On the Study of Some Theoretical Aspects of Genetic Algorithms

Technical Report EVOCINV-02-2003
María Margarita Reyes Sierra and Carlos A. Coello Coello
CINVESTAV-IPN
Evolutionary Computation Group
Departamento de Ingeniería Eléctrica
Sección de Computación
Av. IPN No. 2508
Col. San Pedro Zacatenco
México, D.F. 07300
`mreyes@computacion.cs.cinvestav.mx`
`ccoello@cs.cinvestav.mx`

March 25, 2003

Abstract

In this technical report we study the theoretical results reported so far in the specialized literature regarding to convergence of evolutionary algorithms (genetic algorithms in particular). We only focus our study to those cases in which the model adopted was based on Markov chains. In the work reported here, we will present and extend the model proposed by Günter Rudolph in 1994 to prove convergence of an elitist genetic algorithm. This extended model is then used to predict the expected convergence time of an elitist genetic algorithm with minimum parameters. The theoretical results derived are then empirically corroborated through a set of experiments. Finally, we also discuss the capabilities of a genetic algorithm for optimizing multiple objectives and we propose some possible paths for future research in such area.

1 Introduction

There are optimization problems whose search space is so large that even the most efficient mathematical programming techniques available to solve them require exponential time. It is precisely in these cases in which heuristics have particular relevance. From these heuristics, evolutionary algorithms have shown to be very advantageous in many real-world applications, producing at least sub-optimal solutions in very complex problems [5, 2].

However, due to the stochastic nature of evolutionary algorithms, their behavior is not fully understood. Basically, it turns out to be quite difficult (or even impossible) to determine the conditions under which an arbitrary evolutionary algorithm may succeed or fail in solving a problem. This is the case of genetic algorithms (GAs) which is the type of evolutionary algorithm with which we will be dealing in this document.

Genetic algorithms are a heuristic search technique inspired on natural evolution (i.e., the survival of the fittest). Although GAs were originally conceived as machine learning techniques [6], they have become increasingly popular as optimizers [2].

In their origin, the genetic algorithm (now called "classical") was applied to single-objective optimization problems [5]. Rudolph proved in the mid-1990s convergence of this simple GA to the global optimum of a given function, under certain conditions [15]. This work gave some of the desired theoretical foundations to the behavior of GAs.

In the real world, however, we frequently face problems with multiple objective functions and associated constraints. The potential of genetic algorithms in these problems was hinted long ago [3], and several extensions of a GA were developed in order to deal with multiobjective optimization problems. The theoretical study of the different multiobjective evolutionary algorithms in current use has been fairly limited, but it exists. In this document, we will summarize the theoretical work done on convergence of multiobjective evolutionary algorithms (particularly genetic algorithms) towards the Pareto optimal set of a problem.

Since the model that we will study is based on Markov chains, Section 2 introduces this mathematical tool. In Section 3, the simple (classical) GA is studied, we introduce its mathematical model and we prove its convergence to the global optimum of a given problem. Section 4 extends the model of the previous section so that the necessary convergence time for this algorithm can be estimated. In Section 5, we study convergence of a multiobjective genetic algorithm. Finally, in Section 6, we provide our conclusions and some possible paths for future research.

2 Markov Chains

The analysis of the GAs that will be presented in this document is based on probability theory, specially in Markov chains. Therefore, in order to make this document self-contained, we provide in this section the basics on Markov chains that we will use later on. The material of this section was extracted from [16, 9].

2.1 Basic Definitions

Definition 2.1 *If $\mathcal{S} \neq \emptyset$ is a finite set and $\{X_t : t \in \mathbb{N}\}$ is a sequence of random variables with values in $\mathcal{S}$ with the property:*

$$\mathsf{P}\{X_{t+1} = j | X_t = i, X_{t-1} = i_{t-1}, ..., X_0 = i_0\} = \\ \mathsf{P}\{X_{t+1} = j | X_t = i\} =: p_{ij}$$

for all $t \geq 0$ *and* $i, j \in \mathcal{S}$*, then the sequence* $\{X_t : t \in \mathbb{N}\}$ *is called a* **finite Markov chain** *with state space* $\mathcal{S}$*.*

The number p_{ij} *is called* **transition probability** *of the state* i *to the state* j *in one step. Since we are assuming that such probabilities are independent of* $t \in \mathbb{N}$*, is said that the chain is* **homogeneous***.*

Since $\mathcal{S}$ is finite, the transition probabilities can be gathered in a **transition matrix** $P = (p_{ij})_{i,j \in \mathcal{S}}$. Note that $\sum_j p_{ij} = 1$ for all $i \in \mathcal{S}$.

The row vector $\delta(t)$ with components $\delta_i(t) = \mathsf{P}\{X_t = i\}$ for all $i \in \mathcal{S}$, denotes the distribution of the Markov chain in the step $t \geq 0$. This distribution can be calculed iteratively, since

$$\delta(t) = \delta(t-1)P = \delta(0)P^t, \text{ for all } t \geq 0.$$

This way, a homogeneous Markov chain is completely determined by its initial distribution $\delta(0)$ and its transition matrix P.

Definition 2.2

- *A matrix* $P : n \times m$ *is said to be* **nonnegative** *(*$P \geq 0$*) if* $p_{ij} \geq 0$ *and* **positive** *(*$P > 0$*) if* $p_{ij} > 0$ *for all* $i = 1, ..., n$ *and* $j = 1, ..., m$*.*

- *A nonnegative square matrix is called* **stochastic** *if the sum of each of its rows is equal to one. Thus, the transition matrices are stochastic.*

- *An stochastic matrix* P *is* **primitive** *if*

  $$\exists k \in \mathbb{N} : P^k \textit{ is positive } (P^k > 0)$$

- *It is* **irreducible** *if*

  $$\forall i, j \in \mathcal{S} : \exists k \in \mathbb{N} : p_{ij}(k) > 0,$$

  where $p_{ij}(k)$ *denotes the element* (i, j) *of* P^k*. Therefore, every positive matrix* P *is primitive and every primitive matrix is irreducible.*

- *A matrix* P *is* **reducible** *(of course if it is not irreducible and) if it can be arranged in the form:*

  $$\begin{pmatrix} \mathbf{C} & \mathbf{0} \\ \mathbf{R} & \mathbf{T} \end{pmatrix}$$

  with square matrices $\mathbf{C}$ *and* $\mathbf{T}$*.*

- *Finally, a stochastic matrix* P *is* **diagonal-positive** *if each element of its diagonal is positive, is* **column allowable** *if it has at least one positive element in each column and is* **stable** *if it has identical rows.*

In the following, some results that will be useful in later sections are presented.

Lemma 2.3 *Let P, Q and R stochastic matrices, where Q is positive and R is column allowable. Then the product matrix PQR is positive.*

Proof. [9].

Theorem 2.4 *Let P a primitive stochastic matrix. Then P^k converges when $k \to \infty$ to a positive stable stochastic matrix $P^\infty = \mathbf{1}'p^\infty$, where $\mathbf{1}'$ is a column vector of 1's and $p^\infty = p^0 \cdot \lim_{k\to\infty} P^k = p^0 P^\infty$ has positive entries and is unique regardless of the initial distribution p^0.*

Proof. [9].

Theorem 2.5 *Let $P_{n\times n}$ a reducible stochastic matrix, where $C : m \times m$ is a primitive stochastic matrix and $R, T \neq 0$. Then:*

$$P^\infty = \lim_{k\to\infty} P^k = \lim_{k\to\infty} \begin{pmatrix} C^k & 0 \\ \sum_{i=0}^{k-1} T^i R C^{k-i} & T^k \end{pmatrix} = \begin{pmatrix} C^\infty & 0 \\ R_\infty & 0 \end{pmatrix}$$

is a stable stochastic matrix with $P^\infty = \mathbf{1}'p^\infty$, where $\mathbf{1}'$ is a column vector of 1's and $p^\infty = p^0 P^\infty$ is unique regardless of the initial distributrion p^0 and, moreover, p^∞ satisfies: $p_i^\infty > 0$ for $1 \leq i \leq m$ and $p_i^\infty = 0$ for $m < i \leq n$.

Proof. [9].

2.2 Clasification of States and Chains

Consider a Markov chain with finite state space $\mathcal{S} \neq \emptyset$. In this section the states of a Markov chain are classified accordingly to if it is possible to go from a given state to another state.

Definition 2.6 *We say that the state i* **leads to** *state j and write $i \to j$ if and only if $p_{ij}^k > 0$ for an $k \geq 1$. If $i \to j$ and $j \to i$ we say that state i* **communicates with** *state j and write $i \leftrightarrow j$.*

Beginning with the previous definition, the states are classified into "equivalence classes". Two states are in the same class if they are "comunicated", i.e. if the process can go from one state to the another and viceversa.

The equivalence classes are classified like **ergodic** sets (so called **recurrent**) or **transient** sets. This way, the corresponding states in those classes are called ergodic states and transient states, respectively.

For each finite Markov chain there must be always at least one ergodic set; however, there not need to be transient sets.

Once that a chain leaves a transient set, can never come back to it; while once that it enters into an ergodic set, never can leave it.

In particular, if an ergodic set contains just one state, this state is called an **absorbing** state, because once into it, the Markov chain will stay there for ever.

Theorem 2.7 *A state i is absorbing if and only if $p_{ii} = 1$.*

Proof. [10].

Ergodic sets can be classified in two ways:

1. **Regular**: In this case, there is just one cycle of states, and regardless of the state in which the chain begins, after a sufficient amount of time, the chain can be in any state of the class (primitive transition matrix).
2. **Cyclic** (or **periodic**): In this case, the set is divided in d distinct cycles such that given an initial state, the chain will go through the distinct cycles until it return to the cycle in wich it begins, after exactly d steps.

The Markov chains can be classified according to if it contains transient sets or not:

I **Chains without transient sets**
Without losing generality, let's asume that in this case there is just one ergodic set, i.e. all the set of states of the chain is an ergodic set. A chain that consists of a unique ergodic set is called **ergodic chain**, and it can coindice with some of the following cases:

  I-A The ergodic set is regular.
  I-B The ergodic set is cyclic.

II **Chains with transient sets**
In this case, the chain is moving towards the ergodic sets. The probability that the process enters into an ergodic set tends to 1. In this case we can again classify this kind of chains based in the characteristics of its ergodic sets.

  II-A All the ergodic sets are unit sets. This type of chain is called **absorbing chain**, because it will eventually be trapped into an absorbing state.
  II-B All the ergodic sets are regular but not unit sets.
  II-C All the ergodic sets are cyclic.
  II-D There exist ergodic sets that are both regular and cyclic.

2.3 Absorbing Chains

The absorbing chains are of special interest for us, therefore in this section some important properties of them will be presented [10].

Theorem 2.8 *For each finite absorbing chain, no matter the state in wich it starts, the probability that the process is in an absorbing state after n steps tends to 1 as n tends to infinity.*

Proof. [10].

It is important to consider the canonical form of the transition matrix of a Markov chain. Let's assume that we have s transient states and $r - s$ ergodic states, and that we cluster all the transient sets and all the ergodic sets together, the resulting form is:

$$P = \begin{array}{c} \begin{array}{cc} r-s & s \end{array} \\ \left(\begin{array}{cc} S & O \\ R & Q \end{array} \right) \end{array} \begin{array}{l} \\ \}r-s \\ \}s \end{array}$$

The region O consists completely of zeros. The matrix $Q_{s\times s}$ represents the chain while it is in transient states, the matrix $R_{s\times(r-s)}$ represents the transition from the transient states to ergodic states and the matrix $S_{(r-s)\times(r-s)}$ represents the chain once that it is into an ergodic state.

If we consider an absorbing chain, we have that by definition $S = I_{(r-s)\times(r-s)}$, so its canonical form is:

$$P = \begin{array}{c} \begin{array}{cc} r-s & s \end{array} \\ \left(\begin{array}{cc} I & O \\ R & Q \end{array} \right) \end{array} \begin{array}{l} \\ \}r-s \\ \}s \end{array}$$

From Theorem 2.5 we can see that the powers of Q tend to O.

Definition 2.9 *For an absorbing Markov chain we define the* **fundamental matrix** *to be* $N = (I - Q)^{-1}$.

Definition 2.10 *We define* $\mathbf{n}_j$ *to be a function whose value is the total number of times that the process is in state* s_j*. (This definition is valid just for transient states* s_j*). We define* $\mathbf{u}_j^k$ *to be a function that is 1 if the process is in state* s_j *in the step* k*, and is 0 otherwise.*

It is easy to see that:

$$\mathbf{n}_j = \sum_{k=0}^{\infty} \mathbf{u}_j^k$$

Let $\mathbf{T}$ the set of transient states of the Markov chain. If we denote with $\mathsf{E}_i[\mathbf{n}_j]$ the expected value of $\mathbf{n}_j$ assuming that the process starts in the state s_i, we have the next result:

Theorem 2.11 $\{\mathsf{E}_i[\mathbf{n}_j]\} = N$

Proof. [10].

Definition 2.12 *Let* $\mathbf{t}$ *be a function whose value is given by the number of steps (including the initial state) in wich the process is in a transient state.*

If the process starts in an ergodic state then $\mathbf{t} = 0$. If the process starts in a transient state, then $\mathbf{t}$ gives us the total number of necessary steps for reaching an ergodic state. In an absorbing chain, this is the *time to absorption*.

Let ξ a column vector with all entries equal to 1.

Theorem 2.13 $\{\mathsf{E}_i[\mathbf{t}]\} = N\xi$

Proof. It is easy to see that

$$\mathbf{t} = \sum_{s_j \in \mathbf{T}} \mathbf{n}_j$$

Thus,

$$\{\mathsf{E}_i[\mathbf{t}]\} = \{\mathsf{E}_i[\sum_{s_j \in \mathbf{T}} \mathbf{n}_j]\} = \{\sum_{s_j \in \mathbf{T}} \mathsf{E}_i[\mathbf{n}_j]\} = N\xi.$$

■

Theorem 2.14 $\{\mathsf{V}_i[\mathbf{t}]\} = (2N - I)N\xi - (N\xi)^2$

Proof. [10].

3 The Simple Genetic Algorithm for Global Optimization

Genetic Algorithms are commonly used for solving optimization problems like: $max\{f(b) | b \in \mathbb{B}^l = \{0,1\}^l\}$ assuming that $0 < f(b) < \infty$ for all $b \in \mathbb{B}^l$ and $f(b) \neq const.$

In this section we start our study of the convergence of the GAs. We will consider mainly two cases, the GA **without** elitism, also called Simple GA (SGA) and the GA **with** elistism (EGA).

3.1 Convergence Studies

3.1.1 Genetic algorithm without elitism

In [15], Rudolph models the SGA by means of a finite homogeneous Markov chain. Each state i of the Markov chain corresponds to a possible SGA population, so that the state space is $\mathcal{S} = \mathbb{B}^{nl}$ where n is the number of individuals in the population and l is the lenght of each individual. We define $\pi_k^t(i)$ to be the individual k of the population i in the step t.

Given the nature of the SGA, the transition matrix $\mathbf{P}$ that represents it is defined by:

$$\mathbf{P} = \mathbf{CMS},$$

where $\mathbf{C}$, $\mathbf{M}$ and $\mathbf{S}$ are the transition matrices of the operators of crossover, mutation and selection, respectively.

When *uniform mutation* is used the elements of $\mathbf{M}$ are

$$m_{ij} = p_m^{H_{ij}}(1 - p_m)^{N - H_{ij}} > 0,$$

where p_m is the mutation probability of the SGA, H_{ij} is the Hamming distance between the states i and j, and $N = nl$. This way, we conclude that $\mathbf{M}$ *is positive*.

On the other hand, since what the selection operator does is to give us pairs of individuals either for passing them intact to the next population or for generating (with a certain probability) two new individuals through the crossover operator, the transition matrix of this operator “sorts” the individuals and leaves them ready for generating the next population.

The use of either a proportional or a tournament selection operator [16] determines the existence of a strictly positive probability that the population remains intact, and assures that the diagonal elements s_{ii} of the transition matrix of that operator are positive, thus, the matrix $\mathbf{S}$ *is column allowable*.

In summary, we have that the matrix $\mathbf{M}$ is positive and $\mathbf{S}$ is column allowable. Then, from Lemma 2.3, the matrix $\mathbf{P} = \mathbf{CMS}$ *is positive and therefore primitive*.

Next, we present the corresponding definition of convergence of a SGA [15]:

Definition 3.1 *Let* $Z_t = max\{f(\pi_k^t(i))|k = 1, ..., n\}$ *be a sequence of random variables denoting the best fitness into the population* i *in the step* t*. A genetic algorithm converges to the global optimum, if and only if:*

$$lim_{t\to\infty} P\{Z_t = f^*\} = 1 \tag{3.1}$$

where $f^* = max\{f(b)|b \in \mathbb{B}^l\}$.

This way, we can see that the SGA converges to the global optimum if the probability of being in the population tends to 1 when the number of iterations tends to infinity.

Thus, given the previous definition and using the Theorem 2.4 Rudolph [15] shows that the SGA does not converge:

Theorem 3.2 *The SGA with primitive transition matrix does not converge to the global optimum.*

Proof. Let $i \in \mathcal{S}$ any state in which $max\{f(\pi_k^t(i))|k = 1, ..., n\} < f^*$ and $p_i(t)$ the probability that the SGA is into such state i in the step t. It is clear that $P\{Z_t \neq f^*\} \geq p_i(t) \Leftrightarrow P\{Z_t = f^*\} \leq 1 - p_i(t)$. From Theorem 2.4 the probability of that the SGA is in state i converges to $p_i(\infty) > 0$. Therefore:

$$lim_{t\to\infty} P\{Z_t = f^*\} \leq 1 - p_i(\infty) < 1,$$

i.e. the condition (3.1) is not satisfied. ■

Theorem 3.2 shows that given Theorem 2.4, the transition matrix $\mathbf{P}$ of the SGA converges to a positive matrix, and thus the probability of been into a non optimal state is strictly positive as the number of iterations increase. So the probability of remaining in an optimal state is not 1 in the limit.

3.1.2 Elitist genetic algorithm

Rudolph [15] argues that when the SGA is applied to real-world problems, it normally keeps the best solution found so far along the evolutionary process (this is called elitism in evolutionary computation). Therefore, Rudolph argues that elitism is an important component that has to be considered when modeling a SGA.

Thus, we will now consider to add to the population of the SGA a *super individual* that will not take part of the evolutionary process and that for easiness of notation will be placed in the first lefthand position, i.e. we will be able to access it by $\pi_0(i)$. We will call this new version Elitist Genetic Algorithm (EGA).

The cardinality of the corresponding state space of the Markov chain grows now from 2^{nl} to $2^{(n+1)l}$ since we have 2^l possibles *super individuals* and for each of them we have 2^{nl} possible populations.

The elitist operator will be represented by the matrix $\mathbf{E}$; this matrix is going to update a state that contains an individual better than its current *super individual* replacing them by that individual.

In particular, let:

$$i = (\pi_0(i), \pi_1(i), \pi_2(i), ..., \pi_n(i)) \in \mathcal{S}$$

$\pi_0(i)$ is the super individual of the population (state) i. Now, let
b =argmax$\{f(\pi_k(i))|k = 1, ..., n\} \in \mathbb{B}^l$ be the best individual of the population i excluding the *super individual* and:

$$j \overset{\text{def}}{=} (b, \pi_1(i), \pi_2(i), ..., \pi_n(i)) \in \mathcal{S}$$

then:

$$e_{ij} = \begin{cases} 1 & \text{if } f(\pi_0(i)) < f(b) \\ 0 & \text{otherwise.} \end{cases}$$

The new transition matrix for the EGA results of the product of a matrix composed by 2^l matrices $\mathbf{P}$, one for each possible *super individual* and placed in such a way that the higher the position, the better is the super individual, and the matrix of the elitism operator $\mathbf{E}$:

$$\begin{aligned}
\mathbf{P}^+ &= \begin{pmatrix} \mathbf{P} & & & \\ & \mathbf{P} & & \\ & & \ddots & \\ & & & \mathbf{P} \end{pmatrix} \begin{pmatrix} \mathbf{E_{11}} & & & \\ \mathbf{E_{21}} & \mathbf{E_{22}} & & \\ \vdots & \vdots & \ddots & \\ \mathbf{E_{2^l,1}} & \mathbf{E_{2^l,2}} & \cdots & \mathbf{E_{2^l,2^l}} \end{pmatrix} \\
&= \begin{pmatrix} \mathbf{PE_{11}} & & & \\ \mathbf{PE_{21}} & \mathbf{PE_{22}} & & \\ \vdots & \vdots & \ddots & \\ \mathbf{PE_{2^l,1}} & \mathbf{PE_{2^l,2}} & \cdots & \mathbf{PE_{2^l,2^l}} \end{pmatrix}
\end{aligned}$$

The structure showed of the matrix $\mathbf{P}^{+}$ is due to the fact that the populations are sorted in descending way according to the fitness of their super individual. This way, the blank spaces represent zeros since it is not possible to go from a state to another with a super individual of lower fitness.

The conclusion is that $\mathbf{PE_{11}} = \mathbf{P}$ since such matrices correspond with the populations that have as a super individual the optimum f^*.

On the other hand, making the following definitions:

$$\mathbf{R} = \begin{pmatrix} \mathbf{PE_{21}} \\ \vdots \\ \mathbf{PE_{2^l,1}} \end{pmatrix} \mathbf{T} = \begin{pmatrix} \mathbf{PE_{22}} & & \\ \vdots & \ddots & \\ \mathbf{PE_{2^l,2}} & \cdots & \mathbf{PE_{2^l,2^l}} \end{pmatrix}$$

we conclude that the matrix $\mathbf{P}^{+}$ is reducible to the form:

$$\mathbf{P}^{+} = \begin{pmatrix} \mathbf{P} & \mathbf{0} \\ \mathbf{R} & \mathbf{T} \end{pmatrix}.$$

Thus, we have the next theorem:

Theorem 3.3 *The EGA converges to the global optimum.*

Proof. The submatrix $\mathbf{P}$ contains the transition probabilities between global optimum states. Since $\mathbf{P}$ is a primitive stochastic matrix and $\mathbf{R}$, $\mathbf{T} \neq 0$, Theorem 2.5 guarantees that the probability of remaining in a nonoptimal state converges to zero. Therefore, the probability of remaining in a global optimum state converges to 1. ■

Then, we have shown that the EGA converges, i.e. a Genetic Algorithm that uses elitism converges to the global optimum.

4 Convergence Time

The problem of characterizing the behavior of a GA is complex since it varies with the application domain as well as with the implementation parameters adopted for the GA.

In this chapter will show a model based on Markov chains that is used to estimate the convergence time of a simple genetic algorithm.

The preliminary work reviewed includes the models developed by Carol A. Ankenbrandt [1] and Sushil J. Louis & Gregory J. Rawlins [11]. Such models are based on convergence of alleles and on Hamming distances, respectively.

4.1 Model based on Markov Chains

In Section 2, we showed some results on the fundamental matrix of a Markov chain. As we saw, such a matrix can be used to compute the expected convergence time of the chain. In this section, we will apply such results to the corresponding transition matrix of the EGA.

Rudolph [15] showed that the matrix of the EGA has the form:

$$\mathbf{P^+} = \begin{pmatrix} \mathbf{P} & \mathbf{0} \\ \mathbf{R} & \mathbf{T} \end{pmatrix}$$

where $\mathbf{P}$ is the transition matrix of the SGA and:

$$\mathbf{R} = \begin{pmatrix} \mathbf{PE_{21}} \\ \vdots \\ \mathbf{PE_{2^l,1}} \end{pmatrix} \mathbf{T} = \begin{pmatrix} \mathbf{PE_{22}} & & \\ \vdots & \ddots & \\ \mathbf{PE_{2^l,2}} & \cdots & \mathbf{PE_{2^l,2^l}} \end{pmatrix}$$

where $\mathbf{E_{ij}}$ are the corresponding blocks of the matrix of the elitist operator $\mathbf{E}$.

Since in Rudolph's model the matrix $\mathbf{P}$ corresponds with the populations whose super individual is the global optimum, we can consider that when the chain is in one of those states, the search process has finished. Any further change in the population can be ignored because the super individual will not be any longer modified. Therefore, we can rewrite the matrix like:

$$\mathbf{P^+} = \begin{pmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{R} & \mathbf{T} \end{pmatrix}$$

We can see clearly now that the Markov chain corresponding to the EGA is absorbent. According to Definition 2.9, the fundamental matrix that interests us in this case is:

$$\mathbf{N} = (I - \mathbf{T})^{-1}$$

4.1.1 Study of the matrix of the EGA ($\mathbf{P^+}$)

Since our objective is to know the fundamental matrix $\mathbf{N}$, we will start by studying the structure of the block $\mathbf{T}$. As we saw, once we have the matrix $\mathbf{P}$ we proceed to construct the matrix $\mathbf{P^+}$ in the following way:

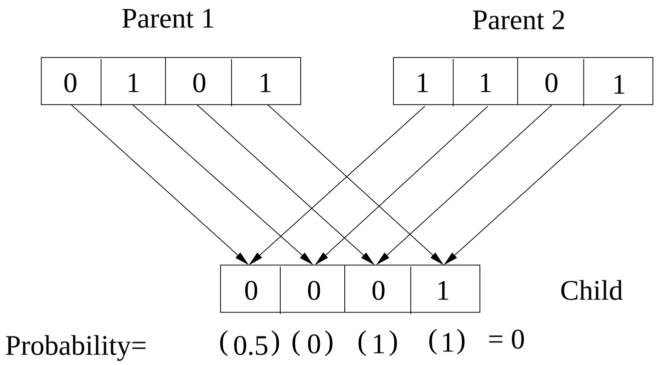
$$\mathbf{P^+} = \begin{pmatrix} \mathbf{P} & & & \\ & \mathbf{P} & & \\ & & \ddots & \\ & & & \mathbf{P} \end{pmatrix} \begin{pmatrix} \mathbf{E_{11}} & & & \\ \mathbf{E_{21}} & \mathbf{E_{22}} & & \\ \vdots & \vdots & \ddots & \\ \mathbf{E_{2^l,1}} & \mathbf{E_{2^l,2}} & \cdots & \mathbf{E_{2^l,2^l}} \end{pmatrix}$$

$$= \begin{pmatrix} \mathbf{PE_{11}} & & & \\ \mathbf{PE_{21}} & \mathbf{PE_{22}} & & \\ \vdots & \vdots & \ddots & \\ \mathbf{PE_{2^l,1}} & \mathbf{PE_{2^l,2}} & \cdots & \mathbf{PE_{2^l,2^l}} \end{pmatrix}$$

Matrix P

In this section we will study the elements of the matrix $\mathbf{P}$. As we know, this matrix is the result of the product:

$$\mathbf{P} = \mathbf{CMS}$$



= 0.5 (0 x 0 + 1 x 0) 0.5 (1 x 0 + 1 x 0)
0.5 (0 x 0 + 0 x 0) 0.5 (1 x 1 + 1 x 1)
= 0.5 (1+0) 0.5(0+0) 0.5(1+1) 0.5(1+1)
= 0.5 (1) 0.5(0) 0.5(2) 0.5(2)
= (0.5) (0) (1) (1) = 0

Figure 1: Example of the way in which uniform crossover works. The symbol $\oplus$ is represented by the letter **x**.

This is the reason why the elements of each of the corresponding matrices will be specified next.

Elements of the matrix of crossover

The elements of 2 types of crossover were modeled: uniform and single-point. For that sake, the following operator was defined:

$\oplus$	0	1
0	1	0
1	0	1

The operator $\oplus$ is nothing more than the negation of an $or - exclusive$, and it will be applied between the bits corresponding to a fixed position of a parent and a possible child. Thus, if the bits are equal, the result is 1 and 0 otherwise.

I Uniform crossover:

When we perform uniform crossover (assuming a crossover percentage of 0.5) each bit of a new child has 0.5 of probability of being equal to the corresponding bit of each one of both parents. Using this fact the following formula was developed:

$$c_{ij} = \prod_{q=1}^{n} (0.5)^{l} \prod_{r=1}^{l} \sum_{s=\phi(q)-1}^{\phi(q)} \pi_s^r(i) \oplus \pi_q^r(j)$$

where $\phi(q) = 2\lfloor \frac{q+1}{2} \rfloor$.

Intuitively, given two parents and a possible child, the probability of obtaining the child from these parents is given by the product of the probability of obtaining, in an independent way, each bit of the child from its parents.

On the other hand, the probability of obtaining a certain bit of the child is: 1 if the corresponding bits of the parents are equal among them and to the bit that it wants to obtain, 0.5 if the bits of the parents are different and 0 if the bits from the parents are equal among them but different from the bit of the child. Figure 1 shows an example of how to compute this probability.

II Single-point crossover.

Assuming that the crossover point is randomly chosen we have that:

$$c_{ij} = \prod_{r=1}^{n} \left[\sum_{k=1}^{l} \frac{1}{l} \left(\prod_{s=1+\alpha}^{(k-1)+\beta} (\pi^s_{\phi(r)}(i) \oplus \pi^s_r(q)) \prod_{s=k-\alpha}^{(l-1)-\beta} (\pi^s_{\phi(r)+1}(i) \oplus \pi^s_r(q)) \right) \right]$$

where $\alpha = (1 - r\mathrm{mod}2)(k-1)$ and $\beta = (1 - r\mathrm{mod}2)(l-k)$.

In this case, if we assume that the crossover point is fixed, we only have to verify if the bits of the first parent going from the beginning of the chain to the crossover point are all equal to those in the child. Then, we have to verify if the remaining bits are all equal to those of the second parent. In this case, the terms of the matrix can only be zero or one. Thus, the total term is the result of the sum of the probabilities for each possible crossover point multiplied by the probability of choosing each point.

Elements of the Mutation Matrix

As we saw before, Rudolph models the GA using uniform mutation. The corresponding elements are:

$$m_{ij} = p_m^{H_{ij}} (1 - p_m)^{N - H_{ij}}$$

where p_m is the mutation probability and H_{ij} is the Hamming distance between populations i and j. Since mutation is applied with probability p_m to each bit, the expression of the element m_{ij} is quite simple.

Elements of the Selection Matrix

The selection operator adopted is the well-known proportional selection. As we know, in such a selection scheme, each individual has a probability of being selected which is proportional to its fitness, thus:

$$s_{ij} = \begin{cases} \frac{\prod_{k=1}^{n} f(\pi_k(j))}{(\sum_{k=1}^{n} f(\pi_k(i)))^n} & \text{if } \pi_k(j) \in \{\pi_r(i) | r = 1, \cdots, n\} \; \forall k = 1, \cdots, n. \\ 0 & \text{otherwise.} \end{cases}$$

where f is the objective function (fitness).

Elements of the $\mathbf{P}$ *Matrix*

Since the matrix $\mathbf{P}$ is the product of the crossover, mutation and selection matrices, we have that:

$$p_{ij} = \sum_{p=1}^{n} (\sum_{q=1}^{n} c_{iq} m_{qp}) s_{pj}$$

which is the reason why:

$$p_{ij} = \sum_{p=1}^{n} \left[\sum_{q=1}^{n} \left(\prod_{r=1}^{n} (0.5)^{l} \prod_{s=1}^{l} \sum_{t=\phi(r)-1}^{\phi(r)} \pi_t^s(i) \oplus \pi_r^s(q) \right) \left(p_m^{H_{qp}} (1-p_m)^{N-H_{qp}} \right) \right] \frac{\prod_{k=1}^{n} f(\pi_k(j))}{(\sum_{k=1}^{n} f(\pi_k(p)))^n}$$

E Matrix

The $\mathbf{E}$ matrix is the following:

$$\begin{pmatrix} \mathbf{E_{11}} & & & \\ \mathbf{E_{21}} & \mathbf{E_{22}} & & \\ \vdots & \vdots & \ddots & \\ \mathbf{E_{2^l,1}} & \mathbf{E_{2^l,2}} & \cdots & \mathbf{E_{2^l,2^l}} \end{pmatrix}$$

Rudolph [15] indicates that this matrix has exactly one 1 by row. Then, $\mathbf{E_{11}}$ is an identity matrix and the matrices $\mathbf{E_{aa}}$ ($a \geq 2$) are identity matrices with some zeros in the diagonal. Next, we will try to clarify these claims and to provide a specific example of this matrix.

As we know:

$$e_{ij} = \begin{cases} 1 & \text{if } f(\pi_0(i)) < f(b) \\ 0 & \text{otherwise.} \end{cases}$$

where b =argmax$\{f(\pi_k(i)) | k = 1, ..., n\} \in \mathbb{B}^l$ y

$$j \overset{\text{def}}{=} (b, \pi_1(i), \pi_2(i), ..., \pi_n(i)) \in \mathcal{S}$$

Then, given a fixed population is clear that its super individual can only improve (Figure 2), so that it only has two choices:

1. Remain intact:
$$f(\pi_0(i)) \geq f(b) \Rightarrow e_{ii} = 1$$

2. Update its super individual by the maximum of its population:
$$f(\pi_0(i)) < f(b) \Rightarrow e_{ij} = 1$$

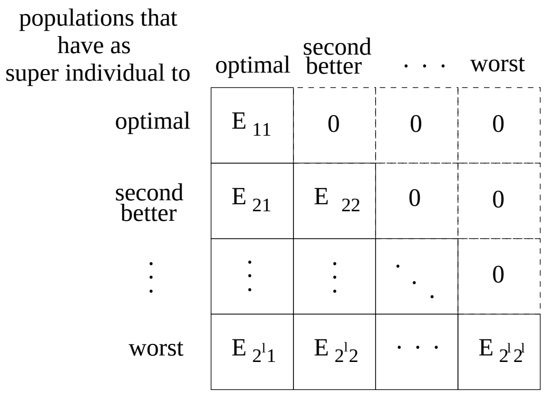


Figure 2: The structure of the elitism matrix is due to the fact that the super individual of a given population can only improve. Therefore, there are blocks of zeros.

From the previous, we conclude that there exists one 1 and only one 1 in the rows of the $\mathbf{E}$ matrix and the remaining elements are zeros.

Now we will proceed to study the structure of the columns. It is clear that no population i will become a population j in which the super individual is of lower quality than the maximum within her (formally, $f(\pi_0(j)) < f(b)$). Here follows that the column corresponding to this population j will consist completely of 0 elements, i.e. $e_{ij} = 0$ for all i. Therefore, the matrix $\mathbf{E}$ has columns of zeros.

We will now study those columns different from zero. Let us remember that in the final structure of the $\mathbf{E}$ matrix each row of blocks i represents the probabilities of transition of all the possible populations but with the ith best individual as a super individual (Figure 2). It is clear that in each row of blocks there exists a population h that contains the second best individual as its maximum. Let us consider that population in the block of rows 2 and in the 2^l, and call them h_2 and h_{2^l}. Clearly, the population h_2 will be intact, but the population h_{2^l} will pass to be h_2 (Figure 3). In conclusion, we have that it is possible to find more of one 1 in the columns that are different from zero.

We will now consider the columns but of blocks. For the first column of blocks (corresponding to the populations that have the optimal solution as their super individual) is clear that in a column corresponding to one population whose maximum is the optimal we will have one 1 in each block, because this means that the super individual of that population will pass to be the optimal in case of not being it, for each of the blocks. Let us say then that all the populations whose maximum is optimal “stay” in the first block of columns. Analogously, the populations whose maximum is the ith best individual “stay” in the ith block. Therefore, in the block of columns i there will be a maximum of $(2^l - i + 1)$ 1’s by column. That is to say, there will be as many 1’s as blocks in the column since in each block there will be a population whose maximum is the individual i.

Then, as we descend throughout the rows of blocks, the populations get distributed along the row, depending on the quality of their maximum individual.

To exemplify the previous, we will assume that within each block the populations

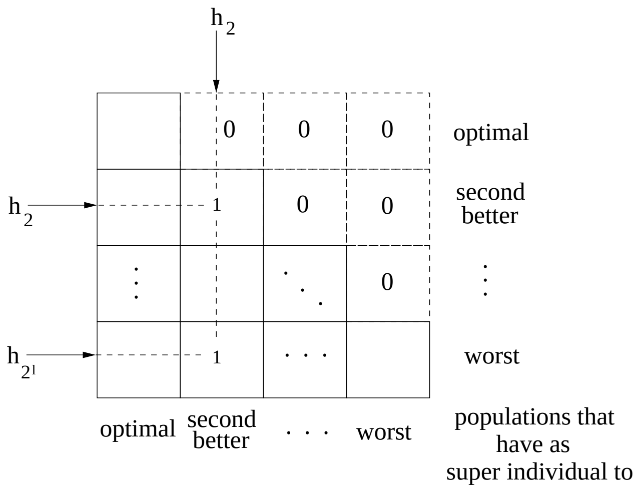


Figure 3: If the populations h_2 and h_{2^l} have as a maximum the second best individual, after applying to them the elitism matrix both will pass to be h_2. This way, the elitism matrix can have more of one 1 in the columns different from zero.

are sorted (by sets within which order is irrelevant) based on the quality of their maximum individual. Let us say that there are three populations and three possible super individuals, the corresponding elitism matrix will have the form depicted in Figure 4.

Obviously, the case that has been used as an example is quite unrealistic, because it would be very difficult to find a problem with 3 possible populations and 3 possible super individuals, particularly if we consider that we are assigning a different maximum individual to each population.

In general, it is clear that the number of populations whose maximum is the ith best individual decreases as i increases (if the maximum individual is i, in the rest of the population we can only find $N - i$ individuals different to it); nevertheless, it does not get down to zero for any value of i.

Now we will show the minimum case that has become commonly associated with the basic (minimum) conditions of a GA.

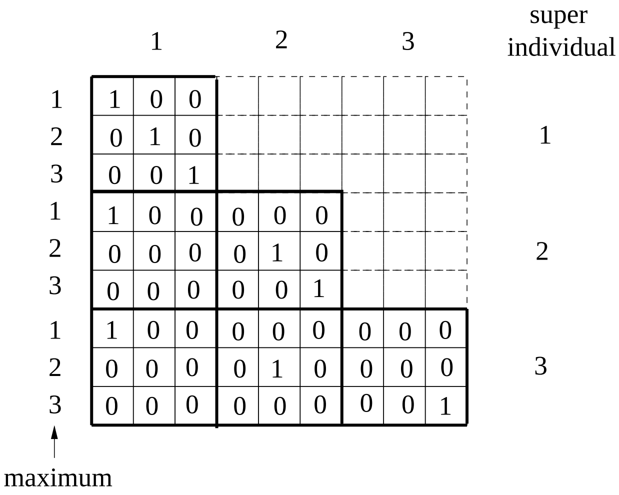
Let us consider the simplest case. Let $l = 2$ and $n = 2$. This gives us a total of $2^{2\times 2} = 2^4 = 16$ populations to consider by the GA. Nevertheless, just with the aim of illustrating our example in an easier way, we will consider the fact that on a formal way the net number of populations is [13]:

$$N = \left(\begin{array}{c} n + r - 1 \\ r - 1 \end{array} \right)$$

where $r = 2^l$.

Therefore in this case we only have $N = 10$ possible populations.

As there exist only 4 ($2^l = 4$) possible individuals, we will say that 4 populations have as their maximum the optimal individual, 3 have as their maximum to the second



$$\begin{array}{c|ccc|ccc|ccc|c}
 & & 1 & & & 2 & & & 3 & & \text{super individual} \\
\hline
1 & 1 & 0 & 0 & & & & & & & \\
2 & 0 & 1 & 0 & & & & & & & 1 \\
3 & 0 & 0 & 1 & & & & & & & \\
\hline
1 & 1 & 0 & 0 & 0 & 0 & 0 & & & & \\
2 & 0 & 0 & 0 & 0 & 1 & 0 & & & & 2 \\
3 & 0 & 0 & 0 & 0 & 0 & 1 & & & & \\
\hline
1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \\
2 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 3 \\
3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & \\
\hline
\uparrow & & & & & & & & & & \\
\text{maximum} & & & & & & & & & &
\end{array}$$

Figure 4: In the elitism matrix the populations get "distributed" throughout the row of blocks according to the quality of their maximum individual as we descend throughout the matrix. The figure shows the case in which we have three possible populations and three possible maximum individuals.

best, 2 to the third best and 1 to the worst individual. This is done with the purpose of modeling the fact that somehow the number of possible populations that have a certain individual as their maximum decreases simultaneously with the quality of such an individual.

The corresponding matrix of elitism is shown in Figure 5. This is a 40×40 matrix in which we can again observe that the populations get "distributed" throughout the rows of blocks based on the quality of their maximum individual, as we descend throughout the matrix. The dimensions of this matrix gives us an idea of how complex this model becomes when we use more realistic parameters for the GA.

Based on the previous, we now know the structure of the $\mathbf{P}^{+}$ matrix. Figure 6 shows the $\mathbf{P}^{+}$ matrix for the case of 3 populations and 3 individuals previously discussed.

4.1.2 Expected Convergence Time

We will use again our two previous examples to illustrate the following procedure. In the case of 3 individuals and 3 populations, whose $\mathbf{P}^{+}$ matrix is provided in Figure 6, the corresponding block $\mathbf{T}$ is shown in the Figure 7.

The package MATHEMATICA 4.0 was used to carry out the computations corresponding to the fundamental matrix $\mathbf{N}$, where

$$\mathbf{N} = (I - \mathbf{T})^{-1}$$

According to Theorem 2.13, once we have the $\mathbf{N}$ matrix, it must be multiplied by a column vector of 1's and the result is a vector whose elements are the expected times of

	1											2											3											4										populations with the super individual
1	1																																							1
1		1																																					
1			1																																				
1				1																																			
2					1																																		
2						1																																	
2							1																																
3								1																															
3									1																														
4										1																													
1	1																																							2
1		1																																					
1			1																																				
1				1																																			
2															1																									
2																1																								
2																	1																							
3																		1																						
3																			1																					
4																				1																				
1	1																																							3
1		1																																					
1			1																																				
1				1																																			
2															1																									
2																1																								
2																	1																							
3																												1												
3																													1											
4																														1										
1	1																																							4
1		1																																					
1			1																																				
1				1																																			
2															1																									
2																1																								
2																	1																							
3																												1												
3																													1											
4																																								1	

maximum

Figure 5: Matrix of elitism for the case of 4 possible individuals and 10 possible populations.

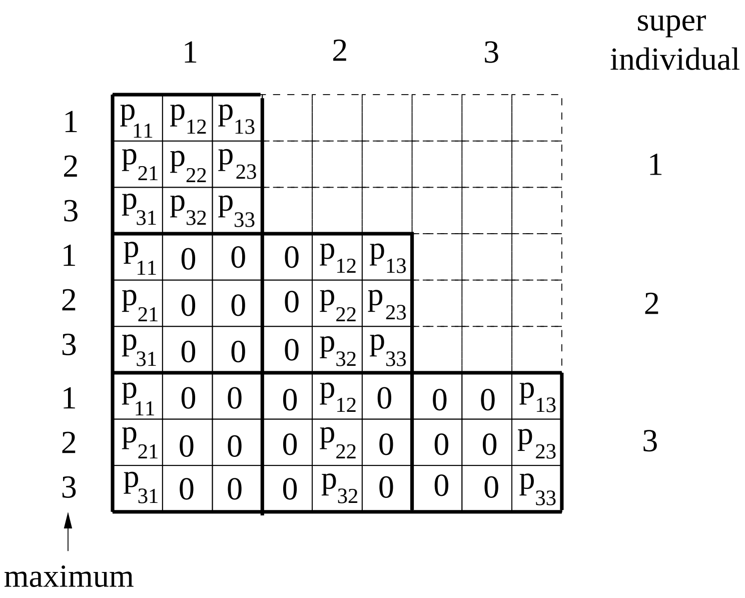


Figure 6: $\mathbf{P}^{+}$ in the case of 3 possible populations and 3 possible individuals.

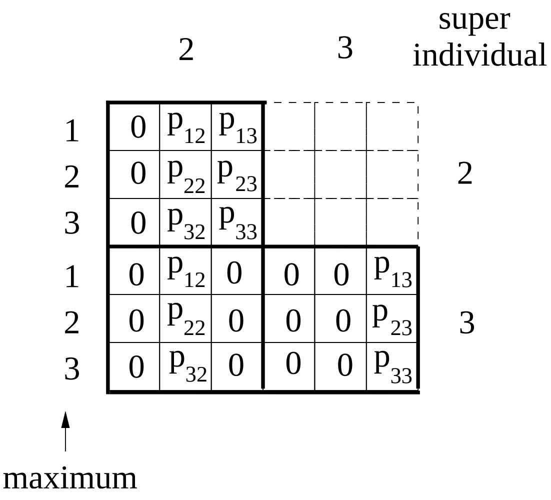


Figure 7: The block $\mathbf{T}$ in the case of 3 possible populations and 3 possible individuals.

absorption or convergence for each one of the transient states, in terms of the elements of the $\mathbf{P}$ matrix.

We shown next the results obtained:

Case I. 3 individuals - 3 populations. We have 6 transient states:

(i) $\frac{1+p_{12}+p_{13}-p_{22}-p_{13}p_{22}+p_{12}p_{23}+p_{13}p_{32}-p_{23}p_{32}-p_{33}+p_{12}p_{33}+p_{22}p_{33}}{1-p_{22}-p_{23}p_{32}-p_{33}+p_{22}p_{33}}$

(ii) $\frac{-1-p_{23}+p_{33}}{1-p_{22}-p_{23}p_{32}-p_{33}+p_{22}p_{33}}$

(iii) $\frac{1-p_{22}+p_{32}}{1-p_{22}-p_{23}p_{32}-p_{33}+p_{22}p_{33}}$

(iv) $\frac{1+p_{12}+p_{13}-p_{22}-p_{13}p_{22}+p_{12}p_{23}+p_{13}p_{32}-p_{23}p_{32}-p_{33}+p_{12}p_{33}+p_{22}p_{33}}{1-p_{22}-p_{23}p_{32}-p_{33}+p_{22}p_{33}}$

(v) $\frac{1+p_{23}-p_{33}}{1-p_{22}-p_{23}p_{32}-p_{33}+p_{22}p_{33}}$

(vi) $\frac{-1+p_{22}-p_{32}}{1-p_{22}-p_{23}p_{32}-p_{33}+p_{22}p_{33}}$

Case II. 4 individuals - 10 populations. We have 30 transient states. The software used could not unfold the results. The obvious cause is that in the expression of the matrix whose inverse we wish to compute there are 60 variables.

It is clear that the use of the elements of the $\mathbf{P}$ matrix makes the problem quite complex. For that reason, we thought of the possibility of using a label called p_i by each i column of the $\mathbf{P}$ matrix in order to simplify the necessary computations. Note that such labels p_i must fulfill: $\sum_{i=1}^{n} p_i = 1$.

Let us assume that we can label each column i of the $\mathbf{P}$ matrix by the term p_i.

In order to clarify our idea, the $\mathbf{P}^+$ matrix and the block $\mathbf{T}$ corresponding to the case of 3 possible individuals and 3 possible populations using the labels p_i are illustrated in the Figures 8 and 9.

Now we show the results obtained using the new matrices:

Case I. 3 individuals - 3 populations. We have 6 transient states: Expected times using a label for each column:

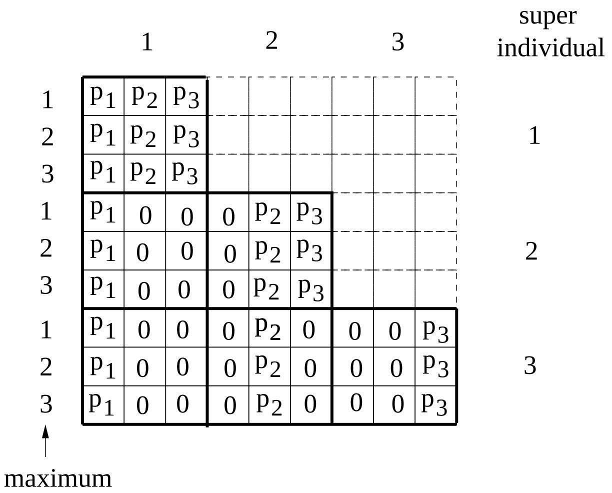
$$\frac{-1}{-1+p_2+p_3}$$

for all the transient states.

Case II. 4 individuals - 10 populations. We have 30 transient states: Expected times using a label for each column:

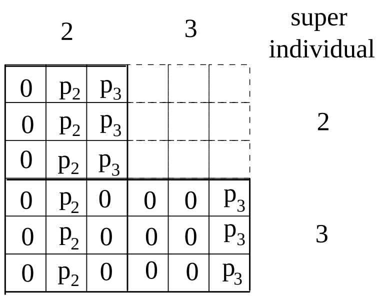
$$\frac{-1}{-1+p_5+p_6+p_7+p_8+p_9+p_{10}}$$

for all the transient states.



$$\begin{array}{c|ccc|ccc|ccc|c}
 & \multicolumn{3}{c}{1} & \multicolumn{3}{c}{2} & \multicolumn{3}{c}{3} & \text{super individual} \\
\hline
1 & p_1 & p_2 & p_3 & & & & & & & \\
2 & p_1 & p_2 & p_3 & & & & & & & 1 \\
3 & p_1 & p_2 & p_3 & & & & & & & \\
\hline
1 & p_1 & 0 & 0 & 0 & p_2 & p_3 & & & & \\
2 & p_1 & 0 & 0 & 0 & p_2 & p_3 & & & & 2 \\
3 & p_1 & 0 & 0 & 0 & p_2 & p_3 & & & & \\
\hline
1 & p_1 & 0 & 0 & 0 & p_2 & 0 & 0 & 0 & p_3 & \\
2 & p_1 & 0 & 0 & 0 & p_2 & 0 & 0 & 0 & p_3 & 3 \\
3 & p_1 & 0 & 0 & 0 & p_2 & 0 & 0 & 0 & p_3 & \\
\hline
\uparrow & & & & & & & & & & \\
\text{maximum} & & & & & & & & & &
\end{array}$$

Figure 8: For the case of 3 populations and 3 individuals, this is the $\mathbf{P}^{+}$ matrix in the case in which a label by column is used.



$$\begin{array}{|ccc|ccc|c}
\multicolumn{3}{c}{2} & \multicolumn{3}{c}{3} & \text{super individual} \\
\hline
0 & p_2 & p_3 & & & & \\
0 & p_2 & p_3 & & & & 2 \\
0 & p_2 & p_3 & & & & \\
\hline
0 & p_2 & 0 & 0 & 0 & p_3 & \\
0 & p_2 & 0 & 0 & 0 & p_3 & 3 \\
0 & p_2 & 0 & 0 & 0 & p_3 &
\end{array}$$

Figure 9: Block matrix $\mathbf{T}$ corresponding to the case of the Figure 8.

4.1.3 Experiments

We carried out the tests corresponding to the case of 10 populations and 4 individuals. In this case, each population consists of 2 individuals and each individual is of length 2. As we mentioned previously, the total number of populations for the GA in this case is of 16.

With the aim of being able to hierarchize the four individuals, we defined the following function:

$$f(x_1x_2) = x_1 + 0.5x_2 + 0.5$$

For example:

$$f(01) = 0 + 0.5 * 1 + 0.5 = 1.0$$

As the function f is strictly positive for any individual, this same function was used as fitness function. In the following table, we show the four possible individuals, their corresponding fitnesses and their hierarchy:

individual	fitness	hierarchy
00	0.5	4
01	1.0	3
10	1.5	2
11	2.0	1

Now, we show the table with the 16 populations organized according to their corresponding maximum individual:

Maximum	No.	*individuals*	
1	1	11	00
1	2	11	01
1	3	11	10
1	4	11	11
1	5	10	11
1	6	01	11
1	7	00	11
2	8	10	00
2	9	10	01
2	10	10	10
2	11	01	10
2	12	00	10
3	13	01	00
3	14	01	01
3	15	00	01
4	16	00	00

As we can see, 7 populations have as their maximum the optimal, 5 have as their maximum the second best, 3 to the third best and one to the worst individual. In addition, within these groups the order is irrelevant.

According to the model described in Section 3, the chain corresponding to the EGA has 16 absorbent (or optimal) states. That is to say, all the possible populations but with the optimal as super individual, and 48 transient states, that is, the sixteen possible populations but with the second, third or worst individual as their super individual.

Using the expressions previously obtained and the procedure described in the previous section, the corresponding matrices of crossover, mutation and selection were constructed, and after that, the corresponding $\mathbf{P}$ matrix. Also, the $\mathbf{E}$ matrix was constructed and finally the $\mathbf{P}^+$ matrix was obtained. From this last one, the block $\mathbf{T}$ was obtained, which gives rise to the fundamental matrix $\mathbf{N}$.

Since in our model the probability of crossover (p_c) was fixed to a value of 1, and due to the fact that the size of the population and the length of the individuals are also fixed, the results only depend on the mutation probability (p_m).

On the other hand, a GA with the conditions imposed by the model was run. Note that within this GA, the super individual (elitist individual) does not have to take part in the evolutionary process. The fixed parameters for the GA were:

population size = 2
chromosome lenght = 2
crossover probability = 1.0

Let g be the random variable whose value is the number of necessary iterations for the convergence of the GA.

Next, we show the expected value of the variable g ($\mathsf{E}[g]$) and the corresponding standard deviation ($\mathsf{D}[g]$) obtained (using MATHEMATICA 4.0) by the theoretical model (TM) developed and the results obtained by the GA. In our experiments, we performed 100 runs with different random seeds:

p_m	*GA*		*TM*	
	$\mathsf{E}[g]$	$\mathsf{D}[g]$	$\mathsf{E}[g]$	$\mathsf{D}[g]$
0.001	345.05	540.001	514.137	660.781
0.005	90.98	131.4615	103.782	132.47
0.01	40.89	54.01943	52.4913	66.4335
0.03	13.0	16.8367	18.3079	22.4155
0.07	5.08	7.036241	8.56138	9.85261
0.1	4.69	6.5	6.38044	7.03
0.2	2.69	3.47	3.87256	3.78054
0.5	1.32	1.847083	2.52747	1.96485

In Figure 10, we show the graph of the values obtained by both methods. As we can see, the values obtained by the theoretical model turn out to be larger than the values obtained by the GA, in all cases.

Perhaps the values that were obtained in the cases of p_m =0.001, 0.005, 0.01 will seem very high, but they are correct from the following point of view: if we remember, the table of all the possible populations, approximately 46% (populations 1 to 7)

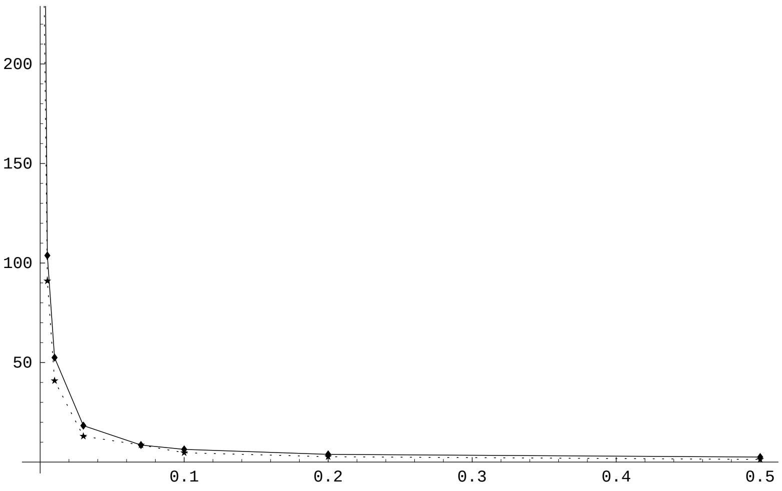


Figure 10: Graph of the values obtained by the GA and the theoretical model developed for the variable g, in terms of the probability of mutation. The continuous line corresponds with the values generated by the theoretical model and the dotted line corresponds to the values generated by the GA.

contains the optimal, and only populations 9 and 11 contain the schemata necessary to give rise to the optimum by means of crossover. In conclusion, approximately 40% of the possible populations do not contain the optimal one nor the schemata necessary to give them rise. Since the population is very small, the number of mutations performed in a short number of generations is practically zero, being this one the only mechanism to be able to converge correctly. This is the reason why we need a high number of generations to find the global optimum in these cases.

Moreover, if in each generation we have 4 opportunities to carry out a mutation and the mutation probability is of 0.001, then we need at least 250 iterations on average to ensure that at least one mutation was carried out. For that reason, for certain populations it will take many iterations to find the optimal solution.

In the following table, we show the populations for which it is particularly difficult to reach the optimal solution. Keep in mind that the super individual does not participate in the evolutionary process.

Maximum	No.	*individuals*	
2	8	10	00
2	10	10	10
2	12	00	10
3	13	01	00
3	14	01	01
3	15	00	01
4	16	00	00

Obviously, the problem previously indicated (i.e., the need of a large number of iterations to ensure that a mutation will be carried out when the mutation probability is low) becomes less critical when we adopt a higher mutation probability. This is reflected in the results previously shown.

On the other hand, the matrix $\mathbf{P}$ corresponding to the value p_m =0.5 was the only one characterized by having a label p_i for each column i. Such property was most likely due to the fact that the probability of passing (by effect of the mutation operator) from the population i to the population j is the same regardless of i and j. In this case, the formula previously obtained gave the correct results, as expected.

4.2 Conclusions

We have seen that the current models to estimate the convergence time of a GA are quite simple and, therefore, very distant from the observed behavior of such type of algorithm.

In this Section, we developed a mechanism to estimate the convergence time of a GA by means of a model based on Markov chains.

In general, it turns out to be very complicated to obtain an expression for the expected convergence time of a GA based on its parameters. This is due to the fact that we need to know every single element of the transition matrix under study. For this reason, the dimensions of such matrix introduce an excessive complexity in our computations. On the other hand, the elements of this matrix are complex enough as to preclude classification. This is the reason why knowing the matrix is a necessary condition.

Our results for the most simple case that can be considered (perhaps argueably) as realistic, led us to conclude that the proposed model is correct. Nevertheless, it is clear that in such case it was relatively easy to obtain the corresponding matrix $\mathbf{P}$, but this is a process that will generally get more complicated as we increase the size of the population and the chromosomic length. For example, let us suppose that the population consists of 50 individuals of length 5, then the corresponding matrix $\mathbf{P}$ is of size $2^{250} \times 2^{250}$. In general, the size of the matrix has an exponential growth.

Therefore, we can conclude that, from a practical point of view, Markov chains are not a recommended theoretical tool for this sort of analysis (i.e., estimation of expected convergence time). Nevertheless, other alternatives exist to which it would be possible to resort (e.g., statistical mechanics [14], geometric interpretation approaches [18], and random search based modelling [19]).

5 The Simple Genetic Algorithm for Multi-Objective Optimization

The theoretical frame corresponding to multi-objective evolutionary optimization is completely different from the one adopted for single-objective optimization. This is because in single-objective optimization, the goal is to obtain a single solution, whereas in multiobjective optimization, we aim to find a set of solutions (the so-called Pareto optimal set).

In this section we provide a brief analysis of the capabilities of the simple GA when dealing with multiobjective optimization problems, starting with some basic definitions.

5.1 Definition of the problem

We are interested in solving problems of the type:

$$\text{(MO)} \qquad \min \vec{f}(\vec{x}) = (f_1(\vec{x}), f_2(\vec{x}), \cdots, f_k(\vec{x}))$$

where $f_i : \mathbb{R}^n \to \mathbb{R}$ for $i = 1, ..., k$ are the objective functions, $\vec{f} : X \subset \mathbb{R}^n \to \mathbb{R}^k$ is the multiobjetive function, and $\vec{x} \in X$ is called the vector of decision variables.

In order to describe the concept of optimality that interests us, we will use the following relation in $\mathbb{R}^k$: we say that $\vec{u} \leq \vec{v}$ if $u_i \leq v_i$ for all $i = 1, \ldots, k$, and that $\vec{u} < \vec{v}$ if $\vec{u} \leq \vec{v}$ but $\vec{u} \neq \vec{v}$. In this last case, we say that $\vec{u}$ **dominates** $\vec{v}$ and instead of $\vec{u} < \vec{v}$ we will write $\vec{u} \prec \vec{v}$.

Summarizing, in the set $\mathcal{F} = \vec{f}(X)$ we can define the following relation:

Definition 5.1 (Pareto Dominance) *We say that a vector* $\vec{u} = (u_1, ..., u_k) \in \mathbb{R}^k$ **dominates** $\vec{v} = (v_1, ..., v_k) \in \mathbb{R}^k$ *(denoted by* $\vec{u} \prec \vec{v}$*) if and only if* $\vec{u} < \vec{v}$*.*

Definition 5.2 *A vector of decision variables* $\vec{x}^* \in X$ *is* **Pareto optimal** *for a MO problem if there does not exist another vector of decision variables* $\vec{x} \in X$ *such that* $\vec{f}(\vec{x}) \prec \vec{f}(\vec{x}^*)$*.*

Let $X^* = \{\vec{x}^* \in X | \vec{x}^* \text{ be Pareto optimal}\}$. The elements of X^* are also called *nondominated* and the set $\mathcal{F}^* = \vec{f}(X^*)$ is called the *Pareto front*.

5.2 Partially Ordered Sets

If the reflective, anti-symmetrical and transient relation $\preceq$ is valid in X then the pair $(X, \preceq)$ is called a partially ordered set. If x and y are two elements of X such that $x \preceq y$ but $x \neq y$, then we say that $x \prec y$.

Definition 5.3 *An element* $x^* \in X$ *is called* **minimal element** *of* $(X, \preceq)$ *if there does not exist* $x \in X$ *such that* $x \prec x^*$*. The set of all the minimal elements is denoted by* $\mathcal{M}(X, \preceq)$*.*

We say that different points $x, y \in X$ *are* **comparable** *when* $x \prec y$ *or* $y \prec x$*; of another way,* x *and* y *are* **incomparable** *which is denoted as* $x \parallel y$*.*

If each pair of different points from $(X, \preceq)$ *is comparable then* $(X, \preceq)$ *is called a totally ordered set or a* **chain***. If each pair of different points from* $(X, \preceq)$ *is incomparable then* $(X, \preceq)$ *is called a* **antichain***.*

Lemma 5.4 *If* $(X, \preceq)$ *is a partially ordered set and* $0 < |X| < \infty$ *then* $\mathcal{M}(X, \preceq)$ *is complete.*

Let now $f : \mathbb{B}^l \to \mathcal{F} = \{f(x) : x \in \mathbb{B}^l\} \subset \mathbb{R}^n$. Note that the Pareto Dominance defines a relation of strict order in $\mathcal{F}$.

Therefore:

$$(\mathcal{F}, \preceq) \text{ is a partially ordered set and}$$
$$\mathcal{M}(\mathcal{F}, \preceq) \simeq \text{Pareto front.}$$

As we have seen, the objectives of the search in this case are the minimal elements of the space search.

Given their natural capacity (due to the use of a population) to handle several possible solutions simultaneously, GAs have become increasingly popular in the solution of multiobjective optimization problems [3].

5.3 Convergence Study

The SGA has been used to solve multi-objective problems by means of the use of Pareto ranking [5, 3] and diversity maintenance mechanisms such as *fitness sharing* and *niching*. Some examples of multi-objective evolutionary algorithms based on the previous concepts are the following: the Nondominated Sorting Genetic Algorithm (NSGA) [17], the Niched-Pareto Genetic Algorithm (NPGA) [8] and the Multi-Objective Genetic Algorithm (MOGA) [4].

As any other evolutionary algorithm, multi-objective evolutionary algorithms can be represented through their corresponding transition matrix. Thus, the model presented in Section 3 is directly applicable to multi-objective evolutionary algorithms.

In what follows, we propose a definition of convergence analogous to Definition 3.1, but extended for multi-objective optimization:

Definition 5.5 *Let* $M_t = \#\{\pi_k^t(i) | \pi_k^t(i) \in \mathcal{M}(\mathcal{F}, \preceq)\}$ *be a sequence of random variables representing the number of minimal elements of the search space into the population represented by the state* i *in the step* t. *A multi-objective genetic algorithm converges to the set of minimal elements of the search space if and only if:*

$$lim_{t\to\infty} P\{M_t = pop_{size}\} = 1$$

where pop_{size} *is the population size of the algorithm.*

Thus, the convergence is in terms of the number of elements of $\mathcal{M}(\mathcal{F}, \preceq)$ that the population of the algorithm contains in a certain step t. This way, each population will be labeled with the number of minimal elements of the search space that contains.

Next, we present a theorem that demonstrates that any evolutionary algorithm with irreducible transition matrix (primitive), does not converge in terms of the previous definition.

Theorem 5.6 *An EA with primitive transition matrix does not converge to* $\mathcal{M}(\mathcal{F}, \preceq)$.

Proof. Let $i \in \mathcal{S}$ be any state in which $M_t < pop_{size}$ and $p_i(t)$ is the probability that the EA is in state i at step t. Clearly, $P\{M_t \neq pop_{size}\} \geq p_i(t) \Leftrightarrow P\{M_t = pop_{size}\} \leq 1 - p_i(t)$. From Theorem 2.4, the probability that the EA is in state i converges to $p_i(\infty) > 0$. Therefore,

$$lim_{t\to\infty} P\{M_t = pop_{size}\} \leq 1 - p_i(\infty) < 1$$

and therefore, the condition of convergence is not satisfied. ■

After proving that any evolutionary algorithm with primitive transition matrix cannot converge to the Pareto front in terms of Definition 5.5, it remains to study the effect of *fitness sharing* in the distribution of the population.

5.3.1 Niching

Since the states of the Markov chain that models a genetic algorithm both in its single-objective and in its multi-objective versions correspond to the population of the algorithm itself, and since the mechanisms of fitness sharing (i.e., niche induction) introduce modifications only to the fitness values of each individual, it should be clear that niching only modifies the matrix representing the selection operator. Nevertheless, the use of fitness sharing introduces some extra complexity into the model because of the changes that it can produce.

For an example of the complexity surrounding the changes in the matrix when fitness sharing is used, refer to Horn's work [7], in which a very simplified model is adopted. Additionally, we suggest to read Mahfoud's work [12], in which a different algorithmic model that uses niching is proposed. An interesting aspect of this work is that Mahfoud shows that the use of Markov chains to model algorithms with niching is unnecessary in a certain particular case.

6 Conclusions and Future Work

In this report, we have provided a theoretical study of the convergence of genetic algorithms both for single-objective as for multi-objective optimization.

The convergence of the elitist GA (for single-objective optimization) was demonstrated, and we then studied the complexity of estimating the expected convergence time of this algorithm. With respect to this point, we could see that although the theoretical model is correct, it turns out to be very complicated to obtain an expression (in terms of the parameters of the GA) that bounds the number of iterations necessary to reach the global optimum. This difficulty is due to the dimensions of the transition matrix corresponding to the evolutionary process. Such dimensions grow exponentially with respect to the value of the parameters adopted for the GA. This means that in order to be able to predict the behavior of the algorithm, it becomes necessary to know its corresponding matrix, which in a real-world situation, would be a very expensive process (computationally speaking).

The most viable alternative in this case is to use another type of theoretical tool to model the GA. Such a tool should not require as much *a priori* knowledge as our Markov chain model to predict some aspects of the algorithm's behavior.

We have also analyzed the capabilities of a simple GA for solving multi-objective optimization problems. Basically, we studied the convergence of a simple GA to the true Pareto front of a problem.

As we could see, the simple GA does not converge to the Pareto front if convergence is defined as having all the members of the population to belong to the Pareto optimal set. This turns out to open several possible areas of study. For example, it would be interesting to know how many elements of the population converge to the Pareto front? Also, we would like to know what is the expected time to obtain the first Pareto optimal solution? In this regard, we propose to study the population dynamics (i.e., how the mechanisms of fitness sharing or any other diversity maintenance mechanism affects the distribution of individuals along the evolutionary process). This is an important topic since one of the aims of multi-objective evolutionary algorithms is to obtain a distribution of nondominated solutions as uniform as possible.

This brief theoretical study has shown that evolutionary algorithms theory in general is a very interesting research area with many open questions. Particularly, the theoretical study of multi-objective evolutionary algorithms has been very scarce so far and many problems remain to be solved (e.g., modelling of parallel multi-objective evolutionary algorithms) [3].

Acknowledgements

The first author acknowledges support from CONACyT through a scholarship to pursue graduate studies at the Computer Science Section of the Electrical Engineering Department at CINVESTAV-IPN. The second author gratefully acknowledges support from CONACyT through project 34201-A.

References

[1] Carol A. Ankenbrandt. An extension to the theory of convergence and a proof of the time complexity of genetic algorithms. In Gregory J. E. Rawlins, editor, *Foundations of Genetic Algorithms*, pages 53–68. Morgan Kaufmann Publishers, 1991.

[2] Thomas Bäck, David B. Fogel, and Zbigniew Michalewicz, editors. *Handbook of Evolutionary Computation*. Institute of Physics Publishing and Oxford University Press, 1997.

[3] Carlos A. Coello Coello, David A. Van Veldhuizen, and Gary B. Lamont. *Evolutionary Algorithms for Solving Multi-Objective Problems*. Kluwer Academic Publishers, New York, May 2002. ISBN 0-3064-6762-3.

[4] Carlos M. Fonseca and Peter J. Fleming. Genetic algorithms for multiobjective optimization: formulation, discussion and generalization. In Stephanie Forrest, editor, *Proceedings of the Fifth International Conference on Genetic Algorithms*, pages 416–423, San Mateo, California, 1993. University of Illinois at Urbana-Champaign, Morgan Kauffman Publishers.

[5] David E. Goldberg. *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley Publishing Co., Reading, Massachusetts, 1989.

[6] John H. Holland. *Adaptation in Natural and Artificial Systems*. Ann Arbor : University of Michigan Press, 1975.

[7] Jeffrey Horn. Finite markov chain analysis of genetic algorithms with niching. In S. Forrest, editor, *Proceedings of the Fifth International Conference on Genetic Algorithms*, pages 110–117. Morgan Kaufmann Publishers, 1993.

[8] Jeffrey Horn, Nicholas Nafpliotis, and David E. Goldberg. A niched pareto genetic algorithm for multiobjective optimization. In *Proceedings of the First IEEE Conference on Evolutionary Computation, IEEE World Congress on Computational Intelligence*, volume 1, pages 82–87, Piscataway, New Jersey, June 1994. IEEE Service Center.

[9] M. Iosifescu. *Finite Markov Processes and Their Applications*. Wiley, Chichester, 1980.

[10] John G. Kemeny and J. Laurie Snell. *Finite Markov Chains*. D. Van Nostrand Company, Inc., Princeton, New Jersey, 1960.

[11] Sushil J. Louis and Gregory J.E. Rawlins. Syntactic analysis of convergence in genetic algorithms. In L. Darrell Whitley, editor, *Foundations of Genetic Algorithms*, pages 141–151, San Mateo, CA, 1993. Morgan Kaufmann Publishers.

[12] S.W. Mahfoud. *Niching Methods for Genetic Algorithms*. PhD thesis, University of Illinois, Urbana-Champaign, 1995.

[13] Allen E. Nix and Michael D. Vose. Modeling genetic algorithms with markov chains. *Annals of Mathematics and Artificial Intelligence*, 5:79–88, 1992.

[14] A. Prügel-Bennett and J.L. Shapiro. An analysis of genetic algorithms using statistical mechanics. *Physical Review Letters*, 72(9):1305–1309, 1994.

[15] Günter Rudolph. Convergence properties of canonical genetic algorithms. *IEEE Transactions on Neural Networks*, 1(5):96–101, 1994.

[16] Günter Rudolph and Alexandru Agapie. Convergence properties of some multi-objective evolutionary algorithms. In *Proceedings of the 2000 Conference on Evolutionary Computation*, pages 1010–1016, Piscataway, New Jersey, 2000. IEEE.

[17] N. Srinivas and Kalyanmoy Deb. Multiobjective optimization using nondominated sorting in genetic algorithms. *Evolutionary Computation*, 2(3):221–248, Fall 1994.

[18] Michael D. Vose. Modeling simple genetic algorithms. In L. D. Whitley, editor, *Foundations of Genetic Algorithms 2*, pages 63–73, San Mateo California, 1991. Morgan Kaufmann Publishers.

[19] Michael D. Vose. *The Simple Genetic Algorithm: foundations and theory*. MIT Press, Cambridge, Massachusetts, 1999.