Journal of Computer-Aided Molecular Design, 9 (1995) 181–202
ESCOM

J-CAMD 282

A genetic algorithm for the automated generation of molecules within constraints

R.C. Glen[a,*] and A.W.R. Payne[b]

[a]*Department of Physical Sciences and* [b]*Information Services Division, Wellcome Research Laboratories, Langley Court, Beckenham, Kent BR3 3BS, U.K.*

Received 13 July 1994
Accepted 22 September 1994

Keywords: De novo drug design; Evolution; Automated structure generation; 3D database; Expert system

Summary

A genetic algorithm has been designed which generates molecular structures within constraints. The constraints may be any useful function, for example an enzyme active site, a pharmacophore or molecular properties from pattern recognition or rule-induction analyses. The starting point may be random or may utilise known molecules. These are modified to 'grow' into families of structures which, using the evolutionary operators of selection, crossover and mutation evolve to better fit the constraints. The basis of the algorithm is described together with some applications in lead generation, 3D database construction and drug design. Genetic algorithms of this type may have wider applications in chemistry, for example in the design and optimisation of new polymers, materials (e.g. superconducting materials) or synthetic enzymes.

Introduction

The design of new drugs involves the discovery of novel chemical entities which have a desired biological profile. A complex relationship exists between the chemico-physical properties of a molecule and the absorption, metabolism, bioavailability, activity and excretion profiles observed in vivo. In order to create new molecules with the desired profiles, hundreds of molecules may be synthesised and tested in many biological test systems with the objective of finding a series that is suitable for development as a medicine.

In recent years, computer-aided drug design methods have addressed some of the simpler aspects of this problem [1]. For example, a pharmacophore may be deduced, an enzyme active site elucidated or a QSAR (quantitative structure–activity relationship) constructed to account for absorption or metabolism.

This central problem of creating new chemical structures is a complex task, however, in many ways it is rule-based and many of the fundamental operations may be embedded in an expert system. For example, the structures must satisfy valency rules (no hexavalent carbons are allowed!). One example of this approach is DENDRAL [2], an expert system for the simulation of mass spectra. The rules adopted for the generation of new structures may be simple and hence result in simple structural modifications, e.g. starting with methane, convert one of the hydrogen atoms to a carbon and fill the empty valency with hydrogen. This gives ethane. A repeat of the process gives propane. Obviously, as more complex structures are conceived, the rules become more involved. By limiting the type of structures that can be created (e.g. no transition metals), a manageable and consistent set of rules for molecule generation may be deduced.

Once an algorithm for the generation of chemical structures has been created, it is necessary to have an optimisation method and constraints which direct the structural modifications towards some goal. For example, the molecule may be required to have a molecular weight in a particular range. If this was between 28–32 Da, then the simple process described above would stop at ethane, which has a molecular weight of 30 Da.

The process of designing drug-like molecules requires the generation of complex structures within many constraints. The '*Chemical Genesis*' algorithm described here

*To whom correspondence should be addressed.

uses a series of rules which consistently produce realistic molecules in three dimensions. These molecules are then evaluated using constraints based on calculated molecular properties which are of use in the prediction of biological activity. Previously, the related problem of molecular similarity [3] was used to evaluate different constraint paradigms with a view to incorporating the most useful properties in the constrained molecular generation algorithm. The assumption that molecular properties useful in similarity searching would be a good starting point for optimisation of molecular structures within constraints seemed a logical one.

A genetic algorithm [4] based on Darwinian selection [5], which uses the operators selection, crossover and mutation, has been adapted to optimise chemical structures such that they are evolved to better satisfy the constraints. This results in the creation of families of structures, some of which may be promising targets for chemical synthesis. In addition, the structure-generation algorithm can produce very large and diverse sets of reasonable chemical structures for searching by 3D database programmes [6]. These databases can then be searched using pharmacophore or active-site constraints.

The benefits of automated structure generation have been recognised by many other groups and a number of recent publications reveal a diverse set of approaches to this problem [7–9]. The application of a genetic algorithm offers fast and powerful optimisation properties as well as the generation of a diverse set of possible structures. Indeed, each run, due to the random nature of the algorithm, generates a new family of structures.

Methods

3D rule-based molecular construction and optimisation using a genetic algorithm

A genetic algorithm (GA) [4] has been devised and implemented that attempts to construct a series of molecules to best fit a set of constraints. The genetic algorithm uses a blind search strategy, requiring no knowledge of the properties of the function which it is optimising, thus enabling the algorithm to be applied to a variety of molecular optimisation problems. We have previously used similar implementations in molecular similarity, pharmacophore elucidation, 3D database searching and conformational analysis [3,10,11]. The constraints may be any useful function, e.g. intramolecular distance constraints between key functional groups, the properties of a protein active site, or shape similarity and charge distribution of active analogues.

The algorithm uses a concept of molecular assembly via substructures which enables novel combinations to result in new molecules which satisfy chemical rules. Substructures are analogous to the amino acid building blocks of proteins (in fact, they can be components of amino acids if we wish to construct a peptide) and these are assembled into novel molecules whose molecular properties are optimised to fit the constraints. In this context, a substructure has a special definition. It is defined as the minimum molecular fragment which is joined to the rest of the molecule only by single bonds. For example, a chlorine substituent, an oxygen in a furanose ring, a methylene in a chain or -CH=N- would be classed as substructures. These can be easily manipulated without interfering with valency rules. A list is kept of the substructures in the molecule and used during modification to maintain the correct valency. The list is updated after any alterations in molecular structure. The assembly of molecular graphs (which this list is) can be in one (e.g. a SMILES [12] string), two (e.g. a chemical-structure diagram) or three dimensions (e.g. a file of atom co-ordinates and bond connectivities). Each of these may be amenable to manipulation by a genetic algorithm. However, molecular properties are dependant on the three-dimensional structure, so at some stage a realistic 3D structure must be generated and its properties calculated. This representation is difficult to represent as a binary string (a useful code used previously for structural manipulation by a GA [3]) and would require rotational and translational fitting to the constraints (introducing additional degrees of freedom). Therefore, the structure manipulations implemented here are made upon the 3D molecular structures themselves. This differs from the situation seen in e.g. DNA coding for amino acids in a protein where modification of the genetic information (DNA) results in changed structure and function (the protein). The analogous situation here is to directly modify the protein (like in site-directed mutagenesis).

When the genetic operators, crossover, mutation and selection are applied to a population of these new molecules, substructure combinations arise which better fit the constraints. One very important aspect is that realistic conformations of the molecules fitting the constraints are selectively bred, so the problem of conformational flexibility does not arise to the extent it does in e.g. 3D database searches.

Definition of constraints

The first step is to define a set of constraints within which families of molecules may be generated.

In the experiments involving the generation of one molecule from another (or a series of molecules), fitting to a pharmacophore, or optimisation within a set of molecular properties, constraints are calculated which help to describe how similar the new molecule is to the target. Here, molecular similarity is the key. In the case of optimisation within an active site or receptor, e.g. an enzyme, molecular complementarity is the key. Thus, the target constraints must be modified for this special case.

These modifications are described later.

The constraints are composed of three sets: scalar, surface and grid-based. The scalar constraints are single values (like surface area), while surface and grid properties are spatially dependant and are calculated on a series of points around the molecules. The constraints to be calculated, their ranges and weights are listed in a *constraints definition file.*

Scalar constraints

The currently available scalar constraints are listed in Table 1. Each of these constraints is defined by a range. If a molecule resides outside the range, then a penalty is applied proportional to the weight. For example, 'molecular weight: 350 > 370 ; 5.0' could be an entry in the constraints definition file and means that a molecular weight is desired between 350–370 Da. Molecules outside this range have (weight × difference from the centre of the constraint range) added to their total score. Some of these constraints are self-explanatory, some require explanation.

A number of scalar constraints were developed during the testing of the algorithm to correct for the occurrence of unreasonable structures.

The dimensions of the active volume describe the volume of interest, for example within an active site. These dimensions define a box inside which the molecule must reside, or a penalty is added. This prevents molecules 'escaping' from surface or grid constraints and evolving in space where no constraints are present. Molecules which exhibit this behaviour are 'sterile'.

The calculated logP is an approximation of the octanol/water partition coefficient. This property often relates to absorption and distribution properties of drug molecules and is very useful in defining the combination of hydrophobic and hydrophilic characteristics that molecules require to be pharmacologically active. Calculated LogP is estimated from a regression equation developed in a manner similar to that described in Refs. 14 and 15 (unpublished data).

TABLE 1
SCALAR CONSTRAINTS USED IN THE GA ALGORITHM

1	Molecular weight range (Da) [13]
2	Dimensions of 'active' volume (X,Y,Z Å)
3	Calculated LogP [14,15]
4	Number of bonds
5	Number of atoms
6	Number of rotatable bonds
7	Dipole moment, calculated from the excess atomic charges (D)
8	Molecular volume (A^3) [16]
9	Surface area (A^2) [17]
10	Polarizability [18]
11	Principal ellipsoid axes (A) [19]
12	Molecular mechanics steric strain per atom (kcal mol^{-1})
13	Feasibility of the structure
14	Chiral centres
15	Conjugated double bonds

$$\begin{aligned}\text{CLogP} = &-0.6911 \times 10^{-4}\ (0.6552 \times 10^{-5})\ \text{AREA}^2 \\ &+ 0.6330 \times 10^{-1}\ (0.3958 \times 10^{-2})\ \text{AREA} \\ &- 8.512(0.9995)\ \text{OVALITY} \\ &+ 1.116(0.2231)\ \text{AlkaneIndicatorVariable} \\ &- 2.256(0.7397 \times 10^{-1})\ \Sigma\text{Partial} - \text{Charges(N,O)} \\ &- 1.462(0.2305)\ \Sigma\text{Partial} - \text{Charges(N)} \\ &+ 6.099(0.9299)\end{aligned}$$

This has recently been replaced by a much simpler equation, based on atom charge and polarizability [18], that will be reported elsewhere. The equation presented here gave a correlation coefficient of r = 0.951, F = 309.4, n = 220 for a diverse set of hetero-atomic structures (including zwitterions).

The number of bonds and atoms is particularly useful in controlling the ratio of ring to chain structures. A lower number of bonds than atoms favours ring formation. This prevents long 'stringy' structures.

Drug-like molecules having high affinity for a receptor are likely to have a small number of rotatable bonds. This minimises the entropy loss on binding. Each rotatable bond (indeed, each additional degree of freedom) is thought to contribute about kT/2 in energy to the entropy term, thus destabilising binding to the receptor. Therefore, the number of rotatable bonds is included as a constraint.

The 'feasibility' is calculated from a lookup table of unlikely four-atom combinations, e.g. X-X-OSP3-F, NSP3-NSP3-NSP3-NSP3, X-OSP3-OSP3-OSP3, etc. This list is expanded as unlikely (or chemically correct but unstable) structures arise. One goal would be to connect the molecule-generation algorithm to a reaction database and to use ease of synthesis as a constraint (other groups developing structure-generation algorithms are including this feature [7,12]). Interestingly, this file may be updated during the course of a run and thus, if unlikely structures appear, the 'environment' may be changed by inclusion of these structures in the file. The new molecules thus evolve away from these unwanted structures.

The chirality constraint was introduced to define the range of chiral centres allowed in molecules. For synthetic as well as pharmacological reasons, it is desirable to have as few chiral centres as possible. In this example, a simple method was implemented which sums the atomic numbers of atoms connected to tetrahedral atoms. These are then compared and, if four different connections are detected, this is assumed to be a chiral centre. More complex chiral centres are obviously not detected; however, this method is very fast and detects the most obvious centres.

The torsion angles are searched for the occurrence of pairs of double bonds, a simple measure of aromaticity. This is desirable in many cases for the production of molecules that are (in general) easier to synthesise.

The molecular mechanics strain per atom is calculated to remove highly strained combinations of fragments or an unlikely geometry which may arise during the evol-

ution of structures. For example, interpenetration of fragments is eliminated as the steric strain/atom is large. The strain is divided by the number of atoms so that all sizes of molecules are equally treated (larger molecules usually have larger calculated strain energy, simply by virtue of their greater size). Due to the diverse nature of structures encountered, structural geometry optimisation by molecular mechanics was developed to cope with almost any eventuality.

Surface properties

The shape and charge distribution may be defined by mapping suitable molecular properties onto an appropriate surface. A series of arrays with locations corresponding to each surface point for each molecular property is calculated.

The template molecule (or molecules) is centred on the origin and surrounded by a surface of points [20,21]. The surface may be configured to be spherical, with a defined radius and point density, or it may be more complex, for example, a solvent-accessible surface at an extra radius. The shape of the template or target molecule is described by a set of distances; each distance is associated with one of the points on the surface. The shape algorithm calculates the distance from a surface point to the nearest atom in the molecule minus the van der Waals radius of that atom. Thus, each point on the surface has an associated point to the molecule–surface distance.

Another approach is to probe the surface with a test atom or molecule and to calculate an interaction energy, for example using a methane or water molecule. This is the approach taken in the GRID programme (Goodford

TABLE 2
DEFINITION OF ATOM TYPES

Atomic number	Atom type name	Number of connections	vdW[a]	D/A[b]	Formal charge	Molecular orbital symmetry and occupancy [27]	Hydrophobic[c]
Carbon							
6	CSP3	4	1.70	N	0	te te te te	y
6	CSP2	3	1.70	N	0	tr tr tr pi	y
6	CSP	2	1.78	N	0	di di pi pi	y
6	CAR	3	1.70	N	0	tr tr tr pi	y
Oxygen							
8	OSP3	2	1.52	B	0	te te te^2 te^2	n
8	OSP2	1	1.50	A	0	tr tr^2 tr^2 pi	n
8	OAR	2	1.50	A	0	tr tr tr^2 pi^2	n
8	O2_N	2(1)	1.60	A	−1.0	tr tr^2 tr^2 $pi^{3/2}$	n
Nitrogen							
7	NSP3	3	1.55	B	0	te te te te^2	n
7	NTRI	3	1.55	D	0	tr tr tr pi^2	n
7	NAM	3	1.55	D	0	tr tr tr pi^2	n
7	NSP2	2	1.55	B	0	tr tr tr^2 pi	n
7	NSP	1	1.60	A	0	di di^2 pi pi	n
7	N3_P	3(4)	1.45	D	1.0	te te te te	n
Phosphorus							
15	PTRI	3	1.80	N	0	di di pi pi	n
15	PTET	4	1.80	N	0	te te te te	n
Sulfur							
16	SSP3	2	1.83	N	0	te te te^2 te^2	n
16	STRI	3	1.83	N	0	te te te pi te^2	n
16	STET	4	1.83	N	0	te te pi pi te te	n
16	SAR	2	1.83	N	0	tr tr tr^2 pi^2	n
16	SSP2	1	1.83	N	0	tr^2 tr^2 te pi	n
Hydrogen							
1	H	1	1.20	N	0	s	–
Halides							
9	F	1	1.40	A	0	te te^2 te^2 te^2	n
17	CL	1	1.76	A	0	te te^2 te^2 te^2	n
35	BR	1	1.95	A	0	te te^2 te^2 te^2	n
53	IODI	1	2.15	A	0	te te^2 te^2 te^2	n
Lone pair							
0	LP	1	0.50	N	0		–

[a] vdW = van der Waals radius [32].
[b] D/A = hydrogen-bond donor or acceptor.
[c] Hydrophobic = simple assignment of hydrophobicity of CX group (X = atom type) for an approximation of desolvation onto a protein surface.

[22]), a method particularly useful in defining favoured binding regions in proteins. This proved to be very time-consuming, as the orientation of the probe needs to be computed at each grid point for each molecule.

The charge distribution of the molecule could be calculated using any preferred method, for example ab-initio [23,24] or semiempirical [25] molecular orbital methods or partial equalisation of orbital electronegativity (PEOE) [26–29]. Electrostatic potential or distributed multipole-derived charges would be preferred, but are presently too time-consuming to calculate [30]. Since the aim here is to develop GA methods to evolve novel molecules within constraints, and as charge distributions change during the course of molecular mutation (and conformation), a method of charge recalculation is required which is very fast (we may have to evaluate hundreds of thousands of molecules during the course of a run). This currently rules out all the other methods (including probably the most desirable, similarity of the wave function [31]), apart from PEOE.

In our algorithm, the atoms in each molecule have been defined by atom types (analogous to force-field atom types, Table 2). The associated electronegativity parameters (required for the calculation of atom charges) are given in Table 3 for each atom type. Only the sigma bonding contributions to atom charges have been used here. This calculation method has deficiencies, particularly for aromatic systems, but is very fast and dipole moments are generally reproduced to within about 10% of the experimental values [26].

Two methods for expressing the charge distribution are used: the first method calculates the electrostatic potential [35–37] at each of the points on the surface using the atom-centred charges and atomic screening constants (the dielectric constant may also be varied). Alternatively, the atom-centred charge density of the non-hydrogen atom nearest a surface point may be associated with that point.

There are obviously fragments, e.g. benzoic acid, which will be ionised at physiological pH (pH 6.8–7.2). To take some aspects of this into account, carboxylic acids, phosphoric acids, sulfonic acids, amines and 2-aminopyridine-like (=N–C–NH_2) fragments (e.g. guanidinium) are detected and ionised before the charge calculation. This obviously changes the logP calculation significantly and also better reflects Coulombic interactions. The PEOE method results (on protonation) in charge accumulating on the amine hydrogens, which would not be detected by the charge screen in the grid and surface calculations (this takes into account non-hydrogen atoms). In this case, the N3_P atom type is detected and used to set the appropriate charge surface points to TRUE.

In all cases, the GA works on the neutral molecules; thus, the molecule is returned to its neutral state on completion of the constraint calculations. The co-ordinates of the points on the surface and the associated shape, partial

TABLE 3
PARTIAL EQUALISATION OF ORBITAL ELECTRONEGATIVITY PARAMETERS[a]

Atom type	1	2	3
CSP3	0.68	7.98	19.04
CSP2	0.98	8.79	19.64
CSP	1.67	10.39	20.56
CAR	0.98	8.79	19.64
OSP3	3.06	14.18	29.51
OSP2	3.75	17.07	31.32
OAR	3.66	10.04	30.50[b]
O2_N	3.06	14.18	29.51
NSP3	2.08	11.54	23.72
NTRI	2.46	12.32	24.86
NAM	2.46	12.32	24.86
NSP2	3.42	12.87	24.86
NSP	3.73	15.68	27.12
N3_P	2.08	11.54	23.72
PTRI	2.39	10.05	20.53[c]
PTET	1.62	6.23	18.10[d]
SSP3	2.39	10.14	20.65
STRI	2.40	7.10	20.64[e]
STET	2.40	7.10	20.64[e]
SAR	2.75	10.88	21.42
SSP2	3.48	12.37	23.20
H	0.37	7.18	12.85[f]
F	3.15	14.07	34.90[g]
CL	3.37	12.19	30.29[h]
BR	3.33	10.92	15.89[i]
IODI	3.17	11.34	14.79[j]

[a] See Ref. 26. There are three parameters required for σ-bonds (and three for π if the π-component is to be included in the calculation). The first three relate to the electronegativities of the negative, neutral and positive species. These are Eqs. 4–6 in Ref. 26. The first column is computed from $1/2E_{iv}^{o}$ (these parameters are annotated as $1/2E_v$ in Ref. 27). The values for the neutral states of Br and I are in Ref. 29. The second column is $1/2(I_o^{iv} + E_o^{iv})$ (and is annotated as $1/2(I_v + E_v)$) from Ref. 26. The third column is from Ref. 29. The value is computed as $1/2(I_{iv}^{+} + E_{iv}^{+})$.
The values for the positive ions of F and Cl are not available. In accordance with Gasteiger et al. [26], we have used the second ionisation potential minus the first ionisation potential [33] and multiplied it by a factor of 1.80 to obtain the desired p-character of the orbitals. These values are then multiplied by the percent s-p characters quoted (in Table 1 of Ref. 26) to give the values in the table.
It should be noted that the hybridisation states for P and S in the references by Hinze et al. [27–29] are not the required states for phosphate (4) and sulfur (3) and so estimates have been used based on the values for the 2-substituted species of sulfur and 3-substituted phosphorus. For atom type PTRI the didipi²pi values are used.

[b] Used 60% of the second parameter to reproduce the dipole moment (0.7 D) [34] of furan.

[c] Used didipi²pi values.

[d] Used tetetete² values (the second parameter is 70% to better fit dipole moments).

[e] Used te²te²tete values (the second parameter is 70% to better fit dipole moments).

[f] H^+ has no Ip. This value was estimated [33] by extrapolation from H^- and $H_{neutral}$.

[g] Used 15% s-character for F.

[h] Used 30% s-character for Cl.

[i] Used 30% s-character for Br.

[j] Used 20% s-character for I. Gasteiger et al. [26] use 13, 17, 17 and 23% s-character for F, Cl, Br and I; the above values were used to better reproduce dipole moments and Mulliken charge distributions in AM1 [25].

charge, electrostatic potential and probe interaction energy descriptions (if there is a supercomputer available!) of the target molecule(s) are now available for comparison with new molecules.

Grid properties

Properties may be calculated on a grid as well. In this method, the molecule is centred at the origin and a grid of points (typically $30 \times 30 \times 30$ points on an orthogonal lattice at 0.5 Å apart for most drug-sized systems) is generated and the grid co-ordinates stored. Molecular shape is described by assigning a logical value to each point. A point inside the molecule is TRUE and one outside the molecule is FALSE (using standard van der Waals radii [32] of atoms).

Atoms are defined as hydrogen-bond donors (D), e.g. –N(H)–, acceptors (A), e.g. =O, or neither (N), e.g. CH_3, and are identified from their atom types (Table 2). (B) is both donor and acceptor, e.g. –OH. For each grid point, the hydrogen-bonding character of the nearest atom to that point is found and two lists are maintained of points adjacent to either donors or acceptors. Thus, the donor list will have a TRUE at a point next to a donor, FALSE for an acceptor, or neither. Points next to donors/acceptors (e.g. OH) have a TRUE. Currently, grid points outside the van der Waals radii having hydrogen-bonding atoms within 3.2 Å are set to TRUE, the rest are set to FALSE. A similar approach is taken for the acceptor list.

Hydrophobic effects are important in many drug–receptor interactions. These are a consequence of the exclusion of water from a hydrophobic surface by a hydrophobic part of a molecule, similar to the agglomeration of oil particles in water. To partially describe this phenomenon (when looking at protein/ligand interactions), atom types have been designated hydrophobic or hydrophilic (Table 2). For each grid point, the hydrophobic/hydrophilic character of the nearest atom to that point is determined from a lookup table and a list maintained of points adjacent to either hydrophobic or hydrophilic atoms. Thus, the hydrophilic atoms have a TRUE at adjacent points. Currently, grid points outside the van der Waals radii having hydrophilic atoms within 3.2 Å are set to TRUE, the rest are set to FALSE.

In order to describe charge distribution, two logical arrays are maintained, a positive charge array and a negative charge array. Each location in the array is associated with a grid point. For those atoms with a partial charge of > 0.15 e, the adjacent point to that atom has its associated positive array location set to TRUE. The procedure is similar for a negative charge array (< -0.15 e). Currently, grid points outside the van der Waals radii having suitable atoms within 3.0 Å are set to TRUE; the rest are set to FALSE.

In the case of an active site, the constraints are modified as complementarity is required, not similarity. For the inside/outside grid, the TRUE and FALSE values are inverted. The negative and positive charge arrays are swapped, as are the hydrogen-bond donor and acceptor arrays. This means, for example, that a donor is now preferred in a position near an acceptor (the score is calculated based on similarity to the inverted grid). The hydrophobic/hydrophilic grid remains unchanged. The same approach to ionisation as described above for surfaces is used for grid properties.

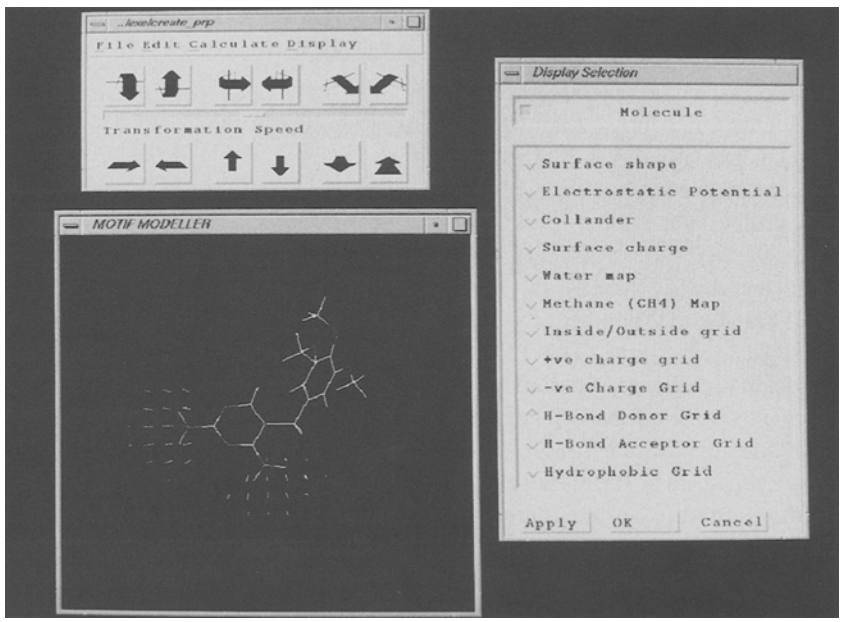

Fig. 1. Property calculation. Trimethoprim (neutral state) is shown with hydrogen-bond donors identified by their associated TRUE grid points, indicated as white dots.

TABLE 4
THE CONSTRAINTS DEFINITION FILE FOR OPTIMISATION OF SIZE AND MOLECULAR WEIGHT

#HEADER				
TITLE	molecular weight (300–350) shape (15 × 9 × 9 Å)			
VERSION	1.0			
#SCALARS				
MOL WEIGHT	330	>	350	; 1.0
STRAIN/ATOM	−5.0	>	5.0	; 2.0
P_AXIS X	14.0	>	16.0	; 1.0
P_AXIS Y	8.0	>	10.0	; 1.0
P_AXIS Z	8.0	>	10.0	; 1.0
#MUTATIONS				
ADD FRAGMENT	; 0.0			
DELETE FRAGMENT	; 1.0			
CHANGE ATOM	; 1.0			
ADD RING	; 0.0			
ROTATE BOND	; 1.0			
BREAK RING	; 0.0			
ADD DOUBLE BOND	; 0.0			
REMOVE DOUBLE BOND	; 0.0			
TRANSLATE MOL	; 0.0			
ROTATE MOL	; 0.0			
INSERT METHYLENE	; 1.0			
CYCLISE	; 0.0			

Implementation of constraints and calculation of a score

A run is initialised by the precalculation of constraints to be used in the optimisation. The algorithms are encoded in C and FORTRAN and are implemented under UNIX with an X-windows interface. Typically, only a few minutes are required to set up a calculation. An example of a property calculation screen is shown in Fig. 1. After calculation, the molecules and associated properties may be listed and displayed graphically.

The constraints to be used in the optimisation (and their associated weighting factors) are defined in a 'constraint definition file', an ASCII file containing the names of the constraints to use, their values or ranges and associated weights. An example is given in Table 4, which is used later to optimise molecules within molecular weight and size constraints.

Also loaded, if required, are the surface and grid points with the properties associated with each point, previously calculated from the template molecule (or pharmacophore or protein active site).

After property calculation, a score is generated by comparing the new molecule to the predefined constraints. The better the fit to the constraints, the lower the score. A more negative score indicates a better fit between the novel molecule and the constraints.

Penalties are added if the new molecule has a high internal strain energy. These are calculated as the sum of the internal strain energies divided by the number of atoms (strain per atom). The score generated for a new molecule in a particular position and conformation is calculated by summing the weighted error terms for each of the constraints (some or all of these constraints may be used in a particular problem).

$$\text{Score} = \sum_{i=1}^{n} W_1 S_{sca} + W_2 S_s + W_3 S_g$$

where S_{sca} is the error in the scalar constraints, S_s is the error in the surface constraints, S_g is the error in the grid constraints, W_x are the weights applied to each property and n is the number of constraints.

The error terms are the root-mean-square differences between the calculated and expected values for each constraint. For the scalar constraints:

$$S_{sca} = \sum_{i-1,n} \sqrt{(S_t - S_e)^2}$$

for n scalar properties, where t is the target and e is the test molecule.

The surface and grid terms are calculated by embedding the new molecule (in its present conformation and position) in the set of points originally associated with the template, for example molecule, pharmacophore or active site (the surface and grid points are always in fixed positions; the molecules move and change). At these points, the required properties are calculated. The scores are the differences between each of the values for the target molecule and the new molecule. An rms difference is not used here, as some of the beneficial changes result in negative values (e.g. hydrogen bonding).

$$S_s = \sum_{i-1,n} \sum_{j=1,p} S_t - S_e$$

for n surface points and p properties. Again, t is the target and e is the test molecule.

$$S_g = \sum_{i-1,n} \sum_{j=1,p} S_t - S_e$$

for n grid points and p properties.

Initiation of the algorithm

The constraints definition file is read and the appropriate scalar, grid and surface constraints loaded. Initial inputs are the number of molecules per generation (1 to 500), the number of generations (1 to 500), the initial seed molecules and a file name for the 'fossil record'. At a number of decision points during the run, random numbers are generated to select courses of action. These random numbers are generated using the system clock time in seconds as a seed and the generator is reseeded every generation. The random numbers are saved and make up the 'fossil record' which can be used to replay the run at a later date, since each run is uniquely determined by the sequence of random numbers.

The algorithm is initiated either from ethane as the seed molecule, a series of random fragments (the database) or from a known starting point. If the fragment (or

ethane) seeds are used to initiate the algorithm, they may be randomly rotated and translated to allow a truly random start. If a known starting point is used (e.g. a known bound inhibitor in an enzyme) this can be randomised (rotation, translation and bond rotation) or left in its present position. It may also be partially 'frozen' by the use of an additional constraint which penalises the fragment if any of the atomic positions or atom types of the input molecule are altered. This is useful in constraining the algorithm to extrapolate about a known series of molecules.

These molecules make up the zeroth generation. Figure 2 is a flow chart showing the sequence of steps involved in a GA optimisation of a molecule within constraints.

The zeroth generation is now evaluated to determine which molecules are best suited to breed the first generation (*selection* operator). The power of the GA lies in the fact that those parents most suitable breed more offspring and hence future generations better fit the constraints. To perform the selection of parents, and the number of offspring they may have, roulette wheel selection [4] is used.

Selection

Molecules are selected from the mating pool of the old population using roulette wheel selection. This is a method for selecting the parents most fit to breed the next generation. Each member of the population is allocated a slot in the roulette wheel and the size of the slot expressed as the proportion of a full rotation of the wheel represents the fitness of the individual. When the wheel is spun (by selecting a random number between zero and one) the probability that any individual is selected as a parent will be dependant on the slot size and therefore the fitness of the individual relative to the rest of the population. The fitness of each individual in the population is modified during the course of the run by *fitness scaling* [4]. Thus, a constant is added to the score of each individual which is dependant on the generation number, the population size and the maximum number of generations in the run. This allows almost random generation of individuals near the start of a run to optimise diversity. As the run progresses, the fitness constant decreases and selection pressure increases until, at the end of the run, the constant is negligible and selection pressure is at a maximum. This forces convergence near the end of the run. The application of fitness scaling gives some improvement in the quality of solutions, preventing far too early convergence based on one very fit individual at the start of the run. The fitness (score) is modified by the addition of the fitness constant, added to the *normalised* scores, calculated from the following equations:

$$Sc = \frac{Press}{pop} \times \frac{maxgen - n}{maxgen}$$

$$Ss = (Sc \times pop) + 1$$

$$Score = \frac{score + Sc}{Ss}$$

where Sc is a selection constant, Press is the selection pressure, set here from 0.1 to 10.0, where lower values increase the selection pressure, Pop is the population size

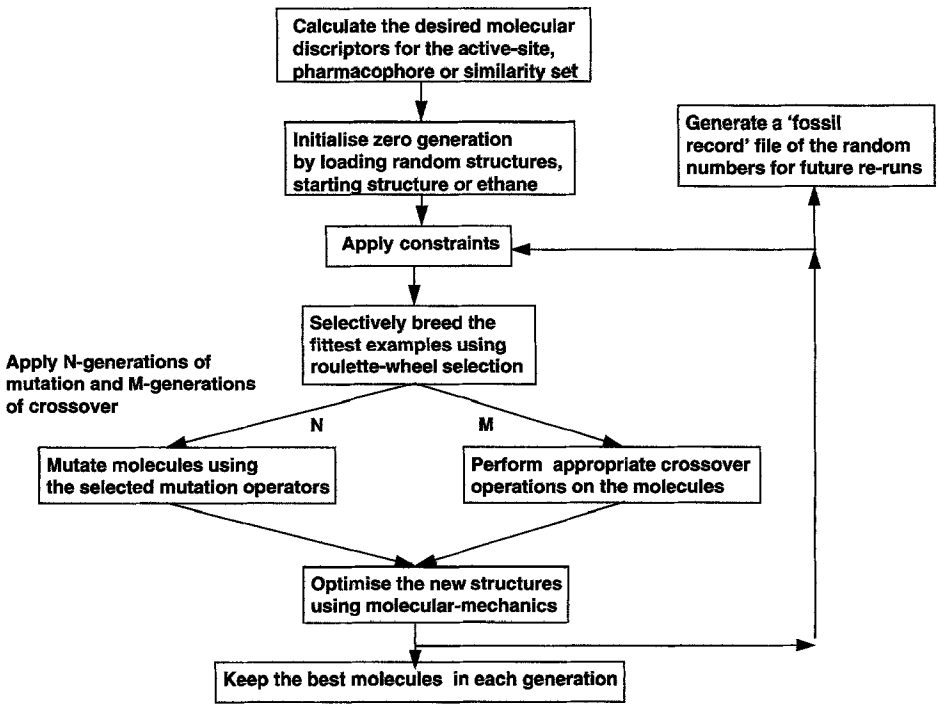

Fig. 2. Flow chart of the steps involved in the optimisation of molecules via the Genetic Algorithm.

for each generation, maxgen is the maximum number of generations in the run, n is the number of the present generation, score is the calculated fitness, and Score is the modified, renormalised score including fitness scaling. The fitness pressure was varied between 10.0 and 0.1. These results are discussed below in the example of the ribose problem.

The algorithm now proceeds to loop over the number of generations selected, modifying molecules in alternate generations via two methods. The first genetic operator is crossover [4], a mechanism by which molecules can perform 'sexual reproduction' and create offspring which are combinations of the parents. This operator is particularly useful in maintaining useful alleles within the population.

Crossover

Crossover is a mechanism for 'inter-breeding' molecules such that their 'genetic material' (substructures) is combined to form new molecules that have some of the characteristics of their parents. This genetic operator is applied every even generation. In order to perform this operation, it is necessary to identify parts of molecules suitable for crossover. Two types of crossover are performed, terminal crossover in which a terminal portion (single bond connection) of a molecule is excised and connected to a similar terminal portion from another molecule. Region crossover involves the excision of an internal portion of a molecule (having two single rotatable bond connections) which is inserted into a molecule which has had a similar region removed.

Both methods start by randomly selecting a crossover type, followed by a random pair of parents from the previous generation. The parents chosen are always different, as during the selection of the second parent, the first chosen is eliminated. Both methods attempt to crossover portions of molecules which are in the same volume of space, thus, the selection of the second fragment is biased towards fragments close in space to the original fragment. This avoids gross changes in the new molecule and prevents the algorithm 'thrashing' around (having difficulty optimising), as the accumulation of small change is usually the best path for optimisation in a GA.

In terminal crossover, all the rotatable bonds (single, acyclic) in the first molecule are identified and one is selected at random. The structure is then split in two and one fragment is selected at random. Its position is defined by a position vector. For the second randomly selected fragment, all the possible sites for bond breaking (of a single rotatable bond) are examined to determine which lies closest to the bond selected in the first fragment (defined by the previous position vector). The two spatially complementary fragments are then rejoined to give two new offspring. The new bonds and angles are adjusted to natural values (from the molecular mechanics files) and the structure regularised by molecular mechanics.

Region crossover involves selecting a fragment at random within the first molecule. This is a fragment as previously defined, i.e. connected by only two rotatable bonds. A position vector for this is calculated and then the fragment is excised. The second molecule for crossover is randomly selected and, as before, single rotatable bonds which correspond spatially (in terms of the position vector) as closely as possible to the first fragment are selected. The two fragments are then inserted where their counterparts were previously, their orientation closely maintained by virtue of their position vectors.

Mutation

The second genetic operator is mutation [4]; currently there are 12 variations of this operator. In each odd generation, the mutation operator is applied to each molecule in turn. The mutation selected is determined by a weighted random number selection between 1 and 12, in the same manner as roulette wheel selection. Mutations with greater weights are selected proportionately more often, e.g., it appears desirable to allow more *'alter atom type'* mutations to more easily find favourable hydrogen-bonding sites. The combination of mutations allows an extremely variable series of molecular structures to be evolved.

Translation mutation

The molecular structure is translated by a random amount along each of the three Cartesian axes. Random numbers are generated in the range −5.0 to 5.0. If n random numbers were used, then a normal distribution would be achieved by virtue of the central limit theorem. In this case, five random numbers are obtained using the computer system time as a seed. These are averaged to give R_{av}. In a similar fashion, a new random number is generated in the range $-R_{av}$ to R_{av}. This results in a skewed distribution biased towards zero, thus favouring small increments. This distribution was found to be more successful in trials optimising molecular position within constraints [3]. This is not surprising, as genetic algorithms optimise best by the accumulation of small changes. This operation is performed for the three Cartesian axes, resulting in a translation vector which may be applied to the molecule.

Rotation mutation

The molecular structure is centred at the origin and a random rotation is applied about each of the Cartesian axes. This is followed by translation of the molecule centre back to its original position.

The rotation increment is obtained by generating five random numbers in the range 0.0 to 2π. These are averaged to R_{av} and used to generate a new random number in the range 0.0 to R_{av}. This results in a distribution of numbers skewed towards zero. The resulting random

number is the rotation increment in radians. This operation is performed for each of the Cartesian axes and the combined rotation matrix is applied to the molecule.

Rotation about a bond

The rotatable bonds in a molecular structure are identified. These are bonds containing two independent sets of fragments. Multiple bonds, bonds in rings and bonds containing terminal atoms are excluded.

One of these rotatable bonds is selected at random using the random-number generator. A random rotation, moving the smaller molecular fragment, is performed about this bond. The rotation increment is obtained by generating five random numbers in the range 0.0 to 2π. These are averaged (R_{av}) and used to generate a new random number in the range 0.0 to R_{av}. This results in a distribution of numbers skewed towards zero. The resulting random number is the torsional rotation increment in radians. These small rotational changes are beneficial as they tend to avoid intramolecular atom clashes and interpenetration of fragments and, as previously stated, GAs usually optimise best via small changes. The structure is regularised using molecular mechanics.

Add a double bond

This can be summarised as follows: convert a single bond to a double bond, modify the atom types at either end of the bond and fill the valencies.

Hydrogen atoms are removed from the structure. Single bonds containing a pair of atoms with unfilled valencies are listed. One bond (which contains atom types from the allowed list) is selected at random and the atom types converted from single to the appropriate double bond atom type (Table 5).

The bond type is reset from one to two. Hydrogen atoms are re-added in the appropriate positions. The substructure assignment is adjusted to include both atoms of the double bond in a single fragment. Fragments having higher numbers are reduced by one to maintain the correct fragment count. If appropriate, nitrogen, sulfur and phosphorus atom types are checked to assign the appropriate type. The structure is regularised using molecular mechanics.

TABLE 5
ALLOWED CONVERSIONS OF SINGLE TO DOUBLE BONDS

Single bond atom type	Double bond atom type
CSP3	CSP2
OSP3	OSP2
NSP3	NSP2
SSP3	SSP2
NSP3	NSP2
NTRI	NSP2
NAM	NSP2

TABLE 6
ALLOWED CONVERSIONS OF DOUBLE TO SINGLE BONDS

Double bond atom type	Single bond atom type
CSP2	CSP3
OSP2	OSP3
NSP2	NSP3
SSP2	SSP3
CAR	CSP3

Remove a double bond

This works opposite to addition of a double bond, i.e., convert a double bond to a single bond, modify the atom types at either end of the bond and adjust the filled valencies.

A double bond is selected at random from the list of available double bonds in the current structure. The bond type is modified from double to single. The atoms at each end of the bond are converted to the equivalent single atom type (Table 6).

The fragment list is reassigned to include the additional fragment. Hydrogen atoms are removed, then added at the appropriate geometries. If appropriate, nitrogen, sulfur and phosphorus atom types are checked to assign the correct types. The substructure list is updated and the structure is regularised using molecular mechanics.

Add fragment

This mutation joins a molecule (read from an ASCII molecule file, currently in SYBYL [38] mol file format) onto the parent structure.

The library contains n molecules (these may be used to initialise the zeroth generation). The contents may vary and can contain any useful set of molecules. For example, amino acids, common acyclic functional groups (e.g. NO_2), polycyclic aromatics, etc. Molecular fragments not easily achievable from simpler structures via the current algorithms may also be included (e.g. H_3PO_4). One current library is shown in Fig. 3.

A molecule is selected at random from the available structures. Two hydrogen atoms, one from the parent and one from the fragment are selected at random. The fragment molecule is translated and rotated in space such that the fragment to bonded hydrogen and the molecule to bonded hydrogen are collinear. A single bond is made from the molecule to the fragment and the two identified hydrogens are eliminated. The appropriate bond length and bond angles about the new bond are set to values obtained from the force field bond and angle files. The fragment numbering is reassigned and hydrogens added in the geometrically correct positions to correct any valence errors. The fragment is then rotated about the new bond by 5° increments (from the starting angle to 360° plus the starting angle). At each point, nonbonded interactions are summed together with the grid and sur-

face shape constraints. The conformation best fitting the constraints with least intramolecular van der Waals strain is chosen. The substructure numbering within the new molecule is reassigned. The structure is regularised using molecular mechanics.

Change atom type

An atom type is modified whilst maintaining valency and geometry rules.

The atom to be altered is selected randomly from all the atoms, including hydrogen. The number of non-hydrogen connections is counted and a set of possible replacements deduced from a series of lists. Replacements must have the same or higher valency. Valency is satisfied by removal and addition of hydrogen (Table 7).

An example is to mutate CSP3 with two non-hydrogen connections to OSP3 (e.g., CSP3 with three non-hydrogen connections mutated to OSP3 would result in one connection left over and is therefore not allowed).

An atom is randomly selected for mutation from all the atoms available. The number of non-hydrogen attached atoms is counted. From the available list of atoms for this atom type with this number of connections, a new atom type is randomly selected. A preset number of attempts at this mutation (here it is set to 10) is attempted. If no suitable replacement is found, this mutation is abandoned for the time being. The attached atoms list (a list of the atoms connected to each atom) is updated and used to deduce nitrogen, sulfur and phosphorus atom types (amides, planar nitrogen types and aromatic oxygen and sulfur types, e.g. furan and thiophene). The structure is regularised using molecular mechanics.

TABLE 7
ALLOWED ATOM TYPE CONVERSIONS

Atom type	Number of connections	Possibilities
CSP3	1	OSP3, NSP3, SSP3, F, CL, BR, IODI
CSP3	2	NSP2, STRI
CSP3	3	NSP3
CSP	1	NSP
OSP3	1	CSP3, NSP3, SSP3, F, CL, BR, IODI
OSP3	2	CSP3, NSP3, SSP3
OSP2	1	CSP2, NSP2, SSP2
OAR	2	CSP3, OSP3, NSP3, SSP3, PTRI
NSP3	1	CSP3, OSP3, SSP3, F, CL, BR, IODI
NSP3	2	CSP3, OSP3, SSP3
NSP3	3	CSP3
NTRI	1	CSP3, OSP3, SSP3, F, CL, BR, IODI
NTRI	2	CSP3, OSP3, SSP3
NTRI	3	CSP3
NAM	2	CSP3, OSP3, SSP3
NSP2	1	CSP2, OSP2, SSP2
NSP	1	CSP
SSP3	1	CSP3, OSP3, NSP3, F, CL, BR, IODI
SSP3	2	CSP3, OSP3, NSP3
STRI	3	CSP2
SSP2	1	CSP2, OSP2, NSP2
SAR	2	CSP3, OSP3, NSP3, SSP3, PTRI
H	1	CSP3, OSP3, NSP3, SSP3, F, CL, BR, IODI
F	1	CSP3, OSP3, NSP3, SSP3, CL, BR, IODI
CL	1	CSP3, OSP3, NSP3, SSP3, F, BR, IODI
BR	1	CSP3, OSP3, NSP3, SSP3, F, CL, IODI
IODI	1	CSP3, OSP3, NSP3, SSP3, F, CL, BR

Remove fragment

This is done as follows: a fragment with only one or two connections is identified and deleted, then the remaining fragments (for a fragment with two connections) are rejoined with the correct geometry.

The fragment to be deleted is randomly selected from the list of contiguous substructures in the molecule having only one or two connections. If this is the only fragment, the mutation is abandoned (otherwise this parent molecule will disappear!). The atoms belonging to the fragment are identified and marked for deletion. Checks are made to detect membership of a ring, or whether this is a terminal fragment having only one non-hydrogen connection. If there are more than two connections to the fragment, the mutation is abandoned. Hydrogen atoms are deleted, followed by deletion of the fragment. If this was a terminal atom with only one connection, then valency is satisfied by the addition of hydrogen at the correct geometry. If the deleted fragment was in a ring, then a new bond is made between the two fragments originally connected to the deleted fragment. Hydrogens are added to the structure, the substructure list is updated and the geometry optimised using molecular mechanics. If there were two connections and the deleted fragment did not belong to a ring, then the position of the second fragment is adjusted to satisfy natural bond length and bond angle values from the appropriate molecular mechanics tables. Hydrogen is added to satisfy the valency. The substructure list is updated and the geometry is further optimised by molecular mechanics.

The fragment list is regenerated, taking into account the missing fragment. If after 10 attempts this mutation fails (only two atoms in the structure or a polycyclic, for example) then it is abandoned.

Add ring

Ring structures are generated in the molecule. Two methods are used, a two-carbon bridge method and a three-carbon bridge method.

In the two-carbon bridge method, all the X-H bond lengths are adjusted to X-CSP3 bond lengths. A list is then created which contains all the pairs of hydrogen atoms which are near the CSP3-CSP3 bond length (1.5 ± 0.15 Å), for which the X-H-H bond angles are reasonable (109.5° ± 10°) and which are on different atoms. A pair is chosen from the list at random and the two hydrogen atoms are converted to CSP3 atom types. A bond is made between them and all hydrogen atoms removed and re-added at the correct geometry. The fragment list is up-

TABLE 8
PARAMETERS USED FOR BOND LENGTH AND FORCE CONSTANT ESTIMATION

Atom type	Radius (Å)	Force constant 1/2K (kcal mol^{-1} Å^{-2})
CSP3	0.771	290.0
CSP2	0.668	580.0
CSP	0.602	870.0
CAR	0.697	580.0
OSP3	0.715	514.6
OSP2	0.615	1029.2
OAR	0.604	1029.2
O2_N	0.630	514.6
NSP3	0.740	368.0
NTRI	0.736	368.0
NAM	0.661	368.0
NSP2	0.676	736.0
NSP	0.679	1104.0
N3_P	0.739	368.0
PTRI	0.935	245.0
STET	0.935	245.0
SSP3	0.905	334.0
STRI	0.865	334.0
STET	0.865	334.0
SAR	0.865	668.0
SSP2	0.855	668.0
H	0.329[a]	161.2
F	0.691	256.0
CL	0.884	190.0
BR	0.969	134.0
IODI	1.065	100.0

[a] Hydrogen has a radius of 1.1 − 0.771 = 0.329 Å to give a softer potential which results in better bond length estimation.

dated and the structure is regularised using molecular mechanics.

In the three-carbon bridge method, all the X-H bond lengths are adjusted to X-CSP3 bond lengths. A list is then created of all the X-H X-H bond vectors which are candidates for the addition of a further CSP3 carbon to complete a ring structure. This uses a three-point three-distance algorithm [39] to determine the best position to place the new atom.

The purpose of this algorithm is to find the position of the two points in space (like the atom positions of a ring-flipping cyclohexane carbon) as a linear combination of the vectors from one of the fixed points (Fig. 4). This may be described as finding the position of the knot in three pieces of string pulled taught above and below the plane.

A CSP3 carbon atom is added at the position of least steric strain, hydrogen atoms are added to satisfy valency rules and the structure is regularised using molecular mechanics.

Break ring

Ring bonds are assigned in the molecule using a spanning-tree algorithm to identify the smallest set of smallest rings (SSSR) [40]. A ring bond is selected (single-bond type) and the bond connecting this pair of atoms is removed from the bond list. One hydrogen is added to one atom of the pair and the structure regularised using molecular mechanics. The other atom of the pair then has a hydrogen added and the structure is again regularised using molecular mechanics. This step-wise addition of hydrogen allows slow relaxation of the structure and prevents hydrogen addition resulting in very high strain energy due to the proximity of other atoms in the structure.

Insert methylene

This mutation inserts a carbon CSP3 between two singly bonded atoms which have at least one hydrogen atom attached to each. This mutation is particularly useful for ring and chain expansion.

A list of single bonds composed of atom pairs, each having at least one attached hydrogen atom, is made up and one selected at random. The bond is removed and a CSP3 inserted between these atoms. New bonds are formed and the geometry is regularised using the natural bond lengths and angles (for a chain) or molecular mechanics (for a ring).

This completes the description of the allowed mutations. However, the molecular geometries must be optimised (in their present conformations), so a molecular-mechanics force field was developed to cope with these

TABLE 9
ATOM TYPES AND ASSOCIATED NATURAL BOND ANGLES AND FORCE CONSTANTS

Atom type	Bond angle (°)	Force constant 1/2K (kcal mol^{-1} deg^{-2})
CSP3	109.5	0.01
CSP2	120.0	0.012
CSP	180.0	0.009
CAR	120.0	0.012
OSP3	110.0	0.014
OSP2	120.0	0.014
OAR	110.0	0.024
O2_N	120.0	0.014
NSP3	108.8	0.012
NTRI	120.0	0.01
NAM	120.0	0.012
NSP2	120.0	0.012
NSP	180.0	0.012
N3_P	109.5	0.012
PTRI	120.0	0.012
STET	109.5	0.012
SSP3	105.0	0.014
STRI	120.0	0.014
STET	109.5	0.01
SAR	92.0	0.01
SSP2	120.0	0.01
H	120.0	0.01
F	120.0	0.01
CL	120.0	0.01
BR	120.0	0.01
IODI	120.0	0.01
LP	107.0	0.01

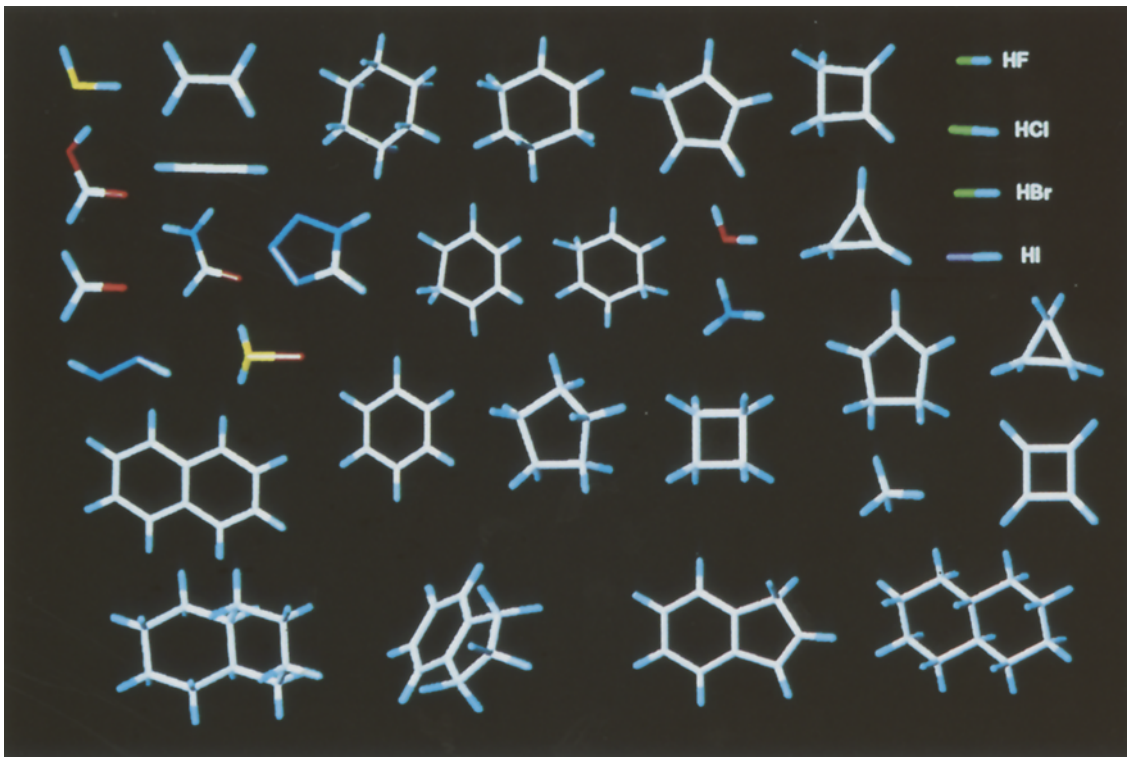

Fig. 3. An example of a useful fragment library of molecules containing common acyclic and cyclic building blocks.

novel structures. Some aspects of this procedure are described, particularly the estimation of unlisted force-field constants.

Molecular mechanics

In order to optimise the geometry (with respect to internal strain energy) of the structurally diverse novel molecules generated, molecular mechanics is used to regularise structures which result from crossover and mutation. This is particularly important in, for example, ring formation. The algorithm used is a derivative of Allinger's MM2 [41–44] with modifications and additions, particularly for heteroatomic molecules. These new features are necessary to enable the optimisation of unusual structures for which accurate force-field parameters are not available. (It is recognised that in such cases, the quality of those parts of the structure will not be as good as in the original MM2 implementation.)

If the natural bond length l_0 is not in the parameter set, then it is estimated by summing the two appropriate values for the atoms involved in the bond from Table 8. These are derived from an averaged set of available carbon-X (X = the other atom type) bond lengths, divided by two, from Ref. 45, e.g. the CSP3-CSP3 bond length would be $0.771 \times 2 = 1.542$ Å.

In the case of the unknown force constants, the values listed are the carbon-X values [46] times two for single bonds. For multiple bonds, these are multiplied by the

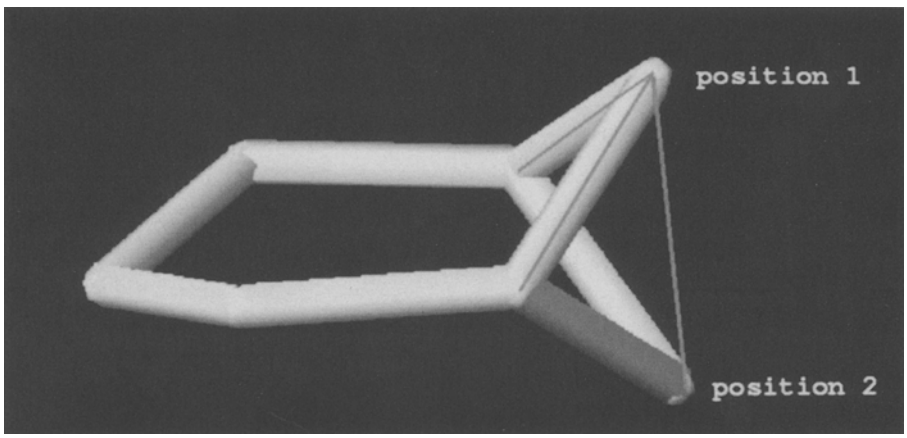

Fig. 4. The two putative positions for an atom addition in ring formation.

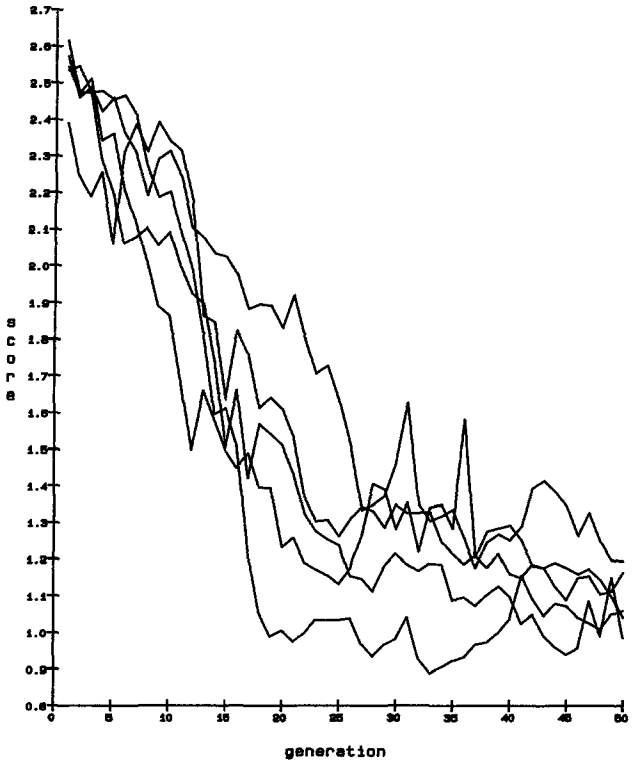

Fig. 5. Convergence using molecular weight and shape constraints with 10 molecules per generation. C1–C5 are indicated by identical lines.

multiplicity of the bond. This gives $1/2K_b$, e.g. the CSP3-CSP3 force constant is $145 \times 2 = 290$ kcal mol^{-1} Å^{-2}.

The bond force constants are similarly estimated by taking the appropriate sum for the bond pair and dividing by two.

For bond angle sets which do not appear in the parameter set, an estimated bond angle and constant for the central atom is selected (Table 9). The bond angles are an average of the values from Ref. 47. The force constants are estimated as averages from the MM2 force constants for similar atom types (Table 9).

If parameters are not available in the parameter list for torsional angles, the alternative energy term (White et al. [48,49]) is used:

$$E_t = \frac{1}{2} k_t (1 + s \cos n\Omega)$$

where s is the sign of the minimum of the energy for this torsion type, and n is the foldness (symmetry) of the rotational energy.

Parameters are required only for the atom pair of the central bond of the torsion angle. These are available in the parameter set for all the possible atom pairs. The mixing of potential forms is undesirable, however, in this case parameters must be available (even if only approximate ones) for the continuation of the algorithm in molecules where parameters are unavailable for the MM2 potential.

The anomeric effect is included by the addition of lone pairs on the hydroxy oxygen (or nitrogen) and the beta-substituted oxygen (or nitrogen). For van der Waals interactions, only pairs within 6.0 Å of each other are included in the nonbonded list, to speed computation. Identified hydrogen bonds use the same equations, but with special parameters which allow interpenetration of atoms [44].

Coulombic interaction energies are obtained using the excess atomic charges calculated by the partial equalisation of orbital electronegativity (PEOE) [26] employing a modified parameter set appropriate to the atom types here (Table 2).

In addition, the improper torsion angle (Θ) is calculated for the planar bond types (e.g. C=C, C=O) and multiplied by a constant to obtain an estimate of the out-of-plane bending energy:

$$E_{oop} = 0.8 \times \Theta$$

Geometry optimisation uses the simplex method of Nelder and Mead [50]. This is a robust optimisation method capable of coping with very distorted geometries. Ten cycles of simplex optimisation after each structure manipulation were found to be adequate for the production of reasonable geometries. Since most of the structural changes are minor, there is a cumulative effect in minimisation as the structures are re-optimised in each subsequent generation.

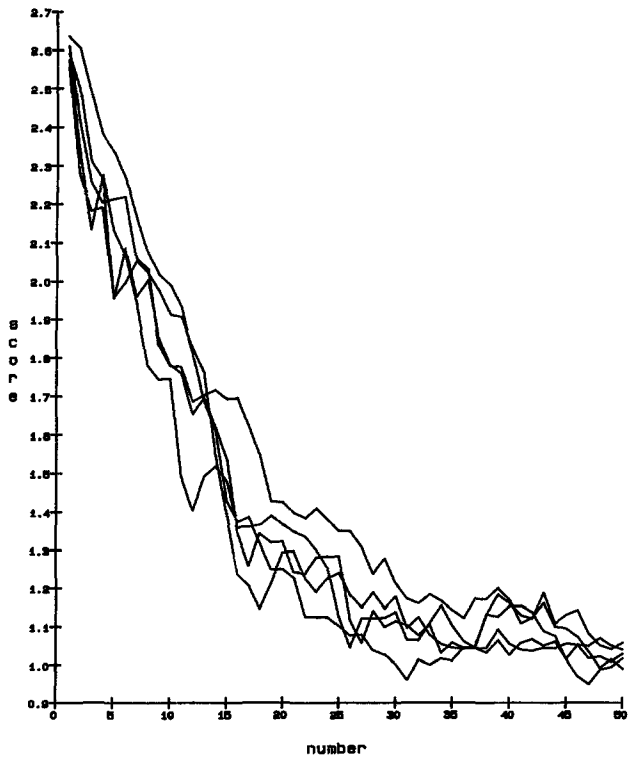

Fig. 6. Convergence using molecular weight and shape constraints with 50 molecules per generation. C1–C5 are indicated by identical lines.

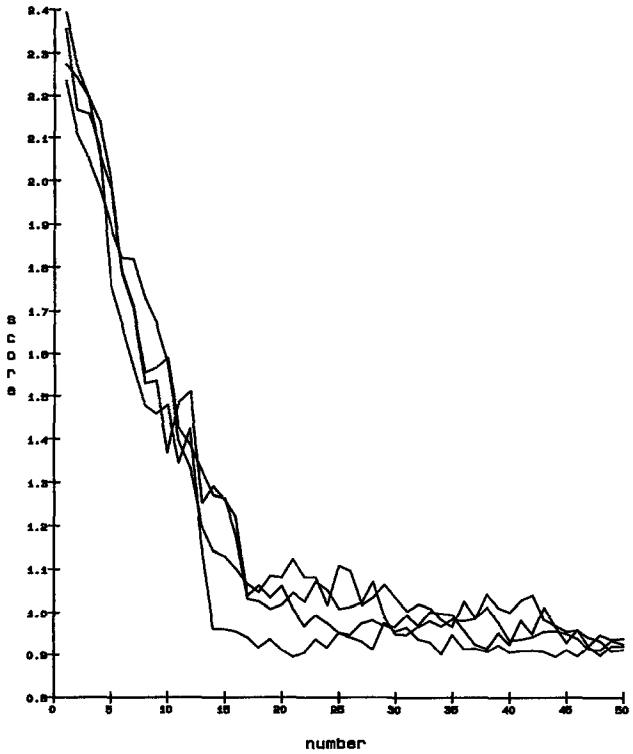

Fig. 7. Convergence using molecular weight and shape constraints with 100 molecules per generation. C1–C5 are indicated by identical lines.

Results

A number of test runs were designed to evaluate the behaviour of the GA in finding optimum structures within constraints. The convergence of the algorithm is tested in the first example, in which three distances and a molecular weight provide simple constraints. This also offers the opportunity to test the effects of population size on the convergence.

Example 1. Molecular weight and shape

The constraints definition file is shown in Table 4. Only some mutations are turned on (a weight of zero means this mutation is never called). The runs were initiated from ethane and since only methylenes may be added and atom types changed, the results will be acyclic structures with interspersed hetero-atoms. The runs were set for 50 generations and the population size was varied between 10 and 100 molecules per generation (selection pressure = 1.0). For each population size, five runs were performed. The results are shown in Figs. 5–7.

With only 10 molecules per generation, the rate of convergence is erratic and unpredictable. Increasing the population size (and so decreasing the selection pressure and increasing the diversity) results in progressively more reproducible optimisation. With 50 or 100 molecules per generation, the algorithm converges at about the 20th generation with a slow improvement in score thereafter. Table 10 lists the dimensions and molecular weight of the molecules from the runs with 100 molecules per generation. Figure 8 shows some of the molecular structures obtained. They fit the predefined constraints and are chemically reasonable (although not very exciting!).

A more difficult optimisation problem is to find molecules resembling ribose, a small compact sugar group forming part of the biologically important nucleosides. This group is frequently mimicked in antiviral compounds

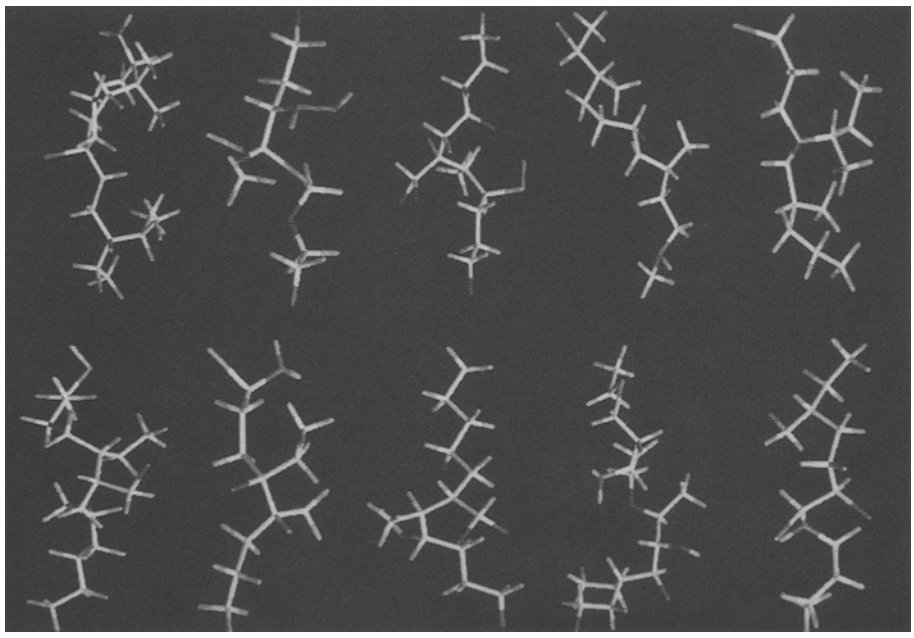

Fig. 8. Some examples of molecules using molecular weight and shape constraints.

TABLE 10
TEN RUNS USING THE GA TO OPTIMISE MOLECULES WITHIN THE MOLECULAR WEIGHT AND SHAPE CONSTRAINTS

Molecule	Molecular weight	X-dimension	Y-dimension	Z-dimension
Target	330–350	14–16	8–10	8–10
1	349	14.06	8.68	8.25
2	350	13.56	7.15	8.02
3	341	15.10	8.26	8.34
4	346	14.09	8.69	8.18
5	348	14.18	7.66	8.55
6	376	12.36	8.79	7.60
7	337	14.23	9.67	8.98
8	348	13.78	8.98	9.91
9	348	14.66	8.32	8.43
10	345	14.22	8.63	8.06

to form, for example, chain terminators of RNA or DNA transcriptase. The algorithm must produce small compact analogues which have electrostatic, steric and hydrophilic similarities to ribose.

Example 2. Replacements for ribose

Ribose (in the furanose form) was placed in a regular grid of dimensions 10×10 Å with 15 points per axis. The molecular properties described above, as well as a number of scalar properties, were calculated on the grid points.

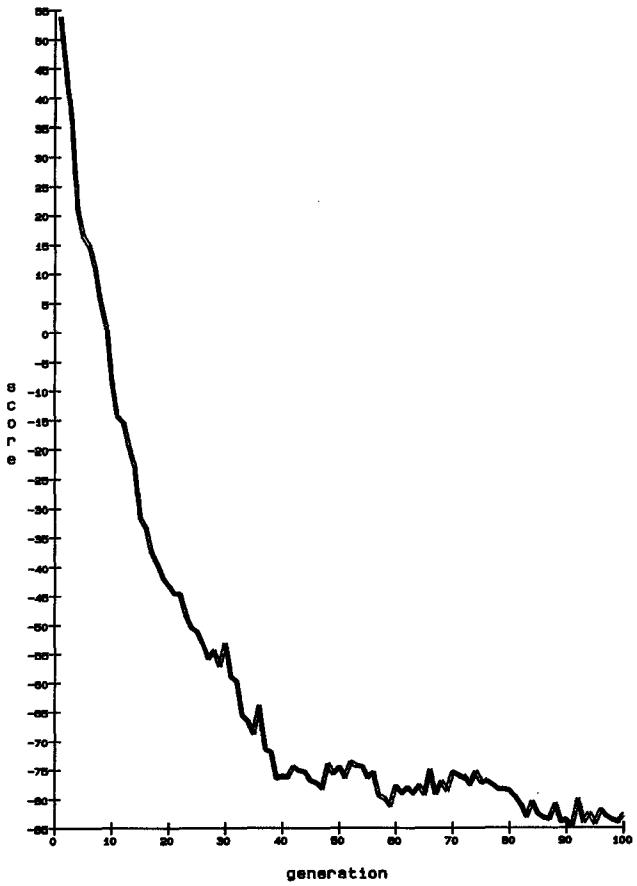

Fig. 9. The evolution of a ribose analogue starting from ethane. The convergence of a run over 100 generations is plotted.

Ranges for these were selected and used as constraints for molecule generation. The constraints are listed in the constraints definition file in Table 11. In this case, a far more complex set of constraints are applied and, in addition, all the mutations are allowed. *Change atom type* and *add fragment* are enhanced to allow for easier selection of ring fragments and hetero-atoms at appropriate positions.

TABLE 11
THE CONSTRAINTS DEFINITION FILE FOR THE GENERATION OF MOLECULES SIMILAR TO RIBOSE

#HEADER				
TITLE	ribose			
VERSION	1.0			
#SCALARS				
STRAIN/ATOM	–5	>	5	; 5
ATOM_COUNT	16	>	22	; 50
BOND_COUNT	16	>	22	; 50
X_BOUNDS	–5	>	5	; 200
Y_BOUNDS	–5	>	5	; 200
Z_BOUNDS	–5	>	5	; 200
LOGP	–2	>	–3	; 2
FEASIBILITY	0	>	0.5	; 200
CHIRAL	0	>	3.0	; 5
#GRID				
FILENAME	ribose			
INSIDE/OUTSIDE	; 0.7			
POS CHARGE	; 0.3			
NEG CHARGE	; 0.3			
H-BOND DONOR	; 0.3			
H-BOND ACCEPTOR	; 0.3			
HYDROPHOBIC	; 0.02			
#MUTATIONS				
ADD FRAGMENT	; 1			
DELETE FRAGMENT	; 1			
CHANGE ATOM	; 2			
ADD RING	; 1			
ROTATE BOND	; 1			
BREAK RING	; 1			
ADD DOUBLE BOND	; 1			
REMOVE DOUBLE BOND	; 1			
TRANSLATE MOL	; 1			
ROTATE MOL	; 1			
INSERT METHYLENE	; 1			
CYCLISE	; 1			

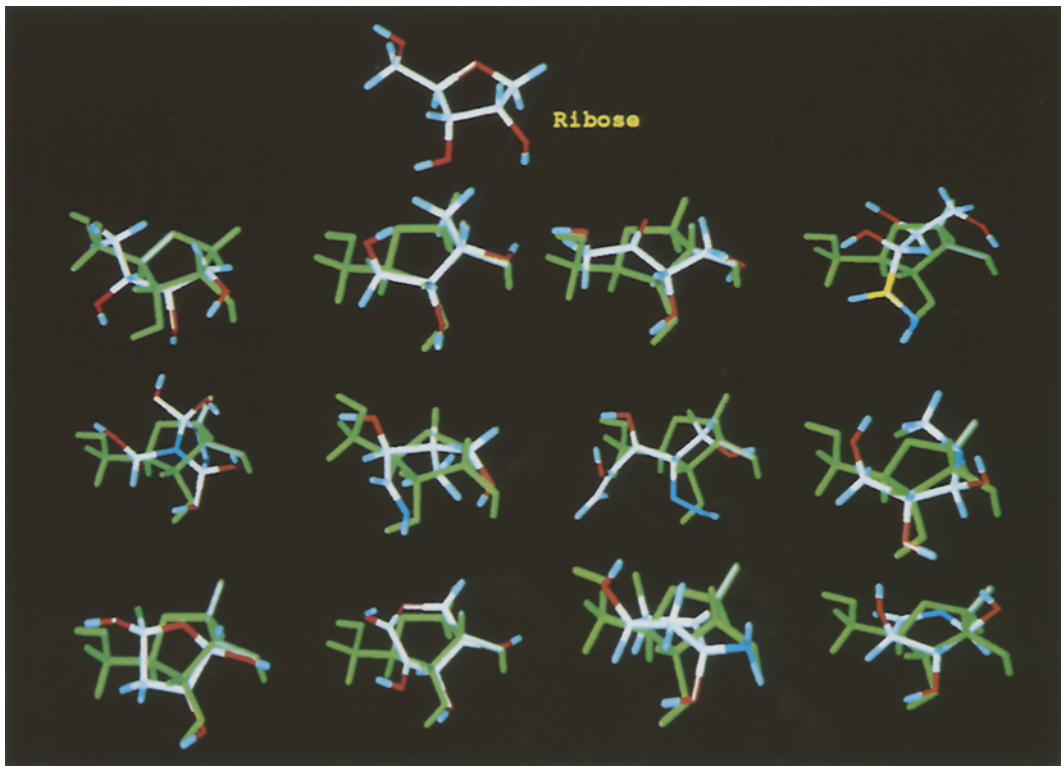

Fig. 10. Ribose (green) is overlayed by the resulting molecules from 12 GA runs. In order to force cyclic structures, in runs 10–12 the atom and bond constraints were adjusted to 19 atoms and 19 bonds with zero range.

The weights for the constraints were adjusted in a number of test runs such that a 'reasonable' series of compounds were produced. Since the constraints are in general logical values, some weighting differences must be present between them to reflect energetically different contributions. This is at present subjective. A more rigorous weighting scheme needs to be developed and is an area of current investigation.

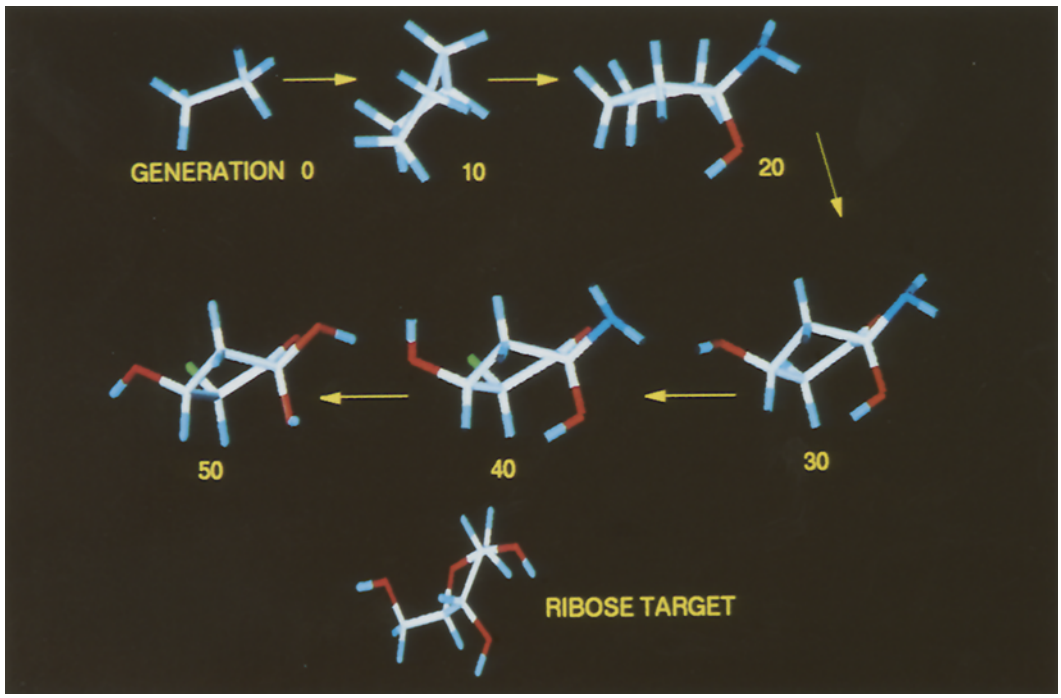

Fig. 11. The evolution of a ribose analogue over 50 generations starting from ethane.

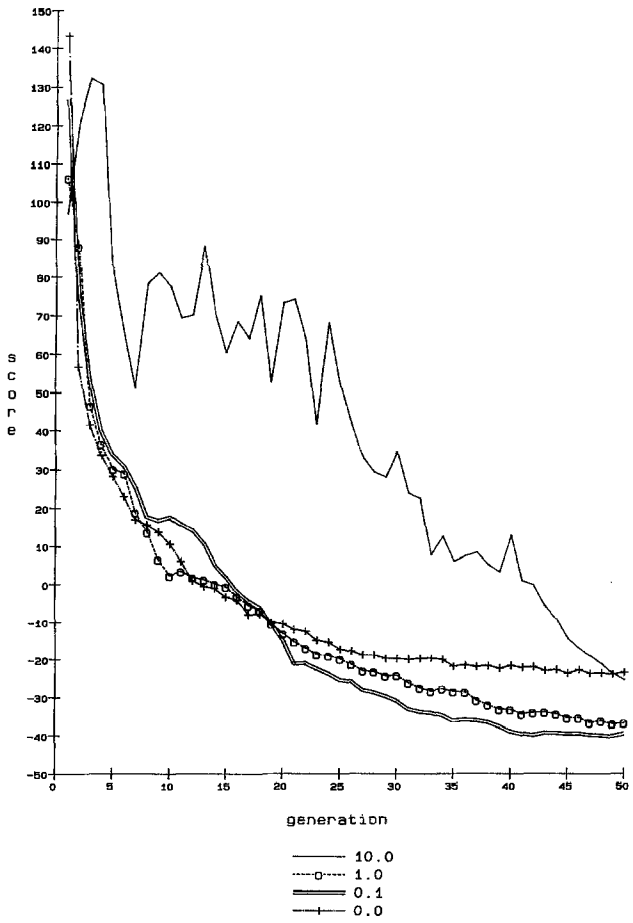

Fig. 12. The effects on convergence of selection pressure using the ribose data. Each line corresponds to the average of five runs. The selection pressure was varied between 10.0 and 0.1.

Twelve test runs were performed with 50 generations of 50 molecules per generation (selection pressure = 1.0) on a Silicon Graphics R4000 processor. The time for generation of each molecule is ca. 0.8 s. In Fig. 9 the convergence of a test run over 100 generations is plotted.

The convergence plot shows the ability of the GA to find increasingly better solutions. It is useful in checking the rate of convergence. Converging too quickly means that the selection pressure is too high (it may be lowered by increasing the population size), resulting in solutions of relatively high score. Slow convergence with too low selection pressure results in thrashing of the algorithm, with no clear solutions emerging. This may also occur if the constraints are contradictory or the search space is too large.

Other runs were very similar and suggested that about 50 generations gave solutions of low score. Overlays of the generated molecules on ribose are shown in Fig. 10.

Since each run generates a different family of solutions, other criteria may need to be applied in the selection of analogues, e.g. ease of synthesis, toxicity, etc. The scores for each GA run may be compared and used as a similarity index. The most dissimilar within a given range could, for example, be selected to produce a diverse series for testing. The list of analogues produced can also be input into a 3D database-searching program (e.g. Tripos Unity [38]), of which there are now a number available. Indeed, during a large run of 500 generations of 500 molecules per generation, 50 000 molecules are produced. If the selection pressure is low, molecules are produced almost randomly and therefore the production of very large databases of chemically sensible molecules is easily achieved.

One interesting aspect of the GA is the path followed during optimisation. The algorithm saves the best new molecules as they are produced and these may be replayed to show the evolutionary process at work. An evolutionary path is shown for an additional run of the ribose problem (Fig. 11).

This is a useful data set to look at the effects of selection pressure. Using the ribose data set, 20 runs were performed, five of each with selection pressures of 10.0, 1.0, 0.1 and 0.0. The results are plotted in Fig. 12. Each set of five has been averaged. A selection pressure of between 0.1 and 1.0 appears to give slightly better convergence (at least, over 50 generations).

Exactly the same approach may be used to generate analogues within a pharmacophore. The constraints may be calculated from a series of fragments positioned in space at appropriate positions to represent key properties or specific functional groups of the pharmacophore. These fragments do not have to be joined.

In more complex problems with a much greater search space (e.g., in the case of a highly constrained protein active-site cavity), where solutions may be difficult to initiate (i.e., obtain a good starting point) in the correct region, some dead-end solutions are optimised only to be discarded later as better families of structures take over. These problems are termed GA-hard [4]. In this case, the GA search space must be large (to 'cast a large net' over the search space) and convergence is required to be slow with a low selection pressure. The next example is of this type, an active site-directed generation of molecules. The enzyme dihydrofolate reductase (DHFR) [51] is a good candidate for testing as it has been exhaustively studied and many good solutions (e.g. the natural substrate folate, methotrexate, and diaminopyrimidine analogues like trimethoprim (TMP)) are known [52].

Generation of analogues in DHFR

DHFR catalyses the conversion of dihydrofolate to tetrahydrofolate in the presence of a cofactor, NADPH. This reaction is critical to the synthesis of precursors of DNA and has been used as a target for antibacterial compounds, principally due to the fortuitous differences between mammalian and bacterial DHFR which offer a window for the selective inhibition of the bacterial

TABLE 12
THE CONSTRAINTS DEFINITION FILE FOR LIGAND GENERATION IN THE ACTIVE SITE OF DHFR

#HEADER				
TITLE	DHFR active site ternary			
VERSION	1.0			
#SCALARS				
STRAIN/ATOM	−5	>	5	; 1
ATOM_COUNT	35	>	45	; 5
BOND_COUNT	40	>	60	; 5
X_BOUNDS	−10	>	10	; 100
Y_BOUNDS	−10	>	10	; 100
Z_BOUNDS	−10	>	10	; 100
LOGP	−2	>	2	; 5
BAD_GROUP	0.0	>	0.5	; 100
CHIRAL	0	>	3	; 5
AROMATIC	20	>	30	; 2
ROTBONDS	8	>	12	; 5
#GRID				
FILENAME	; dhfr20			
INSIDE/OUTSIDE	; 10.0			
POS CHARGE	; 0.5			
NEG CHARGE	; 0.5			
H-BOND DONOR	; 1.0			
H-BOND ACCEPTOR	; 1.0			
HYDROPHOBIC	; 0.1			
#MUTATIONS				
ADD FRAGMENT	;1			
DELETE FRAGMENT	;1			
CHANGE ATOM	;2			
ADD RING	;1			
ROTATE BOND	;1			
BREAK RING	;1			
ADD DOUBLE BOND	;1			
REMOVE DOUBLE BOND	;1			
TRANSLATE MOL	;1			
INSERT METHYLENE	;1			
CYCLISE	;1			

enzyme. Two runs were initiated to explore this site. In the first, using the inhibitor trimethoprim as a 'seed' molecule, the GA attempts to improve upon its affinity at the bacterial enzyme. In the second run, the fragment library is used to seed the active site and the GA builds molecules from a de novo starting point.

The target was an X-ray crystallographic structure of the ternary complex of DHFR [53]. This structure is currently at an early stage of refinement, so hydrogen atoms were added and atom-centred charges assigned using the PEOE method (taking care to properly assign ionised species). This was followed by 20 cycles of simplex geometry optimisation to alleviate strained portions of the molecule (mostly due to hydrogen addition). Geometry optimisation of this type results in only minor movements of the main-chain atoms, with slight movement of the side chains to alleviate strain. In later, highly refined structures, very little movement indeed is seen.

The protein is a large molecule with 1327 heavy atoms, not all of which are needed for the purposes of this algorithm. Atoms within 20 Å of the inhibitor TMP were extracted and used as a template for property calculation. TMP was extracted from this co-ordinate set and the remaining protein active site was used to generate the inverse molecular properties described above.

Example 3. Constrained ligand generation

TMP was used to seed the initial population, with the diaminopyrimidine moiety 'partially frozen' by having a score applied if any atom were changed (this, of course, may be compensated for by particularly good substitutions). This good starting point allows the algorithm to extrapolate about a known inhibitor. These runs comprised 50 generations with 100 molecules per generation. The constraints definition file is listed in Table 12.

The results of six runs are shown in Fig. 13. (These molecules are all shown in their neutral states; the algorithm, as previously described, may ionise appropriate groups.)

The constraints appear to have forced the production of polycyclic analogues by the inclusion of four- and five-membered rings to replace the methylene linker. New functionalities and substituents have been created to interact with additional hydrogen-bonding groups in the active site. Some of these would be difficult to synthesise. However, one really useful feature is the ability of the algorithm to generate novel substituents on the basic framework; this serves as an excellent idea generator for new analogues.

Using the same constraints definition file, the algorithm was initiated from random starting points using the fragment library as seed points. In this case each run generates completely novel structural types. The algorithm starts with a low-molecular-weight fragment (e.g. cyclopentane), randomises its position and orientation, then grows larger structures via crossover and mutation. To illustrate the ability of the algorithm to fill space in the active site, make hydrogen bonds and produce novel structures, a generated ligand is shown in the active site of the DHFR ternary structure. This was generated from ethane over 100 generations (100 molecules per generation). The protein is yellow (DHFR ternary complex), while trimethoprim is shown in purple for comparison (Fig. 14). Hydrogen bonds are drawn as dotted yellow lines. An example of an evolutionary pathway to one of these structures is shown in Fig. 15. The results of some runs are shown in Fig. 16.

Clearly, the algorithm generates many possibilities within the active-site constraints, some of which could serve as starting points for the synthesis of novel drugs.

Conclusions

A genetic algorithm utilising a 'knowledge base' for chemical-structure generation has been developed to evolve molecular structures within constraints. The mol-

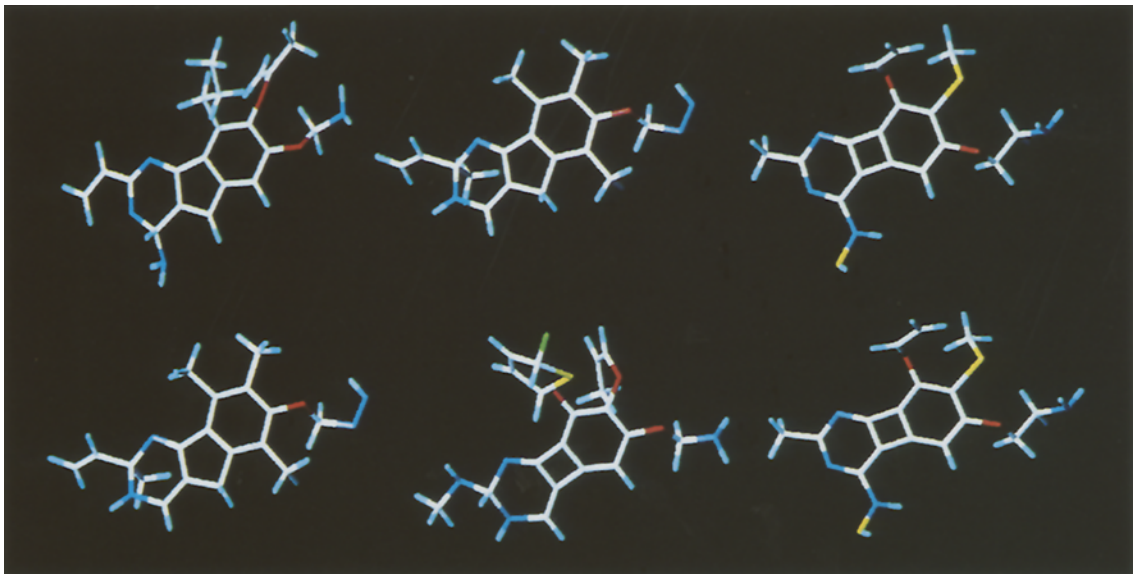

Fig. 13. Some examples of constrained generation of trimethoprim analogues, generated in the active site of DHFR from a TMP seed.

ecule-generation algorithm is fast, effective and stable, resulting in novel compounds that satisfy the rules of chemistry. Of particular note is the powerful optimising ability of the genetic algorithm, even with relatively small population sizes.

Current constraints are rather simple and many of the molecules generated require modification to produce synthesizable and sensible drug-like molecules. However, it is satisfying that molecules which are known inhibitors generate low scores, e.g. trimethoprim in the ternary complex of DHFR/NADPH has a score of −6.7 using the constraints of example 3, while the best four starting molecules had an average score of 26.2, converging to −32.1 after 32 generations.

Future development will concentrate on implementing constraints which better mimic nature (e.g., in the case of evolution within an active site, to generate scores which approximate binding energies). Also, metals and transition states may be useful additions. This work is in progress and will be reported elsewhere. Due to the complexity of the software and the enormous variation in the molecules produced, unexpected algorithmic bugs

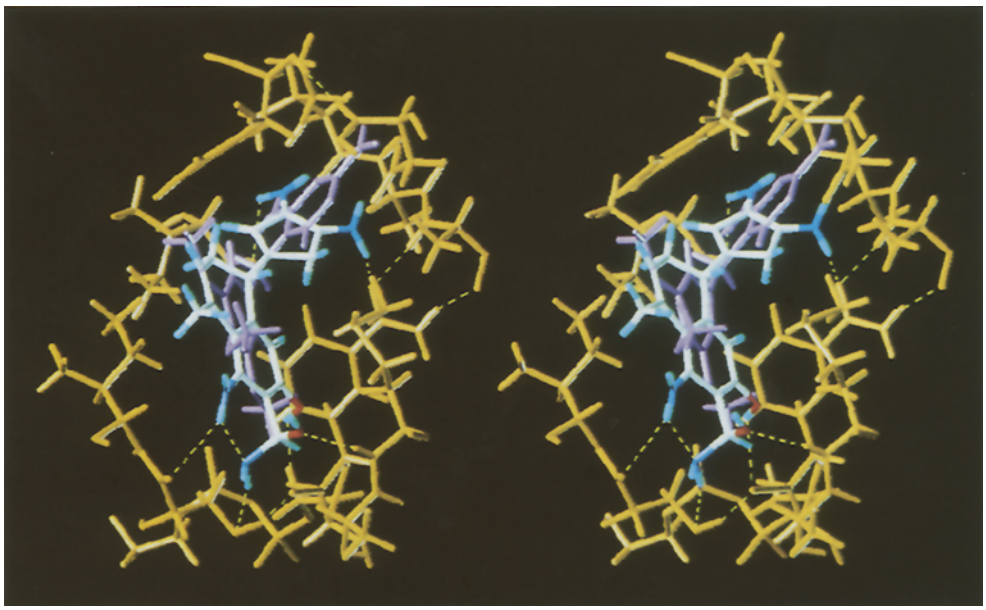

Fig. 14. An example of ligand generation in DHFR (relaxed stereo, coloured in yellow). TMP (from the X-ray structure) is shown in purple; the generated molecule is coloured by atom type. Hydrogen bonds are dotted yellow lines. The molecule evolved from ethane.

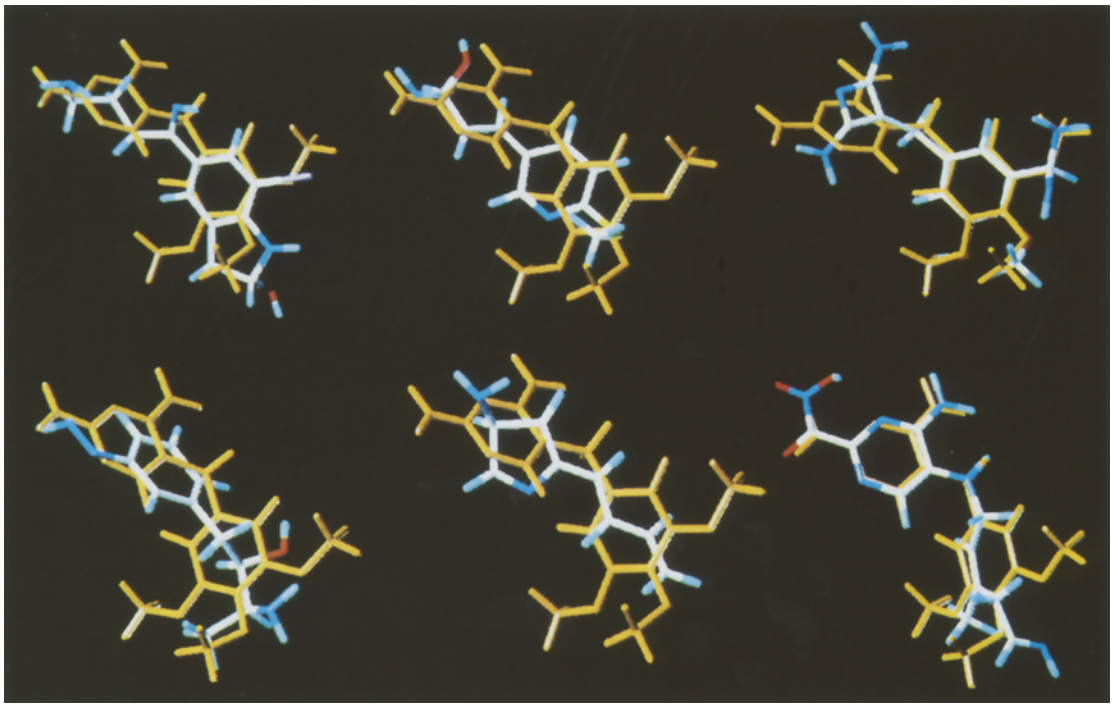

Fig. 15. Some structures generated in the active site of DHFR from random starting points, using the database structures.

do turn up, e.g. the azido (-N=N=N) group appeared and required the bond assignment algorithm to be modified.

Only a small number of applications are presented here. Molecules may also, for example, be generated from pharmacophore models or from overlays of active analogues. In addition, the algorithm may be restricted to only adding or removing fragments from a predefined library to result in molecules made up from easily available starting materials.

The algorithm has been applied to a diverse set of problems and results will be reported in due course.

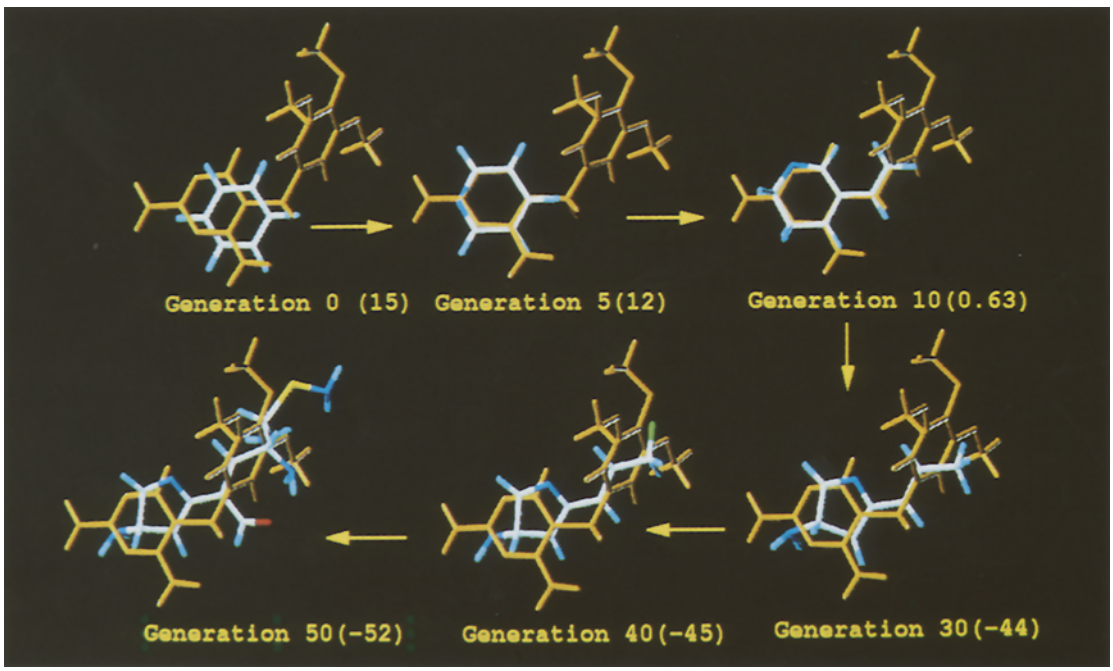

Fig. 16. Example of the evolution of an analogue generated in the active site of DHFR from random starting points, using the database structures. The scores are shown in brackets.

References

1 Glen, R.C., Drug News Perspect., 3 (1990) 332.
2 Lindsay, R.K., Buchanan, B.G., Feigenbaum, E.A. and Lederberg, J., Applications of Artificial Intelligence for Organic Chemistry – The DENDRAL Project, McGraw-Hill, New York, NY, 1980.
3 Payne, A.W.R. and Glen, R.C., J. Mol. Graph., 11 (1993) 74.
4 Goldberg, D.E., Genetic Algorithms in Search, Optimisation and Machine Learning, Addison-Wesley, Reading, MA, 1989.
5 Darwin, C., The Origin of Species, Dent Gordon, London, 1973.
6 Willett, P., Three-Dimensional Chemical Structure Handling, Research Studies Papers, Taunton, 1991.
7 Gillet, V.P., Johnson, A.P., Mata, P. and Sike, S., Tetrahedron Comput. Methodol., 3 (1990) 681.
8 Böhm, H.J., J. Comput.-Aided Mol. Design, 6 (1992) 61.
9 Lewis, R.A. and Leach, A.R., J. Comput.-Aided Mol. Design, 8 (1994) 467 and references cited therein.
10 Brown, R.D., Jones, G., Willett, P. and Glen, R.C., J. Chem. Inf. Comput. Sci., 34 (1994) 63.
11 Jones, G., Brown, R.D., Clark, D.E., Willett, P. and Glen, R.C., In Forest, S. (Ed.) Proceedings of the Fifth International Conference on Genetic Algorithms, Morgan-Kauffman, San Mateo, CA, 1993, pp. 597–602.
12 Weininger, D., J. Chem. Inf. Comput. Sci., 28 (1988) 31.
13 Weast, R.C. (Ed.) Handbook of Chemistry and Physics, 60th ed., CRC Press, Boca Raton, FL, 1980, pp. E70–E72.
14 Klopman, G., Namboordiri, K. and Schochet, M., J. Comput. Chem., 6 (1985) 28.
15 Bodor, N., Gabanyi, Z. and Wong, C., J. Am. Chem. Soc., 111 (1989) 3783.
16 Edward, J.T., J. Chem. Educ., 47 (1970) 261.
17 Pearlman, R.S., SAREA, QCPE Program No. 413, Quantum Chemistry Program Exchange, University of Indiana, Bloomington, IN, 1981.
18 Glen, R.C., J. Comput.-Aided Mol. Design, 8 (1994) 457.
19 Massey, H.S.W. and Kestelman, H., Ancillary Mathematics, Sir Isaac Pitman and Sons Ltd., London, 1964, pp. 849–852.
20 Connolly, M., Molecular Surface Program, QCPE Program No. 429, Quantum Chemistry Program Exchange, University of Indiana, Bloomington, IN, 1983.
21 Richards, F.M., Annu. Rev. Biophys. Bioeng., 6 (1977) 151.
22 Goodford, P.J., J. Med. Chem., 28 (1985) 849.
23 Singh, U.C. and Kollman, P., GAUSSIAN, 80-UCSF, QCPE Program No. 446, Quantum Chemistry Program Exchange, University of Indiana, Bloomington, IN, 1980.
24 Amos, R.D. and Rice, J.E., CADPAC: The Cambridge Analytic Derivatives Package, Issue 4.0, Cambridge, 1987.
25 Stewart, J.J.P., MOPAC, Version 6.0, QCPE Program No. 455, Quantum Chemistry Program Exchange, University of Indiana, Bloomington, IN, 1992.
26 Gasteiger, J. and Marsili, M., Tetrahedron, 36 (1980) 3219.
27 Hinze, J. and Jaffe, H.H., J. Am. Chem. Soc., 84 (1962) 540.
28 Hinze, J., Whitehead, M.A. and Jaffe, H.H., J. Am. Chem. Soc., 85 (1963) 148.
29 Hinze, J. and Jaffe, H.H., J. Am. Chem. Soc., 67 (1963) 1501.
30 Cieplak, P. and Kollman, P., J. Comput. Chem., 12 (1991) 1232.
31 Carbo, R., Leyda, L. and Arnau, M., Int. J. Quantum Chem., 17 (1980) 1185.
32 Weast, R.C. (Ed.) Handbook of Chemistry and Physics, 60th ed., CRC Press, Boca Raton, FL, 1980, p. D-194.
33 Weast, R.C. (Ed.) Handbook of Chemistry and Physics, 71st ed., CRC Press, Boca Raton, FL, 1991, pp. 10-210–10-211.
34 Weast, R.C. (Ed.) Handbook of Chemistry and Physics, 71st ed., CRC Press, Boca Raton, FL, 1991, p. 9-8.
35 Truhlar, D.G. and Politzer, P. (Eds.) Chemical Applications of Atomic and Molecular Electrostatic Potentials, Plenum Press, New York, NY, 1981, pp. 309–334.
36 Giessner-Prettre, C. and Pullman, A., Theor. Chim. Acta, 25 (1972) 83.
37 Giessner-Prettre, C., QCPE Program No. 11, Quantum Chemistry Program Exchange, University of Indiana, Bloomington, IN, 1972.
38 SYBYL v. 6.1 molecular modelling package, Tripos Associates, St. Louis, MO, 1992.
39 Senn, P., Comput. Chem., 15 (1991) 93.
40 Downs, G.M., Gillet, V.J., Holliday, J.D. and Lynch, M.F., J. Chem. Inf. Comput. Sci., 29 (1989) 172.
41 Allinger, N.L., J. Am. Chem. Soc., 99 (1977) 8127.
42 Burkert, U. and Allinger, N.L., Molecular Mechanics, American Chemical Society, Washington, DC, 1982.
43 Norskov-Lauritsen, L. and Allinger, N.L., J. Comput. Chem., 5 (1984) 326.
44 Allinger, N.L., Kok, R.A. and Imam, M.R., J. Comput. Chem., 9 (1988) 591.
45 Weast, R.C. (Ed.) Handbook of Chemistry and Physics, 71st ed., CRC Press, Boca Raton, FL, 1991, pp. 9-1–9-5.
46 Weast, R.C. (Ed.) Handbook of Chemistry and Physics, 71st ed., CRC Press, Boca Raton, FL, 1991, pp. 9-86–9-107.
47 White, D.N.J. and Bovill, M.J., J. Chem. Soc., Perkin Trans. II, (1977) 1610.
48 Bovill, M.J., Chadwick, D.J., Sutherland, I.O. and Watkin, D., J. Chem. Soc., Perkin Trans. II, (1980) 1529.
49 Nelder, J.A. and Mead, R., Comput. J., 8 (1965) 308.
50 Mathews, D.A., Bolin, J.T., Burridge, J.M., Filman, D.J., Volz, K.W., Kaufman, B.T., Beddell, C.R., Champness, J.N.C., Stammers, D.K. and Kraut, J., J. Biol. Chem., 260 (1985) 381.
51 Champness, J.N.C., Kuyper, L.F. and Beddell, C.R., In Burgen, A.S.V., Roberts, G.D.K. and Tute, M.S. (Eds.) Interaction Between Dihydrofolate Reductase and Certain Inhibitors. Molecular Graphics and Drug Design, Elsevier, Amsterdam, 1986, pp. 335–362.
52 Champness, J.N., Stammers, D.K. and Beddell, C.R., FEBS Lett., 199 (1986) 61.