The Influence of Age Assignments on the Performance of Immune Algorithms

Alessandro Vitale[1], Antonino Di Stefano[1], Vincenzo Cutello[2] and Mario Pavone[2]

[1] Department of Electric, Electronics and Computer Science
University of Catania
v.le A. Doria 6, I-95125, Catania

[2] Department of Mathematics and Computer Science
University of Catania
V.le A. Doria 6, I-95125 Catania, Italy
cutello@dmi.unict.it, mpavone@dmi.unict.it

Abstract. How long a B cell remains, evolves and matures inside a population plays a crucial role on the capability for an immune algorithm to jump out from local optima, and find the global optimum. Assigning the right age to each clone (or offspring, in general) means to find the proper balancing between the exploration and exploitation. In this research work we present an experimental study conducted on an immune algorithm, based on the clonal selection principle, and performed on eleven different age assignments, with the main aim to verify if at least one, or two, of the top 4 in the previous efficiency ranking produced on the one-max problem, still appear among the top 4 in the new efficiency ranking obtained on a different complex problem. Thus, the NK landscape model has been considered as the test problem, which is a mathematical model formulated for the study of *tunably rugged fitness landscape*. From the many experiments performed is possible to assert that in the elitism variant of the immune algorithm, two of the best age assignments previously discovered, still continue to appear among the top 3 of the new rankings produced; whilst they become three in the no elitism version. Further, in the first variant none of the 4 top previous ones ranks ever in the first position, unlike on the no elitism variant, where the previous best one continues to appear in $1st$ position more than the others. Finally, this study confirms that the idea to assign the same age of the parent to the cloned B cell is not a good strategy since it continues to be as the worst also in the new efficiency ranking.

1 Introduction

The Immune algorithms are an established and successful computational methodology, which take inspiration from the information processing mechanism of the living things, and from their dynamics. What makes the immune system really challenging from a computational perspective is its ability in recognize, distinguish, detect, and remember (memory) foreign entities to the living organism, as well as its capability in learning, and self-regulation [6]. As in any evolutionary algorithms, also in the immune algorithms the success's key for having efficient performances is given by an appropriate, and proper balancing between the exploration and exploitation mechanisms: the first is carried out by the perturbation and recombination operators, whilst the last one is obtained through the selection process adopted. What makes, however, more efficient and accurate the search process, and at the same time helps to learn information as more as possible, is given by the right maturation time of each solution. Indeed, how long a solution stays, evolves, and matures inside a population is strictly related to a good balancing of the exploration/exploitation processes, and, therefore, plays a decisive role in the performances of any evolutionary algorithm.

In a previous work published in [5], a study on how much time an individual needs to stay into the population to properly explore the search space has been conducted. This study has been performed on an Immune Algorithm (IA), and the outcomes obtained have been summarized in an efficiency ranking of the several age assignment types considered. Further, this ranking was performed and produced on the one-max problem [12, 4], one of the classical toy problems used for understanding the dynamics, variants and search's ability of any stochastic algorithm [2]. In light of this, in this paper we present the same experimental study but tackling a different problem, such as the *NK-model* [8] that is a mathematical model able to describe a *"tunably rugged"* fitness landscape, with the primary goal to verify if the top 4 in the previous ranking, or at least 2 of them, still appear among the top 4 in the new efficiency ranking produced on this model. If so, we give a reliable age's assignment that provides, with high probability, efficient and robust performances in discrete search spaces to can use especially in uncertainty environments. Thus, the same IA proposed in $[1, 11]$ was developed, whose core components are the cloning, hypermutation and aging operators, and eleven (11) age assignment types have been considered for being studied. The *Elitism* and *No Elitism* versions were investigated as well. Inspecting the new outcomes it is possible to assert that (a) in the elitism version, two of the top 4 of the previous ranking appear among the top 3 of the new ranking, but none of them rank in the first position; (b) in the no elitism variant, instead, three of the previous top 4 are still among the first three positions of the new efficiency ranking, and, further, the best of the previous ranking continues to be still the best one; finally, (c) the worst of the previous ranking still continues to be in the last position in the new one.

2 The NK Landscape Model

For validating and generalizing the outcomes obtained, it is needed to avoid that the algorithm is tailored to a specific problem, keeping it instead unaware on the knowledge of the application domain. In this way, the efficiency ranking produced will be suitable and applicable also on other problems. At this end, the NK-model was considered for validating the experiments performed, and the outcomes produced. This model was developed in [8, 9] as a powerful analytic tool able to model and represent the effects of epistatic interactions in population genetic. The basic idea behind this model is that in complex systems with many components (N), the functional contribution of each component is affected by the interaction of one, or more (K) parts of the system. The multidimensional space in which each component is represented by a dimension of the space, to which is assigned a fitness level with respect to a specific property, constitutes the *fitness landscape*, whose topology depends on the interactions and on the degree of interdependence of the functional contribution of the various components. This interdependence degree influences strictly the smoothness or ruggedness of the fitness landscape: if a component is affected by a variety of other parts, then the landscape produced will be quite *rugged* [10]. The NK-model represents, a mathematical model that captures the central factors for an ensemble of tunable rugged fitness landscapes, via changes between overall size of the landscape and the number of its local *hills and valleys*. It is based on two parameters: (1) the number N of components; and (2) a parameter K that measures the richness of interactions among components. Increasing this last parameter means to increase the number of peaks and valleys, and thus raising the ruggedness of the corresponding fitness landscape: moving from single peaks and smooths, to multi-peaks and fully uncorrelated. Any component is assumed to be in a binary state, 0 or 1, and the fitness contribution of such a component depends on its own state, and the one of those K other components explicitly linked to it.

Formally, given a bit string $s = (s_1, s_2, \ldots, s_N)$ of length N, it assigns a fitness contribution ϕ_i to each locus s_i of the N residue chain such that ϕ_i depends on s_i and K other bits $(0 \leq K \leq N - 1)$. There are $2^{(k+1)}$ combinations of states of the $K + 1$ loci that determine the fitness contribution of each locus. For each of this combination is randomly selected a variable in $[0, 1]$ as its fitness contribution. The total fitness of the string s is, then, defined as the average of the fitness contributions of each part, and the ones of K neighbors that affect upon it:

$$F_{NK}(s) = \frac{1}{N} \sum_{i=1}^{N} \phi_i(s_i, s_{i+1}, s_{i+2}, \ldots, s_{i+K}) \tag{1}$$

The goal in dealing with the NK-model is to *maximize* the equation above. Note that the tuning of the parameter K alters how rugged the landscape should be, and then affords us a tunable rugged fitness landscape. Indeed, for $K = 0$ each site is independent of all other ones, and this generate the smoothest landscape; whilst, for $K = N - 1$ the fitness contribution of each site depends on all of other ones, producing then most rugged landscapes, with very many local optima. Solving the NK-model was proved to be NP-complete problem [13]. This model, thanks to its properties, allows us to keep the study, and its validation within the discrete domain, as done in [5], and, in the same time, testing it on a different rugged fitness landscape level.

3 The Immune Algorithm

In order to achieve the prefixed purposes in this research work, we have faithfully developed the immune inspired algorithm proposed in [5, 3], and whose main features are given by the three operators: (1) cloning, which generates a new population centered on the higher affinity values, (2) inversely proportional hypermutation, which explores the neighborhood of each point in the search space, and (3) aging, which introduces diversity in the population and avoid to get trapped into local optima. This algorithm is based on four main parameters (all user-defined), such as population size (d); number of clones to be produced (dup); mutation rate (ρ); and maximum number of generations allowed to a B cell to stay into the population (τ_B). Its pseudo-code is showed in Algorithm 1.

Algorithm 1 The Immune Algorithm (d, dup, ρ, τ_B)

```
t ← 0;
P^(t) ← Initialize_Population(d);
Evaluate_Fitness(P^(t));
repeat
    Increase_Age(P^(t));
    P^(clo) ← Cloning (P^(t), dup);
    P^(hyp) ← Hypermutation(P^(clo), ρ);
    Evaluate_Fitness(P^(hyp));
    (P_a^(t), P_a^(hyp)) ← Aging(P^(t), P^(hyp), τ_B);
    P^(t+1) ← (μ + λ)-Selection(P_a^(t), P_a^(hyp));
    t ← t + 1;
until (stop criterion is satisfied)
```

The immune algorithm begins with the inizialization of the population $P^{(t=0)}$ of size d by generating random solutions in the binary domain $(\{0, 1\})$; after that, for each B cell $(x \in P^{(t)})$, i.e. a candidate solution, the fitness value is computed via the Evaluate_Fitness$(P^{(t)})$ procedure (lines 3 and 8 in Algorithm 1). Inside the evolutionary process the three main operators are performed together to a selection operator, which attempts to exploit as better as possible the information gained during the evolution. The main loop (i.e. the evolution) terminates once a fixed stop criterion is satisfied, that is when it is reached the maximum number of fitness function evaluations allowed (T_{max}).

The first immune operator applied is the *cloning operator*, which simply copies dup times each solution (i.e. each B cell) producing an intermediate population $P^{(clo)}$ of size $(d \times dup)$. Once a clone is produced to this is assigned an age that determines its lifetime into the population. Indeed, based on this assignment, the B cell will live until its age will be greater than τ_B. For doing so, at the beginning of each time step t, the age of all B cells in the population is increased by one (line 5 in Algorithm 1). The assignment of the age to each B cell, together with the aging operator, has the purpose to reduce premature convergences, and keep a right diversity into the population. Then, what age to assign to each clone (or individual) plays a crucial role on the performances of any evolutionary algorithm. Therefore, in order to analyse and confirm if the previous efficiency ranking produced is still valid for this new model, even if in different order, we have conducted the same age assignment study proposed in [5], evaluating the overall performances in terms of performance, convergence and success. Eleven different age assignment types for each clone has been considered in this study:

1. `(type0)` age zero (0);
2. `(type1)` random age chosen in the range $[0, \tau_B]$;
3. `(type2)` random age chosen in the range $[0, \frac{2}{3}\,\tau_B]$. This option guarantees to each B cell to evolve at least for a minimal number of generations (in the worst case $\frac{1}{3}\,\tau_B$);

4. `(type3)` random age chosen in the range $[0, inherited]$, where with *inherited* we indicate the same age of the parent. With this option, in the worst case the clone will have the same age of its parent;
5. `(type4)` random age chosen in the range $[0, \frac{2}{3}\, inherited]$. In this way for each offspring is guaranteed a lower age than the parent;
6. to each clone is assigned the same age of the parent (inherited), but if after M mutations performed on the clone its fitness value improves, then its age is updated as follows:
   (a) `(type5)` zero;
   (b) `(type6)` randomly chosen in the range $[0, \tau_B]$;
   (c) `(type7)` randomly chosen in the range $[0, \frac{2}{3}\, \tau_B]$;
   (d) `(type8)` randomly chosen in the range $[0, inherited]$;
   (e) `(type9)` randomly chosen in the range $[0, \frac{2}{3}\, inherited]$;
7. `(type10)` same age of parent less one $(inherited - 1)$.

We would to highlight, and stress once again, that what we want to prove with this research study is not to get the same and identical efficiency ranking previously produced, but rather verify if the previous best 4, or just some of these, in overall still appear among the top positions in the new ranking. In this way, we may provide one or more age assignments to be considered, for having reliable and efficient performances, especially when it is tackled uncertainty environments.

The *hypermutation operator* acts on each solution of population $P^{(clo)}$ performing M mutations, whose number is determined by an *inversely proportional law*, that is the higher is the fitness function value, the lower is the number of mutations performed on the B cell. Interestingly that this perturbation operator is not based on any mutation probability. In particular, the number of mutations M to perform over a clone $\mathbf{x}$ is determined by $\alpha = e^{-\rho \hat{f}(\mathbf{x})}$, where α represents the mutation rate, and $\hat{f}(\mathbf{x})$ the fitness function value normalized in $[0, 1]$. The number M of mutations is then determined by $M = \lfloor (\alpha \times \ell) + 1 \rfloor$, with ℓ the length of the B cell. In this way, at least one mutation is guaranteed on each B cell; and this happens exactly when the solution is very close to the optimal one. For each clone, the hypermutation operator randomly choose a bit, and it inverts its value (from 0 to 1, or viceversa), and this is repeated without redundancy for M times. At the end, all hypermutated clones produce a new population, labelled $P^{(hyp)}$. During the normalization of the fitness function into the range $[0, 1]$, the best current fitness is decreased by a threshold θ, and it is used in place of the global optima, because usually it is often unknown. In this way, no *a priori* knowledge about the problem is used.

Aging operator acts as last immune operator with the main goal to produce enough diversity into the population in order to avoid premature convergences and then getting trapped into local optima. It eliminates the old B cells from the populations $P^{(t)}$ and $P^{(hyp)}$: every B cell is allowed to stay in the population for a fixed number of generations τ_B; then as soon as a B cell becomes older than τ_B it is removed from the population of belonging independently from its fitness value. There exists a variant of this operator, called elitism version, which makes an exception on the best solution found so far: it is always kept in the population, even it is older than τ_B. In this experimental study, both variants of the aging operator have been taken into account: elitism and no elitism. An analysis on the benefits and efficiency of the aging operator can be found in [7].

After the three immune operators have been performed, a new population $P^{(t+1)}$ for the next iteration is produced by the merging of the best B cells via the $(\mu + \lambda)$-*Selection operator*, which, then, selects the best d survivors to the aging step from the populations $P_a^{(t)}$ and $P_a^{(hyp)}$. Such operator, with $\mu = d$ and $\lambda = (d \times dup)$, reduces the offspring B cell population of size $\lambda \geq \mu$ to a new parent population of size $\mu = d$. The selection operator identifies the d best elements from the offspring set and the old parent B cells, thus guaranteeing monotonicity in the evolution dynamics. Nevertheless, due to the aging operator, it could happen that only $d_1 < d$ B cells survived; in this case, the selection operator randomly generates new $d - d_1$ B cells.

4 Results and Discussion

In this section all outcomes obtained in the experimental study conducted are presented, whose main aim is to investigate if the best age assignments previously produced are still valid in solving a new model, working always in discrete domain (bit strings), but facing a more rugged fitness landscape. For these experiments we have considered the NK-model that is a mathematical model based on two parameters (N and K), from which is possible to produce tunable rugged landscapes. In particular, tuning the parameter K allows to alter the roughness of the landscape: from the smoothest produced for $K = 0$, to the roughest one with many local optima for $K = N-1$. For our test case we have considered $N = 100$, and $K = \frac{N}{2} = 50$. In this way, a more rugged fitness landscape is produced with many local optima, which makes surely harder the search process for the global optima but, on the other hand, allow us to better validate the outcomes obtained. In figure 1 is showed the fitness landscape produced for $N = 12$, and $K = 6 = \frac{N}{2}$. It was considered $N = 12$ due to the high number of the all binary strings (2^N) that affect the given landscape. It gives anyway an idea about the hardness of the landscape of the problem to be solved.

In order to validate all results, and produce the new efficiency ranking, we used the same experimental protocol proposed in [5], that is: $d = \{50, 100\}$; $dup = \{2, 5, 10\}$; $\tau_B = \{5, 10, 15, 20, 50, 100, 200\}$; $T_{max} = 10^5$; and each experiment was performed on 100 independent runs. Because the parameter ρ is related only to the problem dimension (N), after several tests, it was fixed to 5.9 for all experiments. Further, all experiments were performed for both versions of IA: *elitism*, and *no elitism* variants. Since the optimal solution is unknown, we have considered as evaluation measures, in order, respectively: (i) mean of the best solutions found on 100 independent runs; (ii) best solution found; and (iii) standard deviation (σ). Of course, seeing the high number of experiments performed, in this section we report figures and tables of the more meaningful results.

Note that each plot in figures 2 and 3, and for each group of columns inside them (i.e. τ_B values), shows the age assignments in increasing order, as suggested in section 3, from left to right (`type0` leftmost, and `type10` the rightmost one).

Figure 2, plots (a) and (b), shows the results produced by the developed immune algorithm on the 11 age assignment test cases. Each plot reports the *mean* of the best

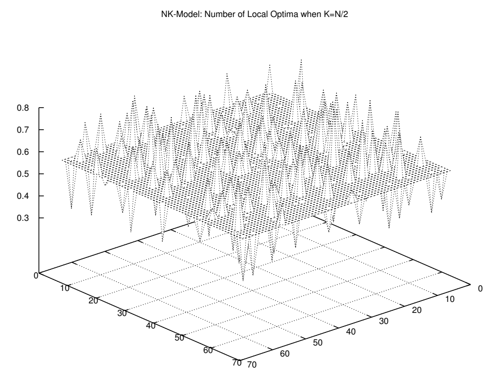
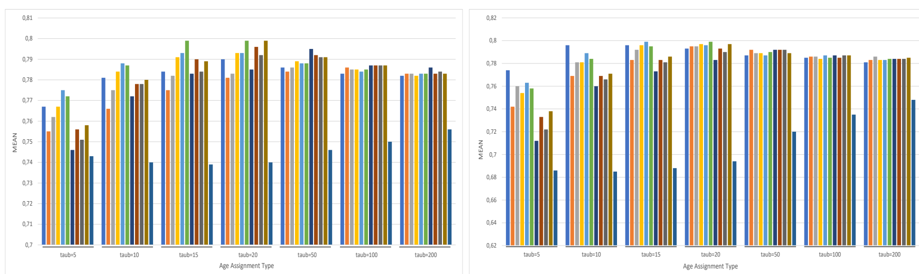
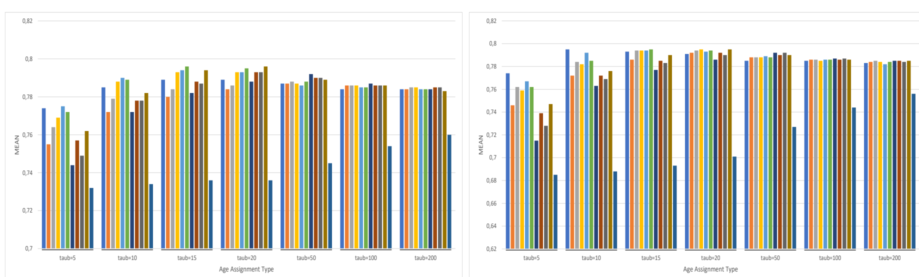


Fig. 1. Rugged fitness landscape produced by setting $N = 12$ and $K = N/2$.



(a)



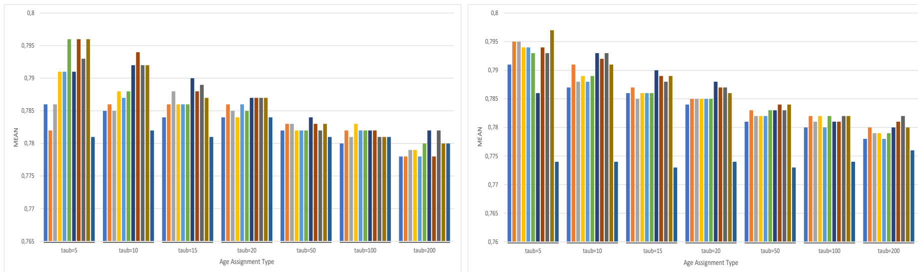
(b)

Fig. 2. Mean of the best solutions versus age types. Results are obtained by both immune algorithm versions: elitism (upper plots) and without elitism (bottom plots). Plot (a) shows the results obtained by setting $d = 50$ and $dup = 2$; whilst the ones of plot (b) using the parameters $d = 100$ and $dup = 2$.

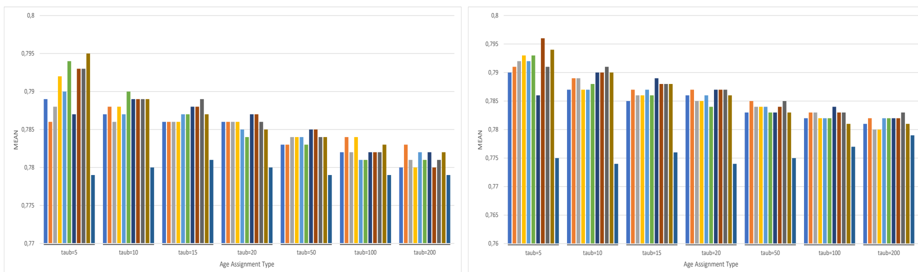
solutions found on the 100 independent runs by varying the τ_B parameter, and for both variants of the algorithm: with elitism (upper plots) and without elitism (bottom plots). In particular, the plot (a) is obtained by setting $d = 50$ and $dup = 2$; whilst plot (b) using the parameters $d = 100$ and $dup = 2$. Inspecting plot (a) of figure 2, is possible to see how at least 2 of the previous best 4 age assignments (`type0`, `type4`, `type3` and `type2`, in the order) often appear among the first positions, although not always as the best ones. This appears clearer if we analyse the no elitism version (bottom plot). In particular, at low values of τ_B all 4 best previous ones appear among the new $4-5$ top positions, and sometimes also in the same order of the previous ranking. It is interesting to note from both plots (a) how for low values of τ_B the performances at various age types are feelingly different; whilst at the increasing of τ_B the performances are roughly equivalents. In both variants, the overall best performances, in terms of mean of best solutions found, are obtained with $\tau_B = 20$. What instead is clear in both plots, and confirms what asserted in the previous efficiency ranking, is that the last age assignment type proposed (`type10`) still continues to be the worst. Same analysis may be done also for the plot (b) in figure 2. In addition to the previous analysis is possible to assert for both versions that all previous 4 top age assignment types appear always among the $4-5$ best (reflecting the same order in some way), except for $\tau_B = 50$. Also in these experiments, the best performances in the overall are obtained by setting $\tau_B = 20$, besides, the last age assignment option (`type10`), that is to assign the same age of parent minus 1, continues to be a bad choice.

Plots (a) and (b) in figure 3 report, instead, the performances of both variants of the immune algorithm on all age assignment test cases when $dup = 10$, and by varying the population size. The upper plots show the results obtained with the elitism version, whilst on the bottom ones the variant without elitism. Analysing the plots (a), it is possible to see for these experiments none of the top 4 previous ones appears in steady way among the new best positions, and when they appear they are in sparse order. This is due to the high number of clones (10 for each B cell) that require a greater diversity into the population, which is better produced by the age assignment types 5 to 9. This statement is also confirmed by having better performances at low τ_B values. However, if we consider only the rank, and possible equivalent ranks, then at least one of the previous best 4 is almost always among the $4-5$ best in the new ranking. The plots (b) show a similar behavior to previous ones, but at the increasing of τ_B values it becomes more pronounced the presence of the previous top age types among the new best ones. Also in these experiments, the `type10` appears as the worst age assignment.

Table 1 reports the efficiency ranking produced for each pair of values (d, dup). In boldface is highlighted when one of the previously best 4 [5] is ranked in the top 4 positions of the new ranking. Looking to the elitism version for $d = 100$, we find `type3` and `type 4` among the first 4 ranked for $dup = \{2, 5\}$, whilst only the `type3` to $4th$ place for $dup = 10$. A different efficiency ranking is, instead, produced for $d = 50$, where for $dup = 2$ the `type3` and `type4` are ranked in $2nd$ and $3rd$ position respectively; whilst, for the last dup value, none of the previously top 4 achieve the first positions. This different behavior may be explained because by setting the population size to 50 the algorithm shows a stronger form of elitism that limits the effect of the age assignments. This means, in a nutshell, that the age type chosen alone is not able to produce a right diversity into the population. Inspecting the no elitism variant is clear how, for $dup = \{2, 5\}$, three over four of the previous ranking appear still among the top 4, as `type0`, `type2`, and `type4`. In particular, for small dup values `type0` is



(a)



(b)

Fig. 3. Mean of the best solutions versus age types. Results are obtained by both immune algorithm versions: elitism (upper plots) and without elitism (bottom plots). Plot (a) shows the results obtained by setting $d = 50$ and $dup = 10$; whilst the ones of plot (b) using the parameters $d = 100$ and $dup = 10$.

Table 1. Efficiency ranking produced by each pairs (d, dup).

type	$d = 50$			$d = 100$		
	$dup = 2$	$dup = 5$	$dup = 10$	$dup = 2$	$dup = 5$	$dup = 10$
	Elitism Ranking					
0	6	8	10	5	8	9
1	10	10	9	10	10	7
2	8	9	8	7	9	9
3	**3**	5	6	**3**	**3**	**4**
4	**2**	6	7	**2**	**4**	8
5	1	1	5	1	1	4
6	9	**7**	2	9	7	4
7	5	3	1	6	4	2
8	**7**	4	3	8	6	3
9	4	2	3	4	2	1
10	11	11	11	11	11	11
	No Elitism Ranking					
0	**1**	7	10	**1**	**4**	10
1	7	6	4	7	8	3
2	**4**	**3**	8	**4**	**3**	7
3	5	**4**	6	5	6	9
4	**2**	**2**	9	**2**	**1**	6
5	3	1	6	3	2	8
6	10	10	5	10	10	5
7	8	8	3	8	7	1
8	9	9	2	9	9	2
9	6	5	1	6	5	4
10	11	11	11	11	11	11

always the best option to choose; `type4` is always the second best; whilst `type3` fluctuates between $3rd$ and $4th$ position. If, instead, the number of clones is increased $(dup = 10)$, then, an opposite ranking to the one showed in [5] is produced, where the previous age types at the bottom of the ranking are now in the best ranks. This is because they are able to produce more diversity than the others, which is a key point when working with a high number of solutions. From these single efficiency rankings, what emerges clearly is that in any ranking the `type10` is always the worst. This confirms once again the outcome obtained in the previous work.

After analysing the single experiments, and individual efficiency rankings, the goal of this research work is to understand which age assignment types show best performances in the overall (i.e. independently of the parameter values used), and if among these appears one or more of the top 4 previous ones. At this end, in table 2 it is reported the summarized and overall outcomes. Thus, once computed the single rankings for each set value of d and dup, in table 2 we summarize everything, reporting the number of times that each age assignment appears among the top k (φ_k) over 6 overall experiments (by varying $d = 50, 100$ and $dup = 2, 5, 10$), and the statistic success rate in using that age assignment type. This last was computed only with respect to the top 4, which means that, inspecting all experiments, a given age type appears SR_{top4} times among the best 4. Obviously, the higher the percentage is, more robust and efficient are the performances using the relative age assignment type. Further, always in table 2, φ_5 indicates how many times the given age assignment appears among the top 5; φ_4 among top 4; and so on. From this table, analysing the elitism version, is possible to

Table 2. Summary of the overall outcomes, where φ_k reports the number of times that each age assignment appears among the top k, and SR_{top4} the statistic success rate that indicates the percentage of how many times the relative age type appears among the best 4.

	Elitism						no Elitism					
type	φ_5	φ_4	φ_3	φ_2	φ_1	SR_{top4}	φ_5	φ_4	φ_3	φ_2	φ_1	SR_{top4}
0	1	0	0	0	0	0%	3	3	2	2	2	50%
1	0	0	0	0	0	0%	2	2	1	0	0	33.33%
2	0	0	0	0	0	0%	4	4	2	0	0	66.67%
3	5	4	3	0	0	66.67%	3	1	0	0	0	16.67%
4	3	3	2	2	0	50.00%	4	4	4	4	1	66.67%
5	6	5	4	4	4	83.33%	4	4	4	2	1	66.67%
6	2	2	1	1	0	33.33%	2	0	0	0	0	0%
7	5	4	3	2	1	66.67%	2	2	2	1	1	33.33%
8	3	3	2	0	0	50%	2	2	2	2	0	33.33%
9	6	6	4	3	1	100%	4	2	1	1	1	33.33%
10	0	0	0	0	0	0%	0	0	0	0	0	0%

note how the best age assignment in the previous ranking (`type0`) appears only one time among the top 5, and never among the first 4; whilst the types 3 and 4, third and second position of the previous ranking, appear in these new experiments among the best 3, respectively 3 and 2 times. Need to be highlight as `type3` is among top 3 for 3 times over 6, but never appear in the first two positions, unlike of `type4` that it ranks in the first two positions 2 times over 6 experiments. Focusing only on the top, none of the previous best 4 is able to outperform the other age assignment types. In particular, `type 5` is the age type that shows the best performances, winning 4 times over 6. In general, it is possible to assert that the `type5`, `type7` and `type9`, the only ones to appear as top, represent an assignment type that introduce diversity into the population, but in the same time leaves enough time for the maturation of all promising solutions. Inspecting the no elitism version, we achieve a different behavior of the performances with respect the other variant (of course, as we expected), where, instead, the previous best one still continues to be the top (2 over 6) also in this case study, together with the `type4` (the previous $2nd$) that appears in the first rank 1 times. Looking to the first 4 ranks, is possible to see as 3 of the previous best are still among the top 4 in the new ranking (`type4`, `type2` and `type0`), together to the assignment `type5`. It is worth noting though that although `type0` is not the best among the top 4, neither among the top 2, it is instead the one that appear in the first position more times than the others.

Finally, keeping in mind the goals of this research work, and following this conducted analysis, it is possible to conclude that, for the elitism variant of the developed immune algorithm, 2 of the previous top 4 (`type3` and `type4`) still appear among the top 4 in the new ranking, and, furthermore, the `type4` is again among the best 2, placing in the first two positions 2 times over 6. For the no elitism variant, instead, three of the previous top 4 (`type4`, `type2` and `type0`) continue to appear in the new top 3 (of course in different order); but if we inspect only the top, we may state that the best previous (`type0`) continues to be it also in this new case study. What emerges clearly, and in common between the two variants of the immune algorithm, are the bad performances produced by the age assignment `type10`, as highlighted also in the previous efficiency ranking.

5 Conclusions

Starting from the well-know assertion that a right balancing between exploration and exploitation is a successful key for any evolutionary algorithm, in this research work we present an experimental study focused on understanding the right maturation of each solution for having a careful search process, and good information learning. Since this study follows a previous one, our main goal is investigate if the best age assignments previously obtained are still validate when is tackled a new and different problem.

An Immune Algorithm (IA) was developed for tackling and solving the NK-model, which is a mathematical model able to capture the fitness interactions producing a tunable rugged fitness landscape. Using this model we are keeping the study on discrete search space (binary strings), as done in the previous study, but with more roughness of the fitness landscape. In order to lead a proper study the tunable parameter (K), with whom is altered the roughness of the search landscape, was set to 50% of the length of the given bit string (N), producing in this way a hard landscape with many local optima.

Eleven (11) age assignment test cases have been studied and analysed with the aim to understand which one shows the best performances; produce an efficiency ranking; but, primarily, investigate if one or more best age types in the previous ranking are still among the top ranks in the new one. An overall of 924 experiments have been performed, and from their analysis we may assert that: (i) for the elitism version, two of the best previous ones (type3 and type4) still appear among the top 3, with, moreover type4 ranked in $2nd$ position 2 times over 6 in the new efficiency ranking; (ii) always for elitism version, looking only the top, any of the best previous ones reach never the $1st$ position, and this is likely due to the lower diversity they produce with respect the other age assignment types; (iii) for the no elitism variant, instead, three of the previous top 4 (type4, type2 and type0) still appear among the best 3 in the new efficiency ranking; (iv) for this last variant, the best of the previous ranking (type0), although it is never the best among the top 3, it is the one that however appears in $1st$ position more than the other age assignment types studied; and, finally, (v) type10 continues to be the worst option as already proved in the previous study.

References

1. P. Conca, G. Stracquadanio, O. Greco, V. Cutello, M. Pavone, G. Nicosia: "*Packing equal disks in a unit square: an immunological optimization approach*", 1st International Workshop on Artificial Immune Systems (AIS), IEEE Press, pp. 1–5, 2015.
2. V. Cutello, A. G. De Michele, M. Pavone: "*Escaping Local Optima via Parallelization and Migration*", VI International Workshop on Nature Inspired Cooperative Strategies for Optimization (NICSO), Studies in Computational Intelligence, vol. 512, pp. 141–152, 2013.
3. V. Cutello, G. Morelli, G Nicosia, M. Pavone, G. Scollo: "*On discrete models and immunological algorithms for protein structure prediction*", Natural Computing, vol. 10, no. 1, pp. 91-102, 2011.
4. V. Cutello, G. Narzisi, G. Nicosia, M. Pavone: "*Clonal Selection Algorithms: A Comparative Case Study using Effective Mutation Potentials*", 4th International Conference on Artificial Immune Systems (ICARIS), LNCS 3627, pp. 13–28, 2005.
5. A. Di Stefano, A. Vitale, V. Cutello, M. Pavone: "*How long should offspring lifespan be in order to obtain a proper exploration?*", 2016 IEEE Symposium Series on Computational Intelligence (SSCI), INSPEC number 16670548 2016, pp. 1–8, 2016.
6. S. Fouladvand, A. Osareh, B. Shadgar, M. Pavone, S. Sharafi: "*DENSA: An effective negative selection algorithm with flexible boundaries for selfspace and dynamic number of detectors*", Engineering Applications of Artificial Intelligence, vol. 62, pp. 359–372, 2017.
7. T. Jansen, C. Zarges: "*On Benefits and Drawbacks og Aging Strategies for randomized Search Heuristics*", Theoretical Computer Science, vol. 412, pp. 543–559, 2011.
8. S. A. Kauffman, S. Levin: "*Towards a Genarl Theory of Adaptive Walks on Rugged Landscapes*", Journal of Theoretical Biology, vol. 128, no. 1, pp. 11–45, 1987.
9. S. A. Kauffman, E. D. Weinberger: "*The NK Model of Rugged Fitness Landscapes And Its Application to Maturation of the Immune Response*", Journal of Theoretical Biology, vol. 141, no. 2, pp. 211–245, 1989.
10. D. A. Levinthal: "*Adaptation on Rugged Landscapes*", Management Science, vol. 43, no. 7, pp. 934–950, 1997.
11. M. Pavone, G. Narzisi, G. Nicosia: "*Clonal Selection - An Immunological Algorithm for Global Optimization over Continuous Spaces*", Journal of Global Optimization, vol. 53, no. 4, pp. 769–808, 2012.
12. J. D. Schaffer, L. J. Eshelman: "*On crossover as an evolutionary viable strategy*", 4th International Conference on Genetic Algorithms, pp. 61–68, 1991.
13. E. Weinberger: "*NP-Completeness of Kauffman's NK model, a Tuneably Rugged Fitness Landscape*", Santa Fe Institute Working Paper, 96-02-003, 1996.