Journal of Computer-Aided Molecular Design, 10 (1996) 337–358
ESCOM

J-CAMD 359

Evolutionary algorithms in computer-aided molecular design

David E. Clark* and David R. Westhead**

Proteus Molecular Design Ltd., Proteus House, Lyme Green Business Park, Macclesfield SK11 0JL, U.K.

Received 22 May 1996
Accepted 12 June 1996

Keywords: Genetic algorithms; Evolutionary programming; Evolution strategies; Drug design; Molecular modelling; Protein folding

Summary

In recent years, search and optimisation algorithms inspired by evolutionary processes have been applied with marked success to a wide variety of problems in diverse fields of study. In this review, we survey the growing application of these 'evolutionary algorithms' in one such area: computer-aided molecular design. In the course of the review, we seek to summarise the work to date and to indicate where evolutionary algorithms have met with success and where they have not fared so well. In addition to this, we also attempt to discern some future trends in both the basic research concerning these algorithms and their application to the elucidation, design and modelling of chemical and biochemical structures.

Introduction

Many of the problems encountered in molecular design are inherently 'hard' from a computational complexity point of view. Thus, there is much interest in the development of effective, heuristic algorithms for search and optimisation. Evolutionary algorithms (EAs) represent an emerging class of algorithms which are proving able to provide optimal, or near-optimal, solutions to a wide range of challenging problems in a variety of disciplines. It is the intent of this review to survey the applications of evolutionary algorithms in the field of computer-aided molecular design (CAMD). (Note that we use CAMD in a very broad sense to include any application of computational techniques in the elucidation, design and modelling of molecular structures.) However, before examining the applications in detail, we shall first provide a brief introduction to the subject of evolutionary algorithms.

Evolutionary algorithms

As the name suggests, evolutionary algorithms are inspired by the mechanisms observed in evolutionary processes in the natural world. The idea of attempting to mimic natural processes in problem solving seems to have been aired first by Box in 1957 [1]. Since that time, three broad classes of EA have emerged: genetic algorithms (GAs), evolutionary programming (EP) and evolution strategies (ES) [2]. In what follows, we shall briefly discuss each of them in turn. For more details, the reader is referred to the excellent reviews of Bäck and Schwefel [2] and Fogel [3].

Genetic algorithms

Perhaps the most popular and well-known class of EA in use at the present time is that of genetic algorithms. The conception and early development of GAs is gen-

*To whom correspondence should be addressed at: Dagenham Research Centre, Rhône-Poulenc Rorer Ltd., Rainham Road South, Dagenham RM10 7XS, U.K.

**Present address: EMBL Outstation, European Bioinformatics Institute, Hinxton Hall, Hinxton CB10 1RQ, U.K.

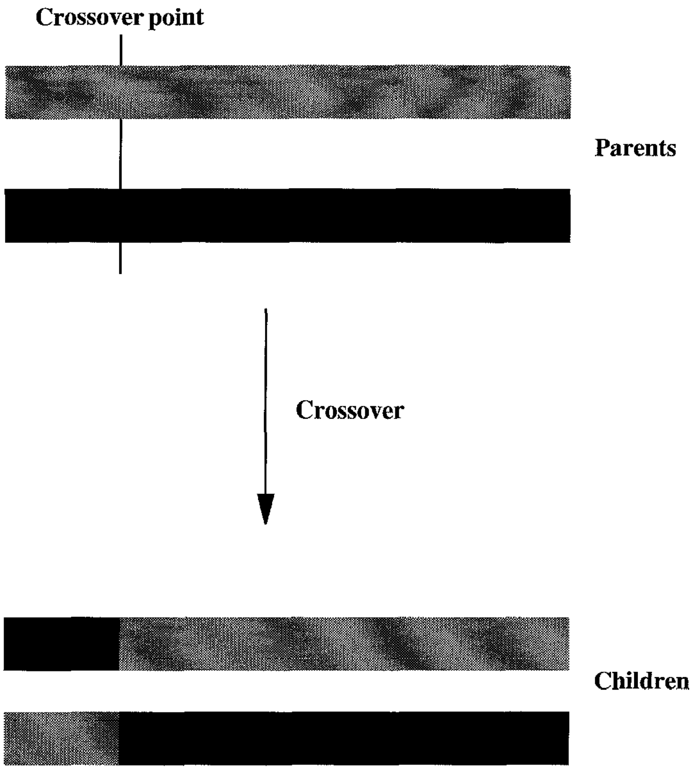

Fig. 1. Illustration of one-point crossover in a GA.

erally attributed to Holland [4,5], although similar ideas were also propounded by Fraser [6].

The simplest GA works with a *population* of individuals, each of which represents an attempt at solving the problem in hand. In general, these attempted solutions will be encoded in a string of variables or *chromosome*. Traditionally, this has taken the form of a bit vector, but real-valued strings may also be used. The 'goodness' of each of these attempted solutions is assessed by a *fitness function* which assigns each member of the population a score or *fitness*. The GA operates by applying *genetic operators* to this population; the most common operators being *selection*, *crossover* and *mutation*.

The purpose of the selection operator is to pick from the population a pair of parents from which further solutions can be 'bred'. In general, selection involves some kind of 'survival-of-the-fittest' mechanism whereby parents are selected with a probability which is proportional to their fitnesses. A popular way of achieving this kind of selection is via a roulette-wheel algorithm [7].

The emphasis laid on the crossover or *recombination* operator is often said to be the feature that distinguishes GAs from other EAs and the utility of crossover is hotly debated by protagonists of GAs and other types of EA, particularly EP [8]. Crossover is the means by which genetic material is passed from the parents to the child solutions and involves the cutting of the parents' chromosomes followed by the recombination of the resulting fragments. Various schemes for crossover have been devised; the simplest is illustrated in Fig. 1 (one point crossover). The result of crossover is that there are two child solutions, each of which contains genetic material from each parent.

The mutation operator simply perturbs a solution's chromosome, perhaps by flipping a bit at random (binary representation) or adding a Gaussian random number

```
1. Randomly initialise a starting population of N vectors
2. Score each population member using a fitness function
3. Select parents according to fitness
4. Breed children by crossover and/or mutation
5. Score children using the fitness function
6. Replace parents by children if children are fitter
7. Go to 3 until termination or convergence
```

Fig. 2. A simple genetic algorithm.

```
1. Randomly initialise a starting population of N vectors
2. Score each population member using a fitness function
3. Breed one child from each parent by mutation
4. Score N children using a fitness function
5. Carry out tournaments among 2N parents + children
6. N members with most wins form new population
7. Go to 3 until termination or convergence
```

Fig. 3. A simple evolutionary programming algorithm.

(real-valued strings). In GAs, mutation is generally invoked with a low frequency and, by analogy with Nature, the effect is usually pathological. However, in some cases, mutation will serve to 'kick' a solution out of a particular local minimum and to increase the diversity of the population.

The repeated application of these operators to a dynamic population of solutions results in the average fitness of the population increasing over time until eventually, the population converges on what is hopefully a (near) optimal solution. Figure 2 shows the sequence of operations in a typical, simple GA. For more details about GAs, the interested reader is referred to Refs. 5, 7, and 9–12. For reviews with a more chemical/chemometric slant, the following works should be consulted: Refs. 13–16. Lucasius and Kateman have also described a software toolkit facilitating GA applications in computational chemistry [17,18].

Evolutionary programming

Evolutionary programming algorithms stem from early research by Lawrence Fogel [19]. After a hiatus of some 10 years, interest in this class of EA resumed in the late 1980s spearheaded largely by David Fogel [3]. Although currently EP has been less widely applied in CAMD than GAs, early results in some areas are promising, as will be detailed later.

As mentioned in the section above, EP algorithms differ from GAs in that they do not employ a crossover operator. In general, EP algorithms operate upon continuous, real-valued strings and children are bred from parents by mutation alone. This mutation generally involves the addition of a randomly generated real number to each element of the string. This random number is traditionally taken from a Gaussian distribution, although the Cauchy distribution has also been used to good effect [20].

When one or more children have been bred from each parent string in this manner, the parents and children compete together for survival into the next generation. This selection process is often carried out by means of *tournaments* in which each individual is compared to a number of randomly selected 'opponents'. If the individual's fitness exceeds an opponent's, the individual receives a 'win'. At the end of this process, the children and parents are ranked according to the number of wins they have achieved and a number is selected from the top of the ranking to form the parent population for the next generation. Here again, as in a GA, a survival-of-the-fittest mechanism drives the evolving population towards better and better solutions. The selection pressure in EP is increased by increasing the number of solutions in each tournament. This drives the algorithm to better solutions more quickly, but can in some cases lead to premature convergence at a local optimum. Decreasing the number of solutions in each tournament allows the algorithm to retain poor solutions for more generations and improves the searching properties of the algorithm. In practical applications it is important to strike a balance between these two effects.

A simple EP algorithm is given in Fig. 3. Various refinements to this are possible. In particular, the generation of multiple offspring for each parent can be useful and the use of self-adaptive mutation parameters has been shown to be particularly effective [21]. For more details about EP algorithms, the reader is referred to Refs. 2, 3, 22, and 23.

```
1. Randomly initialise a starting population of N vectors
2. Score each population member using a fitness function
3. Breed M children from N parents by mutation/recombination
4. Score M children using a fitness function
5. Rank M+N parents + children by fitness
6. Best N members form new population
7. Go to 3 until termination or convergence
```

Fig. 4. A simple evolution strategy.

Evolution strategies

The class of EA that have become known as evolution strategies were initially developed by Rechenburg and Schwefel, who have collaborated in this field since the 1960s. Although developed independently from EP, ES are very similar. The main differences are firstly, that ES may employ various *recombination* operators (such as crossover) in addition to the primary mutation operator and secondly, that ES use a deterministic ranking procedure for selection rather than the probabilistic tournament selection in EP [2]. As with EP, there have been few published applications of ES in molecular design and modelling, but there is no reason to suspect that they should not be as successful as GAs.

An outline of an ES algorithm is given in Fig. 4. Various developments of this basic strategy have been experimented with, particularly the use of multiple parents and offspring. Two approaches have been investigated recently, which have been termed $(\mu+\lambda)$-ES and (μ,λ)-ES, where $1 \leq \mu < \lambda$ [2,3]. In $(\mu+\lambda)$-ES, μ parents create λ children and the next generation is selected from the *union* of these solutions. By contrast, in (μ,λ)-ES, only the λ children compete for survival and the μ parents are replaced at every generation [3]. This latter strategy seems to be the most favourable [2]. Further information about ES can be found in Refs. 2, 3, and 23.

Applications in computer-aided molecular design

The last five years or so have seen a growing number of applications of EAs in CAMD. As hinted above, these have mostly involved GAs, although EP applications are becoming more popular. In what follows, we shall attempt to survey these applications comprehensively. Our intent is to provide an overview of the many application areas and hopefully to whet the appetite of the reader for the field of EAs, thereby stimulating further applications in the CAMD field. Attempting a comprehensive review inevitably leads to something of an imbalance between the length of those sections covering research areas which have attracted many EA applications (e.g., protein folding, molecular docking) and those where perhaps only a single EA approach has been attempted at the present time (e.g., pseudoreceptor modelling). This imbalance should not be interpreted as a bias on the part of the authors, but simply as a reflection of the volume of published EA applications in the respective areas. Other reviews of selected GA applications in molecular recognition, design and modelling have recently been compiled by Willett [24] and by Devillers [25].

Conformational search of small to medium-sized molecules

Conformational search/analysis involves the location and characterisation of the energy minima upon the conformational energy hypersurface for the molecule under study. This type of procedure is of importance for both small and large molecules. In the former case, it is likely that any bioactive conformation will lie somewhere in the vicinity of one of the low-energy minima, while in the latter case, the lowest possible minimum (the so-called 'global energy minimum') may well correspond to the native state of the macromolecule. EAs have been used for the purpose of conformational search/analysis of both small and large molecules. In this section, we describe EA applications in the field of small molecules and in the following section, macromolecular applications including protein folding will be considered.

Probably the most thorough study of the application of GAs to the conformational search of small-to-medium-sized organic molecules was undertaken by Judson et al. [26] as part of a series of investigations into different conformational searching techniques [27–29]. In this study, a set of 72 molecules having 1–12 rotatable bonds were chosen from the Cambridge Structural Database [30]. To perform a conformational search on a given molecule, the appropriate torsion angles were encoded as a binary string. Six bits were allocated to each torsion angle, giving a search resolution of about 5°. A population size of 10 times the number of torsion angles was used. The fitness of each individual in the population (i.e., each conformation) was assessed using the Batchmin implementation of the MM2 forcefield [31]. Two methods of fitness calculation were used: a simple single-point energy calculation and a short gradient optimisation of the conformational energy. For the molecules under study, the former was found to be as effective as the latter in locating low-energy conformers and was significantly more computationally efficient. For the majority of test molecules, the GA was able to locate conformations having energies equal to or lower than the energy of the minimised crystal structure. Also, for molecules with more than eight rotatable bonds, the GA was found to be more computationally efficient than the constrained systematic search method (CSEARCH) implemented within the SYBYL molecular modelling package [32]. A GA was also found to outperform CSEARCH by Clark et al. in a comparison of conformational search algorithms used in conformationally flexible 3D database searching [33]. However, in this instance, the GA was itself outdone by a method known as Directed Tweak [34].

Brodmeier and Pretsch used a GA in combination with an extended version of the MM2 forcefield to carry out conformational search of two alkanes: *n*-decane and 3-methyl-nonane [35]. In both cases, the GA was shown to be capable of locating low-energy conformers more efficiently than two search techniques implemented in commercial software [32]. The study of Brodmeier and Pretsch also emphasised the importance of maintaining diversity in the GA population by means of the mutation operator and a technique known as 'sharing' [35].

A combination of a GA and classical molecular dynamics has proved able to predict successfully the structure of C_{60} (buckminsterfullerene) [36,37]. Starting from an initial population, models are cross-bred by dividing two parents using a randomly chosen plane through the centre of gravity. The top half of one model is then combined with the other, and vice versa, to generate the child models. The children are then subjected to conventional molecular dynamics before selection takes place once more. This scheme proved able to locate the global minimum energy for buckminsterfullerene after 5500 crossover operations. Apparently, conventional MD simulations alone have been unable to predict this structure correctly [37].

Peptide molecules have also been successfully treated by GA conformational search methods. Herrmann and Suhai recently reported the application of a GA method to the energy minimisation of some peptide analogues [38]. In this case, the energy evaluation was carried out using the AM1 Hamiltonian embodied within the MOPAC program [39]. For four simple dipeptide systems, the GA was able to locate the global minimum-energy conformation. For the more complex tetra-alanine case, the global minimum was also located, together with a structure with a lower energy than the reported global minimum-energy conformation.

Somewhat larger peptides have also proved amenable to the GA approach. McGarrah and Judson studied cyclic hexaglycine – a molecule containing 24 rotatable bonds [40]. They suggested that the lowest energy conformation found by the GA was likely to coincide with the global energy minimum for the forcefield in use. Their study attempted to explore how best to tune a GA for the problem of global conformational search. The main conclusion reached was that maintaining diversity in the evolving population was a prerequisite for generating superior individuals. Furthermore, they suggested that using a large population with a smaller number of generations is likely to be more useful than the converse. Finally, when dealing with larger molecules, McGarrah and Judson indicated that frequent gradient optimisation of the conformational energy during the GA search gives a significant advantage over methods which allow the GA alone to optimise the energy. This is in contrast to the work on smaller molecules mentioned above [26].

Finally, the use of a GA to help characterise a solution conformation of a cyclic Arg-Gly-Asp (RGD) peptide analogue has been reported by Sanderson and co-workers [41]. Constraints for the conformational search of the peptide (cyclo-(*S,S*)-2-mercaptobenzoate-arginine-glycine-aspartate-2-mercaptoanilide) were obtained from a 2D ROESY NMR spectroscopy experiment carried out at 600 MHz. A GA developed by Payne and Glen [42] was used to generate low-energy conformations of the peptide satisfying the experimental interproton distance constraints. It was noted that the conformations resulting from the GA approach were 'remarkably similar to those generated from ^{1}H NMR data by the more traditional techniques of distance geometry and molecular dynamics' [41].

Conformational search of macromolecules

DNA

Some of the earliest published work on the application of GAs to conformational analysis was that of Lucasius and co-workers [43]. They described the use of their DENISE program for the generation of DNA conformations fitting experimental NMR data. DENISE employs a recursive strategy in which separate GAs are used for distinct fragments of DNA and the results are later combined and refined by a further GA. The results obtained suggested that the GA was superior to the traditional techniques of restrained molecular dynamics and distance geometry for the generation of conformations satisfying NMR-derived constraints. DENISE was subsequently successfully applied to the determination of the solution structure of a photonucleotide dimer (*cis,syn*-dUp[ ]dT) [44].

RNA

As with all molecules, the understanding of the function of RNAs is greatly facilitated if the relevant 3D structure is available. However, at present, X-ray crystallography and NMR are only powerful enough to provide detailed structural information about small RNA fragments and so workers in the field have resorted to computer modelling based upon such experimental data as are available [45]. Recently, several papers describing the application of genetic algorithms to RNA modelling have been published.

Ogata et al. have reported the development and application of a GA for modelling RNA stem-loops [45]. The RNA conformation is determined by the conformation of its component nucleotides and in the work of Ogata et al., each of these component nucleotides is represented by seven variables: six torsion angles and a pseudorotational phase angle describing sugar puckering. The fitness function consists of three terms: one measuring the degree of satisfaction of known distance constraints, the second measuring repulsion between adjacent nucleotides and the final term constrains the conformation of each nucleotide to be similar to a predefined template. At the end of the GA run, the resulting RNA models are subject to Newton–Raphson minimisation to relieve any steric clashes. The GA was applied to the modelling of two loops for which crystal structure information is available: the anticodon arm and T-arm of tRNA^{Phe}. Using noncrystallographic experimental information to provide constraints, the GA was able to model the anticodon loop to within

1.81 Å rms of the crystal structure and the T-loop to within 1.76 Å rms [45]. At present, the method is limited to loops with less than 20 nucleotides, but it is possible that if more long-range structural constraint information becomes available, then this limit could be exceeded.

The folding of RNA has been studied using GA methods by Gultyaev and co-workers [46–48]. RNA structures are represented as sets of stems which are subject to selection, mutation and crossover operations. Of particular interest is the mutation operation which, by the consecutive removal of a stem and formation of a new stem, attempts to simulate the disruption of intermediates in the folding pathway and the formation of more stable structures. The fitness function guiding the folding process seeks to reflect the kinetic character of stem formation. In RNA structure prediction tests, the GA yielded models that were in greater agreement with phylogenetic data than those resulting from a free-energy minimisation approach [46]. More recently, the GA has been used to simulate the folding pathways for the ColE1 RNA II and insight into experimental observations was gained [47].

Other work using GAs in the field of RNA modelling has been reported by Benedetti and Morosetti [49,50].

Proteins

In recent years evolutionary algorithms have become very popular tools for conformational search studies of proteins. This is largely because of the importance of the protein folding problem, that is, the prediction of the three-dimensional (tertiary) structure of a protein using only the identity of the amino acid residues in the sequence. In its most general form, this problem involves optimisation of a potential energy function over the whole conformational space of the protein chain. This space grows exponentially in size with the number of residues in the sequence, and it is known that the associated optimisation problem is NP-hard for most of the potential energy functions in current use [51]. Here we focus on the application of evolutionary algorithms in protein folding from optimisation over the full unrestricted conformational space of the protein, to simple optimisation of protein side-chain conformations with the protein backbone fixed. Up-to-date reviews of the field which are not restricted to evolutionary algorithms have been prepared by Böhm [52] and also by Scheraga [53]. For a survey of GA applications in protein structure prediction, the reader is referred to the recent review of Pedersen and Moult [54].

One of the earliest applications of evolutionary algorithms in protein folding was due to Dandekar and Argos [55]. In this paper, the potential of a GA to fold a four-stranded β-bundle was investigated. A simple model of the protein, as point amino acids occupying positions on a tetrahedral lattice, was employed. The potential function used contained a term favouring the extended β-strand structure, a term reflecting the tendency of the strands to pack together, and a penalty term for atomic overlaps. The most successful folding was found when the packing tendency of strands was modelled using a potential favouring globular structure by penalising residues a long way from the overall centre of mass.

This early work was the beginning of a series of papers by the same group [55–58], all of which apply GAs to protein-folding problems. In the second paper [56], the method was developed further. The lattice-based approach was abandoned in favour of a complete heavy-atom model of the protein, in which the backbone of each amino acid on the chain was allowed to take on one of seven predefined conformations. The potential energy function was also extended and optimised by studying a sequence designed to form an ideal 4-helix bundle. Results were presented for ab initio folding of three 4-helix-bundle proteins (cytochrome b562, cytochrome c' and hemerythrin) and a small mixed $\alpha + \beta$ protein (crambin). Results for the 4-helix bundle proteins were encouraging; each predicted a fold having the correct main-chain fold and a C^{α} rms deviation from the crystal structure of about 6 Å. Results for crambin were of lower quality. Later papers [57,58] made further developments to the potential function to permit more accurate folding of proteins rich in β-sheet-type structure.

The work by Judson [59,60] is another early application of evolutionary algorithms in protein folding. Here, a very simple two-dimensional model of a protein was used, in which a chain of point-like amino acid residues interact through local (i.e., acting between residues adjacent to each other in the *sequence*) and nonlocal potentials. The author introduced the concept of a ‘folding program’, in which the chain is folded by sequentially bringing together pairs of residues using an additional harmonic term in the energy function and a conjugate gradient minimisation algorithm. The folding program can be considered as analogous to the folding pathway of the protein. The fitness of a folding program was derived from the energy of the final conformation of the protein which it produced, and a GA was used to find folding programs which lead to the global optimum of the conformational energy of the protein. Success was demonstrated for a 19-residue chain for which the global optimum conformation is known. In the latter paper [60], extensive comparisons between the GA and other global optimisation algorithms (simulated annealing and random search) were made. In this work, systems of varying sizes were investigated (19, 37 and 61 residues), and each algorithm was studied in a hybrid form with several local optimisation algorithms.

Like Judson, Unger and Moult [61] used an evolutionary algorithm to fold a simple two-dimensional model of a protein. In contrast to Judson, these authors considered a heteropolymer of two types of residues: hydrophobic and hydrophilic. These residues interact through a very simple potential which gives a fixed negative contribution to each direct contact of nonbonded hydrophobic residues. A chain of 20 residues on a square lattice was studied, this system being simple enough to permit full enumeration of the conformational space and rigorous assessment of algorithms. The authors compared a Monte Carlo simulated-annealing algorithm with a hybrid algorithm, in which the GA operation of crossover was applied to conformations from a population of solutions evolving as in the Monte Carlo procedure. The hybrid algorithm was found to be far superior to the simple Monte Carlo procedure. It is generally believed, because of the complexity of the problem, that successful protein-folding algorithms need to mimic the physical process of protein folding, i.e., in some way follow the pathway by which a real protein folds. The authors argue, using the Schema concept [4], that the GA preserves local favourable structures through the generations, reproducing a pathway in which local regions of the chain fold first, and then assemble into the final structure. This would be consistent with the hierarchical model of protein folding, which has significant support in the literature [62–64].

At about the same time, Sun [65] reported the application of a GA to the folding of real proteins. The reduced model of the protein employed in this work includes heavy atoms on the protein backbone and a united-atom representation of each side chain. The conformation of the protein was represented by a string of pairs of real variables representing the ϕ and ψ dihedral angles of each residue in the chain, and potential energy with both local and nonlocal terms derived from statistical analysis of the database of experimentally determined protein structures was used. The GA used to optimise this potential energy made extensive use of conformational 'dictionaries' to speed the search, by using only previously observed values of ϕ and ψ for each residue. Encouraging results for three small proteins were presented. Mellitin (a 26-residue protein with no tertiary interactions) was folded to 1.66 Å rms from the crystal structure; APP (a 36-residue protein with some tertiary interactions) was folded to 3.93 Å rms; and Apamin, (an 18-residue protein with some tertiary interactions) was folded to 3.46 Å rms from a structure derived from an NMR study. This work was extended in a later paper [66] in which the statistically derived local interaction potential was replaced by one derived from the (ϕ,ψ) energy map for each amino acid under an empirical molecular mechanics forcefield. Improved results were presented for small proteins, along with some attempts to fold larger proteins (cytochrome 256b, ubiquitin and a zinc-finger motif). For the larger proteins the method failed to generate native tertiary structures; the author attributed this problem to deficiencies in the nonlocal interaction term and, to some extent, in the search algorithm. The work of Le Grand and Merz [67,68] represents the first attempt at protein folding using a GA along with a full-atom model of the protein. Here, as in the work of Sun described above, the GA operates on a variable string consisting of the ϕ and ψ angles for each residue. In the first instance, the fitness of the solutions was derived from the AMBER potential energy function [69,70]. This method proved successful in finding energy minima with the correct structure for short peptides with no tertiary interactions; however, for larger proteins results were disappointing. For crambin, the GA found an incorrect structure with an energy lower than the crystal structure by 150 kcal/mol. This problem was attributed to the fact that without explicit water molecules in the simulation, the AMBER potential function contains no term to model the hydrophobic effect (the tendency of hydrophobic side chains to pack in the core of the protein, away from the solvent). With this in mind, the authors also investigated the use of a knowledge-based potential function, derived by statistical analysis of a database of known protein tertiary structures, in which terms representing solvation effects were present. However, GA optimisation of this potential function still produced incorrect structures of lower energy than the crystal structure; generally the structures produced had very poor tertiary packing of the secondary structure elements. The conclusion drawn from these studies was that although the GA was capable of searching the conformational space and finding low-energy solutions, neither the AMBER potential function, nor the knowledge-based potential function, was sufficiently good for protein tertiary structure prediction. Similar work to that of Le Grand and Merz has been reported by Schulze-Kremer [71,72].

In 1994, Bowie and Eisenberg [73] put forward the idea of using a GA for protein folding beginning with a starting population of solutions not generated randomly, as in most folding algorithms, but rather with a generation method biased towards likely structures for the sequence in question. This method is advantageous, they argue, because the size of the conformational space to search is limited from the beginning, thus easing the search problem, and the demands on the potential function are not so great. This function is simply required to provide a downhill route from already promising structures to the correct answer. The population of starting conformations is generated by joining together short segments of structures in a database (9–25 residues in length) for which equal-length parts of the sequence are highly compatible according to the 3D profile method [74]. These structures are then each coded as a string of peptide dihedral angles on which the GA operates. The energy function used consists of several terms, each reflecting a difference between a property of the fold and the average value of the property in the database of native folds. This function was optimised for maximum correlation with the distance matrix error from the native structure of one protein (the helical N-terminal domain of the 434 repressor, 63 residues). The method was tested on two small helical proteins, the 57-residue engrailed homeodomain and a 50-residue fragment of the B-domain of protein A. In both cases, the GA proved capable of finding structures close to the native fold (distance matrix errors in the range 3.0–4.0 Å). However, although a large proportion of the final population consisted of such structures, other structures dissimilar from the native fold, and of lower energy, were also found in each case. Despite this deficiency in the energy function, the authors were able to conclude that the method was successful in finding native folds for the proteins by examining a very much smaller number of structures than would have been necessary by using a random search. Improvements to this algorithm known as 'local moves' – essentially making local changes to the backbone conformation that have no effect on the conformation of distant parts of the chain – have been described in the work by Elofsson and co-workers [75].

The largest protein for which a folding simulation has been carried out using a GA is the 146-residue model of myoglobin used by Gunn and co-workers [76,77]. This protein was first folded to an rms from the native conformation of about 6 Å and this was later improved to about 3.6 Å by changes to the potential function. Although this work involves the GA only as a peripheral part, the algorithm used does have a number of interesting features which may become more common in future applications. Firstly, the algorithm is a hybrid of Monte Carlo simulated annealing and a GA, where, as in the work of Unger and Moult, a population of structures evolve according to the simulated-annealing algorithm with occasional GA-like crossovers. Secondly, a hierarchical model of the protein is used. This involves a crude model comprising rigid secondary structure elements joined by flexible loops, and a model with atomic-level detail defined by ϕ and ψ angles. The crude model is used to permit examination of many structures in a short time, while the more detailed model leads to a better quality in the final structure. Thirdly, the energy function is allowed to change as the simulation proceeds. At the start of the simulation, the terms are weighted so as to favour compact structures using a net hydrophobicity term. This avoids the waste of time spent on examining structures which are not globular. As the simulation proceeds, the weight of the residue-specific pairwise contact term is increased with respect to the net hydrophobicity in order to favour the specific contacts observed in the native structure.

A more recent application of GAs to protein folding is found in the work of Sun, Thomas and Dill [78]. This paper concerns itself mostly with the issue of energy functions, but a GA was used for the conformational search. The energy function is very simple, involving only hydrophobic interactions, excluded-volume, hydrogen-bond interactions between β-strands, and disulphide-bridge terms where appropriate. The protein is again described by ϕ and ψ angles, but the GA only operates on those in loop regions, those in regions of helix or sheet secondary structure being fixed at ideal values. The input to the algorithm requires knowledge of at least the native secondary structure of each residue, and the search is therefore over a restricted part of conformational space. The method was applied to 10 small proteins of which seven were folded successfully. Failures were attributed to minor problems with the potential function, problems with the single-point representation used for side chains, and also to problems with the GA search algorithm, which seemed to encounter difficulties with very long elements of secondary structure.

Another recent application of GA methods in protein folding is due to Pedersen and Moult [79]. In this preliminary report the results of a GA protein-structure prediction method applied to three small polypeptides are described. These predictions were carried out as part of a protein-structure prediction contest in which the native structure of the prediction targets was not known to the competitors. The method was applied to small segments of the target sequences predicted to constitute independent folding units. Results were mixed, with better results for secondary structure than for tertiary packing. Generally, problems were ascribed to the energy function, although in at least one case better results would have been obtained had more GA runs been carried out. It was noted that the GA always found lower energy conformations than the Monte Carlo simulated-annealing algorithm.

A final example related to protein folding using a GA is the work of Del Carpio [80]. The novel aspects of this work are the hybrid nature of the GA, which uses a Newton–Raphson procedure after each genetic operation to optimise side-chain atom positions, and the fact that the GA is implemented to run in parallel with the population distributed between five networked transputers. Using a fitness function incorporating a molecular mechanics-based term for the steric energy, together with terms specifically relating to protein folding, such as compactness and the formation of disulphide bonds, the algorithm proved able to fold crambin to within 6.69 Å rms of the crystal structure.

We now move to the use of GAs in more restricted protein conformational searches than are required for ab initio protein folding. GAs have been used for the problem of positioning all the side chains of a protein given its backbone conformation [81]. The possible side-chain conformations are governed by a preformed library of rotamers determined by a statistical analysis of known protein structures [82]. Using rotamers reduces the size of the combinatorial search space while concomitantly decreasing the search resolution. The search is guided by a forcefield which evaluates the energy of each trial conformation of the protein. For this problem, a simple GA was found to be less effective than a derived algorithm known as 'Selection-Mutation-Focusing' (SMF) [81,82], which gave identical results to the GA but displayed better convergence properties. This is not an unusual result – often, a straightforward GA 'recipe' will not transfer well between applications. The performance of any EA is likely to be enhanced by the inclusion of domain-specific knowledge, heuristics and operators. In the work of Tufféry et al. however, the best results for side-chain placement on a test set of 14 globular proteins were obtained from a deterministic algorithm, rather than an evolutionary approach [81].

Ring and Cohen have applied a GA in their BLOOP program for conformation sampling of loop structures in proteins [83]. Loop conformations can be represented by a series of overlapping tetrapeptides resembling four letters of the alphabet: J, L, U and Z [84]. Thus, an n-residue loop can be represented by n – 3 structural alphabets. Since there are four possibilities, each structural alphabet is conveniently represented by two bits in a bit string. BLOOP generates loop conformations using a GA constrained by the amino acid propensities of the tetrapeptide structural alphabets. The GA takes the sequence of the input loop and generates a population of loop conformations compatible with the sequence. These conformations are then built and refined using other algorithms. In tests using the hypervariable loops of the antilysozyme antibody, HyHel-5, BLOOP was found to model the loops just as well as established procedures such as CONGEN [85,86].

Molecular docking

Broadly stated, molecular docking algorithms seek to predict the bound conformations of two interacting molecules, whether they be a small-molecule ligand and its receptor or two macromolecules. The ability to carry out such predictions reliably and rapidly is a much sought-after goal in CAMD research and thus, much effort has been expended on the docking problem [87–89]. The search spaces involved in docking are enormous, particularly if molecular flexibility is taken into account. There is thus a need for effective heuristic search algorithms to enable thorough sampling of the space. Furthermore, these algorithms must be coupled with accurate objective functions, which are rapidly computable, if the docking problem is to be approached with any hope of success.

Evolutionary algorithms have proved very successful as heuristic search procedures for the docking problem, particularly in the case where the prediction involves a small-molecule ligand binding to a receptor active site. Indeed, given the recent proliferation of papers describing EA applications in this field, one might say that this class of algorithm is now to be preferred for molecular docking. In what follows, we shall review the growing number of EA-based docking methods.

The earliest published application of GAs to molecular docking seems to have been the work of Dixon [90]. These preliminary findings were later augmented by a full paper [91]. Two GA-based methods for docking were described in which the molecular flexibility of the ligand was taken into account by including the ligand's torsion angles in the list of docking variables ('flexible' docking). Algorithms from the DOCK suite of programs were used to characterise the receptor site [92]. In tests on three enzyme–ligand complexes, it proved possible to generate docked conformations in good agreement with the crystallographically determined structures, the best of the methods giving an rmsd accuracy of 1.0 Å or less. The authors note, however, that the method requires the restriction of the ligand's orientation and location to a portion of the active site [91]. A particularly interesting idea presented in this work was the use of a GA to dock a mixture of ligands simultaneously to a single receptor active site. Eight analogues of haloperidol were represented in a population of 761 members and these analogues were allowed to interbreed and compete in binding to the active site of HIV-1 protease. Each population member carried an identification tag to indicate which analogue it was representing; in this way selection was able to take place amongst conformations of the same analogue. Results from these simultaneous docking experiments indicated that for seven out of the eight analogues, comparable results were obtained to those found when docking a single analogue with an identical population size. Thus, considerable time savings can be achieved in this manner. A final finding worthy of note is that a GA method for rigid docking was found to give better scoring hits in less computer time than the matching algorithms currently embodied in the DOCK program [91].

The GAME program of Xiao and Williams [93] has been successfully applied by them to the rigid docking of deoxyguanosine to actinomycin-D [94]. GAME uses a fairly straightforward GA, together with a molecular mechanics potential function to calculate fitness scores. The crystal structure of the deoxyguanosine–actinomycin-D complex contains two deoxyguanosine moieties. It was found to be possible, using GAME, to dock both these into their crystallographically observed binding sites either sequentially or simultaneously.

Judson and co-workers developed their method for GA-based conformational searching (mentioned in an earlier section) into a flexible-docking procedure [95,96]. The method requires the selection of a ligand atom as a 'pivot' that provides a reference point for the translation and rotation of the ligand. A reference position for the pivot is also chosen to be in the vicinity of the expected position in the bound conformation of the ligand. The

GA operates by first docking a small part of the ligand which contains the pivot. The ligand is then 'grown' in an incremental fashion with each submolecule being allowed to explore its low-energy conformations. Finally, the full ligand is allowed to explore its conformational space in the active site. All generated conformations of the full ligand are analysed by clustering and energy minimisation. The algorithm was tested on three proteins and found to generate chemically reasonable conformations [96]. As described, the method is rather time-consuming; the authors report some 7–13 h (SGI Indigo R4400) for the search phase and even more for the analysis. These times could be reduced by using a precalculated grid for the energy evaluations [96]. The authors also found that the inclusion of a continuum solvent model yielded improved results.

The program DIVALI (Docking wIth eVolutionary AlgorIthms) for flexible docking has recently been described by Clark and Ajay [97]. Their GA method is based upon the public domain package GAUCSD [98], but with one particular novel feature that was found to enhance the performance of the algorithm significantly. This feature is known as the 'masking operator' and its effect is to divide the search space into eight parts, each of which can be searched in a separate run. Clark and Ajay describe the search of each of these regions as equivalent to testing a different binding hypothesis. The objective function for DIVALI is based on the AMBER potential function and, in contrast to the work of Judson et al. [96], no treatment of solvent is included. Nonetheless, Clark and Ajay were able to demonstrate good structural agreement between docked and crystallographically determined ligand conformations for four test systems without any adjustment of parameters [97]. CPU times of 2–15 min were reported for flexible docking. Note, however, that an exhaustive analysis would require at least 80 times this amount (eight regions to search with at least 10 docks per region to ensure full sampling by the stochastic GA) [97].

A break in a string of GA docking applications was marked by the publication of the EPDOCK method [99, 100]. In this work, a simple, rapidly computed molecular recognition potential is coupled with an EP searching algorithm to form an extremely fast and reliable flexible-docking program. The EP algorithm benefits from the use of self-adaptive mutation parameters and also from the generation of multiple offspring per parent. The exact number of offspring is varied during the run as an ad hoc function of the number of generations. In tests on the methotrexate-dihydrofolate reductase (3dfr) system, EPDOCK was found to be able to reproduce the crystallographically observed binding mode in 91% of the docking attempts, with each attempt taking only 2.3 CPU min on an R4400 processor [100]. In an extremely challenging example, the docking of the proprietary HIV-1 protease inhibitor AG1343 to HIV-1 protease, the observed binding mode was reproduced in 35 out of 100 attempts, with a further 15 of the attempts yielding a symmetry-related binding mode. The CPU time required in this test case was 8.2 min per dock. The EPDOCK program has also been used in flexible database searching for thymidylate synthase (TS) inhibitors [100]. Nine known TS inhibitors were seeded amongst 3000 compounds randomly selected from a commercially available database. Using fast molecular property screens, 1852 of the 3009 compounds were removed from consideration before docking. All of the top-seven scoring compounds were from among the nine known inhibitors, and the remaining two known inhibitors scored at least 1.25 standard deviations above the mean score.

In a similar manner to the work reported by Sanderson et al. [41] in an earlier section, a GA-based method has been used by Meadows and Hadjuk to dock ensembles of small, flexible ligands to receptor proteins using constraints derived from NMR experiments [101]. Using simulated NOE distances for the streptavidin–biotin complex, the GA was able to generate docked conformations of biotin that were free from van der Waals clashes and almost identical to that observed in the crystal structure.

Over the years, docking algorithms have gradually addressed more and more of the variables involved in the search space. Initially, rigid–rigid docking (i.e., receptor and ligand docked in fixed conformations) was all that was attempted; now, rigid–flexible docking (ligand conformational exploration) is relatively commonplace and manageable. The work of Jones et al. [102] extends this further by including a partial treatment of receptor flexibility during docking. This is accomplished by including in the string of docking variables operated upon by the GA the rotations of single bonds connected to terminal acceptors or donors in the active site. The search is guided by a fast fitness function that comprises an empirical hydrogen-bonding term, including a treatment of desolvation, together with van der Waals energy terms for intermolecular and intramolecular interactions [102]. Results presented by Jones et al. on five test systems showed the GA was capable of reproducing experimental results with a high degree of accuracy. A typical docking run required some 5–8 CPU min on an R4000 SGI Iris computer. The majority of the test cases were shown to be highly reproducible, i.e., a single GA run has a high probability of generating the correct binding mode. The docking of folate to mammalian dihydrofolate reductase was found to be more difficult to reproduce, with 20–30 runs being required to locate the correct binding mode reliably [102].

Moving to larger systems, Duncan has described the use of EP for protein–protein docking [103]. The molecular surfaces of the two proteins to be docked are approximated by expansions of spherical harmonic functions which allows the analytic computation of surface shape properties [104]. An EP algorithm is then used to optimise a scoring function having terms representing surface contact area, shape and electrostatic complementarity and minimisation of surface hydrophobic area, together with penalties for bad van der Waals contacts. This SURFDOCK algorithm has been used to reconstruct successfully the superoxide dismutase homodimer [103] and also to predict the binding of β-lactamase inhibitory protein (BLIP) to TEM-1 β-lactamase [105,106].

De novo molecular design

Computer programs for de novo molecular design seek to generate novel molecules that satisfy a set of design constraints. While most of the activity in de novo design is connected with the generation of small organic molecules for lead discovery, the de novo design of polymers and proteins is also feasible.

Small-molecule design

In the last decade, there has been a dramatic surge of interest in de novo methods for drug design and many different approaches have been reported in the literature [107,108]. In many cases, the design constraints are derived from the active site of the target macromolecule, but de novo design may also be guided by a pharmacophore or CoMFA model inter alia.

GA-based methods for de novo design began to emerge in 1993. At a meeting of the Molecular Graphics Society, descriptions of three different methods were presented together with some early results [109–111]. Brief descriptions of these methods are given in published reports from the meeting [112,113]. Unfortunately, only one of these methods has been fully described in a subsequent publication [114]. The POSIT program [111] has subsequently been marketed by Tripos as 'LeapFrog' [115].

The method reported by Blaney et al. [109] was based upon the SMILES line notation [116,117]. Each molecule in the evolving population is represented by its SMILES string and these strings constitute the chromosomes upon which genetic operations can take place. The fitness of each of the population members is evaluated by flexibly docking the molecule into the relevant active site using a combined distance geometry/DOCK algorithm and scoring its fit. Clearly, the speed of this docking step is of crucial importance [112].

The LeapFrog program incorporates several GA-type features in its strategy for de novo design. A population of candidate molecules is maintained and the user can instruct that operations such as crossover, mutation or selection be performed. One interesting aspect of LeapFrog, in common with other de novo design programs (vide infra), is that the genetic operators are applied to the molecules themselves, rather than some encoding of them, i.e., the 3D molecular structure is both the phenotype and the genotype [112].

The most completely described of the methods is that of Glen and Payne [114]. The GA described in their paper can generate novel molecules to satisfy a wide variety of constraints which are expressed in the fitness function. The crossover and mutation operators developed by Glen and Payne have been specially tailored to the problem of de novo design and, as in LeapFrog, operate directly upon the 3D molecular structure. The crossover operator has two variants: (i) *terminal* crossover, in which a terminal portion of one parent is detached and connected to a similar terminal portion from the other selected parent; and (ii) *region* crossover, in which an internal portion of one molecule is inserted into another molecule that has had a similar region removed. The mutation operator has 12 variations, including translation along a particular axis, rotation about a selected rotatable bond and changing of an atom type [114]. Structures generated by crossover and mutation are optimised using molecular mechanics before their fitness is assessed. Glen and Payne present several examples of successful constrained structure generation with the GA [114].

A slightly different application of a GA in de novo design has been reported by Westhead et al. [118]. In this case, the GA is applied to 'refine' a set of structures generated by a de novo design program, which often vary widely in quality. This set of structures (or a subset thereof) is taken as the initial population for the GA. The fitness function used to score the population members is the same as the scoring function used in the de novo design program that generated the structures [119] and involves a weighted sum of the target sites hit by the structure together with various molecular properties, e.g., the number of rings in a molecule. Selection occurs by a standard roulette-wheel mechanism and crossover is implemented by splicing the two parents across a single bond and reconnecting the resulting fragments. Two forms of the mutation operator have been implemented: a torsional mutation, in which one or more torsion angles in a structure can be randomised, and a rule-based atom-type mutation similar in spirit to that described by Glen and Payne [114]. Two kinds of GA were experimented with and, in this application, a steady-state mode was found to be preferable to a GA using generational replacement with elitism. The structure-refinement GA was successfully applied to the improvement of populations of structures designed to be mimics of distamycin and methotrexate [118,120].

Macromolecule design

In addition to the design of small molecules, GAs have been applied by a number of groups in protein design and engineering. Dandekar and Argos [55] used a simple GA to search the sequence space accessible to a

30-residue peptide using a fitness function reflecting the similarity of the sequence to one typical of a zinc-finger fold. It was found that the GA quickly located suitable sequences from amongst the 20^{30} possibilities. Using the same GA with a fitness function reflecting fold stability (including contributions from solvation effects and secondary structure preferences along with test-case specific terms) the authors were able to design mutations in the λ-repressor sequence for enhanced stability.

In a similar application, Jones [121] used a GA in order to design protein sequences of optimal compatibility with a given three-dimensional structure (fold). The fitness function includes pairwise interresidue interaction and residue-based solvation terms, as well as a term biasing the composition of the designed sequence towards the average composition of a sequence within the desired folding class. A simple GA operates on chromosomes comprising genes identifying the amino acid associated with each position in the desired fold. Three manually designed proteins were used as examples. For each one, it was demonstrated that the GA could produce a sequence more compatible for the chosen fold than was produced by the human designers, according to the fitness function. One example of a natural protein (acylphosphatase) was tried, for which the GA produced an optimal sequence with 27% similarity to the native sequence for the protein. More recently, this approach has been successfully applied to the design of Paracelsin-43, an α-helical protein whose sequence was constrained to retain more than 50% identity to an all-β protein [122].

Related work has been reported by Hellinga and Richards [123], in which an algorithm that they called 'simulated evolution' was used to minimise an empirical potential function over a space in which the protein backbone was fixed but both the sequence and the side-chain conformers were allowed to vary. Evolution in this context refers to random mutation of amino acid identities, and the actual optimisation algorithm they used was simulated annealing, rather than an evolutionary algorithm as defined in this review.

Recent work by Desjarlais and Handel studied the de novo design of hydrophobic cores of proteins [124]. Their GA was designed to generate low-energy core sequences and structures, given a customised rotamer library generated for the protein of interest. Using the program, several variants of the phage-434 cro protein were designed and engineered. Two of these designs, including one with eight changes from the native core sequence, were found to have experimental thermal stabilities comparable to the native protein. As a test, a random design was also generated, which proved to be completely unfolded under the same conditions.

Design of protein sequences was again the subject of the work by Ebeling and Nadler [125]. This time the criterion is to find 'folding' sequences, that is sequences which fold rapidly and reproducibly to a target conformation; this work is therefore closely related to that described above on protein folding. The folding ability of a sequence is optimised by using a two-stage GA. The first stage begins with random sequences and aims to find sequences for which the target conformation constitutes a stable fold. In this stage, the GA operates in sequence space and the fitness function is the stability of the sequence, judged by the proportion of a Monte Carlo simulation for which it remained in the target conformation having begun in this conformation. Sequences derived from this stage are then carried through to a second sequence space GA used to optimise folding ability. Here, the fitness function is the proportion of a Monte Carlo simulation spent in the target conformation when the simulation was started in a random-coil conformation. The algorithm was clearly demonstrated to produce folding sequences.

In a rather different application, Schneider and coworkers have developed a technique for the systematic optimisation of amino acid sequences which they term 'simulated molecular evolution' (SME). The technique combines a (1,λ) evolution strategy with a neural network-based fitness function and has been successfully applied in the classification and rational design of peptide sequences containing proteolytic cleavage sites [126–130].

Finally, a GA approach has been adopted by Venkatasubramanian et al. to design polymers having particular predicted values for a number of important properties, such as density and specific heat capacity [131]. The GA was found to be effective in generating diverse populations of molecules which (almost) satisfy the constraints. This 'fuzziness' was thought to be an advantage, since one cannot always be completely certain of the accuracy of computed property values.

Pseudoreceptor modelling

When seeking to design new drugs in a rational manner, it is highly desirable to possess a detailed 3D model of the target macromolecule, preferably in complex with a known inhibitor. This is the so-called 'direct' design scenario. However, in many cases, the structure of the receptor is not known in such detail and other approaches must be adopted to infer as much information as possible to help in the design process. One such approach is the construction of a *pseudoreceptor*, i.e., a hypothetical model, based upon the structures of some known ligands. The Yak program is perhaps the most well-known computational tool for this purpose [132,133]; other methodologies are also emerging [134,135].

A GA approach for pseudoreceptor construction has been developed by Walters and Hinds [136]. Their GERM (Genetically Evolved Receptor Models) program works by placing a number of explicit atoms around an aligned series of known ligands and then calculating the intermolecular interactions between the ligand atoms and the atoms of the pseudoreceptor. The CHARMm forcefield types of these atoms are encoded in an integer string, which thus represents one particular model receptor [137]. The GA creates an initial random population of these strings and then varies the atom types at each position by means of crossover and mutation operators. The fitness of each model is judged by the quality of the correlation it gives between the calculated intermolecular ligand–receptor energy and the binding affinity for the ligand in question. Once a good coefficient has been obtained, the regression equation may be used to predict approximately the binding affinity of ligands not in the training set by aligning them within the pseudoreceptor and calculating their intermolecular energies. In a test on a set of 22 structurally diverse sweetener molecules with known potencies, 11 of the structures were used to derive a population of 5000 models. This required 10–20 h of CPU time on an SGI Iris 4D/120. Using these models, it proved possible to predict the activity of the 11 molecules not in the training set with an average error of 0.44 log units [136].

Pharmacophore elucidation

An alternative approach to lead generation when no receptor structure is available is to seek to identify a *pharmacophore* from a series of molecules with known activities against the receptor of interest. A pharmacophore is a set of structural features whose relative orientation in space may confer a particular biological activity upon the structure in which they are present. Once a pharmacophore has been obtained, it may be used as a query for 3D database searches, whereby other compounds possessing the pharmacophore may be found [138] or as a basis for de novo design [108]. A number of techniques for pharmacophore elucidation have been suggested over the years [139–145] and recently, GA methods have also been applied.

The first reported application of a GA to the pharmacophore elucidation problem was in the work of Payne and Glen [42]. They used a GA to overlay three NMDA antagonists using a putative three-atom pharmacophore to guide the alignment. This exercise proved to be very computationally demanding, requiring some 10 days of CPU time on an IBM RS6000/320 [42]. More recently, Jones et al. have built upon this work using ideas from their GA docking application mentioned earlier [102,146]. In the GA for pharmacophore elucidation, the molecules are represented by a chromosome that encodes not only variable torsion angles, but also mappings between donor, acceptor and ring centroid features in pairs of molecules. The latter feature means that the GA is truly capable of elucidating pharmacophores rather than simply confirming the viability of a user-specified alignment. The GA uses the molecule in the data set with the fewest features as a reference and then overlays the remaining members of the set in a pairwise fashion. The fitness function used seeks to optimise the number and similarity of the overlaid features, the volume integral of the overlay and the conformational energy of the individual molecules as they are overlaid. In tests on a diverse set of problems, the GA was found to elucidate pharmacophores in good agreement with those determined by other methods, often in a much smaller amount of CPU time [146]. This work has recently been commercialised by Tripos as a program called GASP [147].

Chemical structure handling

One of the fundamental tasks in chemical structure handling is *substructure search*, i.e., the location of a chemical substructure within a larger, complete structure. The ability to perform this kind of search rapidly and reliably is one of the cornerstones of chemical structure database applications. The fact that chemical structures (whether 2D or 3D) can be regarded as mathematical objects known as *graphs* means that the problem of substructure search is equivalent to that of *subgraph isomorphism*, i.e., the location of a smaller graph within a larger graph. In chemical structure handling applications, the subgraph isomorphism algorithm developed by Ullmann [148] has been shown to be very efficient for substructure search [149].

Two groups of workers have sought to apply evolutionary algorithms to substructure searching. Brown et al. developed a GA for graph matching and compared its efficiency to that of the Ullmann algorithm [150]. The comparison was carried out by using three query substructures and 140 structures known to contain one or more of them. Since the GA is nondeterministic, an average CPU time from 10 runs for each query/structure match was computed and compared to the CPU required for a single run of the deterministic Ullmann algorithm. The result of this comparison averaged over all query-structure pairings was that the Ullmann algorithm outperformed the GA by a factor of 3.5. As Brown et al. point out, in real-world applications the GA is likely to be even more disadvantaged because of its inability to detect mismatches (the majority of cases) as quickly as does the Ullmann algorithm. Luke has investigated different evolutionary algorithms for substructure searching using a metric similar to the chemical distance to detect matches [151,152]. He found that an EP algorithm performed particularly well, being able to find all matches of a 7-atom substructure in a database of 520 molecules containing 7 to 26 atoms in 8 CPU s (measured on one node of an IBM SP1 machine). This works out at approximately 0.015 s/structure, which is still appreciably slower than the Ullmann algorithm, which is capable of processing thousands of 2D matches per second.

Another important problem in chemical structure handling is that of determining the maximal common substructure (MCS) of two (or more) compounds [153]. Determination of the MCS allows 'best match' or partial matching of two structures, in contrast to the 'exact match' insisted upon by subgraph isomorphism algorithms. Both types of algorithm are, however, NP-complete. Some early work applying GAs to this sort of problem was carried out by Fontain [154]. He used a genetic algorithm to calculate the minimum chemical distance (MCD) between pairs of 2D structures [154]. The calculation of the MCD requires the maximal structural overlap of the two structures to be determined, and can thus be used as a measure of their constitutional similarity. Fontain found that a GA was very effective in MCD calculation and therefore, in the problem of maximal structural overlap determination.

Other workers have also found GAs to be effective in determining maximal overlaps of chemical structures. Brown et al. found a GA to be very successful in the process of constructing a 'hyperstructure' representation of a set of 2D structures [105,155]. Hyperstructures are constructed by maximally overlapping pairs of structures and storing the common parts only once. Using a conventional maximal overlap set (MOS) algorithm (similar to an MCS algorithm) proved to be too computationally expensive for building hyperstructures from data sets of nontrivial size. Thus, a fast but more approximate 'atom assignment' method was developed, which took no account of bond types during the hyperstructure construction process [156]. Subsequent research produced a GA for hyperstructure generation that was able to match bond as well as atom types during the overlap process. In this way, the GA produced more compact hyperstructures than the atom assignment method, while being more efficient than a conventional MCS algorithm. Wagener and Gasteiger have also implemented a GA approach for MCS determination [157]. They found the method able to determine reliably MCSs for pairs of complex molecules such as morphine and methadone in a relatively short time (48/50 correct answers, each run taking 20 s on a Sparc Station 10/20). The determination of MCSs in this way can provide valuable insights into structural prerequisites for biological activity or give guidance in the design of organic syntheses. As published, the GA considers only 2D structures, but work is in progress to develop a GA for determining maximal common 3D substructures [157].

The field of 3D database searching has also seen some GA applications. As mentioned in an earlier section, Clark et al. found a GA to be effective as a conformational search algorithm for flexible 3D searching, although ultimately not the method of choice [33]. More recently, research has moved on to examine the possibility of matching the molecular *fields* around molecules, rather than their structures per se. In this regard, Wild and Willett have used a GA for aligning molecular electrostatic fields (MEPs) enabling field-based similarity searches to be carried out over 3D structure databases [158]. In their application, the GA operates upon chromosomes which encode the rigid-body rotations and translations needed to overlay the two structures being compared. The fitness function uses Gaussian functions which approximate to the true potential fields, but are very fast to calculate. Experiments have shown the MEP-matching GA to be well-suited to implementation upon parallel computer architectures which may be important in enabling the rapid searching of large compound collections. Further research is examining methods of accounting for conformational flexibility during the field-based search [158].

A novel GA application has been suggested by Hibbert [159], who has developed a GA for generating and displaying 2D chemical structures. Given a molecular formula and information about the types of bonds in which atoms may participate, the GA was able to generate a number of possible isomers for C_6H_6. The program was also used to produce reasonable 2D depictions of anthracene and cubane [159].

Finally, Lushnikov and Sello used a GA to optimise the parameters for a program designed to estimate the nucleophilicity and electrophilicity of atoms in a molecule [160]. Such calculations are of use in computer-assisted synthesis planning and reaction prediction.

QSAR and chemometrics

A typical Quantitative Structure–Activity Relationship (QSAR) model consists of a linear combination of *basis functions*, which are themselves functions of one or more *features*, such as log P, molecular volume, etc. Traditionally, *linear regression* has been used to find a set of coefficients for the basis functions, which together form a model describing the data present in the training set. However, linear regression has several weaknesses, particularly a tendency to *overfitting* when larger numbers of features are involved. Thus, several workers have investigated the use of evolutionary algorithms in the construction of QSARs in a bid to find techniques superior to linear regression.

Rogers and Hopfinger have developed a methodology called 'Genetic Function Approximation' (GFA) [161]. In this approach, a steady-state GA operates upon strings of basis functions, each of which represents a potential QSAR model. The fitness of the models is assessed using a 'lack of fit' (LOF) error measure that resists overfitting and allows user control over the smoothness of fit, i.e., the accuracy to which the model describes the training set data. The algorithm begins with a random population of basis function strings whose coefficients are calculated using least-squares regression. The GFA approach then applies genetic operators of selection, crossover and mutation to these strings. Parents are chosen for breeding in inverse proportion to their LOF score and simple, one-point crossover is used. There are two possible mutation operators: (i) 'new', which appends a new random basis function to the string; and (ii) 'shift', which moves the knot of a spline basis function. No duplicate models are permitted in the population. Over time, the fitness of the models increases and at the end of a run (typically a few CPU minutes), one can either select the best scoring model or peruse a handful of high-scoring models for suitability. The evolution of a population of models is a useful property of an EA approach; traditional approaches yield only a single model. In tests on 'standard' data sets, the GFA algorithm was found to generate models competitive with, or superior to, methodologies based on regression analysis or neural networks [161]. The GFA methodology is now commercially available [162].

Luke has also applied an evolutionary approach to QSAR construction [163], but has chosen to focus upon an EP methodology. The EP algorithm works on a string of basis functions similar to those used by Rogers and Hopfinger [161] and uses a two-term function to measure the fitness of the evolving models. The first term simply measures the rms error of the model-predicted values from the data-set values and the second term is designed to drive the algorithm towards generating models containing a particular number of basis functions. Child models are generated from parents by means of mutating the number of functions in a model or the weights assigned to them in the second term of the fitness function. Selection from the parent + child population is carried out simply by ranking according to fitness. In tests on the same data sets as used by Rogers and Hopfinger [161], Luke found the EP algorithm to be capable of generating QSAR models of similar quality to those found by GFA. In one case, the EP algorithm also located some QSARs not discovered by the GFA. No final conclusions of the relative merits of the EP versus the GFA approach were drawn, but Luke suggests that the EP approach may become more favourable as the number of basis functions increases [163].

In a recent paper, So and Karplus have described a QSAR methodology combining an evolutionary algorithm with an artificial neural network (NN) [164]. The evolutionary algorithm is used to select the descriptors and the neural network then correlates the descriptors with the observed biological activities to yield the QSAR model. A variety of fitness functions was employed including residual rms error, the correlation coefficient and predictive ability, as measured by a cross-validation procedure. Interestingly, So and Karplus experimented with the two evolutionary algorithms mentioned above: the GFA approach of Rogers and Hopfinger [161] and the EP approach of Luke [163]. ·Comparisons of the two methods indicated that while the best model from a GFA population or an EP population were identical, the remaining models in the EP population were substantially superior to the rest of the GFA population. An analysis of this discovery revealed that the models retained by EP were in fact generated during the course of a GFA run, but were subsequently destroyed by crossover or mutation. Thus, So and Karplus chose the EP algorithm for the bulk of their studies. Using the Selwood data set [165], the combined EP–NN approach was able to yield QSAR models claimed to be superior to any others reported in the literature, both in terms of the fit of the training data and their predictive ability. So and Karplus also experimented with composite QSAR models constructed from a number of the high-scoring EP–NN solutions. In principle, predictions from such a model should be more reliable than those from a single model, because more information is embodied in the composite model. The use of several models also allows error estimation. In the case of the Selwood data set however, it was not found possible to generate a composite model whose predictive ability was significantly superior to the best single model.

The MUSEUM (MUtation and SElection Uncover Models) algorithm developed by Kubinyi also leans more towards an EP approach than a GA, as it contains no crossover operator [166]. MUSEUM differs from other evolutionary approaches, however, in that it uses a population of only one member. The algorithm starts with a model containing a number of randomly chosen basis functions and then seeks to generate a better model by applying several steps of mutation. Firstly, one or a few basis functions in the model are mutated; if a better model results from this procedure, then this is stored and used as the basis for the next generation. Otherwise, mutation is applied to several basis functions; if a better model results from this procedure, then this is stored and used as the basis for the next generation. If neither of these random mutation steps improves the model, then a systematic addition and elimination of each variable is performed to seek an improvement. If this systematic procedure fails to locate a better model, then the current model is taken as the best available. This 'best' model then has each of its variables checked for significance; if any is not significant at the 95% level, it is eliminated from the model. Kubinyi used a variant of the Fischer significance value, F, as a fitness function [166]. In presenting his results, Kubinyi suggests that the models generated by MUSEUM are often more relevant than those from other methods, including GFA. In a subsequent paper, Kubinyi reported that combining the evolutionary approach of MUSEUM with a systematic search procedure (for generating models with two or three variables) gave speed ups of up to an order of magnitude over the basic MUSEUM approach [167]. More recently, Kubinyi has found the MUSEUM method to be effective at variable selection in both regression and partial least squares (PLS) analyses [168]. Other workers have also investigated GAs for this purpose [169–171].

Practitioners of chemometrics have also been active in exploring the potential of EAs in their field. For example, evolutionary approaches have been used with success in curve fitting [172–174], clustering [175], wavelength selection [176,177], modelling chromatographic behaviour [178], analysing titration data [179] and determining stability constants [180].

Combinatorial libraries and molecular diversity

One of the most intense areas of research in the pharmaceutical industry at present concerns the application of combinatorial chemistry in lead generation and optimisation. Briefly, combinatorial chemistry involves the rapid synthesis of thousands or millions of compounds using automatic protocols. The resulting compound *libraries* can then be processed through high-throughput assays to locate bioactive molecules or mixtures. Reviews of this rapidly developing area may be found in Refs. 181–185.

Computational methods are proving to be an essential component of combinatorial chemistry research in at least two areas: (i) the management of the enormous amount of structural and other data that are generated by combinatorial experiments; and (ii) the design of combinatorial libraries prior to synthesis [186]. Methods are now being developed and applied which seek to analyse and maximise the molecular diversity contained within any given library or database [187–194]. In this way, the number of molecules needing to be synthesised and tested can be dramatically reduced while keeping the probability of finding an active molecule at an acceptable level. As Kauffman and Macready point out [195], this kind of exercise constitutes an optimisation problem on a high-dimensional molecular fitness landscape. Their theoretical work using a spin glass-like model of molecular fitness landscapes suggests that traditional 'pooling' strategies employed in combinatorial chemistry experiments may be significantly enhanced if coupled with recombination, mutation and selection amongst the pool optimal candidates [195]. It thus seems that evolutionary algorithms may be well-suited to aiding in the task of library design, and other applications in this area have been reported by a number of research groups.

Sheridan and Kearsley have reported their development of a GA for combinatorial library design [196]. Their design problem involved the construction of a tripeptoid library (a peptoid is an N-substituted glycine oligomer) in which the first and second substituent positions in the peptoid were occupied by a primary amine and the third by a primary or secondary amine. Even if only 2500 amines are possible substituents at each position, there are billions of possible combinations. Their GA in effect 'synthesises' molecules in computro by assembling molecular fragments according to predefined rules. These molecules are then scored (or 'assayed'), either according to their similarity to a known target molecule or by using topological SAR information. Successive populations of molecules are bred using standard genetic operators and, at the end of the evolutionary process, the final population is analysed to discover the type of fragments which occur most frequently. This information can then be used to direct the actual synthesis of a relatively small number of compounds for assay. Sheridan and Kearsley found that the GA was able to generate high-scoring molecules quickly, and thus to be a potentially useful tool for combinatorial library suggestion [196].

Rather than use a computational measure as a fitness value, Weber et al. chose to use *experimental* activity data to guide the evolution of candidate molecules [197]. They investigated using a four-component Ugi-type reaction to generate thrombin inhibitors from a starting-materials database of 10 isocyanides, 40 aldehydes, 10 amines and 40 carboxylic acids. These numbers correspond to a potential library of 160 000 reactions. Weber et al. represented each of the starting materials by a bit pattern; four of these patterns were concatenated to form the strings describing individual population members. To initialise the GA, 20 bit strings were generated at random. The corresponding reactions were performed and the crude products tested in a suitable assay. The resulting fitness data were used to guide selection for the first generation, which was bred by crossover and mutation of the bit strings. From the total of 40 parent and child products, the best 20 were chosen to be parents for the next generation. This was repeated for 20 generations, corresponding to a total of 400 Ugi reactions. After only 16 generations, the average EC_{50} value for the best 20 products at each generation was submicromolar. The best compound overall (in generation 18) had a K_i value of 0.22 μM. Thus, by carrying out only 0.25% of the possible total number of reactions, potential lead compounds were rapidly identified. With the increasing use of automated synthesis and assay procedures, this kind of application should become even more rapid and effective [197].

More recently, a GA method has been used by Singh and co-workers to guide the combinatorial synthesis of stromelysin substrates [198]. Here again, the fitness function used to evaluate the population members was an experimentally derived rather than computed value. The GA was initialised with a random population of 60 hexapeptides, each of which was represented by a string of 30 bits (5 bits per amino acid residue gives 32 possible values, more than enough for the 20 naturally occurring amino acids). The peptides coded for by the bit strings were then synthesised and tested for activity in a fluorescence-based assay. The fluorescence values were then used as fitness values for the population members. Subsequent generations were then bred using standard GA operators with an 'elitist' mechanism ensuring the survival of the most fit from one generation to the next. Over five generations, the average fluorescence values and hence, activities, of the population members were observed to increase markedly and new active sequences emerged which had not been present in earlier generations. By using the GA, only 300 peptides out of a possible 64 000 000 needed to be synthesised before potent and selective stromelysin substrates were identified. Here, as in the work of Weber et al. [197], the GA is seen to be an effective means of sampling the high-dimensional molecular diversity space.

Finally, Holliday et al. have reported the implementation of a GA for the computational selection of a maximally dissimilar subset of molecules from a larger collection [188]. However, their implementation was not as successful at dissimilarity selection as another deterministic algorithm devised by them.

Cluster modelling

Another field which has benefited from the application of evolutionary algorithms is the modelling of clusters, both atomic and molecular. Under the latter heading, Xaio and Williams have reported the application of their GAME package [93] to the prediction of the structures of clusters of benzene, naphthalene and anthracene [199]. Homodimers of all three molecules were modelled, as were a benzene trimer and tetramer. It was shown that the GA could efficiently locate solutions in the vicinity of the global minimum which could then be refined by a local search technique.

Most GA applications in this field, however, have been in the field of atomic clusters. Hartke has applied a GA to the optimisation of various clusters, including Si-10 [200,201]. More recently, this GA has been hybridised with a local minimisation technique which has been found to give superior results to the GA alone when applied to the optimisation of Lennard-Jones (Ar)(n) clusters [202]. Mestres and Scuseria have also found GA methods to be useful in the optimisation of a C_8 cluster and rare-gas atomic clusters of up to 13 atoms [203]. Other work has been reported by Zeiri [204].

Miscellaneous applications

In this section, a number of miscellaneous applications of evolutionary algorithms in the field of computer-aided molecular design will be considered briefly.

An interesting application of GAs in the field of protein-structure alignment is due to May and Johnson [205, 206]. Here, the GA operates on a string of variables representing the relative position and orientation of two protein structures. The fitness function reflects the quality of the superposition of the C^{α} atoms of the two structures. Equivalences between pairs of C^{α} atoms are assigned by the dynamic programming method [207] so that a set of minimal interatomic distances is used. The fitness function is derived from the score given by the dynamic programming routine: essentially a sum of the interatomic distances with an additive penalty for the introduction of gaps into the alignment. Unlike many protein-structure alignment methods, this one does not require the user to provide a set of 'seed' equivalences, and the simplicity of the GA makes the method attractive by comparison with some other methods, for instance the double-dynamic programming algorithm [208].

GAs have also been applied to assist in structure determination. For instance, in the assignment of protein 2D NMR spectra, a GA assigned 40–80% of the spin systems correctly when tested on simulated NMR data for three proteins [209,210]. There have also been a number of GA applications in X-ray crystallography, where Chang and Lewis found that a GA was able to determine heavy atom positions consistent with an observed difference Patterson function far more efficiently than a sequential search [211]. In addition, the GA proved to be simple to apply, space-group independent and able to deal effectively with cases involving noncrystallographic symmetry of multiple heavy atoms in the asymmetric unit. A GA was also used by Tam and Compton in their GAMATCH program for indexing crystal faces [212]. In tests on triphenylmethyl chloride crystals, the program gave results that were in excellent agreement with the expected Miller indices. More recently however, Tam and co-workers have achieved even better results in this application area using simulated annealing [213]. A further EA application in crystallography has been reported by Miller et al., who have employed GAs for the de novo phasing of low-resolution X-ray diffraction data from crystals of icosahedral viruses [214]. Finally, a GA has been used recently to help in the analysis by Mossbauer spectroscopy of the mixed valence active center of the diiron-oxo protein uteroferrin [215].

Bush and co-workers have used a GA method to help in the difficult task of predicting the crystal structures of inorganic solids [216]. Their GASSP program takes a two-step approach to structure prediction. In the initial stage, a GA is used to search for a number of plausible structures guided by a sophisticated fitness function based on Pauling's valence rules. The best 10 of these potential solutions are then refined by standard lattice energy-minimisation techniques in the hope of locating the global minimum. This hybrid technique was shown to be capable of predicting the crystal structure of the ternary oxide Li_3RuO_4 in a matter of a few hours computer time [216].

Rossi and Truhlar have reported the application of a GA to the fitting of a set of energy differences obtained from semi-empirical (neglect-of-diatomic-differential-overlap) molecular orbital theory to reference ab initio data [217]. In this way, a set of specific reaction parameters (SRP) were obtained for the combination of $Cl + CH_4$ allowing the modelling of the potential energy surface for this reaction. In spite of the fact that only 13 ab initio points along the distinguished-coordinate reaction path were used in the fitting, the resulting parameters gave an absolute error of only 1.08 kcal over a wide range of energies. Rossi and Truhlar calculated that potential energy surfaces computed using SRPs are 8000 times less expensive to evaluate than their reference ab initio counterparts. They concluded that combining semi-empirical wave functions with an efficient stochastic optimiser (such as a GA) provides a robust and economic protocol for the generation of potential energy surfaces for the reactions of small-to-medium-sized systems [217].

Vivarelli et al. [218] have used a GA to optimise the topology of neural networks used for protein secondary structure prediction. The results obtained for secondary structure prediction from the optimised networks are of similar quality to those obtained by other neural network methods described in the literature. This illustrates that the limiting factor in secondary structure prediction is the information contained in the sequences and not the architecture of the network.

Discussion

It is little more than five years since EAs were first applied to molecular design and modelling problems, yet in that short time, the number and breadth of applications has grown tremendously. We have tried to give some idea of this expansion in the preceding sections. From our survey, it would seem that EAs have been particularly successful in a number of areas such as QSAR, molecular docking, de novo design and protein folding, often outperforming other kinds of algorithm. Other applications, especially in the chemical structure handling field, seem to have fared less well. It is certainly true that where a fast, deterministic algorithm is available for a problem, there may be little to be gained from a stochastic EA method. Where this is not the case, however, EAs have much to offer. The basic methods are conceptually uncomplicated, straightforward to implement and applicable to a wide variety of problems. In particular, EAs do not necessarily require detailed knowledge of the behaviour of the function to be optimised, just its values. This feature enables EAs to cope effectively with discontinuous or badly behaved functions [219].

In many respects, EAs share the elegance and simplicity of another naturally derived stochastic algorithm: simulated annealing [220]. However, EA methods can also offer the benefit of yielding a population of good solutions to a problem, rather than just the 'best' solution. In modelling situations, this is often particularly useful, as it reflects the multiplicity of potentially interesting local minima on an energy surface. Possible drawbacks of EAs are their stochastic nature, which requires that a given experiment be repeated several times to ensure reproducibility and that they often require significant effort to locate a good set of parameters. Furthermore, it may not always be simple to encode the problem to be solved in a suitable form for an EA to work upon. Finally, to achieve the best results, considerable thought may need to be given to the question of the most appropriate implementation of the operators (i.e., mutation and/or crossover) in a given application domain.

In spite of their success in many applications, EAs should not be viewed as a panacea or 'black box', which will automatically improve on results obtained with other search algorithms. Nonetheless, it would seem that EAs have a bright future in computer-aided molecular design applications. The research carried out so far can be built upon and new application areas investigated. In particular, it seems likely that combinatorial library design will see more EA-based methods applied; the same is true of protein-folding simulations. It is possible too that there will be a greater incorporation of EA methods in commercial modelling and design software; to the best of our knowledge, there are only three at present [115,147,162]. In terms of the algorithms themselves and their implementation, the future is likely to see more exploitation of parallel computer architectures; something to which EAs are ideally suited. Some of the work described in this review has already begun to investigate this potential [80,158], as have others [221]. It is also probable that more ES and EP applications will be explored to see if these algorithms are as successful as GAs have been in many areas. Detailed comparisons of EAs with other types of search and optimisation algorithms (e.g., simulated annealing [220] and tabu search [222,223]) would also be helpful to workers seeking the most suitable optimisation procedure for their particular problem. Basic research in EAs will no doubt continue within and without the molecular design field. In particular, the hybridisation of EAs with local optimisation methods seems very fruitful [219,224], as does the combination of various heuristic search techniques [225,226].

Conclusions

In this review, we have briefly described the three main types of evolutionary algorithm: (i) genetic algorithms; (ii) evolutionary programming; and (iii) evolution strategies, and we have surveyed their applications in the field of computer-aided molecular design. In general, these applications have met with encouraging success. On the basis of this, it seems likely that with continuing basic research, EAs will become an increasingly powerful and popular class of algorithms for solving the computationally intensive problems encountered in the design and simulation of new pharmaceuticals, agrochemicals and materials.

Acknowledgements

The authors thank Dr. J.G. Vinter for his encouragement during the preparation of this review. We are also grateful to Drs. C.W. Murray (Proteus Molecular Design Ltd.) and R.A. Lewis (Rhône-Poulenc Rorer Ltd.) for reading and commenting upon the manuscript. Drs. B.T. Luke, X. Yao, K.Y. Tam, T. Kido, R.D. Brown and Professor P. Willett kindly supplied preprints of their work. Other helpful information was provided by Drs. W.A. Warr and H. Rzepa and Professor N.L. Allinger.

References

1 Box, G.E.P., Appl. Stat., 6 (1957) 81.

2 Bäck, T. and Schwefel, H.-P., Evol. Comput., 1 (1993) 1.

3 Fogel, D.B., Evolutionary Computation: Toward a New Philosophy of Machine Intelligence, IEEE Press, Piscataway, NJ, 1995.

4 Holland, J.H., Adaptation in Natural and Artificial Systems, University of Michigan Press, Ann Arbor, MI, 1975.

5 Holland, J.H., Adaptation in Natural and Artificial Systems, 2nd ed., MIT Press, Cambridge, MA, 1992.

6 Fraser, A.S., In Von Foerster, H., White, J.D., Peterson, L.J. and Russell, J.K. (Eds.), Purposive Systems, Spartan Books, Washington, DC, 1968, pp. 15–23.

7 Goldberg, D.E., Genetic Algorithms in Search, Optimization and Machine Learning, Addison-Wesley, Reading, MA, 1989.

8 Fogel, D.B. and Stayton, L.C., BioSystems, 32 (1994) 171.

9 Davis, L. (Ed.) Handbook of Genetic Algorithms, Van Nostrand Reinhold, New York, NY, 1991.

10 Holland, J.H., Scientific American, July (1992) 44.

11 Forrest, S., Science, 261 (1993) 872.

12 Michaelewicz, Z., Genetic Algorithms + Data Structures = Evolution Programs, 2nd ed., Springer, New York, NY, 1994.

13 Cartwright, H.M., Applications of Artificial Intelligence in Chemistry, Oxford University Press, Oxford, UK, 1993.

14 Hibbert, D.B., Chemometr. Intell. Lab. Syst., 19 (1993) 277.

15 Lucasius, C.B. and Kateman, G., Chemometr. Intell. Lab. Syst., 19 (1993) 1.

16 Lucasius, C.B. and Kateman, G., Chemometr. Intell. Lab. Syst., 25 (1994) 99.

17 Lucasius, C.B. and Kateman, G., Comput. Chem., 18 (1994) 127.

18 Lucasius, C.B. and Kateman, G., Comput. Chem., 18 (1994) 137.

19 Fogel, L.J., Owens, A.J. and Walsh, M.J., Artificial Intelligence through Simulated Evolution, Wiley, New York, NY, 1966.

20 Yao, X. and Liu, Y., In Fogel, L.J., Angeline, P.J. and Bäck, T. (Eds.), Evolutionary Programming V: Proceedings of the Fifth Annual Conference on Evolutionary Programming, MIT Press, Cambridge, MA, 1996, in press.

21 Saravanan, N., Fogel, D.B. and Nelson, K.M., BioSystems, 36 (1995) 157.

22 Fogel, D.B., IEEE Trans. Neur. Netw., 5 (1994) 3.

23 Bäck, T., Evolutionary Algorithms in Theory and Practice, Oxford University Press, Oxford, UK, 1995.

24 Willett, P., Trends Biotechnol., 13 (1995) 516.

25 Devillers, J. (Ed.), Genetic Algorithms in Molecular Modelling, Academic Press, London, UK, 1996.

26 Judson, R.S., Jaeger, E.P., Treasurywala, A.M. and Peterson, M.L., J. Comput. Chem., 14 (1993) 1407.

27 Ghose, A.K., Jaeger, E.P., Kowalcyzk, P.J., Peterson, M.L. and Treasurywala, A.M., J. Comput. Chem., 14 (1993) 1050.

28 Treasurywala, A.M., Jaeger, E.P. and Peterson, M.L., J. Comput. Chem., 17 (1996) 1171.

29 Treasurywala, A.M., Jaeger, E.P., Lawless, M., Mathiowetz, A.M. and Castonguay, L.A., CONFANAL, 4: The Use of Molecular Dynamics as a Conformational Sampling Technique for Small Molecules. Paper presented at the Second Electronic Computational Chemistry Conference, Internet, November 1–30, 1995. Abstract at http:/hackberry.chem.niu.edu/ECCC2/abstracts6.html, to be published in J. Mol. Struct. (THEOCHEM).

30 Allen, F.H., Davies, J.E., Galloy, J.J., Johnson, O., Kennard, O., Macrae, C.F., Mitchell, E.M., Mitchell, G.F., Smith, J.M. and Watson, D.G., J. Chem. Inf. Comput. Sci., 31 (1991) 187.

31 Mohamadi, F., Richards, N.G.J., Guida, W.C., Liskamp, R., Lipton, M., Caufield, C., Chang, G., Hendrickson, T. and Still, W.C., J. Comput. Chem., 11 (1990) 440.

32 SYBYL, Tripos Inc., 1699 South Hanley Road, St. Louis, MO 63144, U.S.A.

33 Clark, D.E., Jones, G., Willett, P., Kenny, P.W. and Glen, R.C., J. Chem. Inf. Comput. Sci., 34 (1994) 197.

34 Hurst, T., J. Chem. Inf. Comput. Sci., 34 (1994) 190.

35 Brodmeier, T. and Pretsch, E., J. Comput. Chem., 15 (1994) 588.

36 Deaven, D.M. and Ho, K.M., Phys. Rev. Lett., 75 (1995) 288.

37 Maddox, J., Nature, 376 (1995) 209.

38 Herrmann, F. and Suhai, S., J. Comput. Chem., 16 (1995) 1434.

39 Stewart, J.J.P., J. Comput.-Aided Mol. Design, 4 (1990) 1.

40 McGarrah, D.B. and Judson, R.S., J. Comput. Chem., 14 (1993) 1385.

41 Sanderson, P.N., Glen, R.C., Payne, A.W.R., Hudson, B.D., Heide, C., Tranter, G.E., Doyle, P.M. and Harris, C.J., Int. J. Pept. Protein Res., 43 (1994) 588.

42 Payne, A.W.R. and Glen, R.C., J. Mol. Graph., 11 (1993) 74.

43 Lucasius, C.B., Blommers, M.J.J., Buydens, L.M.C. and Kateman, G., In Davis, L. (Ed.), Handbook of Genetic Algorithms, Van Nostrand Reinhold, New York, NY, 1991, pp. 251–281.

44 Blommers, M.J.J., Lucasius, C.B., Kateman, G. and Kaptein, R., Biopolymers, 32 (1992) 45.

45 Ogata, H., Akiyama, Y. and Kanehisa, M., Nucleic Acids Res., 23 (1995) 419.

46 Gultyaev, A.P., van Batenburg, F.H.D. and Pleij, C.W.A., J. Mol. Biol., 250 (1995) 37.

47 Gultyaev, A.P., van Batenburg, F.H.D. and Pleij, C.W.A., Nucleic Acids Res., 23 (1995) 3718.

48 van Batenburg, F.H.D., Gultyaev, A.P. and Pleij, C.W.A., J. Theor. Biol., 174 (1995) 269.

49 Benedetti, G. and Morosetti, S., Biophys. Chem., 55 (1995) 253.

50 Benedetti, G. and Morosetti, S., Biophys. Chem., 59 (1995) 179.

51 Unger, R. and Moult, J., Bull. Math. Biol., 55 (1993) 1183.

52 Böhm, G., Biophys. Chem., 59 (1996) 1.

53 Scheraga, H.A., Biophys. Chem., 59 (1996) 329.

54 Pedersen, J.T. and Moult, J., Curr. Opin. Struct. Biol., 6 (1996) 227.

55 Dandekar, T. and Argos, P., Protein Eng., 5 (1992) 637.

56 Dandekar, T. and Argos, P., J. Mol. Biol., 236 (1994) 844.

57 Dandekar, T. and Argos, P., J. Mol. Biol., 256 (1996) 645.

58 Dandekar, T. and Argos, P., Int. J. Biol. Macromol., 18 (1996) 1.

59 Judson, R.S., J. Am. Chem. Soc., 96 (1992) 10102.

60 Judson, R.S., Colvin, M.E., Meza, J.C., Huffer, A. and Gutierrez, D., Int. J. Quantum Chem., 44 (1992) 277.

61 Unger, R. and Moult, J., J. Mol. Biol., 231 (1993) 75.

62 Wetlaufer, D.B., Proc. Natl. Acad. Sci. USA, 70 (1973) 697.

63 Karplus, M. and Weaver, D.L., Nature, 260 (1976) 404.

64 Moult, J. and Unger, R., Biochemistry, 30 (1991) 3816.

65 Sun, S., Protein Sci., 2 (1993) 762.

66 Sun, S., Biophys. J., 69 (1995) 340.

67 Le Grand, S.M. and Merz, K.M., J. Global Optimis., 3 (1993) 49.

68 Le Grand, S.M. and Merz, K.M., Mol. Simul., 13 (1995) 299.

69 Weiner, S.J., Kollman, P.A., Case, D.A., Singh, U.C., Ghio, C., Alagona, G., Profeta Jr., S. and Weiner, P., J. Am. Chem. Soc., 106 (1984) 765.

70 Weiner, S.J., Kollman, P.A., Nguyen, D.T. and Case, D.A., J. Comput. Chem., 7 (1986) 230.

71 Schulze-Kremer, S., In Männer, R. and Manderick, B. (Eds.), Parallel Problem Solving from Nature 2, Elsevier/North Holland, Amsterdam, The Netherlands, 1992, pp. 391–400.

72 Schulze-Kremer, S. and Tiedemann, U., Parameterising Genetic Algorithms for Protein Folding Simulation. Presented at Colloquium on Molecular Bioinformatics, Institute of Electrical Engineers, London, U.K., February, 1994. Published in IEE Digest No. 1994/029, IEE Press, London, U.K., 1994.

73 Bowie, J.U. and Eisenberg, D., Proc. Natl. Acad. Sci. USA, 91 (1994) 4436.

74 Bowie, J.U., Lüthy, R. and Eisenberg, D., Science, 253 (1991) 164.

75 Elofsson, A., Le Grand, S.M. and Eisenberg, D., Proteins Struct. Funct. Genet., 23 (1995) 73.

76 Gunn, J.R., Monge, A., Friesner, R.A. and Marshall, C.H., J. Phys. Chem., 98 (1994) 702.

77 Gunn, J.R., J. Phys. Chem., 100 (1996) 3264.

78 Sun, S.J., Thomas, P.D. and Dill, K.A., Protein Eng., 8 (1995) 769.

79 Pedersen, J.T. and Moult, J., Proteins Struct. Funct. Genet., 23 (1995) 454.

80 Del Carpio, C.A., J. Chem. Inf. Comput. Sci., 36 (1996) 258.

81 Tufféry, P., Etchebest, C., Hazout, S. and Lavery, R., J. Comput. Chem., 14 (1993) 790.

82 Tufféry, P., Etchebest, C., Hazout, S. and Lavery, R., J. Biomol. Struct. Dyn., 8 (1991) 1267.

83 Ring, C.S. and Cohen, F.C., Isr. J. Chem., 34 (1994) 245.

84 Ring, C.S., Kneller, D.G., Langridge, R. and Cohen, F.E., J. Mol. Biol., 224 (1992) 685.

85 Bruccoleri, R.E. and Karplus, M., Macromolecules, 18 (1985) 2767.

86 Bruccoleri, R.E. and Karplus, M., Biopolymers, 26 (1987) 137.

87 Blaney, J.M. and Dixon, J.S., Perspect. Drug Discov. Design, 1 (1993) 301.

88 Lybrand, T.P., Curr. Opin. Struct. Biol., 5 (1995) 224.

89 Jones, G. and Willett, P., Curr. Opin. Biotechnol., 6 (1995) 652.

90 Dixon, J.S., In Wermuth, C.G. (Ed.), Trends in QSAR and Molecular Modelling 92 (Proceedings of the 9th European Symposium on Structure–Activity Relationships: QSAR and Molecular Modelling), ESCOM, Leiden, The Netherlands, 1993, pp. 412–413.

91 Oshiro, C.M., Kuntz, I.D. and Dixon, J.S., J. Comput.-Aided Mol. Design, 9 (1995) 113.

92 Kuntz, I.D., Blaney, J.M., Oatley, S.J., Langridge, R. and Ferrin, T.E., J. Mol. Biol., 161 (1982) 269.

93 Xiao, Y.L. and Williams, D.E., Comput. Chem., 18 (1994) 199.

94 Xiao, Y.L. and Williams, D.E., J. Phys. Chem., 98 (1994) 7191.

95 Judson, R.S., Jaeger, E.P. and Treasurywala, A.M., J. Mol. Struct., 308 (1994) 191.

96 Judson, R.S., Tan, Y.T., Mori, E., Melius, C., Jaeger, E.P., Treasurywala, A.M. and Mathiowetz, A., J. Comput. Chem., 16 (1995) 1405.

97 Clark, K.P. and Ajay, J. Comput. Chem., 16 (1995) 1210.

98 Schraudolph, N.N., Technical Report CS92-249, CSE Department, UC San Diego, La Jolla, CA, 1990.

99 Gehlhaar, D.K., Verkhivker, G.M., Rejto, P.A., Sherman, C.J., Fogel, D.B., Fogel, L.J. and Freer, S.T., Chem. Biol., 2 (1995) 317.

100 Gehlhaar, D.K., Verkhivker, G.M., Rejto, P.A., Fogel, D.B., Fogel, L.J. and Freer, S.T., In McDonnell, J.R., Reynolds, R.G. and Fogel, D.B. (Eds.), Evolutionary Programming IV: Proceedings of the Fourth Annual Conference on Evolutionary Programming, MIT Press, Cambridge, MA, 1995, pp. 615–627.

101 Meadows, R.P. and Hajduk, P.J., J. Biomol. NMR, 6 (1995) 41.

102 Jones, G., Willett, P. and Glen, R.C., J. Mol. Biol., 245 (1995) 43.

103 Duncan, B.S., Paper presented at the 13th Annual Conference of the Molecular Graphics Society: Molecular Graphics at the Frontier, Evanston, IL, July 1994. (An abstract of this paper was published in Chem. Design_Autom. News, 9 (1994) 35.

104 Duncan, B.S. and Olson, A.J., Biopolymers, 33 (1993) 219.

105 Strynadka, N.C.J., Eisenstein, M., Katchalski-Katzir, E., Shoichet, B.K., Kuntz, I.D., Abagyan, R., Totrov, M., Janin, J., Cherfils, J., Zimmerman, F., Olson, A., Duncan, B., Rao, M., Jackson, R., Sternberg, M. and James, M.N.G., Nat. Struct. Biol., 3 (1996) 233.

106 Shoichet, B.K. and Kuntz, I.D., Chem. Biol., 3 (1996) 151.

107 Murcko, M.A., In Lipkowitz, K.B. and Boyd, D.B. (Eds.), Reviews in Computational Chemistry, Vol. 11, Wiley, New York, NY, 1997, in press.

108 Clark, D.E., Murray, C.W. and Li, J., In Lipkowitz, K.B. and Boyd, D.B. (Eds.), Reviews in Computational Chemistry, Vol. 11, Wiley, New York, NY, 1997, in press.

109 Blaney, J.M., Dixon, J.S. and Weininger, D., Paper presented at the Molecular Graphics Society Meeting on Binding Sites:, Characterising and Satisfying Steric and Chemical Restraints, York, U.K., March 1993. (Abstracts of this and other papers presented at this meeting are available from Prof. R.E. Hubbard, Department of Chemistry, University of York, York, YO1 5DD, U.K. E-mail: rod@yorvic.york.ac.uk).

110 Glen, R.C., Chemical Genesis: A Genetic Algorithm for Automated Drug Design. Paper presented at the Molecular Graphics Society Meeting on Binding Sites: Characterising and Satisfying Steric and Chemical Restraints, York, U.K., March 1993.

111 Cramer, R.D., POSIT: A Second Generation De Novo Drug Discovery Tool. Paper presented at the Molecular Graphics Society Meeting on Binding Sites: Characterising and Satisfying Steric and Chemical Restraints, York, U.K., March 1993.

112 Cramer, R.D., Chem. Design Autom. News, 8:6 (1993) 32.

113 Slater, T. and Timms, D., J. Mol. Graph., 11 (1993) 248.

114 Glen, R.C. and Payne, A.W.R., J. Comput.-Aided Mol. Design, 9 (1995) 181.

115 LeapFrog, Tripos Inc., 1699 South Hanley Road, St. Louis, MO 63144, U.S.A.

116 Weininger, D., J. Chem. Inf. Comput. Sci., 28 (1988) 31.

117 Weininger, D., Weininger, A. and Weininger, J.L., J. Chem. Inf. Comput. Sci., 29 (1989) 97.

118 Westhead, D.R., Clark, D.E., Frenkel, D., Li, J., Murray, C.W., Robson, B. and Waszkowycz, B., J. Comput.-Aided Mol. Design, 9 (1995) 139.

119 Clark, D.E., Frenkel, D., Levy, S.A., Li, J., Murray, C.W., Robson, B., Waszkowycz, B. and Westhead, D.R., J. Comput.-Aided Mol. Design, 9 (1995) 13.

120 Clark, D.E. and Murray, C.W., J. Chem. Inf. Comput. Sci., 35 (1995) 914.

121 Jones, D.T., Protein Sci., 3 (1994) 567.

122 Jones, D.T., Moody, C.M., Uppenbrink, J., Viles, J.H., Doyle, P.M., Harris, C.J., Pearl, L.H., Sadler, P.J. and Thornton, J.M., Proteins Struct. Funct. Genet., 24 (1996) 502.

123 Hellinga, H.W. and Richards, F.M., Proc. Natl. Acad. Sci. USA, 91 (1994) 5803.

124 Desjarlais, J.R. and Handel, T.M., Protein Sci., 4 (1995) 2006.

125 Ebeling, M. and Nadler, W., Proc. Natl. Acad. Sci. USA, 92 (1995) 8798.

126 Schneider, G., Schuchhardt, J. and Wrede, P., Comput. Appl. Biosci., 10 (1994) 635.

127 Schneider, G. and Wrede, P., Biophys. J., 66 (1994) 335.

128 Schneider, G., Schuchhardt, J. and Wrede, P., Biol. Cybern., 73 (1995) 245.

129 Schneider, G., Schuchhardt, J. and Wrede, P., Biophys. J., 68 (1995) 434.

130 Schneider, G., Schuchhardt, J. and Wrede, P., Biol. Cybern., 74 (1996) 203.

131 Venkatasubramanian, V., Chan, K. and Caruthers, J.M., J. Chem. Inf. Comput. Sci., 35 (1995) 188.

132 Vedani, A., Zbinden, P., Snyder, J.P. and Greenidge, P.A., J. Am. Chem. Soc., 117 (1995) 4987.

133 Greenidge, P.A., Merz, A. and Folkers, G., J. Comput.-Aided Mol. Design, 9 (1995) 473.

134 Hahn, M., J. Med. Chem., 38 (1995) 2080.

135 Hahn, M. and Rogers, D., J. Med. Chem., 38 (1995) 2091.

136 Walters, D.E. and Hinds, R.M., J. Med. Chem., 37 (1994) 2527.

137 Brooks, B.R., Bruccoleri, R.E., Olafson, B.D., States, D.J., Swaminathan, S. and Karplus, M., J. Comput. Chem., 4 (1983) 197.

138 Good, A.C. and Mason, J.S., In Lipkowitz, K.B. and Boyd, D.B. (Eds.), Reviews in Computational Chemistry, Vol. 7, VCH, New York, NY, 1996, pp. 67–117.

139 Sheridan, R.P., Nilakantan, R., Dixon, J.S. and Venkataraghavan, R., J. Med. Chem., 29 (1986) 899.

140 Mayer, D., Naylor, C.B., Motoc, I. and Marshall, G.R., J. Comput.-Aided Mol. Design, 1 (1987) 3.

141 Martin, Y.C., Bures, M.G., Danaher, E.A., DeLazzar, J., Lico, I. and Pavlik, P.A., J. Comput.-Aided Mol. Design, 7 (1993) 83.

142 Hodgkin, E.E., Miller, A. and Whittaker, M., J. Comput.-Aided Mol. Design, 7 (1993) 515.

143 Golender, V.E. and Vorpagel, E.R., In Kubinyi, H. (Ed.) 3D QSAR in Drug Design: Theory, Methods and Applications, ESCOM, Leiden, 1993, ESCOM, Leiden, The Netherlands, 1993, pp. 137–149.

144 Ghose, A.K., Logan, M.E., Treasurywala, A.M., Wang, H., Wahl, R.C., Tomczuk, B.E., Gowravaram, M.R., Jaeger, E.P. and Wendoloski, J.J., J. Am. Chem. Soc., 117 (1995) 4671.

145 Barnum, D., Greene, J., Smellie, A.S. and Sprague, P., J. Chem. Inf. Comput. Sci., 36 (1996) 563.

146 Jones, G., Willett, P. and Glen, R.C., J. Comput.-Aided Mol. Design, 9 (1995) 532.

147 GASP, Tripos Inc., 1699 South Hanley Road, St. Louis, MO 63144, U.S.A.

148 Ullmann, J.R., J. Assoc. Comput. Machin., 23 (1976) 31.

149 Brint, A.T. and Willett, P., J. Mol. Graph., 5 (1987) 49.

150 Brown, R.D., Jones, G., Willett, P. and Glen, R.C., J. Chem. Inf. Comput. Sci., 34 (1994) 63.

151 Luke, B.T., Application of Genetic Methods to Substructure Searches. Paper presented at the 209th ACS National Meeting, Anaheim, CA, April 1995.

152 Luke, B.T., private communication.

153 Brint, A.T. and Willett, P., J. Chem. Inf. Comput. Sci., 27 (1987) 152.

154 Fontain, E., J. Chem. Inf. Comput. Sci., 32 (1992) 748.

155 Brown, R.D., Downs, G.M., Jones, G. and Willett, P., J. Chem. Inf. Comput. Sci., 34 (1994) 47.

156 Brown, R.D., Downs, G.M., Willett, P. and Cook, A.P.F., J. Chem. Inf. Comput. Sci., 32 (1992) 522.

157 Wagener, M. and Gasteiger, J., Angew. Chem. Int. Ed. Engl., 33 (1994) 1189.

158 Wild, D.J. and Willett, P., J. Chem. Inf. Comput. Sci., 36 (1996) 159.

159 Hibbert, D.B., Chem. Intell. Lab. Syst., 19 (1993) 35.

160 Lushnikov, D.E. and Sello, G., J. Chem. Inf. Comput. Sci., 35 (1995) 1060.

161 Rogers, D. and Hopfinger, A.J., J. Chem. Inf. Comput. Sci., 34 (1994) 854.

162 C^{2}GA, Molecular Simulations Inc., 9685 Scranton Road, San Diego, CA 92121, U.S.A.

163 Luke, B.T., J. Chem. Inf. Comput. Sci., 34 (1994) 1279.

164 So, S.S. and Karplus, M., J. Med. Chem., 39 (1996) 1521.

165 Selwood, D.L., Livingstone, D.J., Comley, J.C., O'Dowd, B.A., Hudson, A.T., Jackson, P., Jandu, K.S., Rose, V.S. and Stables, J.N., J. Med. Chem., 33 (1990) 136.

166 Kubinyi, H., Quant. Struct.–Act. Relatsh., 13 (1994) 285.

167 Kubinyi, H., Quant. Struct.–Act. Relatsh., 13 (1994) 393.

168 Kubinyi, H., J. Chemometr., 10 (1996) 119.

169 Leardi, R., J. Chemometr., 8 (1994) 65.

170 Vankeerberghen, P., Smeyersverbeke, J., Leardi, R., Karr, C.L. and Massart, D.L., Chemometr. Intell. Lab. Syst., 28 (1995) 73.

171 Wise, B.M., Holt, B.R., Gallagher, N.B. and Lee, S., Chemometr. Intell. Lab. Syst., 30 (1995) 81.

172 Lucasius, C.B., De Weijer, A.P., Buydens, L.M.C. and Kateman, G., Chemometr. Intell. Lab. Syst., 19 (1993) 337.

173 De Weijer, A.P., Lucasius, C.B., Buydens, L., Kateman, G., Heuvel, H.M. and Mannee, H., Anal. Chem., 66 (1994) 23.

174 De Weijer, A.P., Buydens, L., Kateman, G. and Heuvel, H.M., Chemometr. Intell. Lab. Syst., 28 (1995) 149.

175 Lucasius, C.B., Dane, A.D. and Kateman, G., Anal. Chim. Acta, 287 (1993) 647.

176 Lucasius, C.B., Beckers, M.L.M. and Kateman, G., Anal. Chim. Acta, 286 (1993) 135.

177 Jouanrimbaud, D., Massart, D.L., Leardi, R. and Denoord, O.E., Anal. Chem., 67 (1995) 4295.

178 Marques, R.M.L., Schoenmakers, P.J., Lucasius, C.B. and Buydens, L., Chromatographia, 36 (1993) 83.

179 Parczewski, A., Lucasius, C.B. and Kateman, G., Fresenius J. Anal. Chem., 348 (1994) 626.

180 Hartnett, M.K., Bos, M., Vanderlinden, W.E. and Diamond, D., Anal. Chim. Acta, 316 (1995) 347.

181 Gallop, M.A., Barrett, R.W., Dower, W.J., Fodor, S.P.A. and Gordon, E.M., J. Med. Chem., 37 (1994) 1233.

182 Gordon, E.M., Barrett, R.W., Dower, W.J., Fodor, S.P.A. and Gallop, M.A., J. Med. Chem., 37 (1994) 1385.

183 Terrett, N.K., Gardner, M., Gordon, D.W., Kobylecki, R.J. and Steele, J., Tetrahedron, 50 (1995) 8135.

184 Thompson, L.A. and Ellman, J.A., Chem. Rev., 96 (1996) 555.

185 Gordon, E.M., Gallop, M.A. and Patel, D.V., Acc. Chem. Res., 29 (1996) 144.

186 Rees, P., Sci. Comput. World, February (1996) 25.

187 Martin, E.J., Blaney, J.M., Siani, M.A., Spellmeyer, D.C., Wong, A.K. and Moos, W.H., J. Med. Chem., 38 (1995) 1431.

188 Holland, J.D., Ranade, S.S. and Willett, P., Quant. Struct.–Act. Relatsh., 14 (1995) 501.

189 Shemetulskis, N.E., Dunbar Jr., J.B., Dunbar, B.W., Moreland, D.W. and Humblet, C., J. Comput.-Aided Mol. Design, 9 (1995) 407.

190 Boyd, S.M., Beverley, M., Norskov, L. and Hubbard, R.E., J. Comput.-Aided Mol. Design, 9 (1995) 417.

191 Sadowski, J., Wagener, M. and Gasteiger, J., Angew. Chem. Int. Ed. Engl., 34 (1996) 2674.

192 Ashton, M.J., Jaye, M.C. and Mason, J.S., Drug Discov. Today, 1 (1996) 71.

193 Moreau, G. and Turpin, C., Analusis, 24 (1996) 17.

194 Brown, R.D. and Martin, Y.C., J. Chem. Inf. Comput. Sci., 36 (1996) 572.

195 Kauffman, S.A. and Macready, W.G., J. Theor. Biol., 173 (1995) 427.

196 Sheridan, R.P. and Kearsley, S.K., J. Chem. Inf. Comput. Sci., 35 (1995) 310.

197 Weber, L., Wallbaum, S., Broger, C. and Gubernator, K., Angew. Chem. Int. Ed. Engl., 34 (1995) 2280.

198 Singh, J., Ator, M.A., Jaeger, E.P., Allen, M.P., Whipple, D.A., Soloweij, J.E., Chowdhary, S. and Treasurywala, A.M., J. Am. Chem. Soc., 118 (1996) 1669.

199 Xiao, Y. and Williams, D.E., Chem. Phys. Lett., 215 (1993) 17.

200 Hartke, B., J. Phys. Chem., 97 (1993) 9973.

201 Hartke, B., Chem. Phys. Lett., 240 (1995) 560.

202 Gregurick, S.K., Alexander, M.H. and Hartke, B., J. Chem. Phys., 104 (1996) 2684.

203 Mestres, J. and Scuseria, G.E., J. Comput. Chem., 16 (1995) 729.

204 Zeiri, Y., Phys. Rev., A51 (1995) 2769.

205 May, A.C.W. and Johnson, M.S., Protein Eng., 7 (1994) 475.

206 May, A.C.W. and Johnson, M.S., Protein Eng., 8 (1995) 873.

207 Needleman, S.B. and Wunsch, C., J. Mol. Biol., 48 (1970) 444.

208 Taylor, W.R. and Orengo, C.A., J. Mol. Biol., 208 (1989) 1.

209 Wehrens, R., Lucasius, C., Buydens, L. and Kateman, G., J. Chem. Inf. Comput. Sci., 33 (1993) 245.

210 Zimmerman, D.E. and Montelione, G.T., Curr. Opin. Struct. Biol., 5 (1995) 664.

211 Chang, G. and Lewis, M., Acta Crystallogr., D50 (1994) 667.

212 Tam, K.Y. and Compton, R.G., J. Appl. Crystallogr., 28 (1995) 640.

213 Yiu, K.F.C., Tam, K.Y. and Tsang, S.C., private communication.

214 Miller, S.T., Hogel, J.M. and Filman, D.J., Acta Crystallogr., D52 (1996) 235.

215 Rodriguez, J.H., Ok, H.N., Xia, Y.M., Debrunner, P.G., Hinrichs, B.E., Meyer, T. and Packard, N.H., J. Phys. Chem., 100 (1996) 6849.

216 Bush, T.S., Catlow, C.R.A. and Battle, P.D., J. Mater. Chem., 5 (1995) 1269.

217 Rossi, I. and Truhlar, D.G., Chem. Phys. Lett., 233 (1995) 231.

218 Vivarelli, F., Giusti, G., Villani, M., Campanini, R., Fariselli, P., Compiani, M. and Casadio, R., Comput. Appl. Biosci., 11 (1995) 253.

219 Pal, K.F., Biol. Cybern., 73 (1995) 335.

220 Kirkpatrick, S., Gellatt Jr., C.D. and Vecchi, M.P., Science, 220 (1983) 671.

221 Merkle, L.D., Gates Jr., G.H. and Lamont, G.H., Conformational Search Using a Parallel Fast Messy GA with Migration and Parallel Selection. Paper presented at the, 209th ACS National Meeting, Anaheim, CA, U.S.A., April 1995.

222 Glover, F. and Laguna, M., In Reeves, C.R. (Ed.), Modern Heuristic Techniques for Combinatorial Problems, Blackwell Scientific Publications, Oxford, U.K., 1993, pp. 70–150.

223 Cvijovic, D. and Klinowski, J., Science, 267 (1995) 664.

224 Myung, H. and Kim, J.H., BioSystems, 38 (1996) 29.

225 Montoya, F. and Dubois, J.M., Europhys. Lett., 22 (1993) 79.

226 Kido, T., Takagi, K. and Nakanishi, M., Informatica, 18 (1994) 399.