An Immune Algorithm with Hyper–Macromutations for the 2D Hydrophilic-Hydrophobic Model

Vincenzo Cutello, Giuseppe Nicosia, Mario Pavone
Department of Mathematics and Computer Science
University of Catania
V.le A. Doria 6, 95125 Catania Italy
Email: {cutello, nicosia, mpavone}@dmi.unict.it

***Abstract*— This paper presents an Immune Algorithm (IA) based on Clonal Selection Principle using a new mutation operator, the hypermacromutation, and an aging process to tackle the protein structure prediction problem (PSP) in the 2D Hydrophilic-Hydrophobic (HP) model. The IA presented has only three parameters. To correctly set these parameters we compute the parameter surfaces, the 3D plots of IA success rate in function of the cloning paramater and the maximum age allowed to each B cell. The parameter surfaces show that hypermacromutation and aging operators are key features for generating diversity and searching more properly the funnel landscape of the PSP problem. Experiments show that the Immune Algorithm we propose is very competitive with the state-of-art algorithms for the PSP.**

I. Introduction

Artificial Immune Systems (AIS) represent a new field of Evolutionary Computing that attempts to use theories, principles, and concepts of modern immunology to design Immune System based applications in science and engineering [1], [2].

AIS are adaptive systems inspired by evolutionary mechanisms similar to biological evolution. Thus one wants, first, to understand the dynamics of Immune System (IS) when facing antigenic attacks, and then, to develop new algorithms that mimic the biological IS under study so to catch its ability to solve computational problems otherwise difficult to solve by conventional specialized algorithms.

In nature, the main role of the immune system is to protect the host organism against attacks from antigens (i.e. viruses, bacterias, foreign entities) and to eliminate those cells that have been "infected". To perform these functions, the IS must be in a position to distinguishing between the cells of our organism, *self*, and those that do not belong to it, *nonself*.

The IS provides an excellent example of bottom up intelligent strategy, in which adaptation operates at the local level of cells and molecules, and useful behavior emerges at the global level, the immune humoral response. AIS are proving to be a very general and applicable form of biologically inspired computing. A great deal of work has gone into developing algorithms that extrapolate basic immune processes such as clonal selection, negative selection and immune networks [2].

To date AIS have been applied to areas such as machine learning, network intrusion detection, scheduling, combinatorial optimization problems, fault diagnosis, computer security, virus detection, immunized fault tolerance, design optimization and many other areas [2]. The field of AIS appears to be a powerful computing paradigm as well as a prominent apparatus for improving understanding of biological data and systems.

In this paper we describe an Immune Algorithm based on clonal selection principle [3], [4] using a particular mutation operator, the hypermacromutation operator, to face the *Protein Structure Prediction problem* (PSP) in the 2D HP model. Given the primary sequence of a protein, a sequence of amino-acid, the PSP problem asks to find its 3D naive conformation with minimum energy. Since the protein structure determines its biological function, it is very important to be able to predict the final spatial conformation of the proteins. This work goes in the direction of using biological inspired algorithms to tackle biological problems.

II. The 2D HP Model for the PSP Problem

Essentially, there are five approaches to model the PSP problem: Molecular Dynamics, Monte Carlo, Statistical Model, Probabilistic Road map-Based, Lattice model. The first two methods have been used to analyze the number and the characteristic of folding pathways; the second two are useful tools to study the folding landscape, while the last one has a fundamental theoretical relevance but cannot be applied directly to real proteins. One approach to model the protein folding problem is the well-known Dill's lattice model, the HP model [5]. It models proteins as two-dimensional (2D) *self-avoiding walk chains* of ℓ monomers on the square lattice, that is, two resides cannot occupy the same node of the lattice. There are only two monomer types: the H and the P monomers, respectively for hydrophobic and polar monomers. In this model, each H–H topological contact, that is, each lattice nearest-neighbor H–H contact interaction, has energy value $\epsilon \leq 0$, while all other contact interaction types (H–P, P–P) have zero energy. In general, in the HP model the residues interactions can be defined as follows: $e_{HH} = -\mid \epsilon \mid$ and $e_{HP} = e_{PH} = e_{PP} = \delta$. When $\epsilon = 1$ and $\delta = 0$ we have the typical interaction energy matrix for the standard HP model [5]; while for $\epsilon = 2$ and $\delta = 1$ we have the interaction energy matrix for the shifted HP model [6]. The native conformation is the one that maximizes the number of contacts H–H, i.e. the one that minimizes the free energy function. The Dill's

TABLE I
TORTILLA 2D HP BENCHMARKS.

No.	*Length*	*Protein Sequence*	E^*
1	20	$hphp_2h_2php_2hph_2p_2hph$	-9
2	24	$h_2p_2(hp_2)_6h_2$	-9
3	25	$p_2hp_2(h_2p_4)_3h_2$	-8
4	36	$p_3h_2p_2h_2p_5h_7p_2h_2p_4h_2p_2hp_2$	-14
5	48	$p_2h(p_2h_2)_2p_5h_{10}p_6(h_2p_2)_2hp_2h_5$	-23
6	50	$h_2(ph)_3ph_4p(hp_3)_2hp_4(hp_3)_2hph_4(ph)_3ph_2$	-21
7	60	$p_2h_3ph_8p_3h_{10}php_3h_{12}p_4h_6ph_2php$	-36
8	64	$h_{12}(ph)_2(p_2h_2)_2p_2h(p_2h_2)_2p_2h(p_2h_2)_2p_2(hp)_2h_{12}$	-42
9	20	$h_3p_2(hp)_2hp_2(hp)_2hp_2h$	-10

model has a strong experimental justification. During the folding process of real protein, the hydrophobic residues tend to interact with each other, forming the *hydrophobic kernel* of the native structure, while the hydrophilic resides are on the external surface of protein, forming the interface with the watery environment. The HP model has the great practical advantage of formalizing the protein primary structure as a binary sequence s of H's and P's (i.e., $s \in \{H,P\}^\ell$) and the conformational space as a square lattice. It is worth to say that it is possible to extend the model on triangular 2D lattices and on 3D lattices. Finding the global minimum of the free energy function for the protein folding problem in the 2D HP model is NP-hard [7].

In this work, we present an IA based on clonal selection theory using the HP model as hard benchmarks (see table I). We used the first nine instances of the *Tortilla 2D HP Benchmarks* to test the searching capability of the designed IA. In table I, E^* is the optimal or best known energy values, H_i, P_i e $(\ldots)_i$ indicate i repetition of the relative symbol or subsequence. These instances can be found at [1]

A. The Conformational Space into Lattice

To embed a hydrophobic pattern, $s \in \{H,P\}^\ell$, into a lattice we have the following three methods [8]:

1) *Cartesian Coordinate*: the position of residues is specified independently from other residues;
2) *Internal Coordinate*: the position of each residue depends upon its predecessor residues in the sequence. There are two types of internal coordinate: *absolute directions* where the residues direction are relative to the axes defined by the lattice, and *relative directions* where the residues direction are relative to direction of the previous move.
3) *Distance Matrix*: the location of the a given residue is computed by means of its distance matrix .

Krasnogor *et al.* [8] performed an exhaustive comparative study using the evolutionary algorithms (EA's) with relative and absolute directions. The experimental results shown as the relative directions outperform almost always the absolute directions over square and cubic lattice, while the absolute directions have better performances when facing triangular lattices. Hence, experimental evidence suggested induct us to use the internal coordinates with relative directions. However, in general it is difficult to assess the effectiveness of direction encoding on a EA's performance.

[1] *http://www.cs.sandia.gov/tech_report/compbio/tortilla-hp-benchmarks.html*

III. THE IMMUNE ALGORITHM

A. The Inspiration: Clonal Selection Principle

The theory of clonal selection [9], suggests that among all possible cells, B and T lymphocytes, with different receptors circulating in the host organism, only those who are actually able to recognize the antigen will start to proliferate by duplication (cloning). Hence, when a B cell is activated by binding an antigen, it produces many clones, in a process called clonal expansion. The resulting cells can undergo somatic hypermutation, creating offspring B cells with mutated receptors. Antigens compete for recognition with these new B cells, their parents and with other clones. The higher the affinity of a B cell to available antigens, the more likely it will clone. This results in a Darwinian process of variation and selection, called affinity maturation. The size increase of those populations and the production of cells with longer expected lifetime, assures the organism a higher specific responsiveness to that antigenic attack, establishing a defense over time (immune memory). In particular, on recognition, memory lymphocytes are produced. Plasma B cells, deriving from stimulated B lymphocytes, are in charge of the production of antibodies targeting the antigen.

B. The Algorithm

The designed IA uses only two entities: antigens (Ag) and B cells. The Ag models the hydrophobic-pattern of the given protein, that is a sequence $s \in \{H,P\}^\ell$, where ℓ is the protein length (the number of amino-acids). The B cell population ($P^{(t)}$) represents a set of candidate solution at each time step (or generation) t. The B cell (or a B cell receptor) is a sequence of *relative directions* [8] $r \in \{F,L,R\}^{\ell-1}$; where each r_i, is a relative direction with respect to the previous direction (r_{i-1}), with $i = 2,\ldots,\ell-1$, (i.e., there are $\ell-2$ relative directions) and r_1 the non-relative direction. Hence, we obtain an overall sequence r of length $\ell-1$. The sequence r detects a 2D conformation suitable to compute the energy value of the hydrophobic-pattern of the given protein.

At each time step t, we have a population $P^{(t)}$ of size d. The initial population, time $t=0$, is generated randomly in such a way that each B cell of $P^{(0)}$, and in general all the population used by our IA, represents *self-avoiding* conformations. The function *Evaluate(P)* computes the affinity (fitness) function value of each B cell $\vec{x} \in P$. Hence $f(\vec{x}) = e$ is the energy of conformation coded in the B cell receptor $\vec{x}$, with $-e$ the number of topological contacts $H-H$ in the 2D lattice. Our IA , like all immune algorithms based on the clonal selection principle, is characterized by cloning expansion, the cloning of B cells with higher antigenic affinity. The implemented

IA uses three immune operators, cloning, hypermacromutation and aging.

The cloning operator, simply, clones each B cell dup times producing an intermediate population P^{clo} of size $d \times dup$. Throughout this paper, we will call it *static cloning operator*, as opposed to a *proportional cloning operator* [3], that clones B cells proportionally to their antigenic affinities. Preliminary experimental results using such an operator (not shown in this paper), showed us frequent premature convergence during the population evolution. In fact, proportional cloning gives more time steps to B cells with high affinity values, and the process can more likely be trapped into local minima of the landscape. However, the proportional cloning operator could be used to explore more deeply the attractor basins of conformational space like an implicit local search procedure. The hypermacromutation operator acts on the the B cell receptor of P^{clo}. The number of mutations M is determined by a *mutation potential*. It is possible to define several mutation potentials. In this research paper we tested our IA using only the Hypermacromutation operator, which does not use mutation potentials depending upon costant parameters and it is governed by a simple random process that try to mutate each B cell receptor M times, maintaining the self-avoiding property. In particular, the hypermacromutation operator mutates at most $M_m(\vec{x}) = j - i + 1$ directions, in the range $[i, j]$, with i and j being two random integers such that $(i + 1) \leq j \leq \ell$.

The number of mutations is independent from the fitness function f and any other parameter.

If during this process a constructive mutation occurs, the mutation procedure will move on to the next B cell. We adopted this scheme to slow down (premature) convergence, exploring more accurately the search space. Given a feasible conformation, the hypermacromutation enjoying the self-avoiding property, randomly selects a single direction D in the sequence, and a new direction $D' \neq D \in \{F, L, R\}$. If the new conformation is again self-avoiding then the operator accepts it, otherwise the procedure repeats the process with a new direction $D'' \neq D, D'$. The aging operator eliminates old B cells in the populations $P^{(t)}$, $P^{(hyp)}$ and/or $P^{(macro)}$, to avoid the premature convergence. To increase the population diversity, new B cells are added by the *Elitist_Merge function*. The parameter τ_B sets the maximum number of generations allowed to B cells to remain in the population. When a B cell is $\tau_B + 1$ old it is erased by the current population, no matter what its fitness value is. We call this strategy, *static pure aging*. During the cloning expansion, a cloned B cell takes the age of its parent. After the hypermacromutation phase, a cloned B cell which successfully mutates, that is the new conformation will have a lower energy value, will be considered to have age equal to 0. This scheme intends to give an equal opportunity to each "new conformation" to effectively explore the funnel landscape. We note that for τ_B greater than the maximum number of allowed generations, the IA works essentially without an aging operator. In such a limit case, the algorithm uses a strong elitist selection strategy.

The best B cells which"survived" the aging operator, are selected from the populations $P^{(t)}$, $P^{(hyp)}$ and/or $P^{(macro)}$, in such a way the each B cell receptor is unique, i.e. each conformation is different from all other conformations. In this way, we obtain the new population $P^{(t+1)}$, of d B cells, for the next generation $t + 1$. If only $d' < d$ B cells survived , the *Elitist_Merge* function creates $d - d'$ new B cells (*Birth phase*). The boolean function *Termination_Condition()* returns true if a solution is found, or a maximum number of fitness function evaluations (T_{max}) is reached.

Table II shows the pseudo-code of the proposed Immune Algorithm.

TABLE II
PSEUDO–CODE OF IMMUNE ALGORITHM

```
IMMUNE ALGORITHM(ℓ, d, dup, τ_B, A)
N_c := d × dup;
t := 0;
P^(t) := Initial_Pop(d);
Evaluate(P^(t));
while ( ¬ Termination_Condition() ) do
  P^(clo) := Cloning (P^(t), N_c);
  P^(macro) := Hyper–Macromutation (P^clo, ℓ);
  Evaluate (P^(macro));
  if (A) then
    Pure_Aging(P^(t), P^(macro), τ_B);
  else
    Elitist_Aging(P^(t), P^(macro), τ_B);
  P^(t+1) := Elitist_Merge (P^(t), P^(macro));
  t := t + 1;
end_while
Note: the boolean variable A controls,
the Pure_Aging and Elitist_Aging procedures.
```

C. The IA Dynamics

Now, we show the characteristic dynamics of our Immune Algorithm. We set the population size, d, and the maximum number of fitness function evaluations allowed, T_{max}, to minimal values, respectively $d = 10$ and $T_{max} = 10^5$, with $dup = 1$ and $\tau_B = 15$. All the values reported in this section are averaged on 100 independent runs. In figure 1 we show the average fitness of population P^{macro}, the average fitness of population $P^{(t)}$ and the best fitness value when the IA faces the PSP instance, *Seq2* ($\ell = 24$ and minimum energy value known $E^* = -9$). How we can see, the three curves decrease almost monotonically approximately in the first 35 generations, in the remaining time steps all three curves reach a steady state dynamics. The small oscillations are due to the random nature of the overall process governing the IA. The higher the average fitness of the hypermacromutated clones is, the higher is the diversity in the current population [10].

To analyze the learning process, we use an entropy function, the *Information Gain*. It measures the quantity of information the system discovers during the learning phase [10]. To this end, we define the B cells distribution function $f_m^{(t)}$ as the ratio between the number, B_m^t, of B cells at time t with fitness

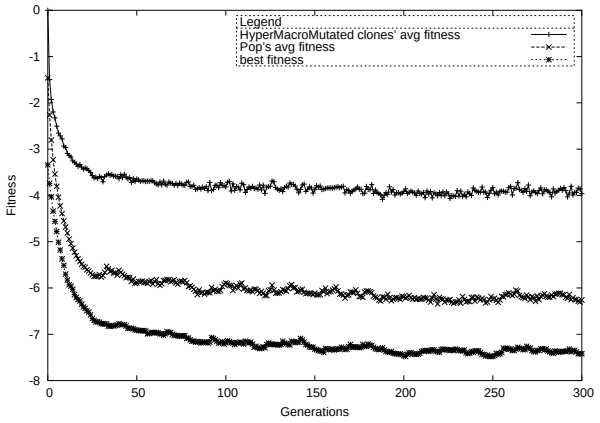

Fig. 1. Average fitness function values of P^{hyp}, $P^{(t)}$ and the best B cell receptor on protein sequence $Seq2$, with parameter values $d = 10, dup = 1$ and $\tau_B = 15$.

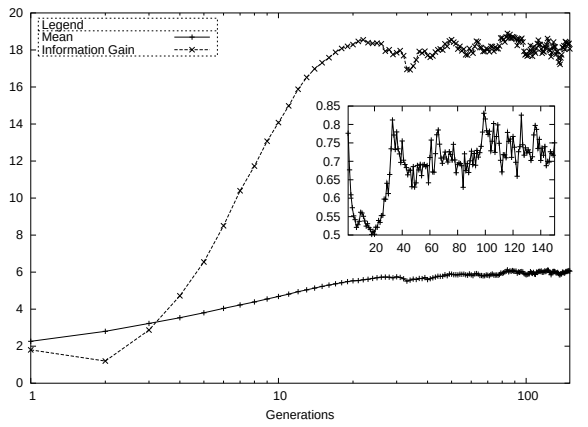

Fig. 2. Information gain, mean versus generations of IA on protein sequence $Seq2$, with parameter values $d = 10, dup = 1$ and $\tau_B = 15$. In the inset plot we report the standard deviation.

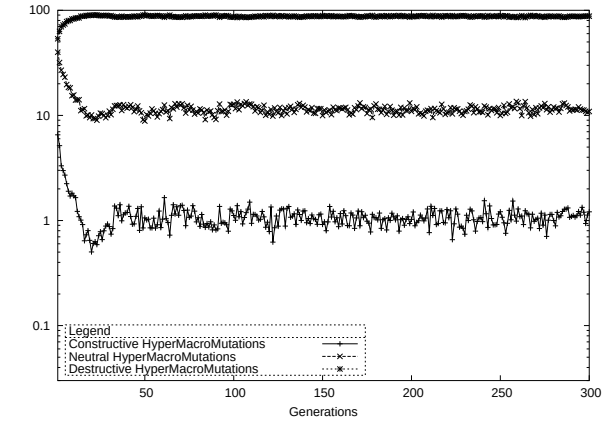

Fig. 3. Constructive, neutral and destructive mutations of IA on protein sequence $Seq2$, with parameter values $d = 10, dup = 1$ and $\tau_B = 15$.

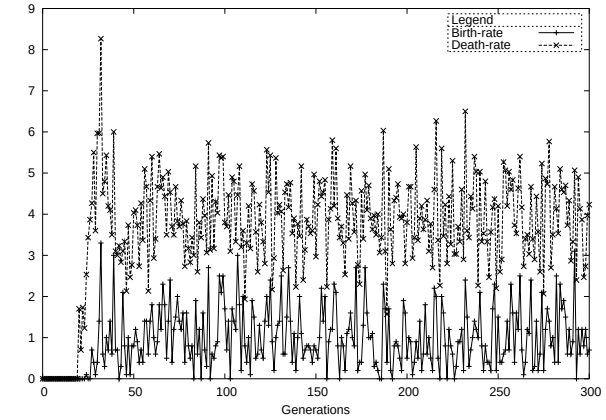

Fig. 4. Birth and age rate versus generations of IA on protein sequence $Seq2$, with parameter values $d = 10, dup = 1$ and $\tau_B = 15$.

function value m, and the total number of B cells:

$$f_m^{(t)} = \frac{B_m^t}{\sum_{m=0}^{h} B_m^t} = \frac{B_m^t}{d}. \quad (1)$$

It follows that the information gain can be defined as:

$$K(t, t_0) = \sum_m f_m^{(t)} \log(f_m^{(t)} / f_m^{(t_0)}). \quad (2)$$

The gain is the amount of information the system has already learned from the given problem instance with respect to the randomly generated initial population $P^{(t=0)}$ (the initial distribution). Once the learning process starts, the information gain increases monotonically until it reaches a final steady state (see fig. 2). This is consistent with the idea of a *maximum information-gain principle* of the form $\frac{dK}{dt} \geq 0$. In x-axis log plot 2, it is evident how the Information Gain is a more informative measure than the average. The standard deviation, the uncertainty over of the population to a given generation (see the inset plot in fig. 2), decreases quickly in the first 20 generations. In fact, the IA converges to the global minimum in this temporal window. After this "threshold" the standard deviation sudden increases producing strong oscillations, that is, strong uncertainty about the current populations for $t > 20$.

The mean value is practically constant during all time steps. For example, in the first time step the IA gains more information than in the second one, because it generates more constructive mutations (see fig. 3), that is, the population at generation $t = 1$ extracts more informative building blocks than the population at the second generation.

In figure 3 we show the constructive, neutral and destructive mutations generated by our IA. A constructive mutation is a mutation that improves the fitness function value of a given B cell receptor, whereas, a destructive mutation will make it worse. A neutral mutation modifies the B cell receptor only producing a new conformation with the same fitness value of the previous B cell receptor. Excluding the first 15 generations, the three values differ of one order of magnitude.

Finally, we report the number of birth and death versus generations. Obviously, in the first $\tau_B = 15$ generations there are no deaths, while the birth process produces new B cells after 25 time steps. This is due to the procedure that eliminates the redundancy from the actual population. This two complimentary processes are the main forces that

generate diversity in the population during the searching in the landscape.

IV. Simulation Results

To show the IA dynamics we performed a set of experiments with the duplication parameter $dup \in \{1, \ldots, 10\}$ and the parameter $\tau_B \in \{1, \ldots, 15, 20, 25, 50, 100, 200, \infty\}$. As a function of dup and τ_B we show the success rate (SR). The obtained 3D plots are the characteristic *parameter surfaces* of the hypermacromutation operator. All the parameter surfaces reported in this section are averaged on 100 independent runs.

The following experiments have been performed to test the robustness of the designed IA. The first experiment (see fig. 5) shows the behavior of the IA with and without elitism. We implemented a simple elitist procedure: when a new population is generated, we do not allow the elimination of B cells with the best fitness function. While in the pure aging, the best B cells can be eliminate as well. The experiment was performed with population size $d = 10$ and maximum number of fitness function evaluations $T_{max} = 10^5$. The plot shows clearly how the IA without the elitist procedure outperforms the IA with elitism. Higher T_{max} values increase the searching ability of the IA with elitism: Table III reports the SR (in boldface) and AES (in italic) values for all the protein instance of the Tortilla Benchmark (all the values are averaged on 30 independent runs).

TABLE III

Comparison of IA performances without and with Elitism.

ℓ	E^*	T_{max}	*Without Elitism*	*With Elitism*
20	-9	2.5×10^5	(**100**, *25418.83*)	(**96.67**, *20508.86*)
24	-9	5×10^5	(**100**, *39410.9*)	(**100**, *37659.73*)
25	-8	5×10^5	(**100**, *79592.1*)	(**96.67**, *58905.31*)
36	-14	10^6	(**16.67**, *466176.4*)	(**36.67**, *310291.36*)
48	-23	10^6	(**6.67**, *483651.5*)	(**3.33**, *277454*)
50	-21	10^6	(**16.67**, *469941.2*)	(**53.33**, *459868*)
60	-36	2×10^6	(*b.f. -34*)	$b.f - 35$
64	-42	2×10^6	(*b.f. -36*)	$b.f. - 39$
20	-10	5×10^5	(**100**, *27852.13*)	(**96.67**, *27719.14*)

To verify the usefulness of the aging operator on the Immune Algorithm performance we turned it off this operator. The plot of figure 6 shows the poor performance of the IA without aging operator..

Figure 7 compares the IA performance with population size $d = 10$ and $d = 100$. Increasing the population dimension increases the SR values.

Figure 8 shows the IA dynamics without the elitist procedure and with population size $d = 100$, $T_{max} = 2.5 \times 10^5$ for the Protein instance 3. The parameter surface illustrates a better IA performance in term of SR when we increase the population dimension and T_{max}, making the IA less sensitive to parameter values settings.

Finally, in table IV we report the comparisons with the state-of-art algorithms for the 2D HP PSP problem. The shown

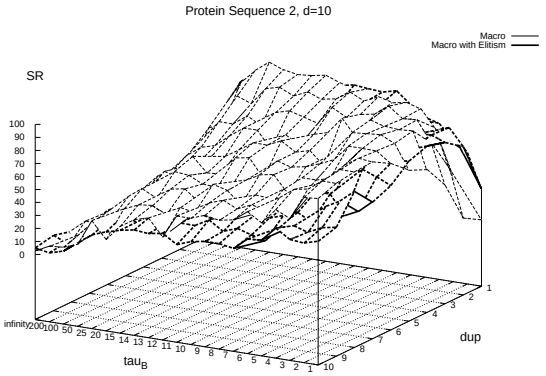

Fig. 5. SR as a function of the values dup and τ_B for the IA with and without Elitism on Protein sequence 2. Population size $d = 10$ and $T_{max} = 10^5$.

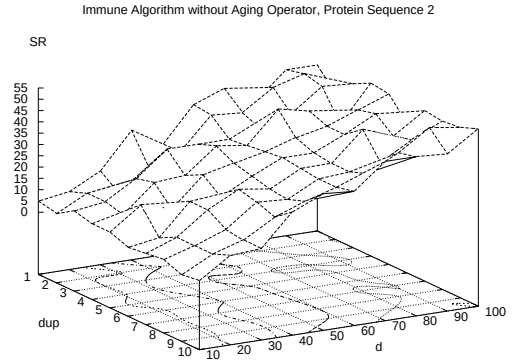

Fig. 6. SR as a function of the values dup and τ_B for the IA without the Aging operator on Protein sequence 2. Population size $d = 10$ and $T_{max} = 10^5$.

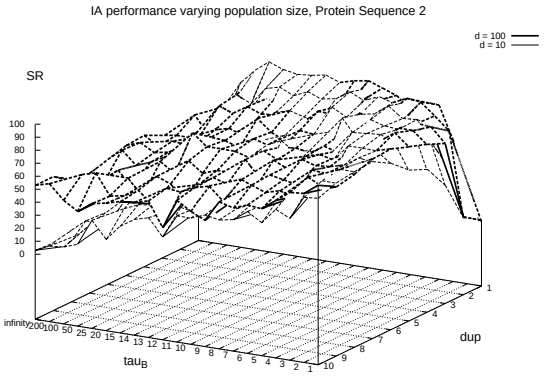

Fig. 7. SR as a function of the values dup and τ_B for the IA with population size $d = 10$ and $d = 100$ on Protein sequence 2.

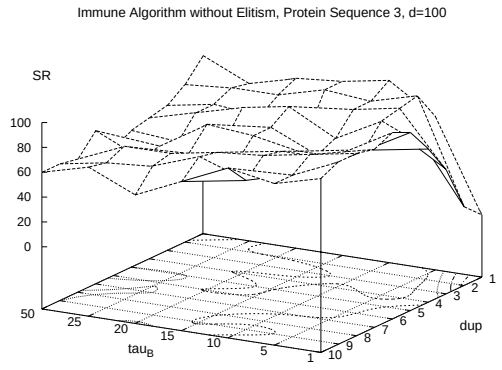

Fig. 8. SR as a function of the values dup and τ_B for the IA with population size $d = 100$ and $T_{max} = 5 \times 10^5$ on Protein sequence 3.

TABLE IV

COMPARISON OF IA AND THE STATE-OF-ART ALGORITHMS.

Sequence		1	2	3	4	5	6	7	8	9
E^*		-9	-9	-8	-14	-23	-21	-36	-42	-10
new ACO	[11]	-9	-9	-8	-14	-23	-21	-36	-42	
CG	[12]	-9	-9	-8	-14	-23	-21	-35	-42	
Tabu Search	[13]	-9	-9	-8	-14	-23	-21		-42	
IA		**-9**	**-9**	**-8**	**-14**	**-23**	**-21**	**-35**	**-39**	**-10**
EMC	[14]	-9	-9	-8	-14	-23	-21	-35	-39	
PERM	[15]	-9	-9	-8	-14	-23	-21	-36	-38	
MMA	[16]	-9		-8	-14	-22	-21		-39	-10
CI	[17]	-9	-9	-8	-14	-23	-21	-35		
GA	[18]	-9	-9	-8	-14	-22	-21	-34	-37	
Pioneer Search	[19]	-9			-14	-23			-37	
ACO + LS	[20]	-9	-9	-8	-14	-23	-21	-34	-32	-10
SC	[19]	-9			-14	-22			-34	
Metropolis MC	[18]	-9	-9	-8	-13	-20	-21	-33	-35	
only LS	[20]	-9	-9	-8	-14	-21	-20	-33	-33	-10
Multiple MC	[18]	-9	-9	-7	-13	-19	-20	-32	-32	
SGA	[19]	-9			-14	-20			-32	

results suggest that the proposed IA using the Hypermacromutation procedure as variation operator is comparable to and, in many protein instances, outperforms the best algorithms.

V. CONCLUSION

The present paper shows as that an Immune Algorithm based on clonal selection principle and a new variation operator, the hypermacromutation, is suitable to face the 2D HP model for the protein structure prediction problem.

The designed IA uses four mechanisms to obtain diversity in the population at each generation: Static cloning operator, Static pure aging operator, Birth phase, population without redundancy.

For the hypermacromutation operator we determined the characteristic parameter surface and the best delimited region on the parameter surfaces that maximize the SR value. This region has been used to predict the best parameter value setting.

When overlapping the parameter surfaces with and without elitism, we discover that the IA performs effectively for low value of T_{max}. Increasing T_{max} the IA with elitism outperforms in term of SR the IA without an elitist strategy.

The experimental results show that the aging operator is a key feature of our IA. We expect that this new kind of operator could be a useful engine to generate diversity and search more properly the landscape of a given computational problem. The IA along with the hypermacromutation operator is comparable to and, in many protein instances, outperforms the state-of-art algorithms for the PSP.

REFERENCES

[1] D. Dasgupta 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] L. N. de Castro and F. J. Von Zuben, "Learning and optimization using the clonal selection principle," *IEEE Trans.Evol. Comput.*, vol. 6(3), pp. 239-251, 2002.

[4] V. Cutello, and G. Nicosia, "The Clonal Selection Principle for in silico and in vitro Computing," in *Recent Developments in Biologically Inspired Computing*, L. N. de Castro and F. J. Von Zuben, Eds., (to appear) 2004.

[5] K. A. Dill, "Theory for the folding and stability of globular proteins," *Biochemistry*, vol. 24(6), pp. 1501-9, 1985.

[6] J. D. Hirst, "The evolutionary landscape of functional model proteins," *Protein Engineering*, vol. 12, pp.721-726, 1999.

[7] P. Crescenzi, D. Goldman, C. Papadimitriou, A. Piccolboni, and M. Yannakakis, "On the complexity of protein folding," *Journal of Computational Biology*,vol. 5(3), pp. 423-466, 1998.

[8] N. Krasnogor, W. E. Hart, J. Smith, and D. A. Pelta, "Protein Structure Prediction with Evolutionary Algorithms," in *Proceedings of GECCO '99*, San Francisco, CA: Morgan Kaufman, 1999, vol. 2, pp. 1596-1601.

[9] F. M. Burnet, *The Clonal Selection Theory of Acquired Immunity*, Cambridge, UK: Cambridge University Press, 1959.

[10] V. Cutello, G. Nicosia, and M. Pavone, "A Hybrid Immune Algorithm with Information Gain for the Graph Coloring Problem," in *Proceedings of GECCO '03*, E. Cantu-Paz *et al.* Eds., Springer-Verlag, 2003, pp.171-182.

[11] A. Shmygelska, and H. H. Hoos, "An Improved Ant Colony Optimization Algorithm for the 2D HP Protein Folding Problem," in *Proceedings of the 16th Canadian Conference on Artificial Intelligence (AI)*, Springer-Verlag, 2003, pp.400-417.

[12] T. C. Beutler, and K. A. Dill, "A Fast Conformational Search Strategy for finding low Energy Structures of model proteins," *Protein Science*, vol. 5, pp. 2037-2043, 1996.

[13] M. Milostan, P. Lukasiak, K. A. Dill, and J. Blażewicz, "A Tabu Search Strategy for Finding Low Energy Structures of Proteins in HP–Model," in *Recomb '03*, Poster No.5-108, 2003.

[14] F. Liang, and W. H. Wong, "Evolutionary Monte Carlo for protein folding simulations," *J. Chem. Phys.*, vol. 115(7), pp.3374-3380, 2001.

[15] H. Hsu, V. Mehra, W. Nadler, and P. Grassberger, "Growth algorithms for lattice heteropolymers at low temperatures," *J. Chem. Phys.*, vol. 118(1), pp. 444-451, 2003.

[16] N. Krasnogor, B. P. Blackburne, E. K. Burke, and J. D. Hirst, "Multimeme Algorithms for Protein Structure Prediction," in *Proceedings of PPSN VII*, Springer-Verlag, Lecture Notes in Computer Science, vol. 2439, pp.769-778, 2002.

[17] L. Toma, and S. Toma, "Contact Interactions Methods: A new Algorithm for Protein Folding Simulations," *Protein Science*, vol. 5, pp. 147-153, 1996.

[18] R. Unger, and J. Moult, "Genetic Algorithms for Protein Folding Simulations," *J. Molecular Biology*, vol. 231(1), pp. 75-81, 1993.

[19] R. Konig, and T. Dandekar, "Improving Genetic Algorithms for Protein Folding Simulations by Systematic Crossover," *Biosystems*, vol. 50(1), pp. 17-25, 1999.

[20] A. Shmygelska, R. Anguirre-Hernandez, H. H. Hoos, "An Ant Colony Optimization Algorithm for the 2D HP Protein Folding Problem," in *Proceedings of ANTS '02*, Springer-Verlag, Lecture Notes in Computer Science, vol. 2453, pp. 40-52, 2002.