Nat Comput (2011) 10:91–102
DOI 10.1007/s11047-010-9196-y

On discrete models and immunological algorithms for protein structure prediction

Vincenzo Cutello · Giuseppe Morelli · Giuseppe Nicosia · Mario Pavone · Giuseppe Scollo

Published online: 22 April 2010

Abstract Discrete models for protein structure prediction embed the protein amino acid sequence into a discrete spatial structure, usually a lattice, where an optimal tertiary structure is predicted on the basis of simple assumptions relating to the hydrophobic–hydrophilic character of amino acids in the sequence and to relevant interactions for free energy minimization. While the prediction problem is known to be NP complete even in the simple setting of Dill's model with a 2D-lattice, a variety of bio-inspired algorithms for this problem have been proposed in the literature. Immunological algorithms are inspired by the kind of optimization that immune systems perform when identifying and promoting the replication of the most effective antibodies against given antigens. A quick, state-of-the-art survey of discrete models and immunological algorithms for protein structure prediction is presented in this paper, and the main design and performance features of an immunological algorithm for this problem are illustrated in a tutorial fashion.

Key words Artificial immune system · Clonal selection algorithm · Dill model · Evolutionary algorithm · Functional model protein · HP model · Immunological algorithm · Protein folding · Protein structure prediction

V. Cutello · G. Morelli · G. Nicosia · M. Pavone · G. Scollo (✉)
Department of Mathematics and Computer Science, University of Catania, Catania, Italy
e-mail: scollo@dmi.unict.it

V. Cutello
e-mail: cutello@dmi.unict.it

G. Morelli
e-mail: morelli@dmi.unict.it

G. Nicosia
e-mail: nicosia@dmi.unict.it

M. Pavone
e-mail: mpavone@dmi.unict.it

1 Introduction

Artificial immune systems (AIS) represent a new field of biologically inspired computing that attempts to use theories, principles, and concepts of modern immunology to design immune system (IS) based applications in science and engineering (Dasgupta 1999; De Castro and Timmis 2002; Nicosia 2004). Thus one wants, first, to understand the dynamics of an IS when facing antigenic attacks, and then, to develop new algorithms that mimic the biological IS under study, so as 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, as generated by viruses, bacterias, foreign entities, etc., and to eliminate those cells that have been "infected". To perform these functions, the IS must be in a position to distinguish between the cells of the host organism, the *self*, and those that do not belong to it, the *nonself*. The IS provides an excellent example of bottom-up intelligent strategy (Cutello et al. 2004; Nicosia 2004), where adaptation operates at the local level of cells and molecules, while useful behaviour emerges at the global level, the immune humoral response. AIS are proving to be a very general and applicable form of bio-inspired computing. A great deal of work has gone into developing algorithms that extrapolate basic immune processes such as clonal selection, negative and positive selection, danger theory and immune networks (De Castro and Timmis 2002; Nicosia 2004).

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 (De Castro and Timmis 2002). The field of AIS appears to be a powerful computing paradigm as well as a prominent apparatus for improving the understanding of biological data and systems.

This paper aims at illustrating, in a tutorial fashion, the effectiveness of using biologically inspired algorithms to tackle biological problems. To this purpose we describe an immunological algorithm (IA) based on the clonal selection principle (De Castro and Von Zuben 2002; Cutello et al. 2004; Nicosia 2004), that is a simpler variant of the IA presented in Cutello et al. (2007b) using a particular mutation operator, the hypermacromutation operator, to face the *Protein Structure Prediction problem* (PSP) in the 2D hydrophobic–hydrophilic (HP) model. Given the primary sequence of a protein, that is a sequence of amino acids, the PSP problem asks one to find its 3D native 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 proteins.

The rest of this paper is organized as follows. In Sect. 2 discrete models for PSP are reviewed, and details are recalled about the standard 2D HP model that relate to the modeling of protein sequence and conformation, energy evaluation, and different ways of embedding the protein conformational space into lattices. Clonal selection is then introduced in Sect. 3, and its operational functioning is illustrated by an IA where it is complemented by hypermacromutation together with an aging selection process. Performance analysis of evolutionary algorithms is reviewed in Sect. 4, where a performance measure is derived from a maximum information-gain principle, and a possible connection with the log-gain regulation principle proposed in Manca (2008) is pointed out. Brief conclusions are drawn in Sect. 5.

2 Discrete models for protein structure prediction

Essentially, there are five approaches to model the PSP problem: molecular dynamics (Levitt 1983), Monte Carlo methods (Covell 1992), statistical mechanical models (Alm and Baker 1999; Muñoz and Eaton 1999), probabilistic roadmap-based (Apaydin et al. 2002; Amato et al. 2003), lattice models (Lau and Dill 1989; Dill et al. 1995). The first two methods have been used to analyze the number and the characteristics of folding pathways; the second two methods are useful tools to study the folding landscape, while the last one has a fundamental theoretical relevance but cannot be applied to real proteins directly. One approach to model the protein folding problem is the well-known Dill's lattice model, the HP model (Dill 1985).

2.1 Protein sequence and conformation in the Dill's model

Proteins are necklaces of amino acids; in the standard Dill's model each amino acid is represented as a bead, while connecting bonds are represented as lines. In this approach, the protein is composed of a specific sequence of only two types of beads, H (bead-Hydrophobic/non-polar) or P (bead-hydrophilic/Polar); that is, the twenty amino acids can be divided into two classes: H and P. This is usually called the HP model (or Dill model) (Dill 1985). We are reducing the alphabet from 20 characters to 2, where protein sequences take the form of nonempty strings over the alphabet $\{H, P\}$.

Hydrophobic amino acids tend to come together to form a compact core that excludes water. Because the environment inside cells is aqueous (primarily water), these hydrophobic amino acids tend to stay at the inside of a protein, rather than on its surface. Hydrophobicity is one of the key factors that determines how the chain of amino acids will fold up into an active protein. The lattice HP model only takes this factor into account.

The whole conformation is embedded in the two (or more) dimensional lattice. The lattice simply divides the space into amino acid-sized units. Bond angles only take a few discrete values, dictated by the lattice structure (e.g., square, triangular, cubic). A lattice site may be either empty or filled by one bead. In particular, in a 2D square lattice, the HP model represents proteins of length ℓ (i.e. the number of amino acids in the protein sequence) as two-dimensional *self-avoiding walk chains* of ℓ beads on the lattice, i.e., two beads cannot occupy the same site of the lattice, and each bead occupies only one lattice site connected to its chain neighbours.

2.2 Protein energy

For each conformation one can evaluate an energy function, which models free energies of protein folds. The simplest form of energy function counts the number of HH-contacts, since each topological HH-contact has a fixed energy value ε, while all other contact interaction types (HP, PP) have zero energy. Two amino acids create an HH-contact if they are topological neighbours and they are not connected by a bond. The goal is to find a conformation with the lowest energy, that is, with a hydrophobic core on the lattice. More generally, in the HP model the residues' interactions can be defined as follows: $e_{HH} = -|\epsilon|$ and $e_{HP} = e_{PH} = e_{PP} = \delta$. When $\varepsilon = -1$ and $\delta = 0$ we have the typical interaction energy matrix for the standard HP model (Dill 1985); while for $\varepsilon = 2$ and $\delta = 1$ we have the interaction energy matrix for the shifted HP model (Hirst 1999). The native conformation is the one that maximizes the number of HH-contacts, i.e. the one that minimizes the free energy function. The Dill's model has a strong experimental

justification. During the folding process of a real protein, the hydrophobic residues tend to interact with each other, thus forming the *hydrophobic kernel* of the native structure, while the hydrophilic residues lie on the external surface of the protein, thereby forming its 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 mentioning 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 (Crescenzi et al. 1998).

In this paper, we present an IA based on clonal selection theory using the HP model as hard benchmarks. We used the first nine instances of the *Tortilla 2D HP Benchmarks* [the first eight sequences are taken from Unger and Moult (1993), sequence 9 is taken from Toma and Toma (1996)] to test the searching capability of the designed IA. Table 1 shows the mentioned instances, where E^* is the optimal or best known energy value, while h^i, p^i and $(\ldots)^i$ indicate i repetitions of the respective symbol or subsequence. These instances can be found at http://www.cs.sandia.gov/tech_reports/compbio/tortilla-hp-benchmarks.html.

2.3 The embedding of a conformational space into lattices

To embed a hydrophobic pattern, $s \in \{H, P\}^{\ell}$, into a lattice we have the following three methods (Krasnogor et al. 1999, 2002; Nicosia 2004; Pavone 2003; Narzisi 2008):

1. *Cartesian Coordinate*: the position of residues is specified independently of 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. (1999) performed an exhaustive comparative study using the evolutionary algorithms (EA's) with relative and absolute directions. The experimental results showed

Table 1 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

that the relative directions outperform the absolute ones almost always, in the case of square and cubic lattices, while the absolute directions feature better performance with triangular lattices. Hence, experimental evidence suggested us to use internal coordinates with relative directions. However, in general it is difficult to assess the effectiveness of direction encoding on an EA's performance.

3 Clonal selection algorithms

The theory of clonal selection (Burnet 1959), suggests that, among all possible cells, B and T lymphocytes, with different receptors circulating in the host organism, only those which are actually able to recognize the antigen will start to proliferate by duplication (cloning). Hence, when a B cell is activated by binding to an antigen, it produces many clones, in a process called clonal expansion. The resulting cells may undergo somatic hypermutation, which creates offspring B cells with mutated receptors. These new B cells compete with their parents and with other clones for antigen recognition. 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 assure the organism a higher specific responsiveness to that antigenic attack, thus establishing a defense over time (immune memory). In particular, upon recognition, memory lymphocytes are produced. Plasma B cells, deriving from stimulated B lymphocytes, are in charge of the production of antibodies targeting the antigen.

The designed IA, see Table 2, uses only two entities: antigens (Ag) and B cells. The Ag models the hydrophobic pattern of the given protein, that is a sequence $s \in \{H, P\}^{\ell}$. The B cell population $P^{(t)}$ represents a set of candidate solutions at each time step (or generation) t. A B cell (or a B cell receptor) is a sequence of *relative directions* (Krasnogor et al. 1999, 2002) $r \in \{F, L, R\}^{\ell-1}$; where each r_i is a relative direction with respect to the previous direction (r_{i-1}), with $i = 2, \ldots, \ell - 1$ (viz. there are $\ell - 2$ relative directions), and r_1 is the first, nonrelative direction. Hence, we obtain an overall sequence r of length $\ell - 1$. The sequence r specifies a 2D conformation suitable to compute the energy value of the hydrophobic pattern of the given protein.

Table 2 Outline of an immunological algorithm

Immune algorithm (ℓ, d, dup, τ_B, T_{max})
$N_c := d \times dup$;
$t := 0$;
$P^{(t)}$:= Initial_Pop(d);
Evaluate($P^{(t)}$);
while ($\neg$ Termination_Condition(T_{max})) **do**
  P^{clo} := Cloning ($P^{(t)}$, N_c);
  P^{macro} := Hyper–Macromutation (P^{clo}, ℓ);
  Evaluate (P^{macro});
  Pure_Aging($P^{(t)}$, P^{macro}, τ_B);
  $P^{(t+1)}$:= Elitist_Merge ($P^{(t)}$, P^{macro});
  $t := t + 1$;
end_while

At each time step t, we have a population $P^{(t)}$ of size d. The initial population, at time $t = 0$, is randomly generated in such a way that every B cell in $P^{(0)}$, and generally in all the populations used by our IA, represents a *self-avoiding* conformation. The procedure *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 HH-contacts in the 2D lattice. Our IA, like all immune algorithms based on the clonal selection principle, is characterized by clonal expansion, that is the cloning of B cells with higher antigenic affinity. The implemented IA uses three immune operators: cloning, hypermacromutation and aging. Their functioning is described below.

3.1 Cloning

The cloning operator, simply, clones each B cell *dup* times, thus producing an intermediate population P^{clo}, which is a multiset of size $d \times dup$. Throughout this paper, we shall call it *static cloning operator*, as opposed to a *proportional cloning operator* [used in the pattern recognition version of CLONALG (De Castro and Von Zuben 2002)], that clones B cells proportionally to their antigenic affinities. Preliminary experimental results using the latter operator (not shown in this paper) showed frequent premature convergence during population evolution. As a matter of fact, proportional cloning gives more offsprings to B cells with higher affinity values, whereby the process is more likely to get trapped into local minima of the landscape (Cutello et al. 2007b). However, the proportional cloning operator may well be used to explore the attractor basins of the conformational space more deeply, just like an implicit local search procedure (Cutello et al. 2005).

3.2 Hypermacromutations

The hypermacromutation operator acts on each B cell receptor of P^{clo}; it tries to mutate the B cell receptor M times, while maintaining the self-avoiding property. The number of mutations M is randomly determined for each B cell as $M = j - i + 1$, with i and j being two random integers such that $0 < i < j < \ell$. The mutations sequentially apply to the $[x_i, x_j]$ contiguous subsequence of $\vec{x}$.

More precisely, the hypermacromutation operator perturbs the B cell population P^{clo}, and produces the new B cell population P^{macro} (again a multiset); each B cell is a feasible candidate solution of the HP model, that is, it is a self-avoiding walk chain on the lattice, $R = \langle r_1, r_2, \ldots, r_{\ell-1} \rangle \in \{L, R, F\}^{\ell-1}$. Hence, given a protein conformation sequence R, the hypermacromutation operator randomly selects a contiguous subsequence $S = \langle s_1, s_2, \ldots, s_M \rangle$ $(2 \leq M \leq \ell - 1)$; then, it randomly selects a perturbation direction, either from left to right $(i = 1 \ldots M)$ or from right to left $(i = M \ldots 1)$; finally, it tries to mutate each relative direction $s_i \in S$, (following the perturbation direction chosen), while maintaining the self-avoiding property. Note that M does not depend on the fitness of the B cell $\vec{x}$, nor on any other parameter, unlike in traditional clonal selection, where M does depend on the fitness of $\vec{x}$. However, if during the B cell mutation process a constructive mutation occurs, viz. one that improves the fitness function value of the given B cell, then the mutation procedure will move on to the next B cell, thus the actual number of mutations is *at most M*. We adopted this scheme to slow down (premature) convergence, by a stepwise exploration of the search space.

Given a feasible conformation, hypermacromutation maintains the self-avoiding property as follows. For each relative direction $D = s_i$ in the subsequence S, it randomly selects a new direction $D' \in \{F, L, R\} \backslash \{D\}$. If the new conformation is again self-avoiding, then

the operator accepts it, otherwise the remaining direction $D'' \in \{F, L, R\}\backslash\{D, D'\}$ is tried; if also the choice of D'' as new relative direction violates the self-avoiding property, then s_i is left unchanged.

3.3 Aging process

The aging process eliminates old B cells in the populations $P^{(t)}$ and P^{macro}, to avoid 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 for a B cell to remain in the population. When a B cell is $\tau_B + 1$ old, it is erased from the current population, regardless of its fitness value. We call this strategy *static pure aging*.

During clonal 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 is meant to give an equal opportunity to each "new conformation" to effectively explore the funnel landscape (Dill and Chan 1997) (see Fig. 1).

Note that, for τ_B greater than the maximum number of allowed generations (e.g., $\tau_B = 200000$), the IA works as if it were without aging. In such a limit case, the algorithm uses a strong elitist selection strategy.

The best B cells which "survived" the aging process are selected from the populations $P^{(t)}$ and P^{macro}, in such a way that each B cell receptor is unique, i.e. each conformation is different from all other conformations. In this way, we obtain the new population $P^{(t+1)}$, which is a set of d B cells, for the next generation $t + 1$. If only $d' < d$ B cells survived, then the *Elitist_Merge* function creates randomly $d - d'$ new B cells (*Birth phase*).

The boolean function *Termination_Condition* (T_{max}) returns true if a solution is found, or a maximum number of fitness function evaluations (T_{max}) is reached. Table 2 displays a succinct outline of the immunological algorithm described so far. Finally, it is worth noting that both the representation and the mutation operator presented above use a discrete coding. Indeed, they work on a three-letter alphabet $\{L, R, F\}$, hence this machinery is only applicable in the context of discrete coding.

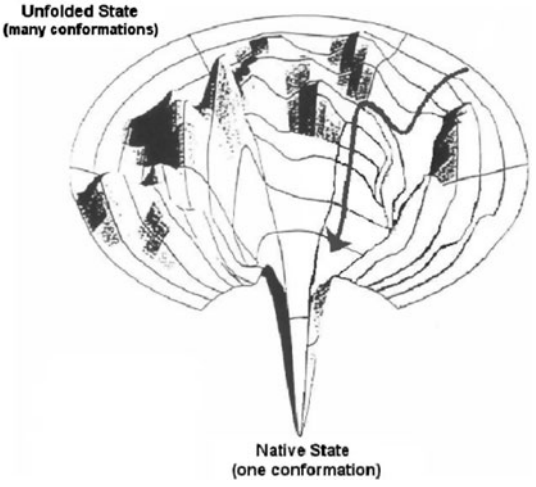

Fig. 1 Energy landscape for the protein structure prediction. The figure is taken from *Dill Research Group*, University of California, San Francisco at http://www.dillgroup.ucsf.edu/

4 Optimal protein structures

To analyze the learning process which takes place in bio-inspired algorithms, we use an entropy function, the *Information Gain*. This measures the amount of information the algorithm discovers during the learning phase (Cutello et al. 2003, 2007a). To this end, we define the B cells distribution function $f^{(t)}{}_m$ as the ratio between the number, B^t_m, of B cells at time t with fitness function value m, and the total number of B cells:

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

where the fitness value $m = 0$ characterizes B cells with no folding, while h is the highest fitness value found. The information gain is then defined as follows:

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

The gain is the amount of information the algorithm 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. This is consistent with the idea of a *maximum information-gain principle* of the form $\frac{dK}{dt} \geq 0$.

It is evident how the Information Gain is a more informative measure than the average. Following the experimental runs reported in Cutello et al. (2007b), we further experimented with $\tau_B = 15$ and found that the standard deviation, that is the uncertainty over the population of a given generation, decreases quickly in the first 20 generations. Indeed, the IA converges to the global minimum in this temporal window. After this "threshold" the standard deviation suddenly increases, further producing strong oscillations; that is to say, 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, that is, the population at generation $t = 1$ extracts more informative building blocks than the population at the second generation. As we mentioned, a constructive mutation is one that improves the fitness function value of a given B cell receptor; consistently, a destructive mutation is one that makes it worse, while a neutral mutation modifies the B cell receptor by only producing a new conformation with the same fitness value of the previous B cell receptor.

It seems interesting to highlight the increasingly relevant role that information-theoretic principles play in discrete models of biological processes. Parallel to the role of the maximum information gain principle in assessing convergence of the global search problem considered here, the *log-gain regulation principle* proposed in Manca (2008) plays the role of cumulating the effects of regulators in MP systems (Metabolic P systems) (Manca 2009, 2010), thereby determining the evolution dynamics of the modeled metabolic system. We plan to investigate this kind of connections even further in forthcoming research on immunological algorithms.

Figure 2 shows the dynamic behavior of the fitness function, by displaying the average fitness of populations P^{macro} and $P^{(t+1)}$, and the best fitness value of population $P^{(t+1)}$. As described in Cutello et al. (2003), when the average fitness of the current population is about the same as the average fitness of the mutated population, then we are close to a premature convergence, or in the best case we are reaching a suboptimal solution. From the

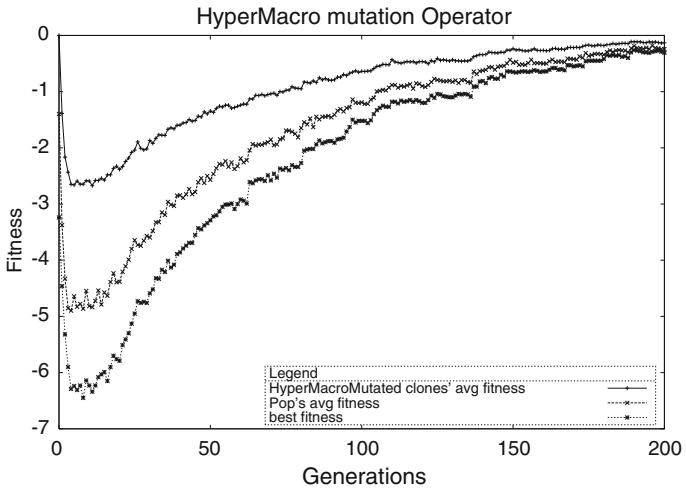

Fig. 2 Average fitness of populations P^{macro} and $P^{(t+1)}$, and best fitness value over generations

plot, instead, it is clear how the proposed IA is able to produce a good diversity in the population, and this is due to the strategies already mentioned above, which are aimed at preventing that the IA may get trapped in local optima. From the same plot it is also possible to see as IA reaches the best solutions in the first few generations, after which the three curves get closer and eventually take almost the same trend.

In Table 3 we show the results obtained by the IA on the first instance of Table 1, with varying values of τ_B, for $dup \in \{2, 5, 10\}$. In this table we report the *Success Rate* (SR), that is how many times does the IA find the native conformation over the overall number of independent run executions, and the *Average Evaluations to Solution* (AES), that is the average number of fitness function evaluations needed to find the native conformation.

Table 3 IA using hypermacromutation on the first instance of Table 1

τ_B	$dup = 2$		$dup = 5$		$dup = 10$	
	SR	*AES*	*SR*	*AES*	*SR*	*AES*
1	8	8293	71	23834.17	**93**	**25338.68**
3	48	10253	**81**	**15845.14**	70	26346.64
5	58	9025	79	15072.77	72	29407.61
8	**58**	**8624.24**	63	18376.73	55	31226.09
10	55	8581.84	66	19448	56	35252.21
15	39	9693.85	51	20767.61	48	33128.46
20	42	9858.19	34	18946.35	42	35198.21
25	38	10186.87	40	21424.07	43	30752.53
50	15	11043.47	20	24709.25	29	37795.21
∞	5	1918.2	10	3550.2	11	5174.36
	36.60	8747.77	51.50	18197.43	**51.90**	**28962**

The experiments reported were obtained by setting the other IA parameters to $d = 10$, $T_{max} = 500000$, with 100 independent run executions. Best *SR*/*AES* values are displayed in boldface

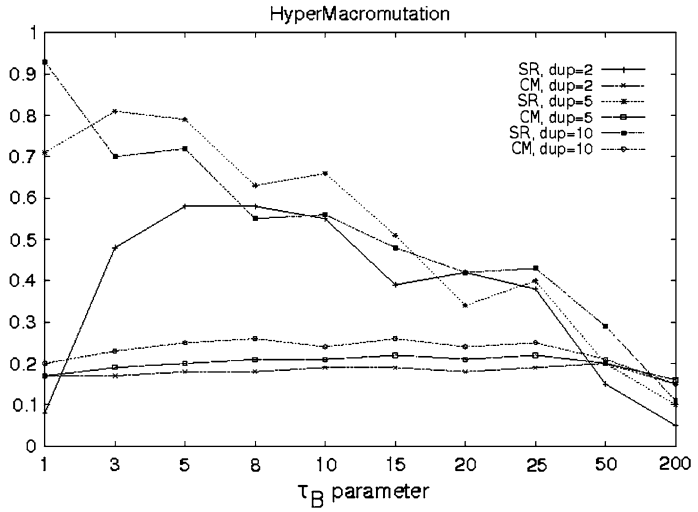

Fig. 3 Success rate versus constructive mutations for $dup \in \{2, 5, 10\}$

Due to the hardness of the PSP search space (rugged funnel landscape, depicted in Fig. 1), the algorithm needs a higher diversity in the population, as one may infer from the results in the table, where a higher number of clones compensates for lower τ_B values. Indeed, the best *SR* and *AES* values were obtained for ($dup = 2$, $\tau_B = 8$), ($dup = 5$, $\tau_B = 3$) and ($dup = 10$, $\tau_B = 1$). The bottom line displays the average values; from these one may see that the best behaviour in terms of *SR* and *AES* values is obtained with $dup = 10$.

In Fig. 3 we compare the number of constructive mutations (CM) with the success rate at different τ_B values. This plot tells that the IA obtains the highest *SR* values for $dup = 5$ and when τ_B is less than or equal to 15; by increasing τ_B, the performance gets worse. Finally, the highest number of constructive mutations was obtained for $dup = 10$. By comparing the CM values for different τ_B values it is apparent that the three curves exhibit a similar behaviour, viz. higher values of τ_B produce a larger rate of constructive mutations. Moreover, for τ_B values greater than or equal to 50, the number of constructive mutations is almost independent of the number of clones produced.

5 Conclusions

The present paper shows that an immunological algorithm based on the clonal selection principle, and equipped with hypermacromutation as variation operator and with an aging mechanism, is suitable to cope with 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. In previous work (Cutello et al. 2007b), a performance analysis of the hypermacromutation operator determined the characteristic parameter surface and the best delimited region on the parameter surfaces that maximize the success rate (SR) value. This region has been used to predict the best parameter value setting. When overlapping the

parameter surfaces with and without elitism, one discovers that the IA performs effectively for low values of T_{max}, the maximum number of fitness function evaluations. By increasing T_{max}, the IA with elitism outperforms the IA without an elitist strategy, in terms of SR.

The aforementioned experimental results show that the aging operator is a key feature of the presented IA. We expect that this new kind of operator could be a useful engine for generating diversity and for better searching 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 PSP.

References

Alm E, Baker D (1999) Prediction of protein-folding mechanisms from free-energy landscapes derived from native structures. Proc Natl Acad Sci USA 96(20):11305–11310

Amato NM, Dill KA, Song G (2003) Using motion planning to map protein folding landscapes and analyze folding kinetics of known native structures. J Comput Biol 10(3):239–255

Apaydin MA, Brutlag DL, Guestrin C, Hsu D, Latombe J-C (2002) Stochastic roadmap simulation: an efficient representation and algorithm for analyzing molecular motion. In: Proceedings of the sixth annual international conference on computational molecular biology (RECOMB). ACM, pp 12–21

Burnet FM (1959) The clonal selection theory of acquired immunity. Cambridge University Press, Cambridge

Covell DG (1992) Folding protein α-carbon chains into compact forms by Monte Carlo methods. Proteins 14(4):409–420

Crescenzi P, Goldman D, Papadimitriou C, Piccolboni A, Yannakakis M (1998) On the complexity of protein folding. J Comput Biol 5(3):423–466

Cutello V, Nicosia G (2004) The clonal selection principle for in silico and in vitro computing. In: De Castro LN, Von Zuben FJ (eds) Recent developments in biologically inspired computing. IGI Publishing, Hershey, pp 104–146

Cutello V, Nicosia G, Pavone M (2003) A hybrid immune algorithm with information gain for the graph coloring problem. In: Cantú-Paz E et al (eds) Proceedings of the genetic and evolutionary computation conference (GECCO). Lecture notes in computer science, vol 2723. Springer, Berlin, pp 171–182

Cutello V, Narzisi G, Nicosia G, Pavone M (2005) Clonal selection algorithms: a comparative case study using effective mutation potentials. In: Jacob C, Pilat ML, Bentley PJ, Timmis J (eds) Proceedings of the fourth international conference on artificial immune systems (ICARIS). Lectures notes in computer science, vol 3627. Springer, Berlin, pp 13–28

Cutello V, Nicosia G, Pavone M (2007a) An immune algorithm with stochastic aging and Kullback entropy for the chromatic number problem. J Comb Optim 14(1):9–33

Cutello V, Nicosia G, Pavone M, Timmis J (2007b) An immune algorithm for protein structure prediction on lattice models. IEEE Trans Evol Comput 11(1):101–117

Dasgupta D (ed) (1999) Artificial immune systems and their applications. Springer, Berlin

De Castro LN, Timmis J (2002) Artificial immune systems: a new computational intelligence approach. Springer, London

De Castro LN, Von Zuben FJ (2002) Learning and optimization using the clonal selection principle. IEEE Trans Evol Comput 6(3):239–251

Dill KA (1985) Theory for the folding and stability of globular proteins. Biochemistry 24(6):1501–1509

Dill KA, Chan HS (1997) From Levinthal to pathways to funnels. Nat Struct Biol 4(1):10–19

Dill KA, Bromberg S, Yue K, Fiebig KM, Yee DP, Thomas PD, Chan HS (1995) Principles of protein folding: a perspective from simple exact models. Protein Sci 4:561–602

Hirst JD (1999) The evolutionary landscape of functional model proteins. Protein Eng 12:721–726

Krasnogor N, Hart WE, Smith J, Pelta DA (1999) Protein structure prediction with evolutionary algorithms. In: Banzhaf W et al (eds) Proceedings of the genetic and evolutionary computation conference (GECCO), vol 2. Morgan Kaufmann, San Francisco, pp 1596–1601

Krasnogor N, Blackburne BP, Burke EK, Hirst JD (2002) Multimeme algorithms for protein structure prediction. In: Merelo JJ, Adamidis P, Beyer H-G (eds) Proceedings of the seventh international conference on parallel problem solving from nature (PPSN VII). Lectures notes in computer science, vol 2439. Springer, Berlin, pp 769–778

Lau KF, Dill KA (1989) A lattice statistical mechanics model of the conformational and sequence spaces of proteins. Macromolecules 22:3986–3997
Levitt M (1983) Protein folding by restrained energy minimization and molecular dynamics. J Mol Biol 170:723–764
Manca V (2008) The metabolic algorithm for P systems: principles and applications. Theor Comput Sci 404(1–2):142–155
Manca V (2009) Log-gain principles for metabolic P systems. In: Condon A, Harel D, Kok JN, Salomaa A, Winfree E (eds) Algorithmic bioprocesses. Natural computing series. Springer, Berlin, pp 585–605
Manca V (2010) From P to MP systems. In: Păun G, Pérez-Jiménez MJ, Riscos-Núñez A, Rozenberg G, Salomaa A (eds) Membrane computing. Lectures notes in computer science, vol 5957. Springer, Berlin, pp 74–94
Muñoz V, Eaton WA (1999) A simple model for calculating the kinetics of protein folding from three dimensional structures. Proc Natl Acad Sci USA 96(20):11311–11316
Narzisi G (2008) Optimization and tradeoffs in protein structure prediction. Dissertation, University of Catania
Nicosia G (2004) Immune algorithms for optimization and protein structure prediction. Dissertation, University of Catania
Pavone M (2003) Biologically inspired algorithms for partitioning, coloring and protein structure prediction problems. Dissertation, University of Catania
Toma L, Toma S (1996) Contact interactions method: a new algorithm for protein folding simulations. Protein Sci 5:147–153
Unger R, Moult J (1993) Genetic algorithms for protein folding simulations. J Mol Biol 231(1):75–81