Theor Chem Acc (2004) 112: 163–178
DOI 10.1007/s00214-004-0601-4

Theoretical
Chemistry Accounts

Regular article

Development and optimisation of a novel genetic algorithm for studying model protein folding

Graham A. Cox, Thomas V. Mortimer-Jones, Robert P. Taylor, Roy L. Johnston

School of Chemistry, University of Birmingham, Edgbaston, Birmingham B15 2TT, UK

Received: 17 December 2003 /Accepted: 4 February 2004 /Published online: 29 June 2004

Abstract. Determination of the native state of a protein from its amino acid sequence is the goal of protein folding simulations, with potential applications in gene therapy and drug design. Location of the global minimum structure for a given sequence, however, is a difficult optimisation problem. In this paper, we describe the development and application of a genetic algorithm (GA) to find the lowest-energy conformations for the 2D HP lattice bead protein model. Optimisation of the parameters of our ''standard'' GA program reveals that the GA is most successful (at finding the lowest-energy conformations) for high rates of mating and mutation and relatively high elitism. We have also introduced a number of new genetic operators: a duplicate predator—which maintains population diversity by eliminating duplicate structures; brood selection—where two ''parent'' structures undergo crossover and give rise to a brood of (not just two) offspring; and a Monte Carlo based local search algorithm—to explore the neighbourhood of all members of the population. It is shown that these operators lead to significant improvements in the success and efficiency of the GA, both compared with our standard GA and with previously published GA studies for benchmark HP sequences with up to 50 beads.

Keywords: Genetic algorithm – Lattice bead model – Protein folding

1 Introduction: the protein folding problem

One of the most interesting and important problems in chemical biology is to establish or predict the 3D local spatial arrangement (''secondary structure'') and folded conformation (''tertiary structure'') adopted by a protein or polypeptide molecule from knowledge of its primary structure—that is the 1D sequence of amino acid residues from which the molecule is built [1, 2]. This sequence–structure correlation is of critical importance if we are to understand how proteins fold and, hence, to investigate sequence–activity relationships for proteins. The ''protein folding problem'' is essentially a search for the biologically active (functional) conformation of a protein (the so-called native state), for a given sequence of amino acid residues. While it was originally assumed that the native state of a protein is the global minimum (GM) on the folding energy hypersurface (the ''thermodynamic hypothesis'' [3]), more recent work indicates that the functional conformation may sometimes be that which is most frequently visited under native conditions (the ''kinetic hypothesis'' [4])—so folding studies should also aim to identify low-lying metastable local minima, in addition to the GM. The major difficulty associated with the molecular dynamics (MD) simulation of protein folding is that proteins typically fold with time scales of the order of seconds or minutes, while MD simulations are limited to under a microsecond. As Levinthal [5] has pointed out, however, even this relatively long experimental folding time (seconds to minutes) is negligible compared with the time which would be required for a protein to explore its folding space randomly, based on typical timescales for torsional motions. The ability of natural proteins to fold reliably to a unique native state has been attributed to the presence of a ''folding funnel'' on the folding free-energy landscape, so that misfolded states are funnelled towards the native state (i.e. protein folding is a far-from-random process) [6]. As well as determining the low-energy protein conformations, it is therefore, important to discover the nature of the folding energy landscape (funnels, heights of potential barriers, etc.) in order to gain a better understanding of the dynamics of protein folding.

1.1 Modelling protein folding

There are a variety of protein models which differ in the way in which they approximate the protein molecule and

Correspondence to: R. L. Johnston
e-mail: roy@tc.bham.ac.uk

how they treat interactions between amino acid residues and with solvents (if included). Owing to the size and complexity of protein hypersurfaces, simplified models have often been employed to study the protein folding process [7].

One of the simplest protein models is the HP lattice bead model [8,9,10], which is a minimalist model of a protein, representing the constituent amino acid residues by either hydrophobic (H) or polar (P) (hydrophilic) beads which lie on a 2D or 3D lattice: square and cubic lattices are most common, though more complex lattices have also been studied. Such coarse-grained protein models, although being less realistic models of actual proteins, can capture some of the important folding behaviour of real proteins, and they have the advantage of being simple, so that their energies may be calculated quickly, making them good for systematic grid searches and for carrying out comparisons of different folding search algorithms. Applications of the HP lattice bead model (and its variants) include studies of minimum-energy structures [8], folding kinetics [11], protein designability and foldability [10, 12, 13], the effect of confinement on protein folding [14], protein biogenesis [9], ligand binding [15] and the evolution of protein functionality [16, 17].

More realistic (though more complex) protein models can be constructed, such as off-lattice bead models (corresponding to a move from a discretised grid into continuous space), united atom models (where each carbon and nitrogen atom in the protein backbone is now treated explicitly and the side chains are represented by one or two ''united atoms'') and all-atom models (in which all the atoms in the protein are treated explicity and the energy is calculated using a molecular mechanics force field [18]). Although all-atom models should be the most accurate representations of real proteins, such calculations are slower than bead or united atom models, as the number of energy terms to be evaluated and the number of structural parameters to be varied are much greater. Indeed, studies of protein folding landscapes and folding dynamics, using off-lattice bead and united atom models, have shown that simple models can reproduce the behaviour of real proteins [19–21] in a generic sense.

Solvent (usually water) effects on protein folding can also be included in a number of ways. Some protein models (such as the HP bead model) include the effect of the solvent implicitly (via the hydrophobicity of the residues), while inclusion of solvent in the all-atom models can be accomplished by introducing an effective solvent (i.e. a dielectric medium) or an explicit solvent model, where the protein is embedded in a large number of water molecules, with protein–water and water–water interactions included in the energy calculation.

Despite the reduction in complexity inherent in the minimalist HP lattice bead model, it has been shown to belong to the set of problems that are ''NP-hard'' [22, 23]. This means that there should be no polynomial algorithm that can solve the protein folding problem (i.e. unambiguously find the lowest-energy folding conformation for a given sequence) exactly. For this reason, researchers have adopted heuristic and approximation algorithms. For the HP lattice bead model and other minimalist models, the approaches adopted include Monte Carlo [24, 25, 26, 27, 28], chain growth algorithms [29, 30, 31], simulated annealing [32], genetic algorithms (GAs) [22, 33, 34, 35, 36 ,37], and ant colony optimization [38, 39, 40].

1.2 GAs for studying protein folding

The GA [41, 42] is a search technique, based on the principles of natural evolution, which uses operators that are analogues of the evolutionary processes of genetic crossover (or mating), mutation and natural selection to explore multidimensional parameter spaces. A GA can be applied to any problem where the variables to be optimised (genes) can be encoded to form a string (chromosome). Each string represents a trial solution of the problem. The GA operators exchange information between the strings to evolve new and better solutions. A crucial feature of the GA approach is that it operates effectively in a parallel manner, such that many different regions of parameter space are investigated simultaneously. Furthermore, information concerning different regions of parameter space is passed actively between the individual strings by the crossover operator, thereby disseminating genetic information throughout the population. The GA is an intelligent search mechanism that is able to learn which regions of the search space represent good solutions.

The design of a protein folding GA (i.e. how the structure is coded and how crossover and mutation are carried out) depends critically on the model used to describe the protein. The optimal values of the GA parameters (population size, mating rate, mutation rate, etc.) also depend on the model adopted, as will the number of generations required to find the lowest-energy folding conformation. A survey of GAs which have been applied to the protein folding problem is presented in reviews by Pedersen [36] and Unger [37] and some of these GAs are discussed briefly in the following.

Studies by Unger and Moult [22] and by Judson et al. [43] have shown that GAs generally perform better than Monte Carlo algorithms for finding low-energy protein conformations—presumably because the GA finds it easier to escape from local minima corresponding to nonoptimal compact structures than Monte Carlo algorithms, which may have to overcome a large energy barrier to escape from such a minimum. All-atom protein folding GAs include the work of Pedersen and Moult [44], who calculated fitness from a potential energy function depending on electrostatic terms and the accessible surface area, with the GA parameters being optimised for a set of fragments of known structure. Pederson and Moult used their GA to study short sequences (with 12–22 residues), using a parallel GA code, finding once again that the GA is significantly more effective than the Monte Carlo method for finding low-energy folded structures. A number of groups have used all-atom GAs to predict the conformations of small oligopeptides, using empirical force fields to obtain the potential energy,

and hence the fitness. GAs have also been used to predict the lowest-energy conformations of side chains, starting from an experimental backbone conformation [44]. In an interesting, alternative application, Jones [45] has applied a GA to protein design — trying to find the optimum amino acid sequence which is consistent with a given folding structure. In this GA, in contrast to the previously mentioned examples, the genetic operators act in sequence space rather than conformation space — i.e. the genes are residues rather than coordinates or torsion angles. In the future, GAs (and other evolutionary algorithms) are likely to play an increasingly important role in the areas of protein design, protein engineering (for example, looking for mutants with specific properties) and protein biogenesis—i.e. using an artificial evolutionary algorithm to simulate the natural evolution of proteins and other biomolecules.

In this study, building on our experience in applying GA to the geometry optimisation of cluster molecules [46] and for the solution of crystal structures from powder diffraction data [47], we have developed a GA for studying the protein folding problem for the 2D square lattice HP bead model.

2 Methodology

2.1 The HP lattice bead model

In the present work, we have adopted the 2D square lattice HP bead model [8, 10], where the H and P beads are constrained to lie on a 2D square lattice and interactions occur only between non-bonded beads that lie adjacent to each other on the lattice (''topological neighbours''), but are not adjacent in the sequence (i.e. they are not directly bonded ''sequence neighbours'') [8]. The values of the H–H, H–P and P–P interactions (ϵ_{ij}) in the standard HP model are [8]

$$\epsilon_{HH} = -1.0, \epsilon_{HP} = 0.0, \epsilon_{PP} = 0.0 \ , \tag{1}$$

so the HP potential can be represented by the interaction vector (−1,0,0).

The energy of the model protein is obtained by summing over these local interactions:

$$E = \sum_{i<j} \epsilon_{ij} \Delta_{ij} \ , \tag{2}$$

where

$$\Delta_{ij} = \begin{cases} 1 & \text{if } i \text{ and } j \text{ are topological neighbours, but} \\ & \text{are not sequence neighbours} \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

It should be noted that the effective attractive (stabilising) interaction between the H beads reflects the fact that (in aqueous solution) the hydrophobic interaction (i.e. the repulsion of hydrophobic residues and water molecules) is the driving force for protein folding and that the native structures of many proteins are compact, with cores which are relatively rich in hydrophobic residues [8, 38]. The reasons for studying the 2D, rather than the 3D lattice bead model are twofold [8]: first, the surface-to-volume ratio of the 2D model approaches realistic ''protein values'' for smaller sequences than in 3D; and second, the computational requirements are greatly reduced. The 2D analogues of protein secondary structure features, such as α-helices and β-sheets, naturally arise in the compact cores of such models, implying that the secondary structure is not driven by specific hydrogen-bonding interactions but is a consequence of the compactness of the core and the presence in the core of hydrophobic groups [18].

2.2 The coordinate system

In this work, we define the folding conformation of the protein using a local coordinate system in which the position of a bead j is defined relative to its predecesors ($j-2$ and $j-1$) [22, 33, 34, 38]. Thus, in two dimensions, the direction of the bond joining the $(j-1)$th and j^{th} beads can be left (0), right (1) or straight ahead (2) relative to the bond joining the $(j-2)$th and $(j-1)$th beads. Each protein conformation is therefore represented by a conformation vector, **c**, which is a string of 0's, 1's and 2's. As the energy of each conformation is invariant to rotation of the whole molecule, we fix the positions of the first two beads in the chain, such that bead 1 lies at the origin (0,0) and bead 2 lies along the x-axis (1,0).

The advantages of using a local, rather than a global (Cartesian), coordinate system (where the conformation vector would consist of successive displacements along the x- or y-axes) are

1. Applying a ''genetic operator'' to a portion of the conformation vector preserves local structure in unaffected portions of the vector.
2. The contribution of these local structure motifs to the energy of the structure (ignoring interactions with beads further along the sequence) is independent of the orientation of the motif in Cartesian space.
3. In the context of the GA, such local structure motifs correspond to ''schemata'' [41]. Through the crossover operation, good (low-energy) schemata have a higher probability of being copied forward into future generations.
4. Some of the invalid conformations (where two or more beads occupy the same lattice site) produced in a global co-ordinate system (e.g. those where a $+x$ move is immediately followed by a $-x$ move) are eliminated in a local coordinate scheme, where such backward moves cannot occur.

Krasnogor et al. [34] have demonstrated that a GA incorporating a local coordinate system almost always outperforms one that employs global coordinates.

In order to calculate the total energy of the protein, within the HP model, the number of H–H contacts must be enumerated, which requires the local coordinate conformation vector to be mapped onto the global cartesian coordinate system corresponding to the 2D square lattice, so that topological neighbours can be identified.

It should be noted that the conformation vector is independent of the sequence of H's and P's — which are represented by a sequence vector, $\boldsymbol{\sigma}$, of 0's and 1's (corresponding to H and P beads, respectively) [8]. The energy of a particular protein structure (which may be represented by the structure vector, **s**) depends on both the conformation and the sequence, as they both determine the number of H–H interactions—i.e. $\mathbf{s} = (\mathbf{c};\sigma)$. In this study, the sequence (σ) is held constant throughout, while the conformation vectors (**c**) are allowed to change, as we aim to find the protein structures corresponding to the lowest-energy conformation(s) for a given sequence.

In our GA program, the GA search space has not been restricted to a single quadrant of configuration space, so it is possible to find mirror image (''enantiomeric'') conformations, where two enantiomers are related by reflection in the xz-plane. In terms of the conformation vector, **c**, the enantiomer of a particular conformation is obtained by converting all 0's into 1's and all 1's into 0's—with the 2's left unchanged. Of course, for a given sequence, enantiomeric conformations will have mirror image local and long-range structures and the same number of H–H contacts, and therefore identical energies.

For the study reported here, we investigated HP bead sequences with 20, 24, 25, 36, 48 and 49 beads. The specific sequences, which are listed in Table 1, are standard benchmark sequences that have previously been used for testing GA and other search algorithms [22, 38, 39, 48]. The table also includes the energy, E^*, of the GM (or global minima — since all of these structures have degenerate global minima—more than one structure with the same lowest-energy) for each sequence. For the larger sequences (HP-36 and above), it is not known for sure whether the reported lowest

Table 1. Benchmark HP sequences used in the present study [48]. The lowest energies reported in the literature for these sequences are indicated by E^*. E^* values in *bold* are optimal solutions found by systematic grid searching

Name	Length	E^*	Sequence
HP-20	20	**−9**	$HPHP_2H_2PHP_2HPH_2P_2HPH$
HP-24	24	**−9**	$H_2P_2(HP_2)_6H_2$
HP-25	25	**−8**	$P_2HP_2(H_2P_4)_3H_2$
HP-36	36	−14	$P_3H_2P_2H_2P_5H_7P_2H_2P_4H_2P_2HP_2$
HP-48	48	−23	$P_2H(P_2H_2)_2P_5H_{10}P_6(H_2P_2)_2HP_2H_5$
HP-50	50	−21	$H_2(PH)_3PH_4P(HP_3)_3P(HP_3)_2HPH_4(PH)_4H$

energies actually correspond to the GM, as no grid search has been performed for these chain sizes.

2.3 The GA

The way in which our GA program operates is shown as a schematic flow diagram in Fig. 1 and the characteristics of the GA are described in the following.

2.3.1 The standard GA

2.3.1.1 Generating the initial population. The initial population corresponds to the starting set of individuals which are to be evolved by the GA. In our GA, the individuals are a set of conformation vectors (strings of 0's, 1's and 2's, as already described). The initial population is formed by the constructor routine, which generates a number of valid conformations at random. In lattice bead models, valid protein conformations correspond to self-avoiding walks on the 2D or 3D lattice. In contrast, invalid conformations correspond to non-self-avoiding walks, where two or more beads occupy one or more sites on the lattice. This is clearly unphysical, and such conformations should be eliminated.

The proportion of invalid conformations increases rapidly with the protein chain length, especially for the 2D model: for example, more than 60% of randomly generated conformations are invalid for a chain of only 15 beads. For this reason, building up a complete, random conformation vector and testing it at the end to check whether the conformation is valid, ("end-checking") rapidly becomes inefficient for longer sequences. The approach we have adopted involves growing the chain one bead at a time, checking the validity of the incomplete conformation at each step and backtracking when an invalid subconformation is generated. This "backtracking" algorithm is much more computationally efficient than the end-checking algorithm. It should be noted that our method is essentially the same as that used by Shmygelska et al. [38] and is similar in principle to the Rosenbluth method which was developed to grow self-avoiding polymer chains, within a Monte Carlo framework [49]. (This approach has subsequently been improved in the prune-enriched Rosenbluth method [27, 50].)

Following initial optimisation studies, in all subsequent calculations (described later) the population size (which is constant within each GA run) was fixed at 200.

2.3.1.2 Fitness. Fitness is an important concept for the operation of the GA. The fitness of a string is a measure of the quality of the trial solution represented by the string with respect to the function being optimised.

In our work, the fitness of the ***i***th individual (structure) is simply related to its energy:

$$F_i = -E_i + 0.01 \quad . \tag{4}$$

Thus, the fitness is a positive quantity, with high fitness corresponding to a large negative energy. The addition of the small constant amount (0.01) is carried out so that even "open" structures with energies $E_i = 0$ will have nonzero fitness and, hence, a finite probability of being selected for crossover.

2.3.1.3 Selection. Selection refers to the way in which individual members of the population are chosen to pass into a temporary "parent population", which is subsequently subjected to a number of genetic operators, as shown in Fig. 1. In this study, we adopted roulette wheel selection [41, 42]: a random number is generated between 0 and F_{total} (the sum of the fitness values of the entire population); if the random number lies in the *i*th interval of cumulative fitness, then the *i*th member of the population is selected for the parent population. In this way, parents are selected until the size of the parent population equals that of the starting population. The fittest individuals will tend to be copied more than once into the parent population, the only restriction being the prevention of the consecutive selection of any given population member (since crossover between identical conformations will lead to no new structures).

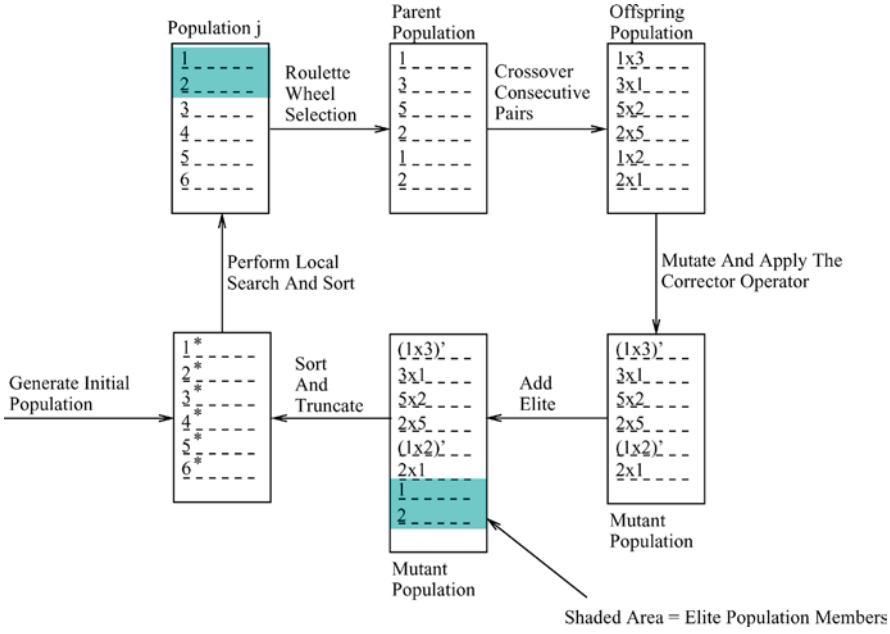


Fig. 1. Schematic flow diagram for the genetic algorithm (GA) program used in this study

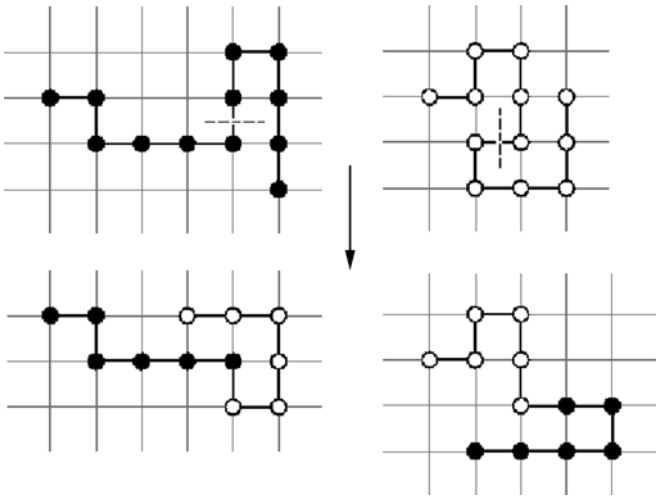


Fig. 2. Demonstration of the preservation of local structure in one-point crossover

2.3.1.4 Crossover. Crossover (or mating) is the way in which "genetic" information from two parent strings is combined to generate "offspring". In this study, the variable mating rate is defined as the percentage of parents in the parent population which undergo crossover. The GA cycles through the parent population, applying the crossover operator to consecutive parent pairs until the correct number of parents have participated in crossover. The two offspring produced from each crossover operation overwrite their parents. (In Fig. 1, the two offspring of parents i and j are labelled $i \times j$ and $j \times i$.) The offspring and unmated parents then pass into the "offspring population". If one or both of the offspring created by the crossover operator are invalid (non-self-avoiding) conformations, further crossovers are carried out (between the same pair of parents) until (in total) two valid offspring have been created. In the (very rare) case where all possible crossovers between a pair of parents lead to either one or no valid offspring, one or both of the parents are copied over unchanged into the offspring population.

Preliminary testing of one-point, two-point and uniform crossover showed that one-point crossover — where two parent strings (conformation vectors) are cut at the same randomly chosen point and complementary portions are combined, to generate two offspring – leads to better performance of the GA, probably owing to the retention of larger local structure motifs (schemata), as shown in Fig. 2, and a reduced tendency to produce invalid offspring conformations.

2.3.1.5 Mutation. While the crossover operation leads to a mixing of genetic material in the offspring, no new genetic material is introduced, which can lead to lack of population diversity and eventually "stagnation" — where the population converges on the same, nonoptimal solution. The GA mutation operator helps to increase population diversity by introducing new genetic material.

In this study, a number of mutation operators were adopted, as shown in Fig. 3. Some of these operators have been utilised as mutations in previous GA and ant colony optimisation studies of protein folding [22, 39], and as move classes in Monte Carlo studies of proteins [11, 14, 26]. Some originate in earlier simulation studies of polymer structures and dynamics [18].

– In-plane rotation involves a $\pm 90°$ or $180°$ rotation, in the xy-plane, of the subchain following a randomly selected bond (say between beads $j-1$ and j). In terms of the conformation vector, this corresponds to a change of the local coordinate direction of bead $j+1$, with the rest of the conformation vector being unchanged — i.e. a single bit change. This mutation, therefore, leaves most of the local structure intact.

– Out-of-plane rotation involves a $180°$ rotation, in either the xz-plane or the yz-plane, of the sub-chain following a randomly selected bond (say between beads $j-1$ and j). [The rotation plane depends on whether the $(j-1)$–j bond points along the x-axis or the y-axis.] In terms of the conformation vector, this corresponds to all of the 0's being changed to 1's and all of the 1's being changed to 0's (with the 2's left unchanged) for the entire subchain starting at bead $j+1$. This mutation, therefore, leads to an inversion of the rotated fragment, thereby generating a diastereoisomer of the original conformation.

– Crank shaft rotation involves a $180°$ rotation, in either the xz-plane or the yz-plane, of a crank shaft local structure motif (corresponding

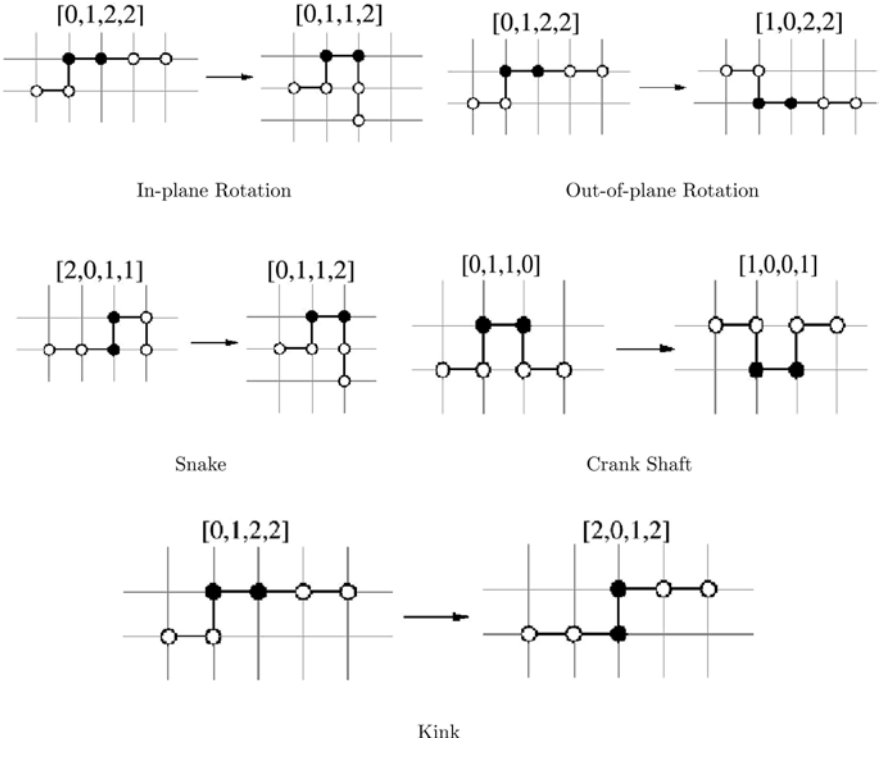


Fig. 3. The mutation operators adopted in this study. In each case, the conformation vectors before and after mutation are given

to the four digit strings ..0110.. or ..1001.. in the conformation vector), which leads to the interconversion of these four digit strings — with the rest of the conformation vector left unchanged.
– Kink motion involves the inversion of a kink (or bend) local structure motif, where the kink bead (say bead j) is moved diagonally across a lattice square, such that it is still bonded to its two neighbours (beads $j-1$ and $j+1$). This only leads to a change of the local coordinate directions of the $(j-1)$–j, j–$(j+1)$ and $(j+1)$–$(j+2)$ bonds, with the rest of the conformation vector left unchanged.
– Snake motion involves the movement of the end of the protein to a neighbouring vacant lattice site (if available), with each of the remaining beads moving to the position of its predecessor. This is analogous to the process of reptation in polymers and is one way in which a dense structure can be mutated with a low likelihood of creating an invalid mutant. In terms of the conformation vector, this mutation corresponds to shifting the vector along by one place and placing the first component of the vector at the end.
The variable mutation rate is defined as the probability of a selected individual undergoing mutation. The GA cycles through the offspring population, generating a random number between 0 and 1, for each individual. The mutation operator acts on individuals where this random number is less than the mutation rate. The mutation operator randomly selects, with equal probability, the mutation type to perform. Individuals selected for mutation are overwritten by the subsequent mutant. Mutants and unmutated individuals pass into the "mutant population".

2.3.1.6 The corrector operator. Since the mutation operator often generates invalid (non-self-avoiding) conformations, a correction operator has been introduced to generate valid conformations from any invalid conformations resulting from mutation. Our corrector operator, which is illustrated in Fig. 4, is based on the approach introduced by Schmygelska and Hoos [39] in their ant colony optimisation study of protein folding for the HP bead model. An invalid conformer can undergo refolding at points of infeasibility (i.e. where two beads lie on top of each other), ensuring that a valid conformer results. The operator starts at the first nonfixed bead and cycles through the conformer placing beads using their corresponding value in the conformation vector. If the placement of the jth bead results in an infeasibility, the bead is randomly repositioned to a valid site, if the bead cannot occupy a valid site, the operator returns to the $(j-1)$th bead and attempts a valid repositioning. The operator continues in this fashion, backtracking as much as necessary until a valid conformation vector is obtained which is as closely related to the initial invalid conformer as possible. As the parent structure, which was mutated, started off as a valid conformation, correction will take place on or after the mutation site. [In Fig. 1, the (possibly corrected) mutants are indicated by primes.]

2.3.1.7 Elitism. In the context of GAs, an "elitist strategy" corresponds to allowing the best individuals in a population to survive unchanged from one generation to the next, thereby ensuring that the best member of the population cannot get worse from one generation to the next. In our GA, elitism is accomplished by specifying the percentage of the best individuals within the jth population which are to be appended to the mutant population, prior to the generation of the $(j+1)^{\text{th}}$ population. (The elite members of the population are indicated by the shaded region in Fig. 1.)

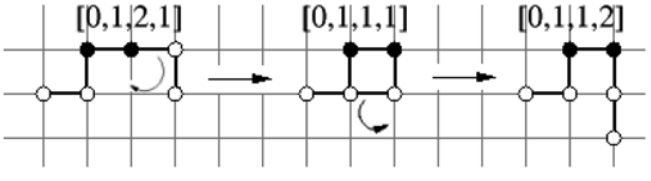


Fig. 4. An example of the application of the corrector operator. An in-plane-rotation mutation converts the valid conformer [0,1,2,1 into the invalid conformer [0,1,1,2, where the second and sixth beads occupy the same lattice site. In this case, the corrector operator refolds the invalid conformer by rotating the sixth bead away from the second bead, to generate the valid conformer [0,1,1,2]

2.3.1.8 "Natural" selection. In biological evolution the concept of the "survival of the fittest" (or best adapted to the environment) is a strong evolutionary driving force. In the case of a GA, although the selection is clearly not "natural", individuals (be they parents, offspring or mutants) are likewise selected to survive into the next generation on the basis of their fitness (their quality with regards to the quantity being optimised). There are many modifications of the natural selection step: here we generate the $(j+1)$th population by sorting the mutant population (including any elite structures) with respect to fitness and truncating it to the original size of the jth population, as shown in Fig. 1. The GA program then continues for a predetermined number of generations (each generation corresponding to a cycle of crossover, mutation and elitism) or until some convergence criterion is reached.

2.3.2 Additional operations

In addition to the previously decribed GA operators, which are found in most GA applications [41, 42], we investigated the following nonstandard operations, to determine to what extent they can improve the success rate and efficiency of our protein folding GA.

2.3.2.1 Duplicate Predator. In recent work, we have extended the analogy between GAs and natural evolution by considering the use of "predators" to remove unwanted individuals or traits from a population [51]. Here we have investigated the application of a "duplicate predator", which deletes ("predates") identical conformations. It should be noted that our duplicate predator is similar in nature to the "pioneer search" strategy introduced by König and Dandekar [35], though they only checked for uniqueness of newly generated individuals every ten generations.

In our study, we define the duplicate predator limit (DPL) to be the maximum number of times that a given structure is allowed to appear in the population in any particular generation. For convenience sake, the normal (predation off) situation is represented by DPL = 0, though in this case there is no restriction on the number of identical stuctures. The duplicate predator serves to increase the diversity (proportion of unique structures) of the population, in order to prevent premature convergence ("stagnation") of the population on a nonoptimal solution.

2.3.2.2 Brood Selection. Rather than generating two offspring from two parents by crossover, a "brood" (of pre-determined size) of offspring may be generated, from which the best offspring can be selected. For large brood sizes, this procedure enables more thorough searching of the possible offspring space of the two parents and is analogous to "soft brood selection" in the field of genetic programming [52]. We have considered two implementations of brood selection:

1. The best two (highest fitness) members of the brood are chosen and they pass into the offspring population. (For a brood size of 2, this is identical to the mating procedure already described.)
2. The offspring in the brood compete with the parents and the best two individuals from this "family" are passed into the offspring population. This approach has built-in elitism, since parents cannot be replaced by less fit offspring.

2.3.2.3 Local search. In problems where the search space is continuous, offspring and mutants invariably occupy states which are not minima, but are rather states which lie within an energy well. In such cases, performing a local minimisation will track each individual to its corresponding local minimum. In the GA context, GAs

incorporating local minimisation correspond to Lamarckian, rather than Darwinian evolution, as individuals pass on a proportion of the characteristics that they have acquired (during the minimisation step) to their offspring. Such Lamarckian GAs, which couple local minimisation with GA searching, have been found to improve GA efficiency for a number of different applications of GAs in global optimisation [46, 53, 54]. Although, owing to the discrete nature of the conformation space of the HP lattice bead model, it is not possible to perform gradient-driven energy minimisations, it is possible to perform a local search whereby a given conformation undergoes a number of folding changes, testing a number of closely related conformations.

In this study, we performed local searching using the "long range move" Monte Carlo type approach introduced by Schmygelska and Hoos [39], though it should be noted that Unger and Moult also introduced a Monte Carlo mutation and Monte Carlo local searching in their original GA study [22]. In our application, a conformation c_1 with energy E_1 is folded at a randomly chosen position (as in the in-plane rotation mutation) by randomly changing one of the digits in the conformation matrix **c**. The new conformation c_2 is accepted if its energy $E_2 < E_1$. For conformation changes where $E_2 \geq E_1$, the conformational change is accepted with a probability

$$p = \frac{E_2}{15E_1} \quad , \tag{5}$$

where the factor of 15 was found to give reasonable acceptance rates (approximately 25%). Each local search corresponds to 30 of these Monte Carlo steps, with a new random fold carried out at a random position each time, starting from the current conformation. As shown in Fig. 1, if included, local searching (followed by sorting according to the fitnesses of the resulting structures) is carried out after the other GA operations.

As the search for the GM generally becomes more difficult for longer HP sequences [22, 38, 39], we adopted an incremental approach, which involves optimisation of the basic GA parameters for smaller sequences and the introduction and optimisation of the additional GA operations (duplicate predation, brood selection and local searching) for the longer sequences. Finally, we will show results obtained using our optimal GA strategy, for all of the sequences listed in Table 1.

3 Results and discussion

3.1 Optimisation of the standard GA parameters

Before considering the effect of the duplicate predator, brood selection and local searching, it was decided to optimise the GA parameters of the standard GA—i.e. those associated with mating, mutation and elitism. Mating, mutation and elitism rates all effect the progress of the GA: high mutation and mating rates rapidly evolve the population whereas lower rates evolve the population more gently. Elitism ensures good structures are passed on to subsequent generations, enabling more thorough searching of regions of the hypersurface around these structures. However, elitism duplicates structures, potentially destroying the diversity within a population.

Our studies into the optimisation of the GA afforded an insight into different evolutionary strategies. Figure 5 shows 3D plots of the percentage success (the percentage of runs finding the global minimum structures); and the average number of structures sampled during successful GA runs, against the mutation and mating rates, for an elitism of 30%, for sequence HP-24. (Similar calculations on the smaller HP-20 sequence showed almost identical trends to those observed for HP-24.) To enable comparison between the success of different strategies, the total number of structures sampled per GA run was capped at 60,000.

For higher elitism, the percentage success increases for higher mutation rates. This suggests that with elitism producing larger quantities of duplicate structures, the mutation operator becomes increasingly more important, as a means of injecting new genetic information into the population. On the other hand, higher elitism leads (on average) to fewer structures having to be sampled before the GM is found, which can be attributed to elitism retaining parents which would otherwise be replaced by poorer offspring, hence focusing the GA on good areas of the search space.

For lower elitism, the percentage success decreases, and the effect of the mutation operator is less pronounced. The percentage success for low mutation rates is higher with less elitism, which produces fewer duplicates. In these instances, crossover between different population members, along with the large number of crossover points, enables a large space to be sampled. On average, more structures are required to be sampled to find the GM when using lower elitism. This is consistent

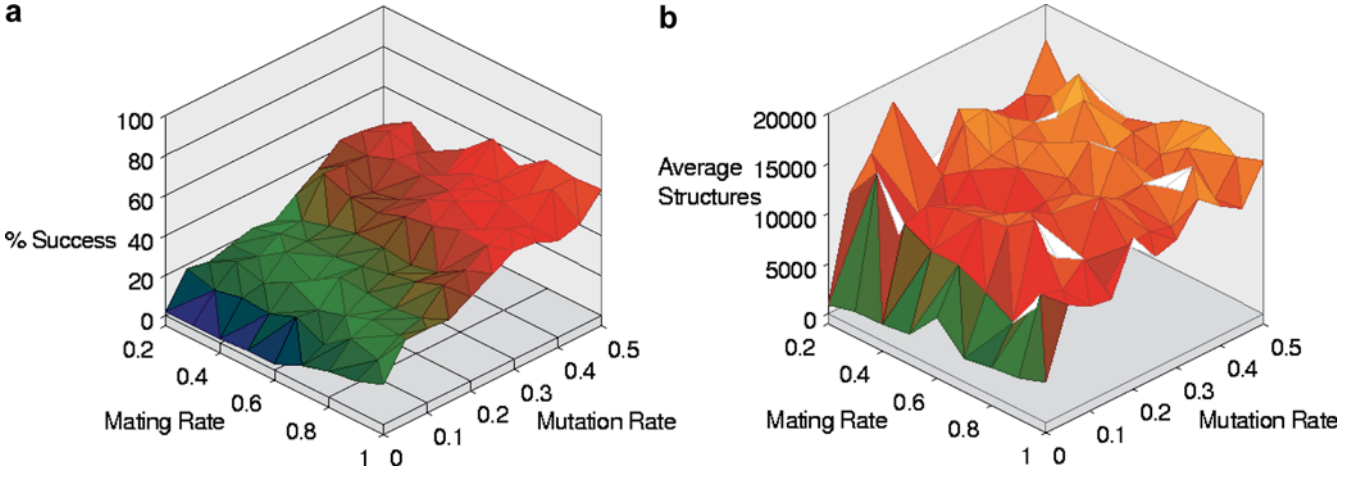


Fig. 5. 3D plots of **a** the percentage success and **b** the average number of structures sampled, as a function of the mating and mutation rates (with an elitism of 30%) for sequence HP-24, using the standard GA

with the GA being less focused on good solutions, and, hence, being affected more by poor mutants or offspring.

The 3D plots in Fig. 5, taken with these other findings, suggest that a good strategy would be to adopt a high rate of mating and mutation, resulting in fast evolution, while using relatively high elitism (30%). This should lead to higher success rates for fewer sampled structures.

3.2 Comparison of the standard GA with random searching

In order to assess the efficiency and scalability of our standard GA program, the success of the GA was compared with what would be expected for a random search of conformation space, for the two smallest benchmark sequences, HP-20 and HP-24. On the basis of the preliminary optimisation discussed previously, for these GA runs we adopted the following GA parameters: mating rate 1.0; mutation rate 0.5; elitism 30%.

For the HP-20 sequence, we performed a systematic grid search which found approximately 4.1890×10^7 valid conformations and four global minima (having energy $E^* = -9$), corresponding to two pairs of enantiomers: one of the global minima is shown in Fig. 6. Thus, if HP-20 conformations are generated at random, we would expect to generate a GM on average once every 1.0472×10^7 conformations. Carrying out 100 GA runs (where the GA stops when a GM is found or when a capping limit of 20,000 structures has been reached) resulted in 56% success. (The capping limit of 20,000 was chosen because preliminary studies showed that 90% of successful runs had found the GM by the time 20,000 structures had been sampled.) In the successful runs, an average of just under 6,000 structures were sampled before finding the GM. Factoring in the unsuccessful runs, a GM structure was found on average every 21,654 conformations studied, representing an improvement by a factor of 484 over a random search.

It should be noted that the HP-20 sequence is end-to-end symmetric (i.e. the sequence of H's and P's is palindromic, as is also the case for HP-24), so the two enantiomeric pairs of GM are in fact equivalent—being related by 90° rotations.

A systematic grid search on the HP-24 sequence found approximately 4.3167×10^9 valid conformations and 38 global minima (having energy $E^* = -9$), corresponding to 19 pairs of enantiomers. Thus, if conformations are generated at random, we would expect to generate a GM on average once every 1.1360×10^8 conformations. Again carrying out 100 GA runs, with a capping limit of 30,000 structures (again corresponding to the limit at which 90% of successful GA runs find a GM conformation) a success rate of 56% was obtained. In the successful runs, an average of just under 11,000 structures were sampled before finding the GM. Factoring in the unsuccessful runs, a GM structure was found on average every 34,504 conformations studied, representing an improvement by a factor of 3292 over random searching.

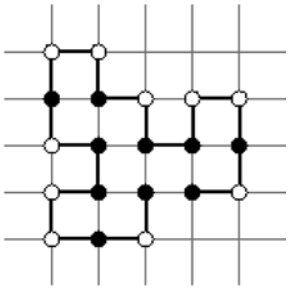


Fig. 6. One of the four global minima (consisting of two enantiomeric pairs) for the HP-20 benchmark sequence. (H beads are shown in *black* and P beads in *white*)

While it is difficult to make absolute comparisons, especially since we are averaging over some runs that did not find a GM, we believe that meaningful statistics can be obtained by capping runs at the number of strutures when approximately 90% of the successful runs have found a GM conformation. The fact that the standard GA is significantly better than statistical (random) is, of course, essential for our future studies, but it is also encouraging that the improvement over random searching increases with increasing sequence length. It is also worth noting that the average number of structures sampled (i.e. the average number of energy evaluations) before finding one of the GM conformations, using our standard GA (approximately 6,000 and 11,000 for HP-20 and HP-24, respectively) are lower than those reported by Unger and Moult [22] in their GA study incorporating local searching (30,492 for HP-20 and 30,491 for HP-24) . (The work of Unger and Moult did show, however, that their GA was more efficient than Monte Carlo searching for finding the GM of HP lattice bead model sequences.) Although our quoted average numbers of function evaluations are for successful GA runs, the success rate of our standard GA is over 50% for both sequences and it should be noted that those reported by Unger and Moult actually represent the best results out of five GA runs. Similarly, our GA outperforms the improved GA of König and Dandekar [35], who incorporated a systematic crossover strategy into their GA. In their study, they required an average of 13,507 energy evaluations to find the GM for the HP-20 sequence.

3.3 The Duplicate Predator

In our preliminary studies, inspection of failed GA runs (where the GM was not found), showed that the diversity (number of unique conformations) within the populations was significantly decreased. This is consistent with high mutation rates yielding higher success rates as the mutation operator will remove some of the duplicates and inject new information into the system.

As already mentioned, an alternative strategy for maintaining population diversity is to introduce a duplicate predator, which only allows a certain number of copies of a particular structure, with any additional copies being "killed" and replaced by new, randomly generated structures.

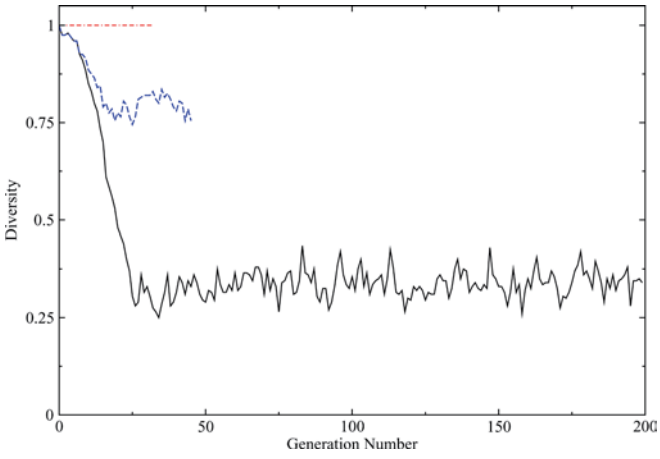


Fig. 7. The effect of the duplicate predator on population diversity during a single GA run for the HP-24 sequence: without duplicate predator (DPL = 0, *solid black line*); DPL = 1 (*dot-dash red line*); DPL = 2 (*dashed blue line*)

Plots of the diversity are shown in Fig. 7 as a function of the generation number in runs of the GA (with and without the duplicate predator) for the HP-24 sequence. (GA parameters: mating rate 1.0; mutation rate 0.5; elitism 30%.) The diversity, d, is defined as

$$d = \frac{N_{\text{unique}}}{N_{\text{pop}}} \quad , \tag{6}$$

where N_{unique} is the number of unique (distinct) structures in the population and N_{pop} is the population size (in this case, $N_{\text{pop}} = 200$). In all cases, the plot terminates when the GM is found. With the duplicate predator off (DPL = 0 — solid black line), the GM is not found after 200 generations and the diversity rapidly falls to approximately 0.25 (i.e. there are only about 50 distinct structures in the population). The dot-dash red line is for extreme predation, where all duplicates are removed (so all members of the population are unique—DPL = 1 and $d = 1$). In this case, the GM is found in 33 generations. When one duplicate is allowed (i.e. DPL = 2, blue dashed line), the GM is found in 46 generations, by which time the diversity has dropped to approximately 0.75 (i.e. there are approximately 150 distinct stuctures in the population). The plots in Fig. 7 are each for single runs of the GA, though with the same random number seed. Inspection of a number of independent runs, however, indicates that these are quite typical results.

The variation in percentage success with the (DPL) for the HP-20, HP-24, HP-25 and HP-36 sequences, where the number of structures sampled per GA run was capped at 60,000, is shown in Table 2. For HP-20, HP-24 and HP-25, where the percentages are relatively high, success is highest (approaching 100% for HP-20 and around 90% for HP-24 and HP-25) when no duplication is allowed (DPL = 1) and falls off as more duplicates are allowed. (It should be noted that DPL = 0 corresponds to no predation and, therefore, the maximum number of identical structures is equal to the population size–which is why the percentage success for DPL = 0 is so low.) The success for the HP-36 sequence is low (typically around 5%) and shows no clear trend with varying DPL.

Table 2. Percentage success obtained for the 20, 24, 25 and 36 bead HP benchmark sequences as a function of the duplicate predator limit (DPL). The GA parameters adopted were those obtained by optimising the GA without duplicate predation: mating rate 1.0; mutation rate 0.5; elitism 30%. The number of structures sampled was capped at 60,000

DPL	HP-20	HP-24	HP-25	HP-36
0	65.0	54.5	24.0	2.5
1	99.5	93.5	86.5	4.0
2	97.5	89.0	71.5	3.5
3	96.0	83.5	64.0	5.5
4	92.0	78.5	55.0	4.5
5	89.5	72.0	56.0	3.0

The GA optimisation tests described in Sect. 3.1 were repeated with the duplicate predator (with DPL = 1), for the HP-24 sequence. Figure 8 shows 3D plots of the GA percentage success and average number of structures sampled (in successful runs) versus the mating and mutation rates, for low (0.5%) and high (30%) elitism. (For a population size of 200, elitism values of 0.5% and 30% correspond to copying the best member or the best 60 members of a given population, respectively, into the next generation.) In most cases, the duplicate predator was found to have a dramatic effect, increasing the success of the GA significantly. This can be seen by comparing the plots in Fig. 8c (where the success is approximately 100% for all mating and mutation rates investigated) and Fig. 5a (where the maximum success is around 60%, for high mating and mutation rates).

By eliminating identical structures, the duplicate predator prevents population stagnation and injects new genetic information into the population (as new random structures replace the predated structures). With new genetic information in the population the mating operator exchanges genetic information between different parents, converging on good solutions. The mutation operator, however, has now become somewhat obsolete, disrupting the offspring and replacing them with poorer structures. (In fact, the duplicate predator can also be considered as a ''kill and replace'' mutation operator.) With the duplicate predator on, Fig. 8 shows that there is a trend towards a decrease in success with increased mutation rates. There are also noticably fewer structures sampled for higher mating rates and lower mutation rates, which is consistent with mutations disrupting the GA while crossover affords the optimal structures by sharing good genetic information between structures.

Figure 8 also shows that percentage success values are higher and that fewer structures need to be sampled for higher elitism values (30%) than for lower values (0.5%). In fact, as mentioned previously, for the higher elitism, the sensitivity to mating and mutation is lost. In particular, the disruption caused by the high mutation rates can be compensated for by high elitism rates which will disregard the poor mutants and reinsert the good parents back into the subsequent population. However, the increased number of structures sampled by the poor mutations is undesirable.

From these studies, we can conclude that, with the duplicate predator on (and DPL = 1), higher success

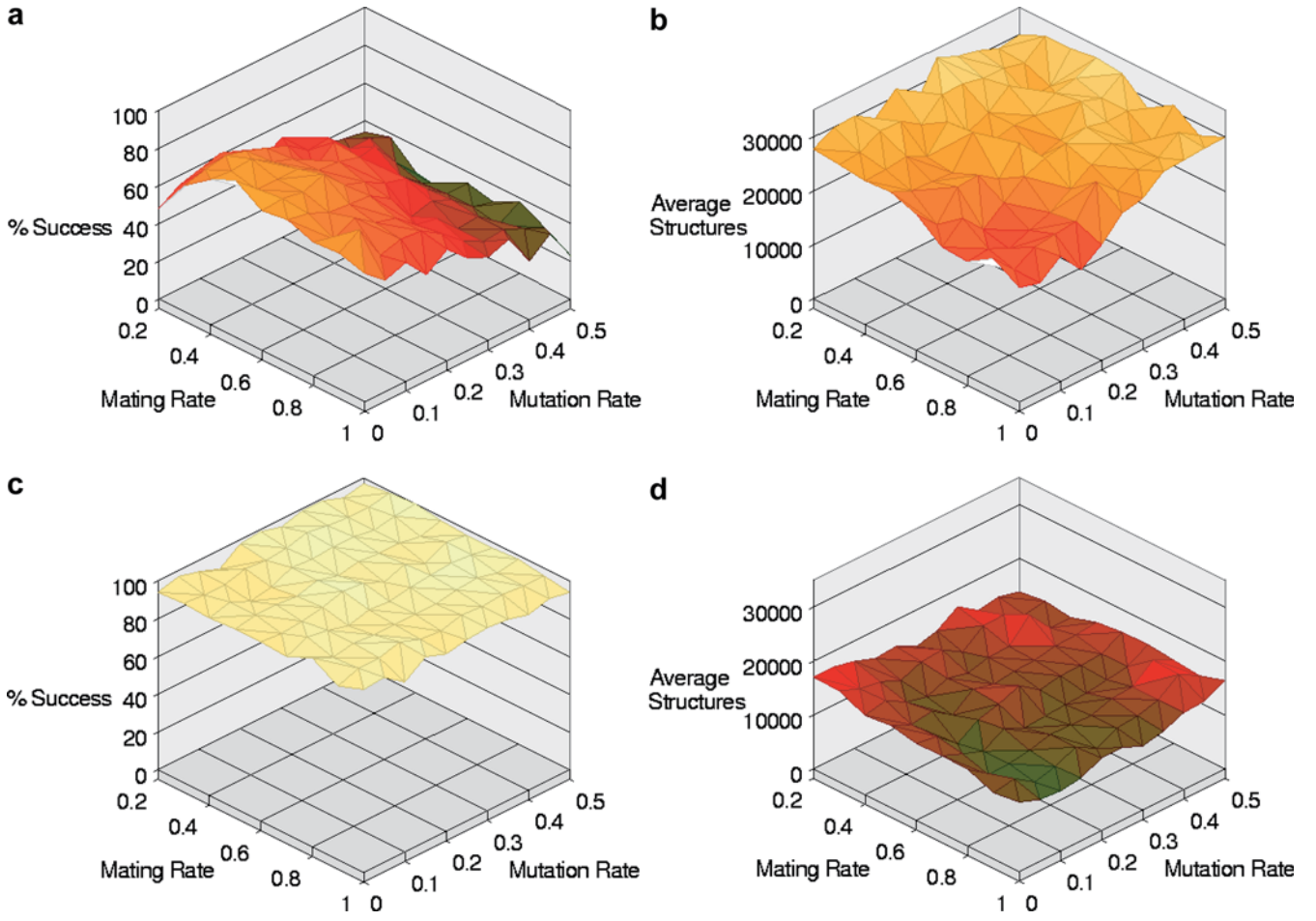


Fig. 8. 3D plots showing the percentage success and average number of structures sampled, as a function of the mating and mutation rates, for the HP-24 sequence, with the duplicate predator on (DPL = 1): **a** percentage success, elitism 0.5%, **b** average structures, elitism 0.5%, **c** percentage success, elitism 30% and **d** average structures, elitism 30%

values are attained than in the absence of predation, often with fewer structures needing to be evaluated. High percentage success is generally favoured by relatively high mating and elitism rates and low mutation rates. In subsequent studies, unless otherwise stated, the following values were adopted: mating rate 1.0; mutation rate 0.1; elitism 30%; DPL = 1.

3.4 Brood selection

We investigated the effect of brood selection (as defined in Sect. 2.3.1) on the percentage success and number of structures sampled (for successful runs) of the GA. Owing to the high success rates achieved using the duplicate predator for the HP-24 sequence, it would be difficult to evaluate the benefit of brood selection for this sequence. Instead, the longer HP-36 sequence was chosen for this investigation. The percentage success values achieved and the average numbers of structures sampled are shown in Fig. 9 as a function of brood size for parents included in the brood and for parents excluded from the brood. Two hundered GA runs were carried out for each brood size, for both parents included and parents excluded from the brood. It should be noted that a brood size of 2, with parents excluded, corresponds to normal crossover (two offspring produced from two parents). However a brood size of 2 with parents included would not make sense, since both parents would pass unchanged into the next generation, without crossover. For this reason, in these brood selection studies, the results for a brood size of 2 (both for parents included and for parents excluded) actually correspond to normal crossover (producing two offspring, which then pass into the offspring population), which is why the percentage success values and average numbers of structures sampled are identical in Fig. 9a and b.

Both graphs in Fig. 9 show a significant increase in percentage success when brood selection is incorporated in the GA (i.e. when the brood size is greater than 2). The brood selection operator effectively allows the GA to perform a local search by exploring a greater number of combinations of the parents' genes (conformation vectors), which is benefical to the performance of the GA. In both cases, the GA success peaks for a brood size of 5, followed by a gradual drop off for increased brood size. Larger brood sizes result in more structures being sampled per mating operation; hence, capping the GA at sampling 60,000 structures results in fewer generations within the GA run. The evolutionary pressure is not high

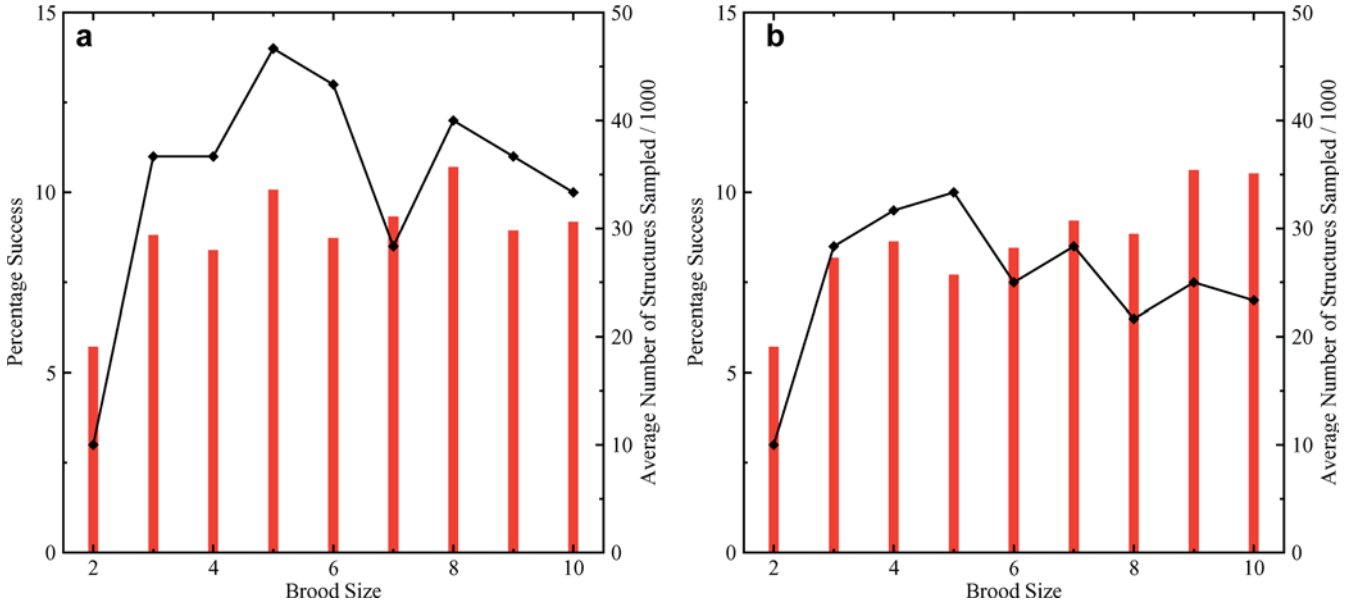


Fig. 9. The percentage success (*lines*) and the average number of stuctures sampled (*bars*), as a function of brood size, for the HP-36 sequence, with the duplicate predator on (DPL = 1): **a** parents included in the brood and **b** parents excluded from the brood. In both **a** and **b**, a brood size of 2 corresponds to normal crossover, where the two parents generate two offspring, which then pass into the offspring population

enough, and the absence of clear guiding pathways leads to a degradation in GA performance.

Higher success values (approaching 15% for a brood size of 5, even for the difficult HP-36 sequence) are achieved when the parents are included within the brood—i.e. parents compete with their offspring to be included in the subsequent generation. Poor offspring will be disregarded, lowering the possible disruption to the population, thereby resulting in an effectively higher elitism rate. Combined with the local search effect of brood selection, this appears benefical to the GA performance.

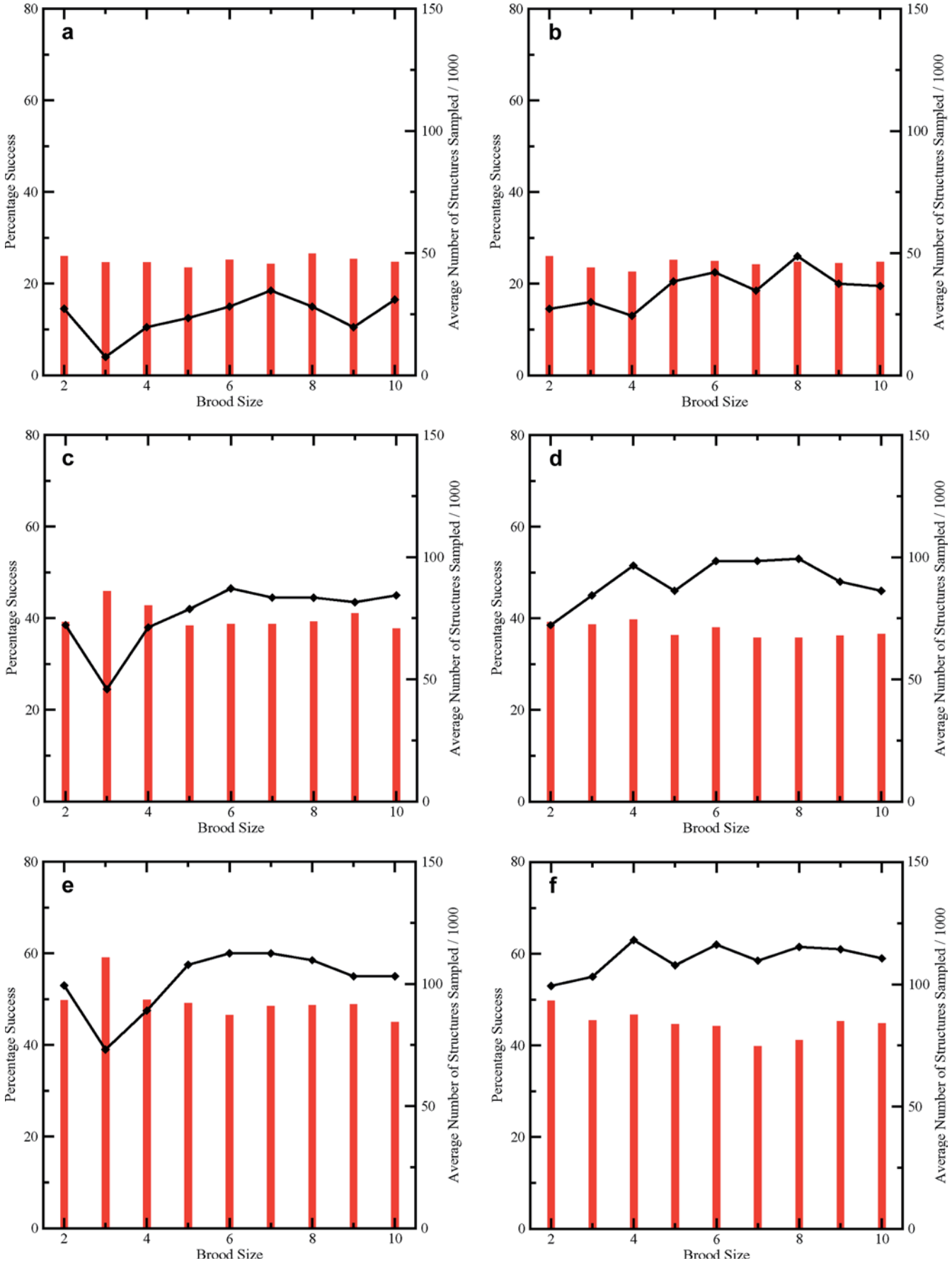
3.5 Local search

To investigate the effects of local search, the previously described brood selection tests were carried out for the HP-36 sequence, with the addition that the local search operator was applied to all population members after sorting and truncation of the population. The number of structures sampled by the GA was capped at 60,000, 120,000 and 180,000 (the other GA parameters were as in the previous study). Again, 200 GA runs were performed for each brood size for both parents included and parents excluded from the brood. Long runs were performed (with the maximum number of generations being 600) and the same runs used to provide the data for capping at 60,000, 120,000 and 180,000 structures. (In all successful cases, the GA run was terminated after the generation in which the GM was found.)

As shown in Fig. 10, applying the local search operator leads to significant improvement of the GA percentage success, with a maximum success of around 60% now achievable and a tendency towards higher percentage success for larger broods. For capping at 60,000 structures, comparison with the previous study (without local search, Fig. 9) shows that there is a small increase in the average number of structures sampled (in successful runs) when local searching is included. As the local searching leads to an increase in structures investigated per generation (there are 30 Monte Carlo steps per local search), this indicates that there is a significant decrease in the average number of generations required to find the GM when local searching is switched on. This is consistent with the work of Shmygelska and Hoos [39], who found that incorporation of a Monte Carlo local search led to a significant improvement in the performance and success of their ant colony optimisation algorithm for protein folding.

The relatively high percentage success values for HP-36, for runs of 120,000 (or fewer) energy function evaluations (i.e. structures sampled), compares well with the earlier GA study by Unger and Moult [22], where the GM for HP-36 was found after over 301,000 energy evaluations . Our GA is also more efficient than the GA (incorporating systematic crossover) of König and Dandekar [35] , who achieved only 4% success in finding the GM for the HP-36 sequence. König and Dandekar capped their GA at 100,000 energy evaluations, but, as shown in Fig. 10, we typically achieve (for brood selection with parents excluded and local searching) 10–20% success for a cutoff of 60,000 total energy evaluations and over 40% success when capping at 120,000 evaluations.

In comparison with brood selection without local searching, Fig. 10 shows that including the parents within the brood now has a slight detrimental effect (compared with excluding them) when local searching is switched on. Allowing parents to populate subsequent populations as offspring increases the elitism and hence the local search is performed on the same structures,


a
b
c
d
e
f
Percentage Success
Average Number of Structures Sampled / 1000
Brood Size
0
20
40
60
80
0
50
100
150
2
4
6
8
10

◄

Fig. 10a–f. The percentage success (*lines*) and the average number of stuctures sampled (*bars*), as a function of brood size, for the HP-36 sequence, with the duplicate predator on (DPL = 1) and local searching. **a** (GA capped at) 60,000 structures, parents included (in the brood), **b** 60,000 structures, parents excluded, **c** 120,000 structures, parents included, **d** 120,000 structures, parents excluded, **e** 180,000 structures, parents included and **f** 180,000 structures, parents excluded

wasting resources. Ensuring new offspring maximises the population diversity and ensures that the local search operator samples new space, thereby maximising the operator's effectiveness.

For larger GA capping values (i.e. allowing the GA to sample more structures), the percentage success increases; however, the large degree of local searching swamps the effects of brood selection and the variation in success with different brood size and brood selection schemes becomes less apparant. When the GA is allowed to sample twice and three times as many structures, the percentage success increases; however, it should be noted that on going from capping at 120,000 to 180,000 structures, the average number of structures sampled increases only slightly. This is because the probability of the GA locating the global minimum decreases with increasing generation number, so only a small number of additional successful runs (which will increase the average number of structures sampled) are added on increasing the capping limit.

The distribution of the generation in which the GM for HP-36 was found in the previously decribed brood selection runs (200 runs per brood size — for brood sizes of 2–10 — for both parents included and parents excluded, with the run finishing when the GM was found or when a limit of 600 generations was reached) is shown in Fig 11. With a limit of 600 generations, the GM was found 2592 times out of 3400 attempts (i.e. 200×17), corresponding to an overall success of 76.2%, with the maximum in the distribution at eight generations. Of the successful runs, 33.7% found the GM in the first ten generations; 84.6% in the first 50 generations and 89% in the first 100 generations. This indicates that for finding global minima it is more efficient to run a large number of short GA runs, rather than a few long runs.

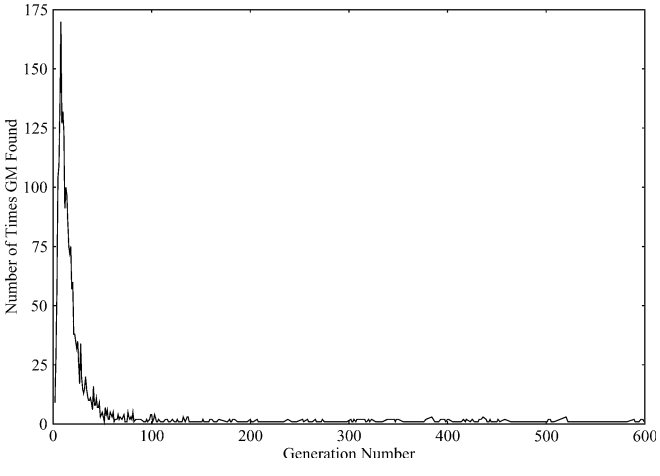


Fig. 11. The distribution of the generation in which the global minima (GM) is found for the HP-36 sequence, from a total of 3400 GA runs (with brood selection), with runs limited to 600 generations

3.6 Analysis of the GM conformations for the HP-36 sequence

As (with the exception of the first two beads, which are fixed, so as to define the coordinate axes) there are three possible positions in which each successive bead can be placed, relative to its predecessors, the total number of conformations for an N-bead sequence in the 2D HP lattice bead model is 3^{N-2}. Although, as discussed in Sect. 2.3, a large proportion of these conformations are invalid, owing to bead overlap, the number of valid conformations still rises rapidly with increasing number of beads. For this reason, no grid search was carried out for the benchmark HP-36 sequence. However, owing to the reasonably high percentage success of our GA, we can say something about the degeneracy of the GM for this sequence.

In our GA runs, we found 383 distinct conformations, all with the same lowest-energy ($E^* = -14$), which is consistent with this being the GM energy, in agreement with previous studies [22, 35, 38, 39, 48]. (Of course, without an exhaustive grid search this cannot be proven to be the lowest-energy possible for this sequence.) These 383 conformations are found to constitute 191 enantiomeric pairs and one odd structure. However, as the mirror image of this odd structure must have the same energy, this just means that we failed to find one enantiomer. We can therefore say that the lower bound on the degeneracy of the GM for the HP-36 sequence is 384 (192 enantiomeric pairs) as it is still possible that we have failed to find other enantiomeric pairs.

The structure of one of the GM conformations for HP-36 is shown in Fig. 12. In fact, all of the GM conformations that we found have a 4×4 square H_{16} hydrophobic core, surrounded by 20 P beads. As noted by Unger and Moult [22], the chain adopts a zig-zag ("helical") conformation. The high GM degeneracy comes about because there is a lot of flexibility afforded

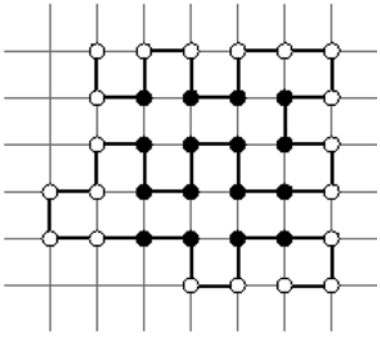


Fig. 12. One of the highly degenerate set of global minima (with energy $E^* = -14$) found for the HP-36 sequence. (H beads are shown in *black* and P beads in *white*)

by the P_3 and P_2 ends of the chain (varying the conformations of these ends does not affect the energy in the HP model), as well as in the two P_4 segments in the chain. The long H_7 segment also allows flexibility within the hydrophobic core, though H-bead flexibility must maintain the same number of H–H contacts if the energy is to remain constant.

3.7 Application of the optimal GA strategy to all benchmark sequences

On the basis of the results presented in previous sections, we have concluded that the optimal strategy for finding the GM for the 2D HP lattice bead model is to incorporate local searching, brood selection (with parents excluded from the brood) and duplicate predation, with a high mating rate, low mutation rate and relatively high elitism. The percentage success (in finding the lowest energies reported in the literature) and the average number of structures sampled are compared in Table 3 in order to find the GM (in successful runs) for all of the benchmark sequences listed in Table 1, using our optimal strategy incorporating local searching, brood selection (brood size 5, parents excluded) and duplicate predation (DPL = 1, mating rate 1.0, mutation rate 0.1, elitism 30%). The total number of function evaluations was not capped, though a limit of 100 generations was imposed. Our average numbers of function evaluations are also compared with those reported (each being the best result obtained from five GA runs) by Unger and Moult [22].

Table 3 shows that our GA is 100% effective in finding the lowest-energy structures previously reported for HP-20, HP-24, HP-25 and even HP-50. For HP-20 and HP-24, the average number of structures sampled in order to find the GM (18,388 and 27,278, respectively) are greater than those obtained (averaging over successful runs) using the standard GA, owing to the inclusion of local searching—but the success rates have gone up from 56% to 100%. Comparison with the results of the standard GA, averaging over all runs, shows that there is actually a decrease in the number of structures sampled for the optimal GA strategy. Table 3 also shows that, in most cases—except for HP-25 and HP-48 (for which they did not find the purported GM, with $E = -23$)— Unger and Moult sampled more structures than in the present study. When considering the results for HP-25, however, it should again be noted that the reported number of structures sampled by Unger and Moult's GA represents the best result from five runs.

It is noticeable from Table 3 that it is more difficult to find the GM for HP-36 and (especially) HP-48 than for the other sequences studied, as reflected in the percentage success values (70% for HP-36 and only 13% for HP-48) and the average numbers of structures sampled in successful runs (over 110,000 for HP-36 and over 260,000 for HP-48). Comparison with Unger and Moult's results confirms the difficulty of finding these global minima, as they required (at best) over 301,000 function evaluations for HP-36 and did not find the GM for HP-48 [22]. As already mentioned, all of the global minima for HP-36 possess a 4×4 square H_{16} core. Similar square hydrophobic cores are found in the GM for HP-25 (3×3 H_9 core) and HP-48 (5×5 H_{25} core). Although we find the GM for HP-25 with 100% efficiency, it is noticeable that the number of structures sampled is significantly higher than for HP-24. It is posssible, therefore, that the lower success rates for HP-36 and HP-48 and the higher number of structures samped for HP-25 are related to the fact that the hydrophobic cores are maximally compact. In these cases, the GA may tend to converge on suboptimal structures, from which it is not easy to find the global minima.

4 Conclusions

In this paper, we have described our GA program for finding the lowest-energy conformations in the 2D HP lattice bead model, for a number of benchmark sequences, up to HP-50. Optimisation of the standard GA showed that the best results were obtained for high rates of mating and mutation and relatively high elitism (30%).

In order to study longer sequences, or to look at more sophisticated protein models, it is important that the GA efficiency is maximised. To this end, we have considered

Table 3. Comparison of the lowest energies found (E) and the efficiency of our GA (employing our optimum GA strategy) with that reported by Unger and Moult [22], for a number of HP benchmark sequences. E values in *bold* are those which are in agreement with the lowest energies reported in the literature. For our GA, we report the percentage success and the (rounded) average number of function evaluations in successful runs of the GA. It should be noted that Unger and Moult did not report their success rate and that their reported number of function evaluations was for the best out of five runs of their GA program

Length	This work			Unger and Moult	
	E	Percentage success	Average number of evaluations	E	Number of evaluations
HP-20	**−9**	100	18,338	**−9**	30,492
HP-24	**−9**	100	27,278	**−9**	30,491
HP-25	**−8**	100	35,128	**−8**	20,400
HP-36	**−14**	70	113,667	**−14**	301,339
HP-48	**−23**	13	261,311	−22	126,547
HP-50	**−21**	100	97,691	**−21**	592,887

a number of new genetic operators. The duplicate predator, which maintains population diversity by eliminating duplicate structures, was found to lead to significantly higher success in finding the GM, often requiring fewer evaluations. With the duplicate predator set to remove all copies, high success was found to be favoured by relatively high mating and elitism rates and low mutation rates.

Brood selection was also introduced and was again found to lead to significant improvement in GA success, owing to more thorough searching of the crossover space of the two parents. The most dramatic increase in the efficiency of the GA, however, was achieved by the addition of a Monte Carlo type local search algorithm, which enables efficient exploration of the local conformation space around population members.

On the basis of the calculations described in this paper, we conclude that the optimum strategy for finding GM for the 2D HP lattice bead model is to incorporate local searching, brood selection (brood size 4–6, with parents excluded from the brood) and duplicate predation (DPL = 1), with a high mating rate (1.0), low mutation rate (0.1) and relatively high elitism (30%). We have found that our GA approach is highly successful in finding the GM for all of the sequences studied, with the exception of HP-48, which appears to be a particularly difficult case. However, for all of the sequences studied, the average number of structures sampled in order to find one of the GM conformations was found to compare favourably with the results of previous GA studies of this model [22, 35].

Research is currently continuing into further improvement of our GA methodology, especially with regard to developing strategies to improve the efficiency of finding global minima for sequences (such as HP-36 and HP-48) with dense hydrophobic cores and for extending our work to longer sequences. Future research will also include the application of the GA to 3D cubic and diamond-type lattices and to more sophisticated protein models. We are also investigating the use of GAs (and other evolutionary computing techniques) to find low-energy folding pathways, so that one can obtain information about folding dynamics, as well as preferred folding conformations.

Acknowledgements. GAC is grateful to the EPSRC for a PhD studentship. TVM-J is grateful to the School of Chemistry, University of Birmingham, for a School studentship, and to Hewlett-Packard Co. (Galway, Ireland) for sponsorship. The authors wish to thank Gareth Rylance for helpful discussions.

References

1. Schulz GE, Schirmer RH (1979) Principles of protein structure Springer, Berlin Heidelberg New York
2. Merz KM (ed.) (1994) The protein folding problem and structure prediction Birkhauser, Boston, MA
3. Anfinsen CB (1973) Science 181: 223
4. Baldwin RL, Rose GD (1999) Trends Biochem Sci 24: 26
5. Levinthal C (1969) In: DeBrunner JTP, Munck E (eds), Mössbauer spectroscopy in biological systems (Proceedings of a meeting held at Allerton House, Monticello, Illinois) University of Illinois Press, Illinois, pp. 22–24
6. Wolynes PG, Onuchic JN, Thirumalai D (1995) Science 267: 1619
7. Dinner AR, Sali A, Smith LJ, Dobson CM, Karplus M (2000) Trends Biochem Sci 25: 331
8. Lau KF, Dill KA (1989) Macromolecules 22: 3986
9. Lau KF, Dill KA (1990) Proc Natl Acad Sci USA 87: 638
10. Chan HS, Dill KA (1991) J Chem Phys 95: 3775
11. Abkevich VI, Gutin AM, Shaknovich EI (1994) J Chem Phys 101: 6052
12. Unger R, Moult J (1996) J Mol Biol 259: 988
13. Cejtin H, Edler J, Gottlieb A, Helling R, Philbin J, Wingreen N, Tang C (2002) J Chem Phys 116: 352
14. Ping G, Yuan JM, Vallieres M, Dong H, Sun Z, Wei Y, Li FY, Lin SH (2003) J Chem Phys 118: 8042
15. Miller DW, Dill KA (1997) Protein Sci 6: 2166
16. Hirst JD (1999) Protein Eng 12: 721
17. Blackburne BP, Hirst JD (2001) J Chem Phys 115: 1935
18. Leach AR (1969) Molecular modelling: principles and applications. Addison-Wesley-Longman, Harlow.
19. Vekhter B, Berry RS (1999) J Chem Phys 111: 3753
20. Miller MA, Wales DJ (1999) J Chem Phys 111: 6610
21. Mukherjee A, Bagchi B (2003) J Chem Phys 118: 4733
22. Unger R, Moult J (1993) J Mol Biol 231: 75
23. Nayak A, Sinclair A, Zwick U (1999) J Comp Biol 6: 13
24. Li Z, Scheraga HA (1987) Proc Natl Acad Sci USA 84: 6611
25. O'Toole EM, Panagiotopoulos AZ (1992) J Chem Phys 97: 8644
26. Ramakrishnan R, Ramachandran B, Pekny JF (1997) J Chem Phys 106: 2418
27. Frauenkron H, Bastolla U, Gerstner E, Grassberger P, Nadler W (1998) Phys Rev Lett 80: 3149
28. Liang F, Wong WH (2001) J Chem Phys 115: 3374
29. Beutler T, Dill K (1996) Protein Sci 5: 2037
30. Gan HH, Tropsha A, Schlick T (2000) J Chem Phys 113, 5511
31. Zhang JL, Lu JS (2003) J Chem Phys 117: 3492
32. Kirkpatrick S, Gellatt CD Jr, Vecchi MP (1983) Science 220: 671
33. Krasnogor N, Pelta DA, Martinez Lopez PE, de la Canal E (1997) Proceedings of engineering of intelligent systems 98 and references therein
34. Krasnogor N, Hart WE, Smith J, Pelta DA (1999) Proceedings of the 1999 international genetic and evolutionary computation conference (GECCO99)
35. König R, Dandekar T (1999) Bio Syst 50: 17
36. Pedersen JT (2000) In: Clark DE (ed) Evolutionary algorithms in molecular design. Wiley-VCH, Weinheim, pp. 223–239 and references therein
37. Unger R (2004) In: Johnston RL (ed) Applications of evolutionary computation in chemistry Structure and bonding, vol 110. Springer, Berlin Heidelberg New York pp 153–175
38. Shmygelska A, Aguirre-Hernández R, Hoos HH (2002) Lect Notes Comput Sci 2463: 40
39. Shmygelska A, Hoos HH (2003) Lect Notes Comput Sci 2671: 400
40. Curley BC, Mortimer-Jones TV, Cox GA, Johnston RL manuscript in preparation
41. Holland J (1975) Adaptation in natural and artificial systems. University of Michigan Press, Ann Arbor, MI
42. Goldberg DE (1989) Genetic algorithms in search, optimization and machine learning. Addison-Wesley, Reading, MA
43. Judson RS, Colvin ME, Meza JC, Huffer A, Guttierrez D (1992) Int J Quantum Chem 44: 277
44. Pedersen JT, Moult J (1997) J Mol Biol 269: 249
45. Jones DT (1994) Protein Sci 3: 567
46. Johnston RL (2003) Dalton Trans 4193, and references therein
47. Harris KDM, Johnston RL, Habershon S (2004) In: Applications of evolutionary computation in chemistry, structure and Bonding vol. 110 (Roy L. Johnston (Ed.), Springer, Berlin Heidelberg New York, pp. 55–94
48. Hart WE, Istrail S HP Benchmarks http://www.cs.sandia.gov/tech–reports/compbio/tortilla–hp–benchmarks. html

49. Rosenbluth MN, Rosenbluth AV (1955) J Chem Phys 23: 356
50. Grassberger P (1997) Phys Rev E 56: 3682
51. Manby FR, Johnston RL, Roberts C (1998) Commun Math Comput Chem 38: 111
52. Altenberg L (1994) In: Kinnear K (ed) Advances in genetic programming. MIT Press, Cambridge
53. Turner GW, Tedesco E, Harris KDM, Johnston RL, Kariuki BM (2000) Chem Phys Lett 321: 183
54. Tuson A, Clark DE (2000) In: Clark DE (ed) Evolutionary algorithms in molecular design. Wiley-VCH, Weinheim, pp 241–264