# A Selection Scheme Based on Competition for Evolutionary Algorithms

Andrea G. B. Tettamanzi
Università degli Studi di Milano, Dipartimento di Scienze dell'Informazione
Via Comelico 39/41, I-20135 Milano, Italy.
E-mail: tettaman@dsi.unimi.it

January 1996[1]

**Abstract**

This paper introduces and studies a local selection scheme for Evolutionary Algorithms based on competition, particularly suited for implementation on massively parallel architectures.

In many tasks, such as evolution of game strategies, genetic programming and evolution of control procedures, competition between two individuals on one problem instance chosen according to some probability can be a valid alternative to defining an appropriate fitness function that embeds *a priori* knowledge of the problem, which often is not within the reach of everybody, as it requires a deep insight of the problem along with some mathematical skills.

The behavior of a population subject to competitive selection is analyzed probabilistically through the study of a statistical indicator under different assumptions about the distribution of fitness values.

[1]This paper is an improved version of [25], which was submitted to the Evolutionary Computation journal, revised, and finally rejected. A summary of this work was then published as [26].

# 1 Introduction

Evolutionary Algorithms [18, 2] (EAs, a general term to encompass genetic algorithms [16, 12], evolution strategies [22, 23], evolutionary programming [11] and genetic programming [17]) are stochastic optimization methods which exploit ideas from natural evolution: An EA maintains a population, $x_t$ for generation $t$, of individuals, chosen from a set $\Gamma$ of all possible individuals. Each individual $\gamma \in \Gamma$ represents a potential solution $S_\gamma$ to the problem at hand; each solution is evaluated to give some measure of its "fitness", $f(S_\gamma)$, or $f(\gamma)$ for brevity; then a new population (generation $t+1$) is formed by selecting the fittest individuals. Some members of the new population undergo mutation or recombination.

This paper deals with a selection methods that can be used to form a population $x_{t+1}$ from $x_t$. Among the classical methods found in the literature [9, 15] is "fitness-proportionate selection", sometimes implemented as "roulette-wheel selection" [12]: an individual $\gamma \in x_t$ is selected for reproduction with probability

$$P[\gamma] = \frac{f(\gamma)}{\sum_{\kappa \in x_t} f(\kappa)}.$$

Another selection method that enjoys large popularity is "linear rank selection" [14], which defines the target sampling rate (TSR) of an individual $\gamma$ as

$$\mathrm{TSR}(\gamma) = \min + (\max - \min)\frac{\mathrm{rank}(\gamma)}{n-1},$$

where $\mathrm{rank}(\gamma) \in \{0, \ldots, n-1\}$ is the index of $\gamma$ when the population is sorted in increasing order on fitness, and $n$ is the population size. The TSR is the number of times an individual should be chosen for reproduction.

The need to exploit massively parallel machines [8, 10] compels the EA implementer to resort to local selection strategies, such as local tournament selection, with all the possible variants that are found in global selection strategies.

Tournament selection, attributed to a never published work by Wetzel and studied for the first time in [7], extracts $k$ individuals from the current population with uniform probability and without replacement and has them play a "tournament". In the deterministic case, the tournament is won by the fittest individual; the tournament can be probabilistic, the probability of an individual winning it being proportional to its fitness. Tournament selection has enjoyed large popularity and has produced much literature (see [3, 4, 20, 13] among others).

On the other hand one of the critical steps in the design of a *good* evolutionary algorithm is to define a fitness function that encapsulates as much as possible any convenient *a priori* knowledge of the problem. In general this task is far from trivial and often it is difficult, as it requires a deep insight of the problem along with some mathematical skills.

The purpose of this paper is to make clear that the explicit definition of a fitness function, although sufficient, is not a necessary condition for having an evolutionary algorithm work. As a consequence, we propose a new selection scheme based on the concept of *competition,* which is a variation of binary tournament selection: We introduce a stochastic operator that determines, according to some procedure that will depend on the problem at hand, a winner out of two given individuals.

The motivation for introducing this concept comes from the observation that competition between two individuals to obtain some scarce resource plays a central role in natural selection, which appears to be a local, distributed and asynchronous process. Examples in Nature of this kind of competition could be a fight to secure enough food, the duels between bucks to gain the right to mate with a given female and so on.

The selection scheme proposed and studied in this paper can be seen as a particular version of binary tournament selection, where the notion of fitness is never used, at least directly.

In Section 2 we define the above stochastic operator and provide its main properties; some application examples will illustrate how the concept of competition thus defined can be taken advantage of in an evolutionary algorithm.

Section 3 studies how the concept of competition connects to that of fitness: In particular we define the *posterior* fitness based on the expected outcomes of competitions and show how competition can be defined based on a given *prior* fitness in a consistent way, i.e. in such a way that the relevant posterior fitness equal the prior fitness up to a multiplicative constant. This will be done for the case in which the fitness of individuals is finite but not bounded from above as well as for the case of bounded fitness.

Finally, in Section 4 we analyze the expected behavior of a population subject to competitive selection through the study of the reproduction rate of an individual with respect to its (prior or posterior) fitness under different assumptions about the distribution of fitness values in the population.

# 2 A competitive model

Let $\Gamma$ be a finite set, whose elements we will call *individuals*. Let us denote by $\Gamma^*$ the space of all populations consisting of any number individuals. More than one individual in a population can share the same genotype and what distinguishes two populations of equal size is only the number of individuals in which each genotype occurs. Therefore, a convenient way to think of a population is as a multi-set, or bag, of genotypes, that is, the equivalence class of $n$-tuples identical up to a permutation of their elements.

Every population $x \in \Gamma^*$ is in a 1-to-1 correspondence with an occupancy function $n_x\colon \Gamma \to N$, such that, for all $\gamma \in \Gamma$, $n_x(\gamma)$ is the number of individuals in $x$ sharing the same genotype $\gamma$, that is the occupancy number of $\gamma$ in $x$. The size of a population $x$ is defined as $\|x\| \equiv \sum_{\gamma\in\Gamma} n_x(\gamma)$.

Let us introduce a stochastic function

$$\mathbf{winner}\colon \Gamma \times \Gamma \to \Gamma$$

which, given two competitors, chooses the winner according to a probability; i.e. $P[\mathbf{winner}(\gamma,\kappa) = \gamma]$ is the probability that $\gamma$ wins a fight with $\kappa$.

Then we must have:

1. $P[\mathbf{winner}(\gamma,\kappa) = \gamma] + P[\mathbf{winner}(\gamma,\kappa) = \kappa] = 1$;
2. $P[\mathbf{winner}(\gamma,\kappa) = \gamma] = P[\mathbf{winner}(\kappa,\gamma) = \gamma]$.

## 2.1 Application examples

**Example 2.1:** A simple example should make clear how the **winner** function can be naturally suggested by certain classes of problems. Suppose that $\Gamma$ is a finite collection of formulas representing analytical functions of one real variable into the real numbers; with a little abuse of notation we can write $\gamma(x) \in R$ for the value of the function represented by $\gamma$ calculated in $x \in R$. We want to find the formula in $\Gamma$ that approximates best a target function $g\colon R \to R$, which we are able to evaluate in any point $x \in R$ by performing some experiment, though we ignore its analytical expression.

In this setting, we may define $\mathbf{winner}(\gamma, \kappa)$ as if the two opponents $\gamma$ and $\kappa$ tried how close they get to $g(x)$ for an $x$ picked at random: Let $X \in R$ be a random variable with standard normal distribution $\mathcal{N}(0,1)$; then

$$\mathbf{winner}(\gamma, \kappa) = \begin{cases} \gamma & \text{if } |\gamma(X) - g(X)| < |\kappa(X) - g(X)| \\ \kappa & \text{if } |\gamma(X) - g(X)| > |\kappa(X) - g(X)| \end{cases}$$

In case of tie, that is when $|\gamma(X) - g(X)| = |\kappa(X) - g(X)|$, we may flip an unbiased coin and choose $\gamma$ if it comes head and $\kappa$ otherwise. Therefore

$$\begin{aligned} P[\mathbf{winner}(\gamma, \kappa) = \gamma] &= P[|\gamma(X) - g(X)| < |\kappa(X) - g(X)|] + \\ &\quad + \frac{1}{2} P[|\gamma(X) - g(X)| = |\kappa(X) - g(X)|]. \end{aligned}$$

Let $A = \{x : |\gamma(x) - g(x)| < |\kappa(x) - g(x)|\}$ and $B = \{x : |\gamma(x) - g(x)| = |\kappa(x) - g(x)|\}$; then

$$P[\mathbf{winner}(\gamma, \kappa) = \gamma] = \int_A \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}x^2} dx + \frac{1}{2} \int_B \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}x^2} dx.$$

**Example 2.2:** This example suggests a way to save a significant amount of computational effort in genetic programming (GP) applications [17]. A GP maintains a population of programs in some appropriate language[2] and calculates fitness of individuals by testing each program on a set of feasible input values, either generated at random or predetermined, and measuring the cumulative error with respect to a specification.

Even for simple problems, assigning a fitness to an individual is a computationally heavy task, especially because the same program must be run several times with different input data in order to obtain a significant estimate of its goodness. That is the reason why adopting competitive selection could in our opinion lead to a substantial improvement in efficiency.

Consider for instance the problem of synthesizing or optimizing a controller for a given plant or system. Suppose thus that $\Gamma$ is the set of possible

[2]Koza and others have used a subset of LISP for a number of reasons, but any other programming language should work as well, and indeed there have been implementations in C++, just to mention a programming language that differs drastically from LISP, or even with an encoding from which a program can be compiled (see e.g. the work by Wolfgang Banzhaf [5]).

controllers; be $S$ the set of possible states of the system to be controlled and suppose that failure and success conditions of a simulation are defined.

Competitions can be performed as follows: An initial state is randomly generated according to some distribution over $S$, which could depend, for example, on the observed frequency of states in the current use of the system; two simulation for the competing controllers $\gamma$ and $\kappa$ are carried out in parallel until one of them fails or succeeds; the competition is then won by the controller that has succeeded or by the one that has not failed. One could imagine setting a time-out, so that if neither controller has prevailed after a certain amount of time, the winner is chosen at random, with equal probability.

This very simple model has been successfully applied by the author to the optimization of programs for a fuzzy processor for control [24].

Notice that the aforesaid would apply with few changes also to those numerous EAs that evolve neural networks.

**Example 2.3:** This last example tries to demonstrate how a suitable **winner** function can be constructed to bypass the actual computation of a particularly heavy fitness function. The problem at hand is that of Portfolio Selection, arising in finance when an investor has to choose how to divide a given amount of funds among several possible assets, i.e. has to select a portfolio. Details on how this problem is approached using EAs can be found in [1]. Here, $\Gamma$ is the set of all the feasible portfolios, and we seek to maximize the expected return while trying to keep the risk as low as possible; it is out of the scope of this paper to explain how this is done, but it suffices to know that the risk $R(\gamma)$ and expected return $E(\gamma)$ associated with portfolio $\gamma$ are calculated on the basis of the historical records $H_{it}$ of the return for each asset $i = 1, \dots, N$ and period $t = 1, \dots, T$. Typical dimensions for this problem are a hundred assets, each with a record of a few hundreds of historical returns. Now, the expected return is usually computed as

$$E(\gamma) = \frac{1}{NT} \sum_{i=1}^{N} \sum_{t=1}^{T} \gamma_i H_{it}$$

and $R(\gamma)$ is some indicator of risk; the fitness of $\gamma$ is a linear combination $z(R(\gamma), E(\gamma))$.

Instead of computing the fitness of two individuals and then compare them, we can define $\textbf{winner}(\gamma, \kappa)$ as follows: choose a period $0 < t \leq T$ and an asset $0 < i \leq N$ at random, then[3]

$$\textbf{winner}(\gamma,\kappa) = \begin{cases} \gamma & \text{if } \gamma_i H_{it} > \kappa_i H_{it} \\ \kappa & \text{if } \gamma_i H_{it} < \kappa_i H_{it}, \end{cases}$$

where $\gamma_i$ indicates a single component of vector $\gamma$. This is computed in $\frac{1}{NT}$ of the time it would take computing the two fitness values; the probability of $\gamma$ winning a competition against $\kappa$ is

$$P[\textbf{winner}(\gamma,\kappa) = \gamma] = P[H_{it} > 0]P[\gamma_i > \kappa_i] + P[H_{it} < 0]P[\gamma_i < \kappa_i].$$

There is no doubt that a competition like this in many cases will kill the wrong portfolio, but one should consider that by the same time the fitness for an individual is computed, as many as for instance 100,000 competitions could have been performed, considering a hypotetical problem involving 100 assets with 1,000 historical returns each.

## 2.2 Implementation

Letting $n$ be the size of a population, we can use the **winner** function to build a general selection scheme as follows:

**for** $i = 1$ **to** $n$ **do**
  **begin**
    $\gamma := \mathit{old\_pop}[\mathcal{U}(1,n)]$;
    $\kappa := \mathit{old\_pop}[\mathcal{U}(1,n)]$;
    $\mathit{new\_pop}[i] := \textbf{winner}(\gamma,\kappa)$;
  **end**;

where $\mathcal{U}(1,n)$ denotes picking an integer from 1 to $n$ with uniform probability.

This is exactly the algorithm for binary tournament selection. The only difference is that here a stochastic **winner** operator is used to determine which individual will reproduce instead of a criterion based on fitness.

[3]This might be justified by saying that the objective function $z$ is separable, but that should not concern the reader at this point; one could always view this as the definition of an alternative index of risk.

This scheme can be easily parallelized: For example in a massively parallel architecture each Processing Element (PE), holding, say, $\gamma$, could pick a $\kappa$ at random among its immediate topological neighbors, then store $\mathbf{winner}(\gamma, \kappa)$ on board in the place of $\gamma$. In a swarm-computing model [19, 6], in which PE's float freely as particles in a gas and every now and then come in contact with one another, two PE's could have their associated genotypes $\gamma$ and $\kappa$ dispute a competition, then substitute both with $\mathbf{winner}(\gamma, \kappa)$.

Let us denote $n_x(\gamma)$ the number of individuals in population $x \in \Gamma^*$ sharing the same genotype $\gamma \in \Gamma$. The expected number of occurrences of $\gamma$ in the new population $y$ obtained from population $x$ by applying the competitive selection scheme given above is

$$E[n_y(\gamma)] = \frac{2n_x(\gamma)}{n} \sum_{\kappa \in \Gamma} n_x(\kappa) P[\mathbf{winner}(\gamma, \kappa) = \gamma]. \tag{1}$$

# 3 Competition and fitness

Competitive selection does not require that a fitness function be explicitly defined. However, one could think that, though never computed in full by the algorithm, fitness is nonetheless there, and determines the outcomes of competitions. There are two complementary ways in which this concept can be made explicit.

On one side, one can start with a definition of the **winner** operator and define the fitness of an individual on the basis of the expected probability of winning a competition against any other individual. Such a fitness measure, obtained by observing the outcomes of competitions, could be called *posterior.*

On the other side, one might imagine to know already how to precisely compute the fitness of any given individual. This fitness measure, given from the outset, could be called *prior.* In this setting, one could define the **winner** operator based on prior fitness.

Now, the question arises if these two views can be reconciled into one consistent framework. In other words, we want to answer the question if there exists a way to assign probabilities to the outcomes of competitions between pairs of individuals on the basis of their prior fitness, such that posterior fitness, obtained through observation of the outcomes, results proportional to prior fitness.

If this is the case (and indeed it is, as it will be shown in the sequel), then we will be able to write, given an evolutionary algorithm making use of competitive selection, an equivalent one that employs a corresponding fitness and a form of probabilistic tournament selection. From the practical standpoint, if the algorithm using competitive selection is faster (because it carries out but a cheap, partial evaluation of fitness), we can use it and get the same results as the latter, but with a smaller expenditure of computational resources.

## 3.1 Posterior fitness

To begin with, we give a definition of posterior fitness for each individual in $\Gamma$ as follows: Imagine having all individuals in $\Gamma$ play an American tournament[4] and call the score $s(\gamma)$ the number of wins gained by $\gamma$ at the end (for sake of simplicity no draw is allowed):

$$s(\gamma) = \sum_{\kappa \in \Gamma} I_{[\mathbf{winner}(\gamma,\kappa)=\gamma]},$$

where $I_{[Q]}$ is the indicator function of predicate $Q$, whose value is one where $Q$ is true and zero everywhere else.

Of course $s(\gamma)$ is a random variable, so we can define the fitness of $\gamma$ as the expected score of $\gamma$:

$$f(\gamma) = E[s(\gamma)] = \sum_{\kappa \in \Gamma} P[\mathbf{winner}(\gamma, \kappa) = \gamma].$$

In the next two subsections, we study how the **winner** operator can be defined (that is, how probabilities for the outcome of competitions can be assigned) on the basis of prior fitness, such that the expected score, that is the posterior fitness, of every individual $\gamma$ results proportional to its prior fitness.

## 3.2 The unbounded prior fitness case

Let us assume that a prior fitness is defined and can be calculated for all individuals in $\Gamma$. In general, we might not know the maximum value that

[4]An *American tournament* is one in which each team encounters all the others (round-robin); here, for the sake of simplicity, we will have each team encounter itself too.

this fitness can attain. This amounts to assuming that fitness is finite but unbounded from above.

We want to come up with a law that expresses the probabilities for the two possible outcomes of competitions in terms of the prior fitness of the two competitors. Besides, we want this law to be *consistent* with the definition of posterior fitness given in Section 3.1.

In mathematical terms, what we are looking for is a function $p: R^+ \times R^+ \to [0,1]$ such that, if we let $P[\mathbf{winner}(\gamma, \kappa) = \gamma] = p(f(\gamma), f(\kappa))$, the following conditions hold:

1. $p(x, y) = 1 - p(y, x)$;
2. $p(0, y) = 0$, for $y > 0$;
3. $x < z$ implies $p(x, y) < p(z, y)$;
4. there exists a constant $C$ such that, for all $\gamma$,

$$\sum_{\kappa \in \Gamma} p(f(\gamma), f(\kappa)) = C f(\gamma).$$

The first condition above derives from the fact that there are only two possible outcomes of a competition, and the probability of either occurring must equal the probability of the other not occurring.

The second condition states that an individual with null fitness cannot conceivably win a competition with another individual having nonzero fitness. This is a rather intuitive assumption which does not seem to require further justification.

The third condition too is quite intuitive, stating that, of two individuals, the one with higher fitness will have greater chances of winning against a third fixed individual.

Condition 4 above is the most important one, and it corresponds to our requirement for consistency: Prior and posterior fitness should be proportional, i.e. equivalent up to a multiplicative constant.

A reasonable additional assumption is that $p$ depends only on the ratio of the two fitness values, not on their absolute magnitude, in the spirit of fitness-proportionate selection. In other words, if we were to replace a fitness function $f: \Gamma \to R^+$ with a new one $\hat{f}: \Gamma \to R^+$ such that, for all $\gamma \in \Gamma$ and $k > 0$, $\hat{f}(\gamma) = k f(\gamma)$, we would expect no change in the way selection

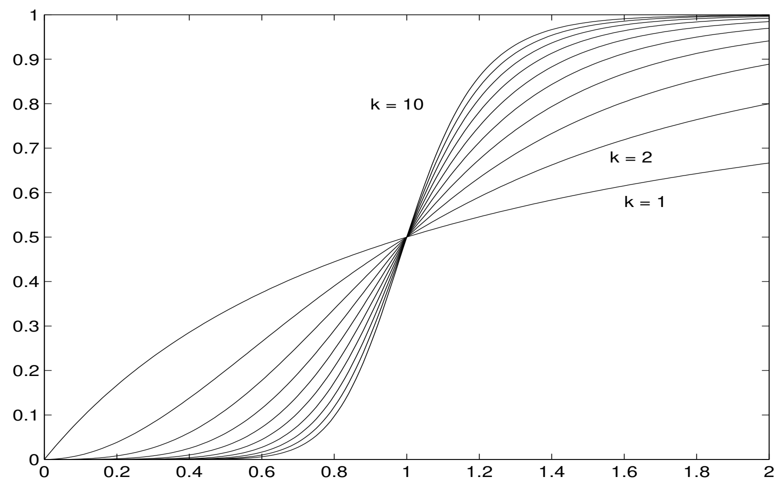


Figure 1: The family of functions $p(x) = 1 - \frac{1}{x^k+1}$ for $k = 1, \ldots, 10$.

works. Therefore, rewriting $p(x, y)$ as $p(x/y)$ yields the following functional equation:

$$\begin{cases} p(x) = 1 - p(x^{-1}) \\ p(0) = 0. \end{cases} \tag{2}$$

**Solutions on $R$.** Functional Equation 2 has at least one continuous and derivable monotone solution over the entire real axis; this solution is

$$p(x) = \frac{2}{\pi} \arctan x.$$

**Solutions on $R^+$.** However, if we restrict our attention to the positive axis, we find a family of continuous and derivable monotone solutions, namely:

$$p(x) = 1 - \frac{1}{x^k + 1}, \tag{3}$$

for $k = 1, 2, \ldots$; the differences in behavior that these solution exhibit when used under the competitive selection scheme will be examined in Section 4.

Notice that, for $k = 1$, Equation 3 can be rewritten as

$$p(x) = 1 - \frac{1}{x + 1} = \frac{f(\gamma)}{f(\gamma) + f(\kappa)}.$$

This is a form of local tournament selection where fitness-proportionate selection is applied to a population of two individuals, namely $\gamma$ and $\kappa$.

The family of functions $p(x)$ in Equation 3 is plotted in Figure 1. As $k$ increases, $p(x)$ allows fewer chances of a win to the worst competitor; in the limit case, for $k = \infty$, the competition becomes deterministic, and the best competitor always wins:

$$\lim_{k\to\infty} p(x) = \begin{cases} 1, & x > 1 \quad (\gamma \text{ always wins}); \\ 1/2, & x = 1 \quad (\gamma \text{ and } \kappa \text{ are even}); \\ 0, & x < 1 \quad (\gamma \text{ never wins}). \end{cases} \tag{4}$$

It is easy to verify that every function $p$ that is a solution of Equation 2 satisfies Condition 4. To show that, we need first prove the following

**Lemma 1** *For any set of positive real numbers $x_1, x_2, \ldots, x_r$, $0 < x_i < +\infty$, $i = 1, 2, \ldots, r$, there exists a constant $0 < C < +\infty$ such that, for all integers $n$, $1 \leq n \leq r$,*

$$\sum_{i=1}^{r} p(x_n/x_i) = Cx_n,$$

*where $p$ is any function satisfying Equation 2.*

**Proof:** To show that such a $C$ exists, we will calculate it explicitly. Let us write $p_{ij}$ instead of $p(x_i/x_j)$ for convenience: We have $r$ equations

$$\begin{aligned} Cx_1 &= p_{11} + p_{12} + \ldots + p_{1r} \\ Cx_2 &= p_{21} + p_{22} + \ldots + p_{2r} \\ &\vdots \\ Cx_r &= p_{r1} + p_{r2} + \ldots + p_{rr}. \end{aligned}$$

First of all observe that $p_{ii} = 1/2$, for $i = 1, 2, \ldots, r$; in addition we know that $p_{ij} = 1 - p_{ji}$, so rewriting all the $p_{ij}$'s where $i > j$ as $1 - p_{ji}$, yields

$$\begin{aligned} Cx_1 &= \frac{1}{2} + p_{12} + \ldots + p_{1r} \\ Cx_2 &= 1 - p_{12} + \frac{1}{2} + \ldots + p_{2r} \\ &\vdots \\ Cx_r &= 1 - p_{1r} + 1 - p_{2r} + \ldots + \frac{1}{2} \end{aligned}$$

In general

$$Cx_n = (n-1) + \frac{1}{2} + p_{n,n+1} + \ldots + p_{nr} - p_{1n} - \ldots - p_{n-1,n};$$

Now summing up all the $r$ equations cancels the terms with opposite signs and yields

$$C(x_1 + x_2 + \ldots + x_r) = \sum_{n=2}^{r}(n-1) + \frac{1}{2}r = \frac{1}{2}r^2,$$

therefore

$$C = \frac{\frac{1}{2}r^2}{x_1 + x_2 + \ldots + x_r}.$$

Then we set $r = \|\Gamma\|$ and substitute $f(\gamma)$ for all $\gamma \in \Gamma$ instead of the $x_i$'s, thus having proven

**Theorem 1** *Let $f\colon \Gamma \to R^+$ such that, for all $\gamma \in \Gamma$, $0 \leq f(\gamma) \leq +\infty$; let $p\colon R^+ \to [0,1]$ satisfy Equation 2; then there exists a constant $0 < C < +\infty$ such that, for all $\gamma \in \Gamma$,*

$$\sum_{\kappa \in \Gamma} p(f(\gamma)/f(\kappa)) = Cf(\gamma).$$

## 3.3 The bounded prior fitness case

Sometimes the nature of the problem allows one to know the maximum value that the fitness can attain; actually we can always reduce to the case of fitness values in $[0,1]$. Therefore we can come up with tighter conditions for $p$:

1. $p(x,y) = 1 - p(y,x)$;
2. $p(0,y) = 0$, for $y > 0$;
3. $p(1,y) = 1$, for $y < 1$;
4. there exists a constant $C$ such that, for all $\gamma$,

$$\sum_{\kappa \in \Gamma} p(f(\gamma), f(\kappa)) = Cf(\gamma).$$

We have to solve this new functional equation:

$$\begin{cases} p(x,y) = 1 - p(y,x) & \\ p(0,y) = 0 & y > 0 \\ p(1,y) = 1 & y < 1 \end{cases} \tag{5}$$

We can take advantage of the fact that we know already solutions for the simpler Equation 2; we shall restrict ourselves to the solutions on $R^+$, because these are the ones we need.

We want to reduce to $p(x,y) = p(z(x,y))$, where the unary $p$ is a solution to Equation 2: In order for $p(x,y)$ to satisfy Equation 5, $z$ must be a solution of

$$\begin{cases} z(x,y) = z(y,x)^{-1} & \\ z(0,y) = 0 & y > 0 \\ z(1,y) = +\infty & y < 1 \end{cases} \tag{6}$$

Simply setting $z(x,y) = x/y$ would not work, because it would violate the condition $z(1,y) = +\infty$ for $y < 1$, so we need a strictly nondecreasing transformation $t$ that maps the interval $[0,1)$ into $[0,+\infty)$, so that we can write the solution to Equation 6

$$z(x,y) = t(x)/t(y).$$

Now, $t$ can be any function satisfying

- $t(0) = 0$,
- if $x < y$ then $t(x) < t(y)$,
- $\lim_{x \to 1} t(x) = +\infty$,

for $x, y \in [0,1)$. The simplest such function $t$ is

$$t(x) = 1 - \frac{1}{x+1} = \frac{x}{x+1};$$

substituting and simplifying then yields

$$p(x,y) = p(z(x,y)) = p(t(x)/t(y)) = p\left(\frac{x+xy}{y+xy}\right) = \frac{\left(\frac{x+xy}{y+xy}\right)^k}{\left(\frac{x+xy}{y+xy}\right)^k + 1}.$$

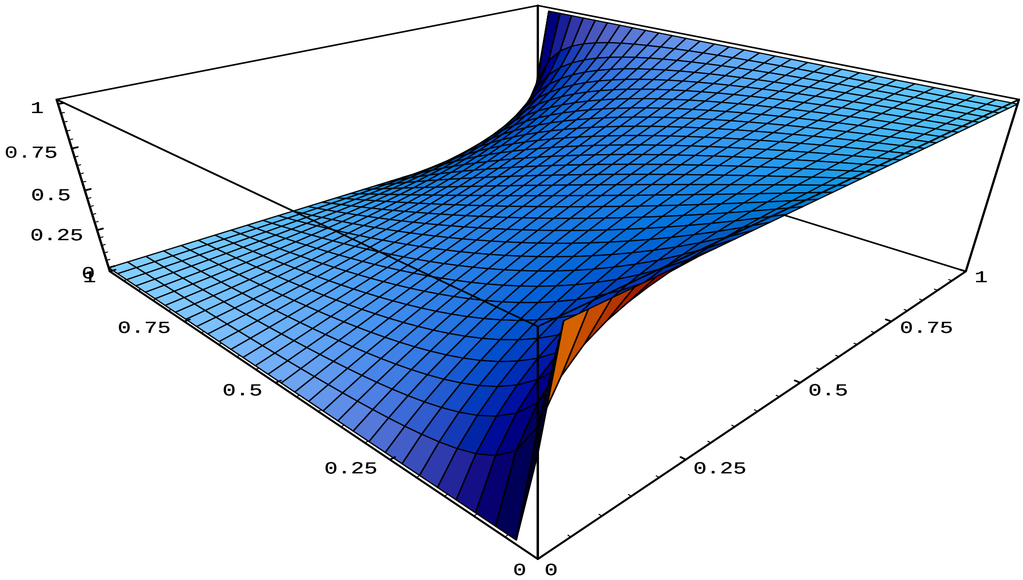


Figure 2: A three-dimensional plot of $p(x,y) = \frac{x+xy}{x+2xy+y}$.

Thus, for instance, with $k = 1$ we have

$$p(x,y) = \frac{x+xy}{x+2xy+y}.$$

This solution is shown in Figure 2.

It is easy to verify that Lemma 1 holds for $0 \leq x_i \leq 1$, $i = 1, 2, \ldots, r$, as well. We are now ready to prove that $p$ satisfies Condition 4.

**Theorem 2** *Let* $f\colon \Gamma \to [0,1]$*; let* $p\colon [0,1] \times [0,1] \to [0,1]$ *satisfy Equation 5; then there exists a constant* $0 < C < +\infty$ *such that, for all* $\gamma \in \Gamma$*,*

$$\sum_{\kappa \in \Gamma} p(f(\gamma), f(\kappa)) = C f(\gamma).$$

**Proof:** Just invoke Lemma 1 by substituting $f(\gamma)$ for all $\gamma \in \Gamma$ instead of the $x_i$'s.

# 4 Analysis of competitive selection

This section studies the behavior of a population subject to competitive selection under different assumptions about the distribution of fitness values.

A selection model can be characterized, among other things, by a graph that relates an individual's probability of being selected to its fitness. For example, with fitness-proportionate selection this graph is a straight line starting at the origin and going up as the fitness increases: This is but another way of stating that an individual's probability of being selected is directly proportional to its fitness. With truncation selection [21], the graph would come out as a step function, null below the truncation threshold and positive constant from there on. With linear rank selection the only thing we are able to say in general is that the graph would be nondecreasing. In all these cases, the actual shape of the graph, i.e. the inclination of the straight line in the fitness-proportionate selection case and the position of the step in the truncation selection case, depends on how fitness of individuals is distributed in the current population.

As an alternative one can study the graph of what could be called the *reproduction rate* of a genotype, that is the expected number of its copies in the next generation divided by the number of its copies in the current generation, depending on its fitness. It should be evident that this is exactly equivalent to the probability of being selected, since the two quantities are directly proportional.

First of all we will establish a methodology that will enable us to calculate, symbolically if we are lucky, or otherwise numerically, the graph of the reproduction rate, given the distribution of fitness in the population. Then the procedure for the rest of this section will consist in making an assumption about the fitness distribution and calculating the corresponding graph of the reproduction rate with respect to fitness.

## 4.1 Methodology

It should be observed, to begin with, that competitive selection behaves quite differently from the classical fitness-proportionate or roulette-wheel selection as to the reproduction rate of individuals. The expected number of occurrences of an individual $\gamma$ in the new population $y$ obtained from population $x$ by means of fitness-proportionate selection is

$$E[n_y(\gamma)] = \frac{nf(\gamma)n_x(\gamma)}{\sum_{\kappa\in\Gamma} n_x(\kappa)f(\kappa)}.$$

Let us denote this expectation with a subscript $f$, to distinguish it from the one related to the competitive scheme introduced in Equation 1, which will be denoted henceforth with a subscript $c$.

**Observation 1** *It is easy to verify that, for all populations $x \in \Gamma^*$ and all individuals $\gamma \in \Gamma$, if $Y$ is a random variable representing a population obtained from $x$ by selection,*

$$0 \leq \quad E_c[n_Y(\gamma)] \quad \leq 2n_x(\gamma); \tag{7}$$
$$0 \leq \quad E_f[n_Y(\gamma)] \quad \leq n. \tag{8}$$

In Inequality 7, equality on the right hand side holds when $P[\mathbf{winner}(\gamma,\kappa) = \gamma] = 1$ for all $\kappa \in x$; this is the competitive equivalent of $\gamma$ having a fitness that is incomparably larger with respect to all the other individuals in population $x$, that is when equality on the right hand side holds for Inequality 8.

Of course equalities on the left hand sides hold when $P[\mathbf{winner}(\gamma,\kappa) = \gamma] = 0$ for all $\kappa \in x$ or, equivalently, $f(\gamma) = 0$.

Let us define the individual reproduction rate as the expected number of occurrences of a genotype $\gamma$ in the the population $Y$ of the next generation per occurrence in the current population $x$; let us denote this quantity by

$$E_\gamma = \frac{E[n_Y(\gamma)]}{n_x(\gamma)}.$$

It is well known (see [12] for instance) that under fitness-proportionate selection,

$$E_\gamma = \frac{f(\gamma)}{\bar{f}_x} = \frac{nf(\gamma)}{\sum_{\kappa\in\Gamma} n_x(\kappa)},$$

that is, $E_\gamma$ is linear against $f(\gamma)$; now we turn our attention to the analysis of $E_\gamma$ against $f(\gamma)$ when competitive selection is adopted.

Let $\phi(t) = P_x[f(\gamma) = t]$ be the probability distribution function of the fitness values of individuals in population $x$; from Equation 1 we know that

$$E_\gamma = \frac{2}{n}\sum_{\kappa\in\Gamma} n_x(\kappa)p(f(\gamma), f(\kappa)).$$

This can be rewritten, using a Lebesgue integral, as

$$E_\gamma = 2\int_0^\infty p(f(\gamma), t)d\phi(t). \tag{9}$$

We may then substitute any $p$ satisfying Equations 2 or 5 along with a particular distribution $\phi$ into Equation 9 to obtain an expression of $E_\gamma$ in terms of $f(\gamma)$.

For example, choosing $p$ from the family of solutions 3, that is,

$$p(t,u) = 1 - \frac{1}{\frac{t^k}{u^k} + 1} = \frac{t^k}{t^k + u^k}$$

yields

$$E_\gamma = 2\int_0^\infty \frac{f(\gamma)^k d\phi(t)}{f(\gamma)^k + t^k} = 2f(\gamma)^k \int_0^\infty \frac{d\phi(t)}{f(\gamma)^k + t^k}.$$

As parameter $k$ increases, the outcome of a competition becomes less and less dependent on chance and more on the actual fitness value of individuals; the limit of this expression as $k$ goes to infinity, which gives the reproduction rate of $\gamma$ when the outcome of each competition is deterministic, can be calculated by recalling Equation 4:

$$\lim_{k\to\infty} 2\int_0^\infty \frac{f(\gamma)^k d\phi(t)}{f(\gamma)^k + t^k} = \phi(f(\gamma)) + 2\int_0^{f(\gamma)} d\phi(t). \tag{10}$$

Of course a fitness distribution for a finite population is discrete, but, provided that the population size $n$ is large enough, we can approximate it with a continuous probability density function and then solve this integral for various fitness distributions. We will carry out the necessary calculations for a few simple cases below.

## 4.2 Uniform fitness distribution

Suppose that fitness values in the current population are uniformly distributed over the interval $[0, b]$; then, substituting $\phi(t) = 1/b$, for $0 \le t \le b$, in Equation 9 yields

$$E_\gamma = \frac{2}{b} f(\gamma)^k \int_0^b \frac{dt}{f(\gamma)^k + t^k} = 2F\left(1, \frac{1}{k}; 1 + \frac{1}{k}; -\frac{b^k}{f(\gamma)^k}\right), \tag{11}$$

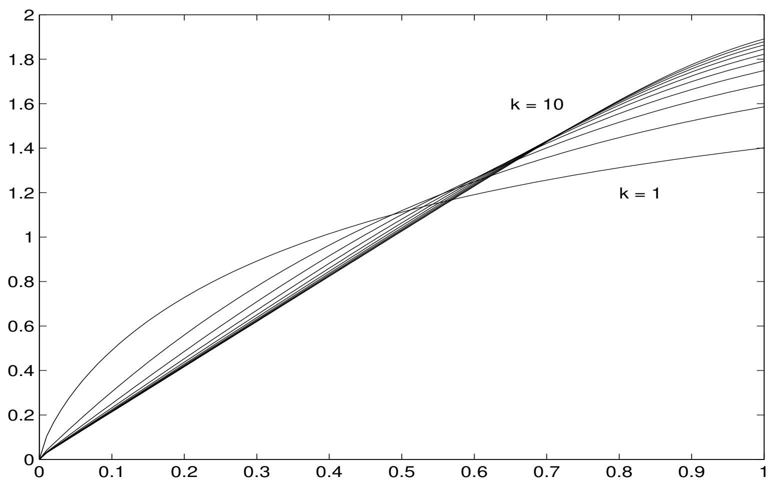


Figure 3: A parametric plot of $E_\gamma$ in the uniform fitness distribution case, for $k = 1, \ldots, 10$.

where

$$F(a, b; c; z) = \sum_{i=0}^{\infty} \frac{(a)_i (b)_i z^i}{(c)_i i!}$$

is the hypergeometric function, also known as the Gauss or Kummer series, and $(a)_i = \Gamma(a+i)/\Gamma(a)$ is the Pochhammer symbol. A parametric plot of Equation 11 is found in Figure 3.

For $k = 1$ Equation 11 becomes

$$E_\gamma = \frac{2}{b} f(\gamma) \ln \left(1 + \frac{b}{f(\gamma)}\right).$$

Thus $E_\gamma = 0$ for $f(\gamma) = 0$; for $f(\gamma) = b$, $E_\gamma = 2 \ln 2 \sim 1.3863$; $E_\gamma = 1$, that is, the occurrences of $\gamma$ are expected to stay constant, when

$$f(\gamma) = \frac{b}{e^{\frac{b}{2}} - 1}.$$

Notice that this value is less than $\frac{b}{2}$, the mean of the uniform distribution: Competitive selection appears to be generous with average individuals and tends to favor them slightly. This is a counterintuitive result, since one would expect that an average individual had exactly a 1/2 probability of winning a competition against another randomly selected individual.

To study what happens to an average fitness individual $\bar{\gamma}$, with $f(\bar{\gamma}) = E_\phi[f] = \frac{b}{2}$, we can substitute $\frac{b}{2}$ to $f(\gamma)$ in Equation 11; after integrating we obtain

$$E_{\bar{\gamma}}(k) = 2F\left(1, \frac{1}{k}; 1 + \frac{1}{k}; -2^k\right).$$

In the same way we are interested in knowing what happens to the best individual $\gamma^*$ in the population, with $f(\gamma^*) = b$: By substituting in Equation 11 and integrating we get

$$E_{\gamma^*}(k) = 2F\left(1, \frac{1}{k}; 1 + \frac{1}{k}; -1\right) = \frac{\psi\left(\frac{k+1}{2k}\right) - \psi\left(\frac{1}{2k}\right)}{k},$$

where $\psi(z) = \Gamma'(z)/\Gamma(z)$ is the digamma function.

The following table lists the values of $E_{\bar{\gamma}}(k)$ and $E_{\gamma^*}(k)$ for $k = 1, \ldots, 10$:

| $k$ | $E_{\bar{\gamma}}(k)$ | $E_{\gamma^*}(k)$ |
|---|---|---|
| 1 | 1.0986 | 1.3863 |
| 2 | 1.1071 | 1.5708 |
| 3 | 1.0900 | 1.6713 |
| 4 | 1.0702 | 1.7339 |
| 5 | 1.0536 | 1.7766 |
| 6 | 1.0410 | 1.8075 |
| 7 | 1.0318 | 1.8310 |
| 8 | 1.0250 | 1.8493 |
| 9 | 1.0202 | 1.8641 |
| 10 | 1.0164 | 1.8762 |

It can be observed that $E_{\bar{\gamma}}(k)$ attains a maximum at $k = 2$, and then it decreases asymptotically to 1 as $k$ increases. This means that the anomaly of an average individual having more than average chances to win tends to disappear as the competition becomes more and more deterministic and the best individual in the population tends to double the number of its occurrences in the next population.

In other words, the uncertainty of the outcome penalizes the best individuals and distributes some of their odds of winning to average individuals, but the worst individuals continue to have very few chances of winning. This is important, in the sense that uncertain outcomes will slow down the overall improvement of a population a little, but they will not able to block it.

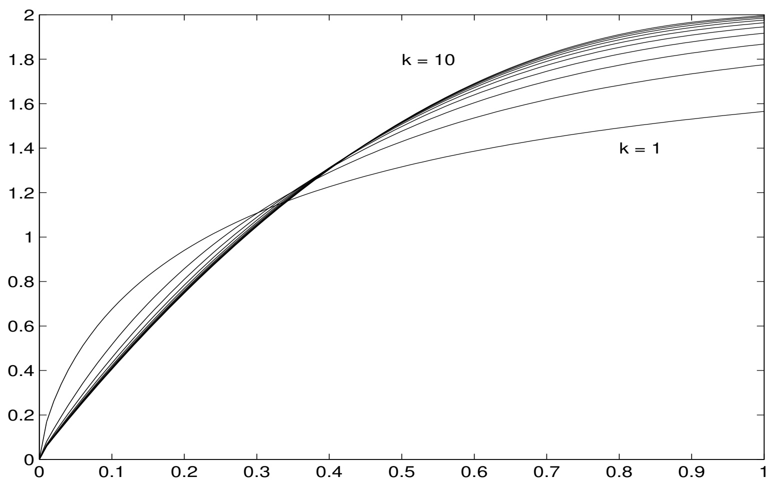


Figure 4: A parametric plot of $E_\gamma$ in the linear fitness distribution case, with $k = 1, \ldots, 10$.

Therefore, if increasing outcome uncertainty allows to substantially reduce computing time, that could result in an overall gain of speed for an evolutionary algorithm.

As $k$ increases, the graph of $E_\gamma$ against $f(\gamma)$ slowly gets flatter and converges to a straight line whose equation can be calculated from Equation 10:

$$E_\gamma = 2\int_0^{f(\gamma)} \frac{dt}{b} = \frac{2}{b} f(\gamma).$$

## 4.3 Linear fitness distribution

Suppose that $\phi(t) = \frac{2}{b}(1 - \frac{t}{b})$ for $t \in [0, b]$ (this could be regarded as a rough approximation of an exponential distribution); then

$$\begin{aligned} E_\gamma &= \frac{4}{b} f(\gamma)^k \int_0^b \frac{1 - t/b}{f(\gamma)^k + t^k} dt = \\ &= 4F\left(1, \frac{1}{k}; 1 + \frac{1}{k}; -\frac{b^k}{f(\gamma)^k}\right) - 2F\left(1, \frac{2}{k}; 1 + \frac{2}{k}; -\frac{b^k}{f(\gamma)^k}\right). \qquad (12) \end{aligned}$$

A parametric plot of Equation 12 is shown in Figure 4.

Here, $f(\bar{\gamma}) = E_\phi[f] = \frac{b}{3}$; substituting in Equation 12 and integrating we obtain

$$E_{\bar{\gamma}}(k) = 4F\left(1, \frac{1}{k}; 1 + \frac{1}{k}; -3^k\right) - 2F\left(1, \frac{2}{k}; 1 + \frac{2}{k}; -3^k\right);$$

in the same way,

$$E_{\gamma^*}(k) = \frac{2}{k}\left[\psi\left(\frac{k+1}{2k}\right) - \psi\left(\frac{k+2}{2k}\right) - \psi\left(\frac{1}{2k}\right) - \psi\left(\frac{1}{k}\right)\right].$$

The following table lists the values of $E_{\bar{\gamma}}(k)$ and $E_{\gamma^*}(k)$ for $k = 1, \ldots, 10$:

| $k$ | $E_{\bar{\gamma}}(k)$ | $E_{\gamma^*}(k)$ |
|---|---|---|
| 1 | 1.1312 | 1.5452 |
| 2 | 1.1537 | 1.7553 |
| 3 | 1.1486 | 1.8484 |
| 4 | 1.1401 | 1.8971 |
| 5 | 1.1330 | 1.9256 |
| 6 | 1.1278 | 1.9438 |
| 7 | 1.1241 | 1.9560 |
| 8 | 1.1214 | 1.9647 |
| 9 | 1.1194 | 1.9710 |
| 10 | 1.1180 | 1.9758 |

As in the uniform distribution case, we observe that $E_{\bar{\gamma}}(k)$ attains a maximum at $k = 2$, and then decreases asymptotically to $\frac{10}{9} = 1.1111\ldots$ as $k$ goes to infinity. Therefore, in this case the anomaly does not disappear and, even with deterministic outcomes, average individuals will have more than equal chances of winning a competition against another randomly selected individual. Those excess chances are taken away from the very worst individuals, rather than from the best ones.

As $k$ increases the graph converges slowly to a convex quadratic curve whose equation is calculated from Equation 10:

$$E_\gamma = \frac{4}{b}f(\gamma) - \frac{2}{b^2}f(\gamma)^2.$$

## 4.4 Other fitness distributions

Of course the uniform and linear fitness distributions studied in the previous subsections are merely theoretical, but allowed us to calculate the integrals

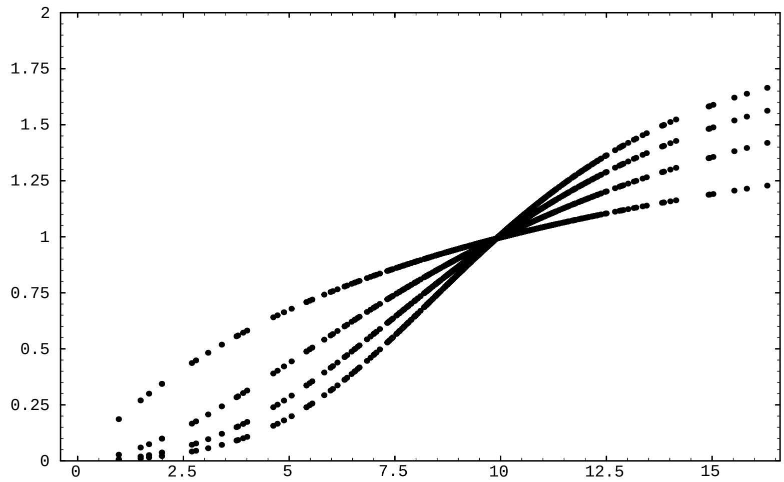


Figure 5: A plot of $E_\gamma$ in a bell-shaped fitness distribution case, with $k = 1, \ldots, 4$, obtained through simulations with a population size of 500 individuals.

involved; more realistic situations can be investigated numerically, by means of simulations.

In fact, the typical situation that can be observed during the resolution of different optimization problems by means of evolutionary algorithms features roughly bell-shaped fitness distributions; in general these "real world" distributions may neither be symmetrical nor unimodal, but nevertheless their overall shape is closer to a bell than to a triangle or a rectangle.

Figure 5 shows simulation curves for a bell-shaped fitness distribution with mean $f(\bar{\gamma}) = 10$, and probability density function

$$\phi(t) = \frac{1}{\pi\big((t-10)^2+1\big)}.$$

The choice of this particular distribution instead of, say, a normal distribution is not dictated by other considerations than convenience, since it is supposed to offer only a summary indication on the behavior of $E_\gamma$ for a large class of realistic situations.

It is still possible, by Equation 4, to obtain the limit behavior for this distribution, whose analytical form is

$$E_\gamma = \frac{2}{\pi}(\arctan 10 - \arctan(10 - x)),$$

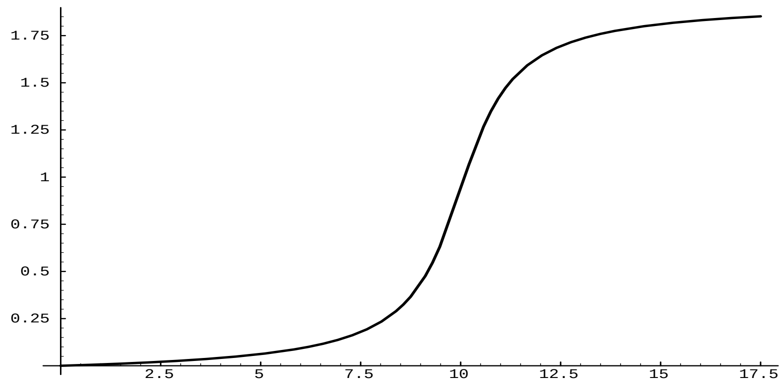


Figure 6: The limit behavior of $E_\gamma$ as $k$ goes to infinity in the bell-shaped distribution case with mean fitness $f(\bar{\gamma}) = 10$.

corresponding to the graph in Figure 6: This result is qualitatively representative of a broad class of bell-shaped and quasi-bell-shaped fitness distributions.

# 5 Conclusions

From the analysis carried out in the previous section, we have seen that the behavior of competitive selection depends both on the choice of the function relating the probability of a win with the prior fitness of two competitors and on the distribution of fitness values in a population; in either case, however, the expected occurrences of an above average genotype in the next generation increase at the expense of below average genotypes; indeed we got that there is a small bias in favor of average individuals. Provided that the probability of schema disruption due to mutation and cross-over is small enough, something similar to Holland's Schema Theorem [16] can be proven. Actually, experiments performed by the author have shown that an evolutionary algorithm making use of this competitive selection scheme performs at its best when the cross-over rate is rather low, i.e. less than 20%. This can be explained by the fact that some time has to be allowed to compensate for statistical deviations in the outcomes of competitions, before the genetic material of an individual is exchanged or modified.

In addition, competitive selection, like other selection schemes, does not exhibit that property of fitness proportionate selection that many judge undesirable, which allows individuals much fitter than the average to rapidly take over the whole population, thus causing premature convergence; on the other hand, in the lower part of the fitness distribution, competitive selection behaves more or less like fitness proportionate selection.

But the important thing about competitive selection is that

1. it provides us a way of circumventing the difficult and often critical task of defining an appropriate fitness function, because all is needed is just an ordinal criterion to decide which one of two individuals is better, and
2. for some problems it suggests randomization as a suitable way to reduce computing time while at the same time upholding the requirements for evolution.

# 6 Acknowledgements

The author would like to thank Prof. Gianni Degli Antoni for his patient support and incitement and Prof. Vincenzo Capasso for the useful discussions and criticism.

A warm acknowledgement goes to three anonymous reviewers that gave a very useful feedback and made substantial suggestions to improve the clarity of the presentation.

# References

[1] S. Arnone, A. Loraschi, and A. Tettamanzi. A genetic approach to portfolio selection. *Neural Network World, International Journal on Neural and Mass-Parallel Computing and Information Systems*, 3(6):597–604, 1993.

[2] T. Bäck and H.-P. Schwefel. An overview of evolutionary algorithms for parameter optimization. *Evolutionary Computation*, 1(1):1–23, Spring 1993.

[3] J. Baker. Adaptive selection methods for genetic algorithms. In J. J. Grefenstette, editor, *Proceedings of the First International Conference on Genetic Algorithms*, pages 101–111, Hillsdale, NJ, 1985. Lawrence Erlbaum Associates.

[4] J. Baker. Reducing bias and inefficiency in the selection algorithm. In John J. Grefenstette, editor, *Genetic algorithms and their application: Proceedings of the Second International Conference on Genetic Algorithms*, pages 14–21, Hillsdale, NJ, 1987. Lawrence Erlbaum Associated.

[5] W. Banzhaf. Genetic programming for pedestrians. In S. Forrest, editor, *Proceedings of the Fifth International Conference on Genetic Algorithms*, page 628, San Mateo, CA, 1993. Morgan Kaufmann.

[6] G. Berry and G. Boudol. The chemical abstract machine. *Theoretical Computer Science: the Journal of the EATCS*, 96(1):217, 1992.

[7] A. Brindle. Genetic agorithms for function optimization. Technical Report TR81-2, Department of Computer Science, University of Alberta, Edmonton, 1981.

[8] R. Collins and D. Jefferson. Selection in massively parallel genetic algorithms. In R. K. Belew and L. B. Booker, editors, *Proceedings of the Fourth International Conference on Genetic Algorithms*, pages 249–256, San Mateo, CA, 1991. Morgan Kaufmann.

[9] K. Deb and D. E. Goldberg. A comparative analysis of selection schemes used in genetic algorithms. In Gregory J. E. Rawlins, editor, *Foundations of Genetic Algorithms*, pages 69–93, San Mateo, CA, 1991. Morgan Kaufmann.

[10] M. Dorigo and V. Maniezzo. Parallel genetic algorithms: Introduction and overview of current research. In J. Stenders, editor, *Parallel Genetic Algorithms: Theory and Applications*, pages 5–42, Amsterdam, 1992. IOS Press.

[11] L. J. Fogel, A. J. Owens, and M. J. Walsh. *Artificial Intelligence through Simulated Evolution.* John Wiley & Sons, New York, 1966.

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

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

[14] J. Grefenstette and J. Baker. How genetic algorithms work: A critical look at implicit parallelism. In J. D. Schaffer, editor, *Proceedings of*

*the Third International Conference on Genetic Algorithms*, pages 20–27, San Mateo, CA, 1989. Morgan Kaufmann.

[15] P. J. B. Hancock. An empirical comparison of selection methods in evolutionary algorithms. In T. C. Fogarty, editor, *Evolutionary Computing: AISB Workshop*, volume 865 of *Lecture Notes in Computer Science*, pages 80–94, Berlin, 1994. Springer-Verlag.

[16] J. H. Holland. *Adaptation in Natural and Artificial Systems.* The University of Michigan Press, Ann Arbor, 1975.

[17] J. R. Koza. *Genetic Programming: on the programming of computers by means of natural selection.* The MIT Press, Cambridge, Massachussets, 1993.

[18] Z. Michalewicz. *Genetic Algorithms + Data Structures = Evolution Programs.* Springer-Verlag, Berlin, 1992.

[19] R. Milner. *Communication and Concurrency.* International Series in Computer Science. Prentice-Hall, Englewood Cliffs, 1976.

[20] H. Mühlenbein. Parallel genetic algorithms, population genetics and combinatorial optimization. In J. D. Schaffer, editor, *Proceedings of the Third International Conference on Genetic Algorithms*, pages 416–421, San Mateo, CA, 1989. Morgan Kaufmann.

[21] H. Mühlenbein and D. Schlierkamp-Voosen. The science of breeding and its application to the breeder genetic algorithm (bga). *Evolutionary Computation*, 1(4):335–360, Winter 1993.

[22] I. Rechenberg. *Evolutionsstrategie: Optimierung technischer Systeme nach Prinzipien der biologischen Evolution.* Fromman-Holzboog Verlag, Stuttgart, 1973.

[23] H.-P. Schwefel. *Numerical optimization of computer models.* Wiley, Chichester; New York, 1981.

[24] A. Tettamanzi. An evolutionary algorithm for fuzzy controller synthesis and optimization. Submitted to the 1995 IEEE International Conference on System, Man and Cybernetics, Vancouver, Canada, October 22–25, 1995.

[25] A. Tettamanzi. A distributed model for selection in evolutionary algorithms. Technical Report 110-94, Dip. di Scienze dell'Informazione – Università degli Studi di Milano, Milano, Italy, 1994.

[26] A. Tettamanzi. Genetic programming without fitness. In J. Koza, editor, *Late Breaking Papers at the Genetic Programing 1996 Conference*, pages 193–195. Stanford University Press, Palo Alto, CA, 1996.