacement strategies given the same parental selection strategy.

3 Exact Models

The following replacement strategies can all be modelled exactly using the analysis above.

3.1 Replace Worst

In this case a nonmember will be replaced with probability 1.0 until takeover has occurred. Using p_{select} for the probability that a member of the optimal class is chosen for copying, the elements of P are given by:

$$p_{ij} = \begin{cases} 1 & i \in \{0, N\}, \quad j = i \\ p_{select} & i = 1, \ldots, N-1, \ j = i+1 \\ 1 - p_{select} & i = 1, \ldots, N-1, \ j = i \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

The first term notes that states X(t) = 0 and X(t) = N are absorbing, which holds for all the strategies. The second term relates to a member of the optimal class being selected for copying. Since a non-member is always replaced (unless X(t) = N) the number of members increases by one. The third term is the probability that a nonmember is selected for copying. Again a nonmember is replaced so X(t+1) = X(t).

Finally the fourth term states that the matrix is in this case bidiagonal, since the number of class members varies by at most one per selection-replacement event and for this strategy can never decrease.

3.2 Replace Random

In this case the probability of picking a member for replacement is simply i/N for the optimal class, and $(N-i)/N$ for the others. The elements of P are given by:

$$p_{ij} = \begin{cases} 1 & i \in \{0, N\}, \quad j = i \\ p_{select}\left(\frac{N-i}{N}\right) & i = 1, \ldots, N-1, \quad j = i+1 \\ (1 - p_{select})\left(\frac{i}{N}\right) & i = 1, \ldots, N-1, \quad j = i-1 \\ 1 - p_{select}\left(\frac{N-i}{N}\right) - (1 - p_{select})\left(\frac{i}{N}\right) & i = 1, \ldots, N-1, \quad j = i \\ 0 & \text{otherwise} \end{cases} \tag{5}$$

Comparing (5) with (4), the probability of increasing the number of members of the optimal class is altered to allow for the fact that the count only increases if a member is chosen for copying (p_{select}) and a non-member is replaced $(N-i)/N$.

There is an extra term since the count can now be reduced with probability $(1 - p_{select})$ [non member copied]$\times i/N$[member deleted].

The probability that the system stays in the same state is reduced to allow for this (since the terms must sum to 1.0), and the final term states the fact that the matrix is tridiagonal.

3.3 Elitist Replace Random

This replacement strategy is as above, with the proviso that if there is only one instance of the optimal class then it is never chosen for replacement. This is catered for by adding an extra set of terms to (5) for the case when i is one, so that P_{ij} beomes:

$$p_{ij} = \begin{cases} 1 & i \in \{0, N\}, \quad j = i \\ p_{select} & i = 1, \quad j = 2 \\ 1 - p_{select} & i = 1, \quad j = 1 \\ p_{select}\left(\frac{N-i}{N}\right) & i = 2, \ldots, N-1, \; j = i+1 \\ (1 - p_{select})\left(\frac{i}{N}\right) & i = 2, \ldots, N-1, \; j = i-1 \\ 1 - p_{select}\left(\frac{N-i}{N}\right) - (1 - p_{select})\left(\frac{i}{N}\right) & i = 2, \ldots, N-1, \; j = i \\ 0 & \text{otherwise} \end{cases} \tag{6}$$

3.4 Kill Tournament

With a strategy of selecting a member for replacement by a “kill tournament” a number of possibilities arise. As for the parent selection tournament, it is possible to pick a number of parents at random and replace the worst, or to pick two parents at random and replace the worse with some probability $0.5 \leq d \leq 1.0$. Other authors have suggested tournaments between the parents and their offspring to decide which should be in the next population (see [13] for a good discussion). Assuming a random choice of participants in the kill tournament, the probability p_{kill} of deleting a member of the optimal class will again be a function of i and is defined by analogous expressions to those for parent selection. For a n–way tournament with the worst always replaced, all of the participants in the tournament must be of the optimal class if one is to be replaced, i.e. $p_{kill} = (i/N)^n$. For binary kill tournaments the probability of deleting a member of the optimal class is given by:

$$\begin{aligned} p_{kill} &= \frac{i^2}{N^2} + \frac{2i(N-i)(1-d)}{N^2} \\ &= \frac{(2d-1)i^2}{N^2} + \frac{2i(1-d)}{N} \end{aligned} \tag{7}$$

The elements of P are given by:

$$p_{ij} = \begin{cases} 1 & i \in \{0, N\}, \quad j = i \\ p_{select}(1 - p_{kill}) & i = 1, \ldots, N-1, j = i+1 \\ p_{kill}(1 - p_{select}) & i = 1, \ldots, N-1, j = i-1 \\ 1 - p_{select}(1 - p_{kill}) - p_{kill}\,(1 - p_{select}) & i = 1, \ldots, N-1, j = i \\ 0 & \text{otherwise} \end{cases} \tag{8}$$

3.5 Elitist Kill Tournament

Again it is possible to implement an elitist version of this strategy, which can be modelled by adding terms to (8) for the case when i is one, yielding:

$$p_{ij} = \begin{cases} 1 & i \in \{0, N\}, \quad j = i \\ p_{select} & i = 1, \quad j = 2 \\ 1 - p_{select} & i = 1, \quad j = 1 \\ p_{select}(1 - p_{kill}) & i = 2, \ldots, N-1, \quad j = i+1 \\ p_{kill}(1 - p_{select}) & i = 2, \ldots, N-1, \quad j = i-1 \\ 1 - p_{select}(1 - p_{kill}) - p_{kill}(1 - p_{select}) & i = 2, \ldots, N-1, \quad j = i \\ 0 & \text{otherwise} \end{cases} \tag{9}$$

4 Approximate Models

For all the replacement strategies considered above the Markov models are exact,which is confirmed by comparisons of predicted values for P_{ij}^t against simulation results. However for some strategies exact models are not possible, as it is necessary to estimate the probability that the member about to be deleted belongs to the optimal class.

4.1 Replace Oldest

4.1.1 The Generic Form

For a FIFO strategy, it is necessary to estimate the probability p_{old}, that the oldest member of the population belongs to the optimal class. If the system is in one of the two absorbing states ($i \in \{0, N\}$) then this probability is constant with values 0.0 and 1.0 respectively. If the system is not in one of these states, then there will be stochastic fluctuations, but p_{old} will be some function of i. The generic form for the elements P_{ij} is:

$$p_{ij} = \begin{cases} 1 & i \in \{0, N\}, \quad j = i \\ p_{select}(1 - p_{old}) & i = 1, \ldots, N-1, j = i+1 \\ (1 - p_{select})p_{old} & i = 1, \ldots, N-1, j = i-1 \\ 1 - p_{select}(1 - p_{old}) - p_{old}(1 - p_{select}) & i = 1, \ldots, N-1, j = i \\ 0 & \text{otherwise} \end{cases} \tag{10}$$

4.1.2 A Linear Approximation for p_{old}

In order to model p_{old} in an evolving population we consider the probability $p_{class}(x)$, that an individual of age x ($0 \leq x \leq N-1$) belongs to the optimal class. The age of an individual here is taken to mean the number of individuals inserted into the population after it. The simplest model is to assume that $p_{class}(x)$ is independent of x, in which case the model is identical to that for Replace–Random (5). However simulation reveals that this is not the case, which corresponds to the intuition that if i is increasing with time (i.e. the selection/replacement mechanism is increasing the mean fitness of the population) then newer members of the population are more likely to belong to the optimal class than older members.

The model investigated in this paper makes the simple assumption that $p_{class}(x)$ decreases linearly with age x across the population, so it can be modelled as $p_{class}(x) = a + bx$, where a and b, are constants.

The value of a is $p_{class}(0)$, which by definition is the probability that the member copied at the last time step belonged to the class. Since the number in the class varies by at most one per time-step, this can be approximated using the expressions derived above for p_{select}, i.e:

$$a = p_{select}(t-1) \approx p_{select}(t) \tag{11}$$

Note that in all of these expressions the value of p_{select} is a function of i . Since there is only one individual of each age, the expected number of members of the optimal class for each age is equal to the probability $p_{class}(x)$. If we sum over all ages between 0 and N–1, the expected number of members in the population is the sum of the expected number for each age i.e. the sum of p_{class}. Recognising that this sum must equal i gives:

$$\begin{aligned} i &= \sum_{x=0}^{N-1} p_{class}(x) \\ &= Na + b\sum_{x=0}^{N-1} x \\ &= Np_{select} + bN(N-1)/2 \\ \therefore b &= \frac{2(i - Np_{select})}{N(N-1)} \end{aligned} \tag{12}$$

Substituting these values for a and b in the linear model gives the probability that the oldest member of the population belongs to the optimal class as:

$$\begin{aligned} p_{class}(N-1) &= a + b(N-1) \\ &= p_{select} + 2(i - Np_{select})/N \\ &= 2i/N - p_{select} \end{aligned} \tag{13}$$

4.1.3 Applying the Approximation

Substituting this approximation for p_{old} into (10) gives the transition matrix for this model of a Replace–Oldest strategy as:

$$p_{ij} = \begin{cases} 1 & i \in \{0, N\}, \quad j = i \\ p_{select}(1 - 2i/N + p_{select}) & i = 1, \ldots, N-1, \quad j = i+1 \\ (1 - p_{select})(2i/N - p_{select}) & i = 1, \ldots, N-1, \quad j = i-1 \\ 1 - 2p_{select}^2 + 2i(2p_{select} - 1)/N & i = 1, \ldots, N-1, \quad j = i \\ 0 & \text{otherwise} \end{cases} \tag{14}$$

Finally it should be noted that if a stochastic binary selection tournament (3) is used with a with Replace–Oldest strategy, the probability of deleting a member of the optimal class for $0 < i < N$ (13) reduces to:

$$P(deleting\, class\, member) = p_{old} = \frac{2i(1-s)}{N} + \frac{(2s-1)i^2}{N^2} \tag{15}$$

Comparing this with the equivalent probability for a stochastic binary kill tournament, (7), shows that the two are the same provided $s = d$. This shows that *provided the linear model is accurate*, when using a binary selection tournament with a probability $0.5 \leq s \leq 1.0$ of selecting the fitter, a Replace–Oldest policy is equivalent (at least in terms of takeover) to a binary Kill Tournament where the worst is deleted with the same probability.

4.2 Elitist Replace Oldest

It is possible to implement elitist versions of a Replace–Oldest strategy. Unlike the previous elitist strategies cases the Markov models cannot be exact since it is necessary to approximate the probability that the oldest member of the population belongs to the optimal class. As noted in the previous section, the linear approximation for p_{class} used here results in a probability of deleting a member of the optimal class which is the same as for binary kill tournaments, so the elitist version of Replace–Oldest will have the same form as (9).

4.3 Conservative Selection

4.3.1 Generic form

This operator was developed in [14] for use in noisy and/or non-stationary environments, and combines a FIFO replacement strategy with a modified deterministic binary tournament selection operator. In this case one of the two members of each tournament is always the oldest in the population, thus elitism is implicitly assured, at least under recombination and selection. Thus in these models this is an elitist strategy.

If the oldest member is the optimal class, it will win the tournament, so the number in the class can never decrease. The number will only increase if the oldest is out of the optimal class (which has probability $1 - p_{old}$) and the other (randomly chosen) candidate in the tournament belongs to the optimal class (which has probability i/N), which gives for the generic form:

$$p_{ij} = \begin{cases} 1 & i \in \{0, N\}, \quad j = i \\ (i/N)\,(1 - p_{old}) & i = 1, \ldots, N-1, \quad j = i+1 \\ 1 - (i/N)\,(1 - p_{old}) & i = 1, \ldots, N-1, \quad j = i \\ 0 & \text{otherwise} \end{cases} \tag{16}$$

4.3.2 Using Linear Approximation for p_{old}

Using the linear model $p_{class} = a + bx$ and binary deterministic tournament selection gives for a :

$$\begin{aligned} a &\approx p_{select} \\ &= P(both\,in\,class) + P(oldest\,in,\ random\,not) + P(random\,in,\ oldest\,not) \\ &= p_{old}\ (i/N) + p_{old}\ (1 - i/N) + (1 - p_{old})\ i/N \\ &= p_{old} + \frac{i}{N}(1 - p_{old}) \end{aligned} \tag{17}$$

Again summing expectations, subsituting for a and rearranging gives:

$$\begin{aligned} i &= Na + b\sum_{x=0}^{N-1} x \\ &= Np_{old} + i - ip_{old} + bN(N-1)/2 \\ \therefore b &= \frac{2p_{old}(i-N)}{N(N-1)} \end{aligned} \tag{18}$$

This gives the probability that the oldest member is of the optimal class as:

$$\begin{aligned} p_{old} &= p_{class}(N-1) = a + (N-1)b \\ &= p_{old} + \frac{i}{N}(1 - p_{old}) + (N-1) \times \frac{2p_{old}(i-N)}{N(N-1)} \\ &= i/(2N-i) \end{aligned} \tag{19}$$

Using these approximations the transition matrix (16) becomes:

$$p_{ij} = \begin{cases} 1 & i \in \{0, N\}, \quad j = i \\ \frac{2i(N-i)}{N(2N-i)} & i = 1, \ldots, N-1, \quad j = i+1 \\ 1 - \frac{2i(N-i)}{N(2N-i)} & i = 1, \ldots, N-1, \quad j = i \\ 0 & \text{otherwise} \end{cases} \tag{20}$$

5 Results

The probability of a single individual completely taking over populations of size $N = 50$, 100 and 500 in t time steps (i.e. P^t_{1N}) was calculated using the models above. This will be referred to as $P_{takeover}$. In all cases a binary tournament (with $s = 1$ except where explicitly stated) was used to select the individual to be copied. The Kill Tournaments deleted the worst of two randomly chosen members (i.e. $d = 1$).

Under these circumstances the approximate models developed for the two Replace–Oldest variants are the same as for Kill Tournaments, and so they were not calculated. For the Replace–Oldest and Conservative strategies results were also obtained from simulations, in order to test the validity of the approximate models. These are the mean of 1000 runs: that is to say that the results presented for the probability of takeover at time t are the proportion of runs in which the population already consisted of N members of the optimal class by that time-step. Since the effects of crossover and mutation are ignored for the purposes of this analysis, the representation of individuals becomes irrelevant, so the initial population members are assigned random fitnesses over the range 0.0-1.0, with the constraints that all fitnesses are unique (although this is not strictly necessary) and exactly one individual belongs to the optimal class with fitness 1.0.

5.1 Quality of Linear Approximation

The linear approximation for the probability that the oldest member is of the optimal class can be evaluated by comparing the modelled and simulated results for the two FIFO variants and Conservative selection. In the case of FIFO, the approximation predicts that the simulated results should match the modelled results for the appropriate variant of Kill Tournament.

Figure 1 shows the relevant takeover curves for a population of size 100. For the Replace–Oldest strategy, empirical results are shown where the initial copy of the fittest class is placed either at the end of the population (v.1), or at random (v.2). The predicted curves for Elitist Replace–Oldest are not shown as they are very close to the non-elitist version.

Examination of Figure 1 for the Conservative operator shows a close match between simulated and predicted results, although the initial rise in $P_{takeover}$ is faster in practice than

Figure 1: P(takeover) vs. time - Empirical vs. Theoretical Results

in the model. The simulated results for elitist FIFO match the predictions accurately over most of the range, but the initial rise is again slightly faster than the model, and in this case the final part of the curve is flatter in practice than predicted.

For the non-elitist FIFO results, the $P_{takeover}$ curves are similar in practice to those predicted up to a certain point, at which the simulation results stop improving, i.e. the approximation overestimates the asymptotic value for $P_{takeover}$.

In the simulations the population was usually initialised such that the instance of the optimal class was treated as the "youngest" member, mimicking the situation where a new fittest point is discovered during the search. This makes no difference to the elitist FIFO and Conservative Replacement. For non-elitist FIFO, this curve is shown in Figure 1 as v.1. Initialising so that the fittest member was placed in a random position, changes the asymptotic value for $P_{takeover}$ from 91.6% to 51.1%. This is shown in Figure 1 as v.2 By contrast the approximation reaches its maximum probability of 99.505% after 1330 evaluations.

The accuracy of the approximate linear model for p_{class} (and hence p_{old}) for elitist FIFO and Conservative Replacement suggests that the errors in the aproximation arise when the number of members of the optimal class is small. This can be explained by considering the derivatives of p_{select} with respect to i . For a deterministic binary tournament these are:

$$p_{select} = 2i/N - i^2/N^2, \qquad dp_{select}/di = 2/N - 2i/N^2, \qquad d^2p_{select}/di^2 = -2/N^2 \quad (21)$$

The first derivative is always positive except in the limit $i = N$, since the probability of selecting a member of the optimal class increases with their count. The negative sign of the second derivative shows that the rate at which this changes decreases with i. Since the value of i changes by at most one per time step, the approximation $p_{select}(t-1) \approx p_{select}(t)$ is least accurate when the value of p_{select} is changing most rapidly, i.e. at low values of i. By restricting the changes in i to positive values, the approximation becomes an inequality

$(p_{select}(t-1) \leq p_{select}(t))$ and is more accurate. This holds for elitist FIFO when $i = 1$ and for all values of i with Conservative Replacement.

5.2 Comparison of Strategies

5.2.1 Probability of Takeover

Figure 2: P(takeover) vs. time - Comparison of Strategies

In Figure 2 takeover curves (empirical where appropriate) are shown for a population of 100. Also shown is $P_{takeover}$ for a Replace–Worst strategy in combination with stochastic selection with $s = 0.6$ (for $0.6 < s < 1.0$ the curves lie between the two lines plotted). The Elitist versions for predicted Kill Tournament and empirical Replace–Oldest are not shown for the sake of clarity, as they are both very similar to non-elitist Kill-Tournament results.

The results show that changing the replacement strategy has a marked effect on the takeover time, even for the same parent selection mechanism. Replacing the worst member of the population produces very fast takeover by copies of the fittest member of the population, such that even if the parent selection pressure is drastically reduced (i.e. $s = 0.6$) the takeover still occurs faster than for the other replacement methods. Although it is not shown in Figure 2 for the sake of clarity, the curve was plotted for Replace–Worst with *no* parental selection pressure (i.e. $s = 0.5$). This showed very similar performance to Kill Tournaments and the empirical results for elitist FIFO, and was markedly faster than elitist Replace–Random.

As noted above, the simulations of FIFO with elitism exhibited behaviour which matched closely that of deterministic binary Kill Tournaments, as predicted. Adding elitism to the Kill Tournaments had very little effect. All of these strategies exhibit takeover of the population by the optimal class, although less rapidly than Replace-Worst. Replacing the oldest with the modified selection tournament (Conservative Replacement) also guarantees

Algorithm		Pop. 50		Pop. 100		Pop. 500	
		50%	99%	50%	99%	50%	99%
Worst		121	225	280	490	1816	2871
Random		633	–	1493	–	10000	–
Random	Elitist	420	775	990	1702	6609	10172
Kill	Tournament	216	468	505	822	3337	4747
Kill Tourn.	Elitist	216	357	504	783	3337	4719
Oldest	(Simulation)	214	–	490	–	3102	–
Oldest	Elitist (Sim.)	214	401	490	856	3101	5038
Conservative	Model	307	529	721	1166	4821	7061
	(Simulation)	306	517	693	902	4463	6560

Table 1: Time to Reach 50% and 99% Takeover Probability

takeover, although at a slower rate, and elitist Replace–Random is yet slower.

In contrast, with random initialisation, replacing the oldest or a random member without elitism led to a loss of the optimal value in 49% of the runs performed, regardless of the initial population size. This effect has been noted by other authors e.g. [11]. The position in the population makes no difference to a Replace–Random strategy, and even with the population initialised so that the fittest member was the "youngest", 9% of the Replace–Oldest runs lost the fittest member before it was copied and could multiply.

5.2.2 Speed of Takeover

In Table 1 the times taken for $P_{takeover}$ to reach 50% and 99% are given for both the models and the simulations for population sizes of 50, 100 and 500.

The figures in Table 1 demonstrate that for the landmarks given, a Replace-Worst policy will meet them in approximately half the time taken by most of the other strategies, and roughly a third of the time taken by the Random–Elitist policy.

The Replace–Worst, Kill Tournament, Elitist and Conservative strategies always achieve takeover, but the time-scales in which this happens are very different. As has been noted by a number of authors e.g. [9, 6], the Replace–Worst strategy exhibits a much harder "push" towards takeover, and does so very early on. This is likely to lead to premature convergence since a smaller number of points on the search space will have been sampled, and is one of the reasons for the common use of large populations with the GENITOR algorithm. A related effect has also been noted in the context of self-adaptation, e.g. in [12] where it was found that replacing the worst led to premature decrease in mutation rates and stagnation of the search.

5.2.3 Rate of Approaching Convergence

It should be noted that the values of $P_{takeover}$ for the models are given to a finite degree of precision. For parental selection with replacement, (as here) the values only approach 1.0 asymptotically. This is because when the final member of the non-optimal class is chosen for replacement, there is a finite non-zero probability that both candidates in the tournament will be the non-optimal member.

The rate at which $P_{takeover}$ approaches 1.0 can be explained by considering two factors which affect the time taken to replace the last inferior member. The first factor is the expected waiting time between selections of the individual for replacement. This is a single time step if the worst is deleted, and exactly N time steps for FIFO and Conservative strategies. For Replace–Random and Kill-Tournaments, the waiting time comes from a random distribution with mean N (but with an infinitely long right tail). The second factor is the probability that the member being replaced will be selected twice for the tournament. The latter probability is $1/N$ for the conservative operator, which is larger than the value of $1/N^2$ for the others.

In short, once there is a single inferior individual left in the population, the probability that it will still be present decreases for each generation (N time steps) by a factor of $1/N^{2N}$ for Replace–Worst, $1/N^2$ for FIFO, Kill-Tournaments and Replace–Random and $1/N$ for Conservative. This is a further demonstration of the differing selection pressures exhibited by the different strategies.

5.3 Discussion

The results demonstrate an important facet of search in SSGAs, namely that large variances in performance can arise from the nature of the replacement algorithms.

For randomly initialised FIFO and Replace–Random strategies variations in performance will arise from the fact that with 49% probability a sole instance of the fittest population member will be lost from the population. This puts an increased emphasis on the ability of the reproductive operators to rediscover good points. One attempt to tackle this problem can be seen in the CHC algorithm, where the task of preserving fit individuals is left to a specifically elitist algorithm [8]. The results presented here demonstrate that increasing the population size has no effect on the likelihood that a new "fittest" individual will be lost before it is copied.

Another source of variation is the rapid increase in the probability of takeover for Replace–Worst for values of s greater than 0.5. This means that frequently only a relatively small proportion of the search space will have been searched. The performance achieved will therefore depend greatly on the (random) choices of initial population and bits mutated during the early stages of the search.

In practice, the test runs with Replace–Worst showed $P_{takeover} > 0.99$ after a number of evaluations which was between 5 and 6 times the population size. This number will also include a number of re-evaluations, and so it can readily be seen why users of GENITOR-like algorithms often use large population sizes, prevention of incest or prohibition of duplicate copies in order to preserve diversity and delay takeover.

By contrast the conservative strategy displays guaranteed takeover (in the absence of mutation) by the best individual discovered during the search, whilst showing a much later "push" than either Replace–Worst or FIFO. The implicit elitism effect will reduce a source of variance in performance due to loss of good solutions, and the delayed takeover means that a larger proportion of the search space will be sampled before the algorithm converges. This will reduce variance arising from the initial choice of population.

6 Conclusions

Selection is a vital force in any evolutionary algorithm, and an understanding of the nature of its effects is necessary if effective algorithms are to be developed. For Generational Genetic Algorithms selection has been well studied, and methods developed which reduce much of the noise inherent in the stochastic algorithm e.g. Baker's SUS [3]. Unfortunately the very nature of Steady State Genetic Algorithms precludes the use of such methods and those available are inherently more noisy. In this paper five selection strategies have been studied using a Markov Chain analysis of the takeover probability vs. time, in order to understand the different sources of the noise.

For Replace–Oldest and Replace–Random strategies, the variations in performance arise in part from losing the only copy of the current best in the population, which happens approximately 50% of the time. The extent to which this will affect the quality of the solutions obtained in a "real" Genetic Algorithm will depend in part on the ability of the reproductive operators to rediscover the lost class. This will in turn depend on the match between the search landscape and the probability distribution functions induced by the action of the reproductive operators on the population.

A common way of avoiding the problem of losing the best member of the population is to incorporate elitism - usually in the form of a Replace–Worst replacement strategy. As has been documented by other authors, this causes an increased selection pressure which can lead to premature convergence. The results obtained here show that for a selection only algorithm, takeover will occur after a number of evaluations which is only 5-6 times the original population size.

By comparison the fifth strategy investigated - "conservative" selection - suffers from neither of these problems. It exhibits takeover by the optimal class, but over a longer timescale, and so is likely to sample a larger proportion of the search space before convergence.

Development of the transition matrices for the Markov model used involved a linear approximation for the probability that the oldest member of a population belongs to the optimal class. The results obtained show that this was surprisingly accurate for Conservative replacement and Replace–Oldest with elitism, although it over estimated the asymptotic probability of takeover for the non–elitist version.

One consequence is that given this approximation, when using binary tournament selection, a Replace–Oldest strategy is equivalent to replacement by Kill Tournament provided the parameters of the two tournaments are the same.

References

[1] Hideki Asoh and Heinz Mühlenbein. On the mean convergence time of evolutionary algorithms without selection and mutation. In Yuval Davidor, editor, *Proceedings of the Third Conference on Parallel Problem Solving from Nature*, pages 88–97. Springer Verlag, 1994.

[2] Thomas Bäck. Generalised convergence models for tournament and (μ, λ) selection. In Larry J. Eshelman, editor, *Proceedings of the Sixth International Conference on Genetic Algorithms*, pages 2–8. Morgan Kaufmann, 1995.

[3] James E. Baker. Reducing bias and inefficiency in the selection algorithm. In John J. Grefenstette, editor, *Proceedings of the Second International Conference on Genetic Algorithms*, pages 14–21. Lawrence Erlbaum, 1987.

[4] Tobias Blickle and Lothar Thiele. A comparison of selection schemes used in genetic algorithms. Technical Report TIK Report 11, December 1995, Computer Engineering and Communication Networks Lab, Swiss Federal Institute of Techmology, 1995.

[5] Uday Kumar Chakraborty. An analysis of selection in generational and steady state genetic algorithms. In *Proceedings of the National Conference on Molecular Electronics*. NERIST (A.P.) India, 1995.

[6] Uday Kumar Chakraborty, Kalyanmoy Deb, and Mandira Chakraborty. Analysis of selection algorithms: A Markov Chain aproach. *Evolutionary Computation*, 4(2):133–167, 1997.

[7] Kenneth De Jong and Jayshree Sarma. Generation gaps revisited. In L. Darrell Whitley, editor, *Foundations of Genetic Algorithms 2*, pages 19–28. Morgan Kaufmann, 1992.

[8] Larry J. Eshelman. The CHC adaptive search algorithm: how to have safe search when engaging in non-traditional genetic recombination. In G. Rawlins, editor, *Foundations of Genetic Algorithms*, pages 263–283. Morgan Kaufmann, 1990.

[9] David E. Goldberg and Kalyanmoy Deb. A comparative analysis of selection schemes used in genetic algorithms. In G. Rawlins, editor, *Foundations of Genetic Algorithms*, pages 69–93. Morgan Kaufmann, 1991.

[10] David E. Goldberg and Philip Segrest. Finite Markov chain analysis of genetic algorithms. In J. J. Grefenstette, editor, *Proceedings of the Second International Conference on Genetic Algorithms*, pages 1–8, Hillsdale, New Jersey, 1987. Lawrence Erlbaum Associates.

[11] Peter J. B. Hancock. An empirical comparison of selection methods in evolutionary algorithms. In Fogarty, editor, *Evolutionary Computing : Proceedings of the 1994 AISB workshop*, pages 80–94. Springer Verlag, 1994.

[12] Jim Smith and T.C. Fogarty. Self adaptation of mutation rates in a steady state genetic algorithm. In *Proceedings of the Third IEEE International Conference on Evolutionary Computing*, pages 318–323. IEEE Press, 1996.

[13] D. Thierens. Selection schemes, elitist recombination and selection intensity. In Thomas Bäck, editor, *Proceedings of the Seventh International Conference on Genetic Algorithms*, pages 152–159. Morgan Kaufmann, 1997.

[14] Frantisek Vavak. *Genetic Algorithm Based Self-Adaptive Techniques for Direct Load Balancing in Nonstationary Environments*. PhD thesis, University of the West of England, 1997.

[15] Daryl Whitley and J. Kauth. Genitor: A different genetic algorithm. In *Proceedings of the Rocky Mountain Conference on Artificial Intelligence*, pages 118–130, 1988.