Probabilistic Crowding: Deterministic Crowding with Probabilistic Replacement

Ole J. Mengshoel
Department of Computer Science
University of Illinois
Urbana, IL 61801
mengshoe@cs.uiuc.edu
(217) 244-1640

David E. Goldberg
Department of General Engineering
University of Illinois
Urbana, IL 61801
deg@illigal.ge.uiuc.edu
(217) 333-0897

Abstract

This paper presents a novel niching algorithm, probabilistic crowding. Like its predecessor deterministic crowding, probabilistic crowding is fast, simple, and requires no parameters beyond that of the classical GA. In probabilistic crowding, subpopulations are maintained reliably, and we analyze and predict how this maintenance takes place.

This paper also identifies probabilistic crowding as a member of a family of algorithms, which we call integrated tournament algorithms. Integrated tournament algorithms also include deterministic crowding, restricted tournament selection, elitist recombination, parallel recombinative simulated annealing, the Metropolis algorithm, and simulated annealing.

1 INTRODUCTION

The two main objectives of niching algorithms are (i) to converge to multiple, highly fit, and significantly different solutions, and (ii) to slow down convergence in cases where only one solution is required. Different algorithms have been developed to fulfil these objectives [Goldberg and Richardson, 1987] [Harik, 1995] [Mahfoud, 1995]. One of these algorithms is known as deterministic crowding [Mahfoud, 1995]. Strengths of deterministic crowding are that it is simple, fast, and requires no parameters in addition to those of a classical GA. Deterministic crowding has also been found to work well on test functions as well as in applications. However, deterministic crowding also has some weak points. There is a lack of analysis of convergence; as a result it is not entirely clear what deterministic crowding computes. Considering the internal workings of the algorithm, the main problem appears to be that there is no restorative pressure—species of higher fitness tend to win over species of lower fitness—thus niches may get lost even though they should not be according to their fitness.

This paper introduces a new niching algorithm, *probabilistic crowding*. As the name suggests, probabilistic crowding is an offspring of deterministic crowding, and as such inherits many of its pleasant characteristics. The main difference is the use of a probabilistic rather than a deterministic acceptance function. No longer do stronger individuals always win over weaker individuals, they win proportionally according to their fitness, thus we get a restorative pressure. Using a probabilistic acceptance function is shown to give stable, predictable convergence according to the niching rule, a gold standard for niching algorithms. We show this both analytically and experimentally.

A second purpose of this paper is to briefly review the family of algorithms to which both deterministic and probabilistic crowding belongs, *integrated tournament algorithms*. Other members of this class are restricted tournament selection [Harik, 1995], elitist recombination [Thierens and Goldberg, 1994], parallel recombinative simulated annealing [Mahfoud and Goldberg, 1995], the Metropolis algorithm [Metropolis et al., 1953], and simulated annealing [Kirkpatrick et al., 1983]. Common to these algorithms is that competition is localized and occurs between what we might call a family of similar individuals. It turns out that slight variations in how the family is formed is crucial to whether one obtains a niching algorithm or not, and more generally this class of algorithms is interesting because it is very efficient and gives a wide spectrum of functionality which can be attained by changing a few parameters.

The rest of this paper is organized as follows. Section 2 presents integrated tournament algorithms. Section 3 introduces the probabilistic crowding algorithm. In Section 4, we analyze certain variants of probabilistic crowding. Section 5 and 6 gives empirical evidence that probabilistic crowding works well, while Section 7 concludes and points out directions for future research.

2 INTEGRATED TOURNAMENT ALGORITHMS

In traditional GAs, mutation and recombination is done first, and then selection (or replacement) is performed second, without regard to similarity between individuals. Many algorithms, such as probabilistic crowding, deterministic crowding, parallel recombinative simulated annealing, restricted tournament selection, the Metropolis algorithm, and simulated annealing work differently, although this distinction has not always been clearly expressed in the literature. What these algorithms, which we shall call integrated tournament algorithms, have in common is that the processes of mutation, recombination, and replacement are all integrated. Intuitively, integrated tournament algorithms can give niching through local tournaments: Similar individuals compete for spots in the population, and fit individuals replace those that are less fit, at least probabilistically. The exactly nature of the tournament depends on the algorithm, and is a crucial factor in deciding whether we get a niching algorithm or not. For instance, elitist recombination [Thierens and Goldberg, 1994] is an integrated tournament algorithms, but it is not a niching algorithm.

An early integrated tournament algorithm is the Metropolis algorithm, which originated in physics [Metropolis et al., 1953], and consists of generation and acceptance steps [Neal, 1993]. In the generation step, a new state (or individual) is generated from an existing state; in the acceptance step, the new state is accepted or rejected with some probability. Two common acceptance probability distributions are the Metropolis and the Boltzmann distributions. The Boltzmann distribution is

$$\Pr(E_j) = \frac{\exp(-E_j/T)}{\exp(-E_j/T) + \exp(-E_i/T)}, \qquad (1)$$

where E_i and E_j are the energies of the old and new states (individuals) respectively.

Simulated annealing is essentially the Metropolis algorithm with temperature added. The temperature controls the probability of accepting a higher-energy (less fit) state (individual). At high temperature, this probability is very high, but it decreases with the temperature. Simulated annealing consists of iterating the Metropolis algorithm at successively lower temperatures, and this way it finds an estimate of the global optimum [Kirkpatrick et al., 1983] [Laarhoven and Aarts, 1987]. Both the Metropolis rule and the Boltzmann rule achieve the Boltzmann distribution

$$\Pr(E_i) = \frac{\exp(-E_i/T)}{\sum_j \exp(-E_j/T)}, \qquad (2)$$

where $\Pr(E_i)$ is the probability of having a state i with energy E_i at equilibrium, T is temperature. If cooling is slow enough, one is guaranteed to find the optimum.

Within the field of genetic algorithms proper, an early integrated tournament approach is preselection. Cavicchio introduced preselection, in which a child replaces an inferior parent [Goldberg, 1989]. DeJong turned preselection into crowding [DeJong, 1975]. In crowding, an individual is compared to a randomly drawn subpopulation of c members, and the most similar member among the c is replaced. Good results with $c = 2$ and $c = 3$ were reported by DeJong on multimodal functions.

In order to integrate simulated annealing and genetic algorithms, the notion of Boltzmann tournament selection was introduced [Goldberg, 1990]. Two motivations for Boltzmann tournament selection were asymptotic convergence (as in simulated annealing) and providing a niching mechanism. The Boltzmann (or logistic) acceptance rule, shown in Equation 1, was used. Boltzmann tournament selection was the basis for parallel recombinative simulated annealing (PRSA) [Mahfoud and Goldberg, 1995]. PRSA also used Boltzmann acceptance, and introduced the following two rules for handling children and parents: (i) In double acceptance and rejection, both parents compete against both children. (ii) In single acceptance and rejection, each parent competes against a pre-determined child in two distinct competitions. Like simulated annealing, PRSA uses a cooling schedule. Both mutation and crossover are used, to guarantee convergence to the Boltzmann distribution at equilibrium. Three different variants of PRSA were tested empirically with good results, two of these have proofs of global convergence. Deterministic crowding [Mahfoud, 1995] is similar to PRSA. Differences are that deterministic crowding matches up parents and children by minimizing some distance measure, and it uses the deterministic acceptance rule of always picking the best fit individual in each parent and child pair.

Another integrated tournament algorithm is the gene-invariant GA (GIGA). In GIGA, children replace the parents [Culberson, 1992]. Parents are selected, a family constructed, children selected, and parents replaced. Family construction amounts to creating a set of pairs of children, and from this set one pair is picked according to some criterion, such as highest average fitness or highest maximal fitness. The genetic invariance principle is that the distribution over any one position on the gene does not change over time. GIGA with no mutation obeys the genetic invariance principle, so the genetic material of the initial population is retained. In addition to selection pressure provided by selection of better child pairs in a family, there is selection pressure due to sorting of the population combined with selection of adjacent individuals.

Restricted tournament selection is another integrated tournament algorithm [Harik, 1995]. The approach is a modification of standard tournament selection, based on local competition. Two individuals $\mathbf{x}$ and $\mathbf{y}$ are picked, and crossover and mutation is performed in the usual way, creating new individuals $\mathbf{x}'$ and $\mathbf{y}'$. Then w individuals are randomly chosen for $\mathbf{x}'$, and among these the closest one, $\mathbf{x}''$, competes with $\mathbf{x}'$ for a spot in the new population. A similar procedure is applied to $\mathbf{y}'$. The parameter w is called the window size. The window size is set to be a multiple of s, the number of peaks to be found: $w = c \times s$, where c is a constant. Restricted tournament selection illustrates that integrated tournament algorithms only need to be have their operations conceptually integrated; the key point is that individuals compete locally (with similar individuals) for a spot in the population.

In summary, important dimensions of integrated tournament algorithms are the form of the acceptance rule, whether temperature is used, which operators are used, and whether the algorithm gives niching or not. The importance of the distinction between probabilistic and deterministic acceptance is as follows. It seems easier to maintain a diverse population with probabilistic acceptance, and this is the goal of niching algorithms. Processes similar to probabilistic acceptance occur elsewhere in nature, for instance in chemical reactions and in statistical mechanics.

Concerning operators, one important distinction is whether similar individuals are brought together to compete implicitly or explicitly. The *implicit* approach, of which PRSA, deterministic crowding, probabilistic crowding are examples, integrate the operations of variation and selection. The *explicit* approach, examples of which are crowding and restricted tournament selection, search for similar individuals in the population. So in addition to variation and selection, there is a search step. Note that explicit versus implicit is a matter of degree, since even deterministic crowding with crossover searches for a given child for the closest among the parents. Whether the integrated tournament algorithm gives niching or not depends on the nature of the family competition. If the family competition is based on similarity, such that two or more similar individuals compete for a place in the population, the result is niching, else no niching is obtained. For example, deterministic crowding, restricted tournament selection, and probabilistic crowding are niching algorithms, while elitist recombination and GIGA are not.

3 PROBABILISTIC CROWDING ALGORITHM

Probabilistic crowding is based on the deterministic crowding algorithm [Mahfoud, 1995]; the two main differences being (i) the probabilistic acceptance rule and (ii) the fact that we have a variant without crossover at all.

Let $\mathbf{x}$ and $\mathbf{y}$ be two similar individuals that have been picked to compete to replace one of these two individuals in the next generation. Similarity comes about implicitly, when mutation only is employed, or explicitly, by using a distance measure in connection with crossover or explicit search for family members. In probabilistic crowding, $\mathbf{x}$ and $\mathbf{y}$ compete in a probabilistic tournament. The probability of $\mathbf{x}$ winning is given by:

$$p_{\mathbf{x}} = p(\mathbf{x}) = \frac{f(\mathbf{x})}{f(\mathbf{x}) + f(\mathbf{y})}, \tag{3}$$

where f is the fitness (or objective) function. Notice that probabilistic crowding is primarily a distance-based niching algorithm, since competition occurs within families, between similar individuals. Here, the family consists of two members $\mathbf{x}$ and $\mathbf{y}$, but this can easily be generalized to larger families.

Three variants of the probabilistic crowding algorithm have been investigated: Variant M (with mutation only), Variant M&C (with mutation and crossover), and Variant C (with crossover only). These variants differ in the way in which one attains $\mathbf{x}$ and $\mathbf{y}$; and also when crossover is used a measure of distance is needed.

One of the most important questions to ask about an integrated tournament algorithm is what the characteristics of its steady-state (equilibrium) distribution are. In particular, we are interested in this for niches. A niche is a set of fitness function values that have the same local optimum under some local search algorithm; see [Mahfoud, 1995] for details. The notation $\mathbf{x} \in \mathbb{X}$ will be used to indicate that individual $\mathbf{x}$ is a member of niche $\mathbb{X}$. Let q be the number of niches, and $\mathbb{X}_i$ the i-th niche. The *niching rule*

$$N_i = \frac{f_i}{\sum_{j=1}^{q} f_j} \tag{4}$$

gives allocation of N_i individuals to $\mathbb{X}_i$. Here, f_j is a measure of fitness in niche $\mathbb{X}_j$, for example fitness of best fit, average fitness, or fitness sum. The niching rule, which can be derived from the sharing rule [Goldberg and Richardson, 1987], is considered a gold standard for niching algorithms. In the following we will see how probabilistic crowding gives this rule as a special case.

4 ANALYSIS OF PROBABILISTIC CROWDING

We analyze probabilistic crowding, first the special case with two niches, second the more general case with several niches. Two kinds of analyses are provided: at steady state and of the form of convergence of the population. We assume some variation operator, which typically would be mutation or crossover. In the analysis we consider one representative per niche; for example if the niche is $\mathbb{X}$, the representative is $\mathbf{x}$. We perform a deterministic analysis, thus focusing on the mean in the stochastic processing of a GA.

4.1 TWO NICHES, SAME JUMP PROBABILITIES

Suppose we have a variation operator that results in two types of jumps; short jumps and long jumps. When an individual is treated with a short jump it stays within its niche, when it is treated with a long jump it jumps to some other niche. The probabilities are p_s and p_l respectively, and $p_s + p_l = 1$. That is, we either jump short or long.

Consider mother $\mathbf{m}$ (individual before variation operator was applied) and daughter $\mathbf{d}$ (individual after variation operator was applied). Suppose we have niches $\mathbb{X}$ and $\mathbb{Y}$, and think about how $\mathbb{X}$ can gain individuals from one generation to the next. (i) The first possibility is $\mathbf{m} \in \mathbb{X}$. The first case is that the daughter $\mathbf{d}$ stays in the mother $\mathbf{m}$'s niche if a short jump is made; in this case it doesn't matter whether $\mathbf{m}$ or $\mathbf{d}$ win since both are in the same niche. The second case is that the daughter jumps and loses. (The case where the daughter jumps and wins is a loss for $\mathbb{X}$, and is not participating in the difference equation below.) (ii) The second possibility is that $\mathbf{m} \in \mathbb{Y}$. Now, gain for niche $\mathbb{X}$ happens when the daughter jumps to $\mathbb{X}$ and wins.

The above argument can be formalized in a difference equation. Let the proportion of individuals in niche $\mathbb{Z}$ at generation t be $Z(t)$. By assumption we have two niches, $\mathbb{X}$ and $\mathbb{Y}$, and the proportions of interest at time t are denoted $X(t)$ and $Y(t)$ respectively. Note that $X(t) + Y(t) = 1$ for any t. This gives rise to the following difference equation

$$\begin{aligned} X(t+1) &= p_s X(t) + p_l p_{\mathbf{x}} X(t) + p_l p_{\mathbf{x}} Y(t) \quad (5) \\ &= X(t) - p_l X(t) + p_l p_{\mathbf{x}}. \end{aligned}$$

We will solve this equation in two ways—considering the steady state and getting a closed form formula. At steady state we have $X(t+1) = X(t) = X_{ss}$, substituting this into Equation 5 leads to

$$\begin{aligned} X_{ss} &= X_{ss} - p_l X_{ss} + p_l p_{\mathbf{x}} \quad (6) \\ &= p_{\mathbf{x}} \\ &= \frac{f(\mathbf{x})}{f(\mathbf{x}) + f(\mathbf{y})}, \end{aligned}$$

where $\mathbf{x} \in \mathbb{X}$, $\mathbf{y} \in \mathbb{Y}$. In words, we get the niching rule of Equation 4 at steady state, as one would hope for.

Now we turn to obtaining a closed form formula. Considering two niche proportions $X(t)$ and $Y(t)$, we have

$$\begin{aligned} X(t+1) &= p_s X(t) + p_l p_{\mathbf{x}} X(t) + p_l p_{\mathbf{x}} Y(t) \quad (7) \\ Y(t+1) &= p_s Y(t) + p_l p_{\mathbf{y}} Y(t) + p_l p_{\mathbf{y}} X(t) \end{aligned}$$

The solution to the above system of difference equations can be written as:

$$\begin{aligned} X(t) &= p_{\mathbf{x}} + p_s^t X(0) - p_s^t p_{\mathbf{x}} X(0) - p_s^t p_{\mathbf{x}} Y(0) (8) \\ Y(t) &= p_{\mathbf{y}} - p_s^t X(0) + p_s^t p_{\mathbf{x}} X(0) + p_s^t p_{\mathbf{x}} Y(0), \end{aligned}$$

and we see how as t tends to infinity we get the niching rule, expressed as $p_{\mathbf{x}}$ and $p_{\mathbf{y}}$, for both niches.

For illustration, suppose $X(0) = Y(0) = \frac{1}{2}$ in the initial population. This gives solutions:

$$\begin{aligned} Y(t) &= p_{\mathbf{y}} + (\frac{1}{2} - p_{\mathbf{y}}) p_s^t \quad (9) \\ X(t) &= p_{\mathbf{x}} + (\frac{1}{2} - p_{\mathbf{x}}) p_s^t, \end{aligned}$$

and again we see how we get the desired result as t tends to infinity. Also note that a smaller p_s gives faster convergence to the niching rule proportion. In summary, we see that initialization does not affect the fact that the niching rule is achieved in the limit.

4.2 TWO NICHES, DIFFERENT JUMP PROBABILITIES

Here we relax the assumption of equal jump probabilities for the two niches. Rather than jump probabilities p_s and p_l, we have jump probabilities p_{ij} for jumping from niche $\mathbb{X}_i$ to niche $\mathbb{X}_j$, where $i, j \in \{0, 1\}$. We also use the notation $X_i(t)$ for the proportion of individuals in niche $\mathbb{X}_i$ at time t. The facts $p_{11} + p_{12} = 1$ and $p_{21} + p_{22} = 1$ are used below, too.

We obtain an expression for $X_1(t+1)$ using reasoning similar to that used for Equation 5. At steady state we have $X_1(t+1) = X_1(t) = X_1$, leading to

$$X_1 = p_{11} X_1 + p_{\mathbf{x}} X_1 - p_{11} p_{\mathbf{x}} X_1 - p_{21} p_{\mathbf{x}} X_1 + p_{21} p_{\mathbf{x}}$$

which after some manipulation simplifies to the following allocation ratio for niche $\mathbb{X}_1$

$$X_1 = \frac{p_{\mathbf{x}_1}}{p_{\mathbf{x}_1} + \frac{p_{12}}{p_{21}} p_{\mathbf{x}_2}} = \frac{p_{\mathbf{x}_1}}{p_{\mathbf{x}_1} + \rho_{12} p_{\mathbf{x}_2}}. \tag{10}$$

Here, $\rho_{12} = \frac{p_{12}}{p_{21}}$ is denoted the transmission ratio from $\mathbb{X}_1$ to $\mathbb{X}_2$. In general, we have that ρ_{ij} is the transmission ratio from niche $\mathbb{X}_i$ to $\mathbb{X}_j$. Clearly, ρ_{12} is large if the in-flow into $\mathbb{X}_2$ is large relative to the out-flow from $\mathbb{X}_2$. We may compare Equation 10 to the niching rule proportion N_1 according to Equation 4:

$$N_1 = \frac{p_{\mathbf{x}_1}}{p_{\mathbf{x}_1} + p_{\mathbf{x}_2}}. \tag{11}$$

Obviously, letting $\rho_{12} = 1$ in Equation 10 gives the same allocation as Equation 11. $\rho_{12} > 1$ means that niche $\mathbb{X}_2$ will have a larger subpopulation at equilibrium than under perfect niching, giving $\mathbb{X}_1$ a smaller subpopulation, while $\rho_{12} < 1$ means that $\mathbb{X}_2$'s subpopulation at equilibrium will be smaller than under perfect niching, giving $\mathbb{X}_1$ a larger subpopulation.

The size of a niche as well as the operators used will have an impact on p_{12} and p_{21}. Disregarding the effect of operators, a large $\mathbb{X}_2$ niche will have $\rho_{12} > 1$ and therefore give a smaller subpopulation for $\mathbb{X}_1$. Likewise, a small $\mathbb{X}_2$ niche will have $\rho_{12} < 1$ and thus a larger subpopulation for $\mathbb{X}_1$.

Along similar lines, the ratio for niche $\mathbb{X}_2$ turns out to be

$$X_2 = \frac{p_{\mathbf{x}_2}}{\frac{p_{21}}{p_{12}} p_{\mathbf{x}_1} + p_{\mathbf{x}_2}} = \frac{p_{\mathbf{x}_2}}{\rho_{21} p_{\mathbf{x}_1} + p_{\mathbf{x}_2}},$$

with $\rho_{21} = \frac{p_{21}}{p_{12}}$.

Finally, note that the same result can be established by solving these two simultaneous difference equations:

$$\begin{aligned} X_1(t+1) &= p_{11}X_1(t) + p_{12}p_{\mathbf{x}}X_1(t) + p_{21}p_{\mathbf{x}}X_2(t) \\ X_2(t+1) &= p_{22}X_2(t) + p_{21}p_Y X_2(t) + p_{12}p_Y X_1(t), \end{aligned}$$

which yields fairly complex solutions which can be solved by eliminating all terms with generation t in the exponent. These solutions can then be simplified, giving exactly the same result as above.

4.3 MULTIPLE NICHES, DIFFERENT JUMP PROBABILITIES

Here we generalize from two to q niches. Let the probability of transfer from the i-th to j-th niche under the variation operator be p_{ij}, where $\sum_{j=1}^{q} p_{ij} = 1$. The probability of the individual $\mathbf{x}_i \in \mathbb{X}_i$ winning over the individual $\mathbf{x}_j \in \mathbb{X}_j$ is

$$p^*_{ij} = \frac{f(\mathbf{x}_i)}{f(\mathbf{x}_i) + f(\mathbf{x}_j)}.$$

We can now set up the following system of i difference equations:

$$X_i(t+1) = \sum_j p_{ij} p^*_{ij} X_i(t) + \sum_j p_{ji} p^*_{ij} X_j(t)$$

Unfortunately, these equations are hard to solve. But by introducing the assumption of local balance (known as detailed balance in physics [Laarhoven and Aarts, 1987]), progress can be made. The condition is [Neal, 1993, p. 37]

$$X_i p_{ij} p^*_{ji} = X_j p_{ji} p^*_{ij}. \tag{12}$$

The local balance assumption is that individuals (or states) are in equilibrium: The probability of an individual $\mathbf{x}_i$ being transformed into another individual $\mathbf{x}_j$ is the same as the probability of the second individual $\mathbf{x}_j$ being transformed into the first individual $\mathbf{x}_i$. We can assume this is for a niche rather than for an individual, similar to what we did above, thus giving Equation 12. On the left-hand side of Equation 12 we have the probability of escaping niche $\mathbb{X}_i$, on the right-hand side of Equation 12 we have the probability of escaping niche $\mathbb{X}_j$. Simple rearrangement gives

$$X_i = \frac{p_{ji} p^*_{ij}}{p_{ij} p^*_{ji}} X_j = \rho_{ji} \frac{f(\mathbf{x}_i)}{f(\mathbf{x}_j)} X_j, \tag{13}$$

where $\mathbf{x}_i$ and $\mathbf{x}_j$ are representatives for niches $\mathbb{X}_i$ and $\mathbb{X}_j$ respectively. Now we introduce $X_1 + \cdots + X_q = 1$. Consider the k-th niche, and express all other niches, using Equation 13, in terms of this niche. After some manipulation, this gives

$$X_k = \frac{f(\mathbf{x}_k)}{\sum_{i=1}^{q} \rho_{ki} f(\mathbf{x}_i)}, \tag{14}$$

where we define $\rho_{kk} = 1$. This is the most general theoretical result in this paper, and it generalizes the niching rule of Equation 4. Notice how the transmission ratio ρ_{ki} from $\mathbb{X}_k$ to $\mathbb{X}_i$ generalizes the transmission ratio ρ_{12} from Equation 10.

In order to validate the theoretical developments so far, we perform experiments in the next two sections.

5 EXPERIMENTS USING IDEALIZED OPERATORS

The purposes of these experiments are: (i) Check that the deterministic difference equation analysis models the stochastic situation well; (ii) Check that the approach of picking a candidate from each niche is reasonable in the analysis. In order to achieve these goals, we use quite large population sizes in the following.

The experiments in this first section are done using a fitness function with only q discrete niches each of

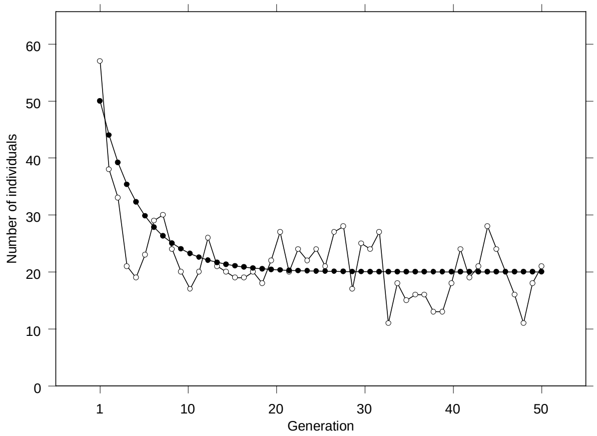

Figure 1: Predicted results (solid circles) versus experimental results (open circles) for probabilistic crowding.

size one, mutation probability p_l idealized as uniform change to one of the other niches and the probabilistic crowding acceptance rule to choose the winner.

5.1 TWO NICHES, SAME JUMP PROBABILITIES

We use Equation 9, here with $f(\mathbf{x}) = 1, f(\mathbf{y}) = 4$. This gives

$$X(t) = \frac{1}{5} + \left(\frac{1}{2} - \frac{1}{5}\right) p_s^t = \frac{1}{5} + \frac{3}{10} p_s^t$$

and

$$Y(t) = \frac{4}{5} + \left(\frac{1}{2} - \frac{4}{5}\right) p_s^t = \frac{4}{5} - \frac{3}{10} p_s^t.$$

We let $q = 2$, $p_s = 0.8$, use population size 100, and let the GA run for 50 generations. A plot of experimental versus predicted results for niche $\mathbb{X}$ is provided in Figure 1. A 'mutation' probability $p_l = 1 - p_s = 0.2$ might seem high, but recall that this operation gives jumps between niches, and is not the usual bit-wise mutation. In the figure, we notice that the experimental results follow the prediction very well. There is some noise, but this is as expected, since a probabilistic acceptance rule is used.

5.2 MULTIPLE NICHES, SAME JUMP PROBABILITIES

Consider the function $f(x) = x$, where x is an integer between 1 and 8 inclusive, so $q = 8$. Here we can use Equation 14 with $\rho_{ji} = 1$, giving

$$X_i = \frac{f(x_i)}{\sum_{i=1}^{q} f(x_i)}, \tag{15}$$

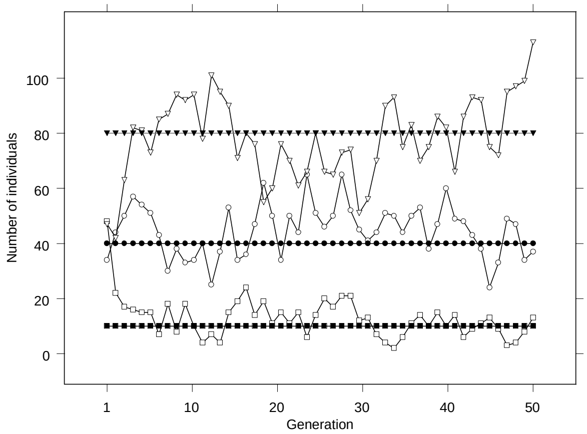

Figure 2: Predicted (solid circles) and experimental (open circles) results for niches $\mathbb{X}_1$, $\mathbb{X}_4$, and $\mathbb{X}_8$. Predicted is steady-state expected proportion of population; experimental is measured proportion starting at generation one.

with for example $X_1 = 1/36$, $X_4 = 4/36$, and $X_8 = 8/36$.

A plot of experimental versus predicted results for $p_s = 0.8$ is provided in Figure 2. A population size of $n = 360$ is used, and the GA is run for 50 generations. With the proportions just mentioned for X_1, X_4, and X_8, we get predicted population sizes $nX_1 = 10$, $nX_4 = 40$, and $nX_8 = 80$. Again, we notice that the empirical results follow the predicted results very well, although there is a certain level of noise also in this case, as expected.

An analysis of the amount of noise can be performed as follows. Consider the GA's operation as n Bernoulli trials where the probability of picking from the i-th niche is given by Equation 15, so $p_i = X_i$. This gives a binomial distribution, where $\mu_i = np_i$, $\sigma^2 = np_i(1 - p_i)$. For niche $\mathbb{X}_1$ we get $\sigma \approx 3.1$, and similarly for $\mathbb{X}_4$ and $\mathbb{X}_8$ we get $\sigma \approx 6.0$ and $\sigma \approx 7.9$ respectively. The fact that the noise increases with the fitness of a niche, as seen in Figure 2, is therefore not surprising.

6 EXPERIMENTS USING TRADITIONAL OPERATORS

In this section, we introduce traditional mutation and crossover operators into the probabilistic crowding experiments. A population size of $n = 200$, and 100 generations is used. Two probabilistic crowding variants are investigated, variant M and variant M&C. For the variant M, mutation probability 0.1 is used; for the variant M&C, crossover probability 0.6 and mutation probability 0.3 is used.

Interval of	**F1**			**F2**		
function domain	**P**	**M**	**M+C**	**P**	**M**	**M+C**
0.00 – 0.04	0.00083	0.0027	0.0019	0.0012	0.0016	0.0019
0.04 – 0.08	0.041	0.027	0.032	0.061	0.062	0.078
0.08 – 0.12	0.12	0.14	0.11	0.17	0.26	0.34
0.12 – 0.16	0.0 41	0.021	0.025	0.061	0.055	0.047
0.16 – 0.20	0.00083	0.0014	0.0014	0.0012	0.0017	0.0022
0.20 – 0.24	0.00083	0.0011	0.0026	0.0012	0.0020	0.0023
0.24 – 0.28	0.041	0.021	0.024	0.058	0.043	0.049
0.28 – 0.32	0.12	0.16	0.17	0.16	0.23	0.19
0.32 – 0.36	0.041	0.025	0.029	0.055	0.046	0.042
0.36 – 0.40	0.00083	0.0019	0.0024	0.0011	0.0012	0.0019
0.40 – 0.44	0.00083	0.0016	0.0018	0.00097	0.0012	0.0018
0.44 – 0.48	0.041	0.024	0.029	0.046	0.032	0.030
0.48 – 0.52	0.12	0.13	0.10	0.12	0.11	0.088
0.52 – 0.56	0.041	0.029	0.029	0.041	0.026	0.028
0.56 – 0.60	0.00083	0.0018	0.0017	0.00078	0.0019	0.0026
0.60 – 0.64	0.00083	0.0015	0.0024	0.00067	0.0016	0.0015
0.64 – 0.68	0.041	0.024	0.025	0.031	0.018	0.015
0.68 – 0.72	0.12	0.18	0.16	0.080	0.043	0.033
0.72 – 0.76	0.041	0.024	0.027	0.026	0.018	0.015
0.76 – 0.80	0.00083	0.0020	0.0032	0.00048	0.0010	0.00055
0.80 – 0.84	0.00083	0.0014	0.0030	0.00039	0.0011	0.0011
0.84 – 0.88	0.041	0.026	0.028	0.017	0.012	0.011
0.88 – 0.92	0.12	0.14	0.15	0.044	0.024	0.0096
0.92 – 0.96	0.041	0.027	0.032	0.014	0.0086	0.0066
0.96 – 1.00	0.00083	0.0015	0.0024	0.00025	0.00055	0.0014

Table 1: Experimental results for probabilistic crowding on the F1 and F2 functions, aggregated over all generations. The ‘P’ columns show predicted allocations, while the ‘M’ and ‘M+C’ columns show actual allocations for probabilistic crowding with mutation only and with mutation and crossover respectively.

Table 1 shows results from the experiments. Results are presented for the F1 and F2 test functions [Goldberg and Richardson, 1987], where

$$\begin{aligned} \mathrm{F1}(x) &= \sin^6(5\pi x) \\ \mathrm{F2}(x) &= e^{-2(\ln 2)\left(\frac{x-0.1}{0.8}\right)^2} \sin^6(5\pi x). \end{aligned}$$

The question is whether one should focus on niche allocation, peak allocation, or a combination of both. The last alternative is chosen here, since the strength of probabilistic crowding is how individuals are evenly allocated. According to this, the functions have been split up into 25 equally-sized intervals on the X-axis. These intervals are presented as rows in the table. The columns present results for test functions F1 and F2.

The main result in Table 1 is that for both F1 and F2, the probabilistic crowding variants M and M+C give allocation of trials close to that predicted. This confirms that Equation 14 (and its special cases as presented) can be applied also when classical GA operators are used, at least for a certain class of fitness functions. There are two qualifications here. First, note that there is a small ‘smoothing’ effect for the intervals of lowest fitness, for example the interval $0.00 - 0.04$. This is mostly due to allocations in early generations, when individuals are distributed more uniformly, according to the uniform random initialization. Second, there is a slightly higher allocation than predicted to intervals of high fitness, such as the intervals $0.08 - 0.12$ and $0.28 - 0.32$. It is currently not clear what the reason is for this slightly higher allocation.

7 CONCLUSION AND FUTURE WORK

A novel niching algorithm, probabilistic crowding, has been introduced. Probabilistic crowding has been shown theoretically as well as empirically to give reliable niching according to a novel, general niching rule, a rule which generalizes the niching rule known from previous research. The two core ideas in probabilistic crowding are (i) to hold tournaments between similar individuals, and (ii) to let tournaments be probabilistic. These two principles leads to a niching algorithm which is simple, predictable, and fast. Probabilistic crowding also does not have any additional parameters compared to classical GAs, which is often an advantage. The algorithm belongs to a family of algorithms, integrated tournament algorithms, which are also defined and presented in this paper.

Future work includes the following. First, experiments on harder fitness functions, such as complex Bayesian networks, would be interesting. Second, the relationship to Metropolis and simulated annealing could be investigated further, in particular the use of temperature. A third issue is population sizing. Fourth, the approach of holding a probabilistic tournament could be extended to include the distance function and possibly also the mating step.

Acknowledgments

Ole J. Mengshoel gratefully acknowledges support from Dr. David C. Wilkins, in part by ONR Grant N00014-95-1-0749, ARL Grant DAAL01-96-2-0003, and NRL Grant N00014-97-C-2061. The software for this work used the GAlib genetic algorithm package written by Matthew Wall at the Massachusetts Institute of Technology.

Dr. David E. Goldberg's work was sponsored by the Air Force Office of Scientific Research, Air Force Materiel Command, USAF, under grant number F49620-97-1-0050. Research funding for this project was also provide by a grant from the US Army Research Laboratory Program, Cooperative Agreement DAAL01-96-2-003. The US Government is authorized to reproduce and distribute reprints for governmental purposes notwithstanding any copyright notation thereon.

The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies and endorsements, either expressed or implied, of the Air Force Office of Scientific Research or the US Government.

References

[Culberson, 1992] Culberson, J. (1992). Genetic invariance: A new paradigm for genetic algorithm design. Technical Report 92-02, University of Alberta, Department of Computer Science.

[DeJong, 1975] DeJong, K. A. (1975). *An Analysis of the Behavior a Class of Genetic Adaptive Systems*. PhD thesis, University of Michigan.

[Goldberg, 1990] Goldberg, D. (1990). A note on Boltzmann tournament selection for genetic algorithms and population-oriented simulated annealing. *Complex Systems*, 4(4):445–460.

[Goldberg, 1989] Goldberg, D. E. (1989). *Genetic Algorithms in Search, Optimization & Machine Learning*. Addison-Wesley, Reading, MA.

[Goldberg and Richardson, 1987] Goldberg, D. E. and Richardson, J. (1987). Genetic algorithms with sharing for multimodal function optimization. pages 41–49, Hillsdale, New Jersey. Lawrence Erlbaum Associates.

[Harik, 1995] Harik, G. R. (1995). Finding multimodal solutions using restricted tournament selection. In Eshelman, L., editor, *Proceedings of the Sixth International Conference on Genetic Algorithms*, pages 24–31, San Francisco, CA. Morgan Kaufmann.

[Kirkpatrick et al., 1983] Kirkpatrick, S., Gelatt Jr., C. D., and Vecchi, M. P. (1983). Optimization by simulated annealing. *Science*, (220):671–680.

[Laarhoven and Aarts, 1987] Laarhoven, P. J. M. v. and Aarts, E. H. L. (1987). *Simulated Annealing: Theory and Applications*. Mathematics and Its Applications. D. Reidel Publishing Compahy, Dordrecht, Holland.

[Mahfoud, 1995] Mahfoud, S. W. (1995). *Niching methods for genetic algorithms*. PhD thesis, University of Illinois at Urbana-Champaign, Urbana, IL, USA. IlliGAL Report 95001.

[Mahfoud and Goldberg, 1995] Mahfoud, S. W. and Goldberg, D. E. (1995). Parallel recombinative simulated annealing: A genetic algorithm. *Parallel Computing*, 21:1–28.

[Metropolis et al., 1953] Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. N., Teller, A. H., and Teller, E. (1953). Equation of state calculations by fast computing machines. *Journal of Chemical Physics*, 21(6):1087–1092.

[Neal, 1993] Neal, R. M. (1993). Probabilistic inference using Markov chain Monte Carlo methods. Technical Report CRG-TR-93-1, Dept. of Computer Science, University of Toronto.

[Thierens and Goldberg, 1994] Thierens, D. and Goldberg, D. E. (1994). Elitis recombination: An integrated selection recombination GA. In *Proceedings of the First IEEE Conference on Evolutionary Computation*, pages 152–159, Orlando, FL.