Proceedings of the 2007 IEEE Symposium on
Foundations of Computational Intelligence (FOCI 2007)

On the Convergence of Immune Algorithms

Vincenzo Cutello, Giuseppe Nicosia, Mario Romeo
Department of Mathematics and Computer Science
University of Catania
V.le A. Doria 6, 95125, Catania, Italy
Telephone: +39 095 738 –3074 Fax: +39 095 330094
Email: {cutello, nicosia, romeo} @dmi.unict.it

Pietro S. Oliveto
School of Computer Science
University of Birmingham
Edgbaston, Birmingham B15 2TT, U.K.
Email: P.S.Oliveto@cs.bham.ac.uk

Abstract— **Immune Algorithms have been used widely and successfully in many computational intelligence areas including optimization. Given the large number of variants of each operator of this class of algorithms, this paper presents a study of the convergence properties of Immune Algorithms in general, conducted by examining conditions which are sufficient to prove their convergence to the global optimum of an optimization problem. Furthermore problem independent upper bounds for the number of generations required to guarantee that the solution is found with a defined probability are derived in a similar manner as performed previously, in literature, for genetic algorithms. Again the independence of the function to be optimised leads to an upper bound which is not of practical interest, confirming the general idea that when deriving time bounds for Evolutionary Algorithms the problem class to be optimised needs to be considered.**

I. Introduction

Evolutionary Algorithms (EAs) have been used widely in many areas (e.g., numerical and combinatorial optimization, machine learning, constraint satisfaction) since the seventies and eighties [10]. Nevertheless, the study of their computational complexity is a fairly new field spawned in the early nineties. By using the theory of Markov chains, various results related to the convergence of EAs and various time bounds for the first hitting times of simple EAs on pseudo-boolean functions were given [9]. Building on this first block of theory, recently the first combinatorial optimization problems with practical applications have been tackled in evaluating the performance of the (1+1)-EA. Among these problems particularly worth of a mention are maximum matching [12], minimum spanning tree [13], the NP-complete partition problem [11] and the NP-hard subset sum problem [8]. For the last problem, populations and crossover have also been considered.

In this paper we attempt a first general step in the theoretical analysis of Immune Algorithms (IAs) [16], [4], [21], a class of EAs inspired by the natural Immune System approach [1], [2], [14]. The field of IAs is fairly new, although various successful results have been achieved in different areas. As done previously for other sub-classes of EAs, we analyse the convergence properties of IAs. The two previous papers regarding the convergence of IAs, are related to specific algorithms which had previously proved to be useful in literature such as MISA (Multi-objective Immune System Algorithm) [17] and BCA (B-Cell Algorithm) [18]. Here, instead, we will concentrate on examining general conditions which are sufficient for proving the convergence of IAs, rather than designing a specific proof for each algorithm, following the consideration that the intensive elaboration of convergence issues *"has finally led to simple proofs which do not require Markov theory any more"* [9]. In such a way only algorithms that do not satisfy the given conditions need to be examined with specific techniques concerning their convergence to the global optimum.

In section 2 we introduce IAs and describe how they are inspired by the immune system of vertebrates. In section 3 we introduce the concept of stochastic convergence, present the general IA which will be considered in this research paper and examine its convergence properties. In section 4 we discuss bounds for the convergence in probability of the IA in a similar manner as done previously for genetic algorithms (GAs) [7], [6], [5] and compare the GA time bounds with the ones obtained for the IA. As this is just a first step towards the theoretical analysis of IAs, in the final section we discuss ideas for future work.

II. Immune Algorithms

Immune Algorithms are randomized algorithms inspired by immunology and by immune functions and principles observed in nature [21].

The immune system of vertebrates, hence of humans also, is composed of a large quantity of cells, molecules and organs cooperating in the effort of keeping the organism in good health by fighting diseases which may cause illness. The immune cells considered in Artificial Immune Systems (AIS) are *lymphocytes*, white blood cells whose major concern is to fight *antigens* (Ags), molecules belonging to foreign agents such as bacteria or viruses which have introduced themselves in the organism.

The lymphocytes, B-cells and T-cells according to the organ in which they develop, firstly need to recognise the antigens. This task is performed using cell receptors called TCRs for T-cells and BCRs or Antibodies (Abs) for B-cells. *Recognition* occurs if the shapes of a cell receptor and that of an antigen are *approximately complementary*. In this case the lymphocyte recognising the antigen binds to it, hence activating the *immune response*.

1-4244-0703-6/07/$20.00 ©2007 IEEE

B-cells and T-cells are distinguished in different types according to the kind of antigens they are able to recognise. Immune cells do not only recognise perfectly matching antigens but are also capable of recognising foreign agents within a region of complementarity, the *affinity threshold*.

The *immune response* in fighting a disease begins by reproducing the cells which are able to recognise and bind with the antigens (*clonal expansion*). The clones then undergo high mutation rates. This phenomenon has been given the name *hypermutation*. The cells obtained through this process, having the greater affinity with the antigen, live longer (i.e. have a higher life time) so to be still in the organism in case a future attack occurs (*memory cells*). While the cloning proliferation rate is *directly proportional* to the affinity with the antigen, the hypermutation rate is *inversely proportional* to such an affinity, so that the nearer the cell is to antigen complementarity the lower is the hypermutation rate. On the other hand if a cell's antigen affinity is very low, high hypermutation rates are applied in hope to raise the affinity values quickly. This process is called *affinity maturation* while the composition of antigen recognition, clonal expansion and memory cell creation, is called *clonal selection*.

Just like GAs have been inspired by the Darwinian theory of evolution, IAs derive from the immune system principles [21]. Considering antigen recognition, the idea of imitating such a process for solving pattern recognition problems is quite straightforward, and it has proved successful [19], [4]. In a similar manner, with the introduction of operators resembling cloning expansion and memory cell creation, IAs for solving optimization problems have spawned. These algorithms present a wide range of hypermutation operators inspired by the concept of affinity maturation, from the *Inversely Proportional Hypermutation* [16], [4], [15], derived directly from the somatic process described above to the *fitness function independent hyperMacromutation operator* [3].

Following strategies used in theoretical literature of EAs, where only the mutation operator is determinant to estimate if an EA visits the global optimum in finite time, in this paper we consider a simple hypermutation operator to gather some information about the convergence properties of IAs.

III. Limit behaviour of Immune Algorithms

A. Convergence measures of Evolutionary Algorithms

An EA is said to *converge* to the global optimum of a given optimization problem if it can be assured that the algorithm finds the solution in a finite number of steps and if such a solution will be kept in the population afterwards.

Since the state transitions of an EA are of stochastic nature, the deterministic concept of convergence cannot be used to determine the time limit behaviour of this kind of algorithms. Two commonly used measures of stochastic convergence are *complete convergence* and *convergence in mean* [22] :

Definition Let X be a random variable and $(X_t : t > 0)$ a sequence of random variables. Then the sequence X_t is said to **converge completely** to X, if for any $\epsilon > 0$

$$\lim_{t\to\infty} \sum_{i=0}^{t} P(|X_i - X| > \epsilon) < \infty.$$

Definition Let X be a random variable and $(X_t : t > 0)$ a sequence of random variables. Then the sequence X_t is said to **converge in mean** to X, if

$$\lim_{t\to\infty} E[|X_t - X|] = 0.$$

Now the following definitions for the convergence of an EA can be given [9]:

Definition Let $X_t : t \geq 0$ be the sequence of populations generated by an EA and let F_t be the fitness value of the best individual in the population at time step t. An EA is said to **converge completely** to the global optimum f^* of the optimization problem defined by the function $f : X \to R$ if the non-negative random sequence $D_t = f^* - F_t$ converges completely to zero.

Definition Let $X_t : t \geq 0$ be the sequence of populations generated by an EA and let F_t be the fitness value of the best individual in the population at time step t. An EA is said to **converge in mean** to the global optimum f^* of the optimization problem defined by the function $f : X \to R$ if the non-negative random sequence $D_t = f^* - F_t$ converges in mean to zero.

B. The Immune Algorithm

The behaviour of a general IA per generation is described in table I. A description of the most common variants of each operator follows.

1) Cloning: The cloning operator generates a new population $P^{(clo)}$ of copies of the individuals in the current population. Commonly used cloning operators are the *static cloning operator* [16], which simply clones each B cell dup times producing an intermediate population $P^{(clo)}$ of size $d \times dup = Nc$, and the *proportional cloning operator* (used in the pattern recognition version of CLONALG [4]), which clones B cells proportionally to their antigenic affinities. In *probabilistic cloning* [3], instead, the B cells are chosen from the current population depending on a clonal selection rate p_c.

2) Hypermutation: The hypermutation operator acts on the current population of clones $P^{(clo)}$ by applying, on each individual, a number of "single" mutations M determined by a random process, called *random mutation potential*. It is possible to define several mutation potentials [20]. The most common are:

- *Static Hypermutation*: the number of mutations is independent from the fitness function f, so each B cell receptor at each time step will undergo at most $M_s(\vec{x}) = c$ mutations.
- *Proportional Hypermutation*: the number of mutations is proportional to the fitness value. For each B cell $\vec{x}$ the mutations are at most $M_p(f(\vec{x})) = (E^* - f(\vec{x})) \times (c \times \gamma)$, where E^* is the minimum fitness function value known for the current instance (for a minimization problem).

TABLE I

PSEUDO–CODE OF THE IMMUNE ALGORITHM

```
Immune Algorithm($\tau_B$)
$t := 0$;
$P^{(t)}$ := Initialize_Pop();
while ( ¬ Termination_Condition() ) do
    Evaluate($P^{(t)}$);
    $P^{(clo)}$ := Cloning ($P^{(t)}$);
    $P^{(hyp)}$ := Hypermutation ($P^{(clo)}$);
    Evaluate ($P^{(hyp)}$);
    $(P_a^{(t)}, P_a^{(hyp)})$ := Aging($P^{(t)}, P^{(hyp)}, \tau_B$);
    $P^{(t+1)}$ := Selection ($P_a^{(t)}, P_a^{(hyp)}$);
    $t := t + 1$;
end_while
```

- *Inversely Proportional Hypermutation*: the number of mutations is inversely proportional to the fitness value. In particular, at each time step t, the operator will perform at most $M_i(f(\vec{x})) = ((1 - \frac{E^*}{f(\vec{x})}) \times (c \times \gamma)) + (c \times \gamma))$ mutations.
- *Convex Hypermutation*: each "gene" x_i, in the B cell $\vec{x}$, depending on the hypermutation rate p_h executes the hypermutation of convex combination: $x_i' = (1 - \beta)x_i + \beta x_k$, where β is a random value $\beta \in \{0, \dots, 1.0\}$, and x_k is randomly selected in $\vec{x}$.
- *Hypermacromutation*: the number of mutations is independent from the fitness function f and the parameter c. In this case, two integers, i and j such that $(i+1) \leq j \leq \gamma$ are randomly chosen and the operator mutates at most $M_m(\vec{x}) = j - i + 1$ values, in the range $[i, j]$.

3) Aging: The aging operator eliminates old individuals.

The *static pure aging operator* uses a parameter τ_B for the maximum number of generations the B cells are allowed to remain in the population. When a B cell is $\tau_B + 1$ old it is erased from the current population, no matter what its fitness value may be. During the cloning expansion, a cloned B cell inherits the age of its parent. After the hypermutation phase, only the cloned B cells which have gained a higher fitness value, will be given $age = 0$. An elitist version of this operator is obtained by giving the best individual of the population in each generation $age = 0$.

In the *stochastic aging operator*, the probability that a B cell remains in the current population is governed by the following law with parameter τ_B, (expected B cell mean life):

$$P_{live}(\tau_B) = e^{-\frac{\ln(2)}{\tau_B}}.$$

An elitist version of this operator is obtained by giving the best individual in the population per generation $P_{live} = 1$ and thus $P_{die} = 0$.

4) Selection: The selection operator generally substitutes the worst individuals in the population with new randomly generated individuals (*birth phase*), although even this operator presents variants, such as *no redundancy* where it is avoided that copies of the same individual survive for the next generation.

The IA used in this paper is kept as general as possible and follows the scheme presented in table I by considering any of the above described variants of each operator except for hypermutation: here a very simple operator which randomly selects $r \leq \gamma$ digits (with γ being the length of the candidate solution) from each individual of the population and flips them independently is used.

To keep the operator as general as possible the chosen bits are not necessarily distinct even though, in practice, mutations of the same bit in one single macromutation are usually avoided. This means that if the hypermutation operator, for example, randomly selects $r = 2$ and then randomly chooses the same bit twice, the resulting string is left unchanged. In the rest of the paper, when the considered hypermutation operator only flips distinct bits, it will be carefully pointed out.

C. Convergence of Immune Algorithms

The convergence of EAs, in general, has been proved under certain assumptions. In the following we will show which conditions, that are sufficient for the convergence of an EA, can be applied to the general Immune Algorithm taken into consideration in this paper. Afterwards we will discuss its convergence to the global optimum.

A single iteration of a general EA can be described as follows:

$$\forall i \in \{1, \dots, m\} : x_i{}' = mut(reco(mat(x_1 \dots, x_n)))$$

$$(y_1, \dots, y_n) = sel(x_{\pi(1)}, \dots, x_{\pi(q)}, x_1', \dots x_m')$$

Here $(x_1, \dots, x_n) \in \chi^n$ is the current parent population and

$$mat : \chi^n \rightarrow \chi^\rho, rec : \chi^\rho \rightarrow \chi, mut : \chi \rightarrow \chi, sel : \chi^k \rightarrow \chi^n$$

are functions representing respectively the mating process and the recombination, mutation and selection operators.

In [9] it is shown that under the following conditions such an EA converges both *in mean* and *completely* to the global optimum of any optimization problem:

Condition 1: Every individual in the population can be changed to an arbitrary other individual in one single mutation with probability $p > 0$.

Condition 2: The best individual in the population survives in each generation with probability $p = 1$.

Formally, the two conditions can be described as:

$$\forall x, y \in \chi \quad P\{y = mut(x)\} \geq \delta_m > 0 \qquad (1)$$

$$P\{v_n^*(sel(x_1, \dots x_k)) = v_k^*(x_1, \dots, x_k)\} = 1 \qquad (2)$$

Here v_i^* returns the best individual of a population of i individuals.

If only condition 1 is valid it can be proved that the EA visits the global optimum after a finite number of steps with probability $p = 1$ regardless of its initialisation, but not its convergence since it cannot be guaranteed that the optimum does remain in the population forever after it has been found. If condition 2 also holds, then it can be proved that the EA converges to the global optimum.

A general IA can be similarly described as follows:

$$\forall i \in \{1, \dots, m\} : x_i^{'} = hyp(clone(x_1 \dots, x_n)))$$

$$(x_1^{''}, \dots, x_k^{''}) = aging(x_1, \dots, x_n, x_1^{'}, \dots x_m^{'})$$

$$(y_1, \dots, y_n) = sel(x_1^{''}, \dots, x_k^{''})$$

Here the aging operator may or may not be used just like not all EAs use crossover, and k may be greater or smaller than n.

By examining the process above, it is easy to see that the aging operator actually takes part in the *selection mechanism* of the evolutionary process since it decides whether or not an individual is to survive and live for the next generation according to its age. This means that, while condition 1 can be applied to the IA just by considering the hypermutation operator, the aging operator needs to be considered in the formal description of condition 2. Hence,

$$P\{v_n^*(aging(x_1, \dots x_n, x_1^{'} \dots x_m^{'})) = v_k^*(x_1^{''}, \dots, x_k^{''}))\} = 1$$

$$P\{v_n^*(sel(x_1^{''}, \dots x_k^{''})) = v_k^*((y_1, \dots, y_n))\} = 1$$

describe condition 2 in a correct way when considering IAs.

Theorem 3.1: The IA considered in this paper converges both completely and in mean to the global optimum of an optimization problem whatever is its initialisation, as long as an elitist aging operator is applied.

a) Proof.: In order to prove the theorem we need to show that both conditions 1 and 2 are satisfied by the IA.

Since neither Cloning nor Aging do modify existing individuals or create different ones, only two operators may be responsible for introducing the optimum in the population for the first time: the hypermutation operator and/or the selection operator.

Let us consider bit-strings of length γ, with each point of the search space represented by a vector $\{0,1\}^\gamma$. If an individual of the population compared to the string representing the optimum matches in $\gamma - c$ bits, hence mismatches in c bits, the probability of the hypermutation operator of reaching the global optimum in one step is:

$$P_c^{(\gamma)} = \frac{c!}{\gamma^c}\frac{1}{\gamma} \tag{1}$$

Here the favourable choices $c!$ are the different permutations of c elements out of the γ^c possible choices. This probability needs to be multiplied by the probability that the operator actually randomly chooses to flip c bits. $\frac{1}{\gamma}$ is the probability that $r = c$ with r being the randomly chosen number of bits to be mutated. (For example if the first two bits, i.e. $c = 2$, of a 3-bit string need to be mutated to reach the global optimum, then the 2! favourable choices are $\{1,2\}$ and $\{2,1\}$ out of the $3^2 = 9$ combinations which are $\{1,1\}, \{1,2\}, \{1,3\} \dots \{3,1\}, \{3,2\}, \{3,3\}$. This probability has to be multiplied by the probability that the operator randomly chooses to mutate 2 bits rather than 1 or 3. Such a probability is $1/\gamma = 1/3$).

In extending equation (1) to strings belonging to an alphabet of cardinality K, with each point of the search space represented by a vector $\{0, 1, 2, \dots, K-1\}^\gamma$, the probability that each of the digits to be mutated actually turns into the correspondent digit of the optimal string needs to be considered:

$$P_c^{(\gamma)} = \frac{c!}{\gamma^c}\frac{1}{(K-1)^c}\frac{1}{\gamma} \tag{2}$$

Here $\frac{1}{K-1}$ is the probability that a single digit mutates to the correct value.

As $P_c^{(\gamma)}$ is always positive, *condition 1* is proved. There is also a probability $P_s > 0$ that the optimum is randomly introduced in the population by the selection operator, that should be considered, although $P_c^{(\gamma)}$ is enough to prove condition 1.

To prove that *condition 2* also holds we need to take into consideration all the operators acting on the population and show that none of them will ever be responsible for the loss of the optimal solution once it has been found.

The cloning operator creates copies of individuals but does not modify the values of any element so it cannot lose the optimum.

The hypermutation operator only acts on the individuals of the population $P^{(clo)}$ introduced by the cloning operator but does not modify individuals which have been created by any other operator, including itself.

The aging operator does get rid of old individuals but the best candidate solution of each generation is given $age = 0$ (or $P_{die} = 0$ according to which operator is used). Hence, it is impossible that Aging loses the copy of the optimum, unless it deletes individuals of $age = 0$ which seems to be pointless (by definition of Aging operator).

At last the selection operator, gets rid of the least fit individuals, so the optimum does not risk to be lost. In the *no redundancy* variant if there is more than one optimum in the population then only one optimal individual will survive the selection process. This is sufficient to prove *condition 2* and the proof of the theorem follows.

∎

In such a way the convergence of a great range of IAs can be proved as long as an operator (usually hypermutation or selection) satisfies condition 1 and as long as it can be shown that condition 2 also holds. This is usually done by verifying the elitism of the aging operator and very rarely (in the case of IAs) of Selection. For instance the proof of theorem 3.1, with minor modifications, would also hold for the B-cell algorithm previously proved to converge by using Markov chains in [18]. The B-cell algorithm does not use an Aging operator and employs Hypermacromutation rather than the hypermutation operator considered in theorem 3.1. It would be sufficient to show that hypermacromutation satisfies condition 1 to obtain the proof.

Theorem 3.2: The IA considered in this paper does not converge completely to the unique global optimum of an optimization problem regardless of its initialisation if the hypermutation operator only flips distinct bits and a non elitist variant of an aging operator is applied.

b) Proof.: In theorem 3.1 *condition 1* has been proved as long as the IA uses an operator (i.e. hypermutation or selection) which is able to reach any other possible individual of the search space. This, as discussed previously, assures the IA will visit the optimum. To prove convergence *condition 2* also needs to be satisfied. It is important to notice that *condition* 2 is *sufficient* for convergence, but not *necessary*. To prove that an EA does not converge to the global optimum it is sufficient to prove that whenever the optimum has been found there will always exist a successive generation in which the EA does not have the optimum in the population. If there is a probability, no matter how low, that the IA will lose the optimum without having found another optimal solution in the mean time, then it does not converge with probability one and far less completely. Such a probability is guaranteed by the non-elitism of the aging operator, since when the individual representing the optimum reaches a sufficient age it will be removed from the population.

Let at generation t, the population $P^{(t)}$ consist of X_1 optimal individuals and $X_2 = P^{(t)} - X_1$ non-optimal individuals. At time $t + \tau_B$, all the X_1 optimal solutions will have been surely removed from the population (for simplicity static aging is considered). The proof consists of showing that there always is a positive probability that all the optimal solutions have been removed from the population before a new one has been introduced, whatever the number of optimal individuals and the values of the non-optimal ones. The cloning operator will create a population of clones consisting of copies of the X_1 optimal solutions and of the X_2 non optimal solutions. Let the clones of the X_2 non-optimal solutions be m_{X_2}. Since the hypermutation operator always flips at least one bit, the clones of the optimal solutions will not produce global optima. On the other hand each of the m_{X_2} clones may turn into the global optimum at the next step with probability p, such that $p \leq p_{d=1}$. Here, $p_{d=1}$ is the probability of reaching the optimum from the most likely position, which is at distance $d = 1$. Hence, the probability of none of them turning into the global optimum is higher than $(1-p_{d=1})^{m_{X_2}}$. This probability is minimised when m_{X_2} is maximised. This occurs when there is only one global optimum in the population $P^{(t)}$ (i.e. $X_1 = 1$ and $X_2 = n - 1$). If a new global optimum has not been generated, the selection operator will take, at most, all the X_1 optimal individuals to the next generation (and their age will be incremented by 1), together with the best non optimal individuals. Furthermore it may introduce some new randomly created individuals. So, a lower bound on the probability of the hypermutation operator of not generating the global optimum before all the global optimums in the current population are lost is:

$$p_{NO} \geq (1 - p_{d=1})^{m_{n-1}\tau_B} > 0.$$

Also the probability of the selection operator of not randomly introducing the global optimum in τ_B generations is to be considered together with the described above probability. Although it is expected that for non-trivial functions the former probability is very low. Hence the optimal solutions will be eventually lost (i.e. in finite time). The proof follows.

■

A hypermutation operator that only flips distinct bits is chosen for the theorem because it simplifies the proof. In the extreme case that all the individuals in the population are all optimal individuals there will be no optimum in the population after at most τ_B generations unless a new optimal solution is introduced by the selection operator which is rather unlikely. Furthermore, in practical applications, the hypermutation operator chooses distinct bits. On the other hand, theorem 3.1 holds with both kinds of hypermutation.

In the general IA, the possibility of using stochastic selection in the evolutionary process is not considered since it is not very common to find such an operator in an IA. In such a case condition 2 for theorem 3.1 would not hold unless some elitist mechanism is combined with the probabilistic selection.

IV. Convergence in probability

For genetic algorithms, and evolutionary algorithms in general, it is possible to calculate the number of iterations that guarantee the obtainment of an optimal solution with a fixed probability δ, with $0 < \delta < 1$. This upper bound $t(\delta)$, is independent of the problem being tackled.

In [7] the authors use Markov chains to detect bounds for a genetic algorithm designed for binary coded problems by ensuring that every possible population has been visited by the algorithm with a probability of at least δ. While, in [6] the authors extend their results from binary representations to alphabets of cardinality $K = 2^x$, in [5] a tighter upper bound has been derived by guaranteeing that *all possible individuals*, have been seen at least once with probability δ, rather than all possible populations:

$$\tilde{t}_1(\delta) = \text{INT}\left[\frac{\ln(1-\delta)}{n\ln\left[1 - \min\left\{\left(1-\mu\right)^{\gamma-1}\left(\frac{\mu}{K-1}\right), \left(\frac{\mu}{K-1}\right)^{\gamma}\right\}\right]}\right] \tag{3}$$

Here n is the size of the population and γ the string length.

The goal of the following analysis is to find an upper bound for the number of generations necessary to guarantee a visit of the considered IA to a global optimum in t_1 generations under probability δ.

Let $\{0, 1, 2, ..., K-1\}^\gamma$ be the vector representing the search space of the optimization problem with an alphabet of cardinality K and solutions of length γ.

As discussed in section III-C, the probability that each of the digits to be mutated actually turns into the correspondent digit of the optimal string is:

$$P_c^{(\gamma)} = \frac{c!}{\gamma^c}\frac{1}{(K-1)^c}\frac{1}{\gamma} \geq \frac{\gamma!}{\gamma^\gamma(K-1)^\gamma}\frac{1}{\gamma} \tag{4}$$

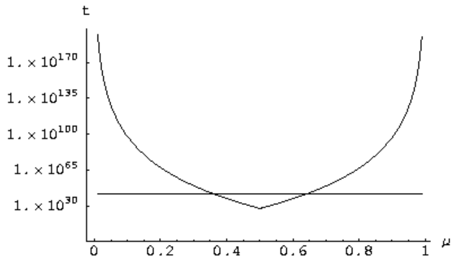

Fig. 1. The upper bound (the straight line) for the immune algorithm, and the upper bound for the genetic algorithm versus μ values. Both algorithms use a binary alphabet ($K = 2$). The analysis is performed with $\delta = 0.9, n = 1000$, and $\gamma = 100$.

Here $\frac{1}{K-1}$ is the probability that a single digit mutates to the correct value. The proof of the inequality is trivial since for $K = 2$

$$\frac{\gamma!}{\gamma^\gamma} = \frac{c!}{\gamma^c}\frac{(c+1)}{\gamma}\cdots\frac{(\gamma-1)}{\gamma}\frac{\gamma}{\gamma} \leq \frac{c!}{\gamma^c}$$

and for $K \geq 2$, $(K-1)^\gamma \geq (K-1)^c$.

The probability that the hypermutation operator does not convert a specific string into the optimal one with a population of n individuals and in t generations is at most $(1 - P_{IA})^{tn}$, hence the probability that the optimum has been seen in t generations time is $P_t \geq 1 - (1 - P_{IA})^{tn}$.

Now it is possible to fix the number of generations t_1 for probability P_t to be greater than δ:

$$1 - (1 - P_{IA})^{t_1 n} \geq \delta.$$

Thus

$$t_1 \geq \frac{log(1-\delta)}{nlog(1-P_{IA})} = t_{IA} \tag{5}$$

Here t_1 is the number of required generations for the Immune Algorithm to see the global optimum with a probability of at least δ. By solving the inequality $t_{IA} \leq t_{GA}$, it turns out that, as long as γ is sufficiently large (i.e. $\gamma >> 1$) the upper bound for the IA is lower than that of the GA for values of μ satisfying:

$$\begin{cases} \mu < \frac{1}{e} & \text{if } \mu \leq \frac{K-1}{K} \\ \mu > 1 - \frac{1}{e}\frac{1}{K-1} & \text{if } \mu \geq \frac{K-1}{K} \end{cases} \tag{6}$$

In particular for a binary alphabet we have

$$\begin{cases} \mu < \frac{1}{e} & \text{if } \mu \leq \frac{1}{2} \\ \mu > 1 - \frac{1}{e} & \text{if } \mu \geq \frac{1}{2} \end{cases} \tag{7}$$

The results of such a comparison can be viewed numerically in figure 1. In this plot we show the upper bound computed for the immune algorithm (equation 5), and the upper bound obtained in [5] for the genetic algorithm versus the mutation probability μ. The value chosen for the population size n is typical both for genetic and immune algorithms and the string length γ should be sufficiently large. Since the hypermutation operator does not depend on a mutation probability μ, its corresponding curve is a straight line.

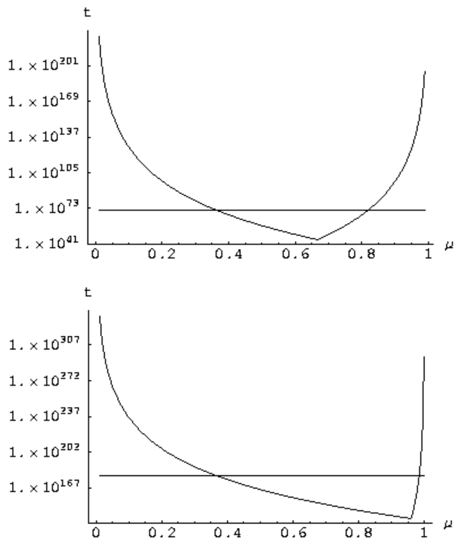

Fig. 2. The upper bound (the straight line) for the immune algorithm, and the upper bound for the genetic algorithm versus μ values. Upper plot: both algorithms use an alphabet of cardinality $K = 3$. Lower plot: both algorithms use an alphabet of cardinality $K = 25$.

Although the analysis has theoretical significance, its problem independence represents the main weakness of the results since more realistic bounds, hence of greater interest, can be obtained by taking into consideration the problem class of functions to be optimised. In [23] it has already been discussed how in the best case (i.e. $\mu = 0.5$ giving the lowest bound-value) the required number of generations is the same as that of a Random Algorithm (RA) choosing independently n random individuals per generation. Furthermore for low values of μ, which are the ones of practical interest, the required number of generations grows to infinity. For the IA the last consideration does obviously not apply but the number of generations required to guarantee the optimum is still higher than that of the RA. Figure 2 shows the upper bounds for alphabets of cardinality $K = 3$ and $K = 25$ with the same parameters used for figure 1.

V. Conclusion

In this paper a first general step towards the theoretical analysis of IAs has been performed. Conditions for the convergence of such a class of algorithms are given, and examples of how to evaluate such conditions are applied to a general IA. An analysis of the time needed to guarantee an optimal solution using a simple Hypermutation operator confirms the general knowledge regarding EAs that the analysis of their complexity needs to be related to the problem class to be optimised and shows that such a consideration is also valid in the theory of IAs.

Analyses of simple EAs on simple boolean functions have given a broader insight of how useful different operators may be and of what kind of landscapes they are effective on. Since such an approach has progressively led to the analyses of

EAs on combinatorial optimization problems with practical applications, a similar strategy in analysing IAs is worth considering. The examination of simple AIS inspired operators could give a better insight of how IAs actually work and when one of the many variants of each operator should be chosen rather than another. Furthermore it would be interesting to understand if so many of the existing variants of each operator are really useful or not.

REFERENCES

[1] D. Dasgupta D. (ed.), *Artificial Immune Systems and their Applications*, Berlin, Germany: Springer-Verlag, 1999.

[2] L. N. De Castro and J. Timmis, *Artificial Immune Systems: A New Computational Intelligence Paradigm*, London, UK: Springer-Verlag, 2002.

[3] V. Cutello and G. Nicosia and M. Pavone, *A Hybrid Immune Algorithm with Information Gain for the Graph Coloring Problem*, GECCO '03, Chicago, IL, USA, LNCS, Springer, vol. 2723, pp.171-182, 2003.

[4] L. N. de Castro and F. J. Von Zuben *Learning and optimization using the clonal selection principle*. IEEE Trans. on Evol. Comp., vol. 6, no. 3, pp. 239-251, 2002.

[5] D. Greenhalgh and S. Marshall, *Convergence Criteria for Genetic Algorithms*, SIAM J. Comput., vol. 30, No. 1, pp. 269-282, 2000.

[6] H. Aytug and S. Bhattacharrya and G. J. Koehler, *A Markov chain analysis of genetic algorithms with power of 2 cardinality alphabets*. European J. Oper. Res. vol. 96, pp. 195-201, 1996.

[7] H. Aytug and G. J. Koehler, *Stopping criteria for finite length genetic algorithms*. INFORMS J. on Comp., vol. 8, no. 2, pp. 183-191, 1996.

[8] J. He and X. Yao, *Drift analysis and average time complexity of evolutionary algorithms*, Artificial Intelligence, vol. 127, no. 1, pp. 57-85, 2001.

[9] G. Rudolph, *Finite Markov Chain Results in Evolutionary Computation: A Tour d'Horizon* Fundamenta Informaticae, vol. 35, pp. 67-89, 1998.

[10] D. E. Goldberg, *Genetic Algorithms for Search, Optimization, and Machine Learning*, Addison-Wesley Pub. Co., 1989.

[11] C. Witt, *Worst-Case and Average-Case Approximations by Simple Randomized Search Heuristics*, in Proc. of the 22nd Annual Symposium on Theoretical Aspects of Computer Science (STACS '05), LNCS, Springer, vol. 3404, pp. 44-56, 2005.

[12] O. Giel and I. Wegener, *Evolutionary algorithms and the maximum matching problem*, in Proc. of the 20th Annual Symposium on Theoretical Aspects of Computer Science (STACS 2003), LNCS, Springer, vol. 2607, pp. 415-426, 2003.

[13] I. Wegener and F. Neumann, *Randomized local search, evolutionary algorithms, and the minimum spanning tree problem*, GECCO'2004, Seattle, WA, USA, LNCS, Springer, vol. 3102, pp. 713-724, 2004.

[14] G. Nicosia and V. Cutello and P. J. Bentley and J. Timmis, *Artificial Immune Systems*, Third Int. Conf. (ICARIS 2004), Catania, Italy. LNCS, Springer-Verlag, vol. 3239, 2004.

[15] V. Cutello and G. Nicosia and M. Pavone and J. Timmis, *An Immune Algorithm for Protein Structure Prediction on Lattice Models*, IEEE Transactions on Evol. Comp., vol. 10, 2006 (to appear).

[16] V. Cutello and G. Nicosia, *An Immunological Approach to Combinatorial Optimization Problems*, Advances in Artificial Intelligence, IBERAMIA (2002). LNAI, Springer vol. 2527, pp. 361-370, 2002.

[17] M. Villalobos-Arias and C. A. Coello Coello and O. Hernández-Lerma, *Convergence Analysis of a Multiobjective Artificial Immune System Algorithm*, In Nicosia *et al.* (eds) Proc. Int. Conf. Artificial Immune Systems (ICARIS 2004), LNCS, Springer, vol. 3239, pp. 226-235, 2004.

[18] E. Clarke and A. N. W. Hone and J. Timmis, *A Markov Chain Model of the B-cell Algorithm*, ICARIS 2005, LNCS, Springer, vol. 3627, pp. 318-330, 2005

[19] G. Nicosia and F. Castiglione and S. Motta, *Pattern Recognition by primary and secondary response of an Artificial Immune System*, Theory in Biosciences, vol. 120, no.2, pp. 93-106, 2001.

[20] V. Cutello and G. Narzisi and G. Nicosia and M. Pavone, *Clonal Selection Algorithms: A Comparative Case Study using Effective Mutation Potentials*, ICARIS 2005, LNCS, Springer, vol. 3627, pp.13-28, 2005.

[21] G. Nicosia, *Immune Algorithms for Optimization and Protein Structure Prediction*, PhD Thesis, Department of Mathematics and Computer Science, University of Catania, Italy, December 2004.

[22] T. Back and D. B. Fogel and Z. Michalewicz, *Handbook of Evolutionary Computation*, Bristol, UK, IOP Publishing, 1997.

[23] M. Safe and J. A. Carballido and I. Ponzoni and N. B. Brignole, *On Stopping Criteria for Genetic Algorithms.*, Advances in Artificial Intelligence - SBIA 2004. 17th Brazilian Symposium on Artificial Intelligence. Proceedings (Lecture Notes in Artificial Intelligence Vol.3171), 2004, p 405-13.