

Molecular structure matching by simulated annealing. I. A comparison between different cooling schedules

M.T. Barakat and P.M. Dean

Department of Pharmacology, University of Cambridge, Tennis Court Road, Cambridge CB2 1QJ, U.K.

Received 13 September 1989

Accepted 30 November 1989

Key words: Molecular similarity; Molecular matching; Simulated annealing

SUMMARY

This paper outlines an application of the theory of simulated annealing to molecular matching problems. Three cooling schedules are examined: linear, exponential and dynamic cooling. The objective function is the sum of the elements of the difference distance matrix between the two molecules generated by continual re-ordering of one molecule. Extensive tests of the algorithms have been performed on random coordinate data together with two related protein structures. Combinatorial problems, inherent in the assignment of atom correspondences, are effectively overcome by simulated annealing. The algorithms outlined here can readily optimize molecular matching problems with 150 atoms.

INTRODUCTION

The identification of structurally matched regions between two molecules is a complex but important problem. A solution is necessary for any quantitative study of the relationship between structure and activity. Methods to examine structural matches fall into 3 types: (a) a comparison of bonding skeletons by molecular common subgraph procedures [1]; (b) a comparison of molecular surface similarities [2]; (c) a comparison of corresponding atom positions [3]. If these methods of searching for similarity are to be optimized, then consideration has to be given to the nature of the optimization problem; optimization in continuous space requires different approaches to its solution from optimization in discrete space. For example, if we were to compare the electrostatic potential, or electron densities, on a defined molecular surface, continuous optimization procedures would be required [4,5]. On the other hand, if unassigned atom coordinate positions are to be matched then optimization procedures for discrete space are needed. A large and detailed review on molecular similarity is provided elsewhere [6]. Each type of comparison is needed for different circumstances in any complete analysis of how molecular properties determine the course of mo-

lecular interactions; obviously these methods for comparison are of great importance for the development of effective procedures for drug design.

Structure matching of the atom coordinate positions between two molecules involves two conceptual problems. Firstly, if the atom correspondences between the two structures are known then all that needs to be done is to minimize the difference between the corresponding coordinates by a translation and rotation step of one molecule. This corresponds to overlaying the structures so that predetermined atoms in each molecule lie close to one another. Secondly, if there are no predetermined atom correspondences between dissimilar molecules they have to be found *before* the structures can be overlayed. This paper tackles the latter problem and outlines a method to determine the optimum positional correspondences in matching two molecular structures.

Consider two non-identical molecules A and B, containing N_A and N_B atoms, respectively; the atoms in each molecule are numbered separately in some arbitrary order. Then if the order of A is fixed, the order of B has to be permuted so that the difference in atom positions between A and re-ordered B is a minimum. The problem can be stated formally: let the configuration space \mathbb{Z} denote the set of all possible discrete configurations (orderings) π , and let an objective function be E such that at each configuration π_i , the function is assigned a value E_{π_i} . The aim is to find an optimal configuration, π_{opt} , for which

$$E_{\pi_{opt}} = \min \{E_{\pi_i} \mid \pi_i \in \mathbb{Z}\} \quad (1)$$

The objective function E is a measure of the dissimilarity between A and B. The size of the configuration space for two molecules, where $N_B \geq N_A$, is $N_B!/(N_B - N_A)!$ and can be estimated from Stirling's formula; for example $10! \approx 10^6$, $20! \approx 10^{18}$ and $30! \approx 10^{32}$. Therefore the configuration space exhibits a combinatorial explosion as the number of atoms increases; the problem of finding π_{opt} belongs to the large class of NP-complete problems. The phrase NP-complete describes a large class of non-deterministic polynomial time-complete problems; these are characterised by an exponential increase in computing time needed to solve them as the size of the problem increases. This class of problems is defined precisely by Papadimitriou and Steiglitz [7] and numerous cases of NP-complete computing problems are considered by Sedgewick [8]. Exhaustive tree searches, even with branch-and-bound pruning, are of little help for molecules with more than 15 atoms [3]. The term branch-and-bound describes an algorithm involving a tree search where branches of possibilities are explored. These branches are terminated when the value of the objective function has become worse than the best solution found so far; the branch is then 'bounded' and this rules out the need to explore large sections of the potential solution tree.

Simulated annealing

A possible general, but approximate, solution to NP-complete problems has been proposed by Kirkpatrick et al. [9]. This technique of simulated annealing replaces the exponential increase in computing time with N, required for an exhaustive search, to small powers of N. Their procedure has points in common with the physical problem of cooling an ensemble of particles to reach the frozen state. The objective function, E, to be minimized is handled as though it were a statistical mechanical quantity of energy whose behaviour can be followed by Monte Carlo methods employing the Metropolis algorithm.

Consider an ensemble at high temperature, with the objective function of the current state, s , calculated to be E_s . A random perturbation is applied to the system so that a new state, s' , with a value of $E_{s'}$ is produced. The difference, ΔE , between the two states s and s' is calculated. If ΔE is negative, the perturbation is accepted unconditionally and the new state $E_{s'}$ is maintained. However, if ΔE is positive, the acceptance of state s' depends on the Metropolis condition [10] related to the Boltzmann probability distribution

$$P(s'|s) = e^{-\Delta E/kT}; \quad kT = T \quad (2)$$

where $P(s'|s)$ is the probability of accepting state s' which has an energy greater than the previous state s by an amount ΔE ; k is the Boltzmann constant and t is a temperature in Kelvins; kt is replaced by an annealing ‘temperature’, T , having the same units as E . In the Metropolis condition, the state s' is accepted when ΔE is positive only if a random number generated from a uniform distribution between $[0,1]$ is less than $P(s'|s)$. Another random perturbation is performed and the process repeated at the same temperature until equilibrium is reached; this occurs when the probability distribution is acceptably close to the Boltzmann distribution. The temperature is reduced by a small amount and the cycle repeated. This progressively restricts the size of accepted uphill transitions between successive Markov chains. The cycles are continued until the change in E , from one cycle to another, is insignificant. The system is then said to be frozen and the desired configuration has been found. The optimization should be ergodic; the same end point can be reached from any initial state at high temperature.

Simulated annealing is a general technique that can be used to minimize any problem in a discrete space. It has been successfully applied to: the travelling salesman problem (TSP) [11,12], the placement of standard cells in circuit design [13], the minimization of the Hamiltonian for flips of spins in spin glasses [11,14], the quadratic assignment problem [15,16], the football pools [17] and crystallographic refinement of proteins [18,19]. A considerable number of algorithms exist for simulated annealing and they differ in important respects; the principal differences are how they handle the temperature decrement, the length of the Markov chains and the stop criterion. A ready-made algorithm such as that provided by Numerical Recipes [20], contains many pitfalls for the unwary since it is written only for TSP-like problems employing a similar construct for the objective function. Difficulties in an application of the ready-made algorithm arise from: (1), the need to choose an appropriate objective function; (2), the need to generate a suitable set of moves to modify the configuration; (3), the need to develop an appropriate cooling schedule for the particular problem. Bounds states that ‘the main difficulty is that choosing an annealing schedule for a particular purpose is still something of a black art’ [21].

Molecular matching problem

In this paper, a number of optimization algorithms for simulated annealing have been used to investigate molecular matching. The objective function is the degree of dissimilarity in corresponding atom positions between molecules A and B. Four situations arise that are of practical importance in drug design: (a) when $N_A = N_B$ and the whole of A is compared with the whole of B; (b) when $N_B \geq N_A$ and the structure of B is searched for a region that resembles only a defined region of A; (c) when $N_B > N_A$ and the structure of B is searched for a region that resembles the

whole of A; (d) when $N_B \geq N_A$ and the structure of B is searched for a region that resembles *any* unspecified region of A.

Here we address the cases a, b, and c; c is a special case of b. The situation for case d is formidable more complex since null correspondences have to be introduced and the size of the unspecified region is unknown. A blind-searching technique, employing continuous minimization combined with cluster analysis, for an analogous problem of searching for similarity between molecular surfaces, has been proposed for case d [4].

This paper investigates the performance of three algorithms for the molecular matching problem; they are tested extensively on data derived from a random uniform distribution of points in three dimensions and on the matching of C_α -atoms from a globular protein without taking into account any bonding information or any predetermined atom correspondences. The algorithms have a linear, exponential, or dynamic cooling schedule. The accompanying paper [22] expands these findings by testing the best algorithm found here against data designed to produce difficult configuration landscapes.

THEORY

The objective function for matching atom positional correspondences

The atoms in molecules A and B are in a rigid conformation and are numbered $i = 1 \dots N_A$ and $i = 1 \dots N_B$; each atom has Cartesian coordinates (x_i, y_i, z_i) . The distance matrix for A is computed with elements

$$d_{ij}^A = [(x_i^A - x_j^A)^2 + (y_i^A - y_j^A)^2 + (z_i^A - z_j^A)^2]^{1/2} \quad (3)$$

The matrix is symmetrical with diagonal elements of zero, so only the sub-diagonal matrix needs to be computed. The distance matrix for B, with elements d_{ij}^B can be similarly calculated.

For a particular ordering of A and B the difference distance matrix (DDM), when the number of atoms $N_A = N_B$, is composed of elements

$$\Delta d_{ij}^{AB} = |d_{ij}^A - d_{ij}^B| \quad (4)$$

The sum of these elements, E, for the sub-diagonal DDM is given by

$$E = \sum_{i=2}^{N_A} \sum_{j=1}^{i-1} \Delta d_{ij}^{AB} \quad (5)$$

At each step of the minimization procedure, the order of the atoms in B is rearranged from a state s to a state s' , changing the objective function by an amount

$$\Delta E = E_{s'} - E_s \quad (6)$$

If ΔE is negative the new state s' is a better match than the previous state s and is accepted directly. However, if ΔE is positive, the Metropolis test (Eq. 2) is used to decide whether s' can be accepted.

The new ordering, π_i , for the atoms in molecule B is generated by swapping a randomly selected pair of atom numbers from a uniform distribution [1, N_A]. A single pair swap creates s' from s . This procedure differs significantly from the strategy for changing a configuration in the TSP given in Numerical Recipes [20] and it is also computationally more efficient.

Substantial savings in computer time can be achieved by computing the whole of the DDM once at the start. Subsequently only the difference caused by swapping the elements of the corresponding two rows, or columns, of the matrix needs to be computed. If the proposed move is accepted, then the current value of E is incremented by ΔE and the new elements of the DDM are stored. This procedure circumvents the need to calculate the whole of the DDM each time.

A slightly modified procedure for calculating the objective function is introduced when $N_B > N_A$. In this case a randomly chosen subset of atoms in B, containing N_A members, is needed to calculate the DDM. A swap is only allowed between a pair of atoms if one or both are taken from the subset; a move which would select a pair of atoms outside the subset is prohibited.

In addition to the objective function, a further parameter is determined at the start and end of the annealing process, the DDM statistic, DDM_{stat} [23]

$$DDM_{stat} = \left[\sum_{i=2}^{N_A} \sum_{j=1}^{i-1} (\Delta d_{ij}^{AB})^2 \right]^{1/2} / N_A \quad (7)$$

This indicates the goodness of fit between the distance matrix for A and the corresponding re-ordered distance matrix, or a subset of it when $N_B > N_A$, for B. The lower the DDM_{stat} , the better the match. The DDM_{stat} values quoted in this paper refer to coordinates of A and B which have been scaled into a unit cube.

Basic algorithm for simulated annealing

The algorithm can be expressed in pseudo FORTRAN as:

M = number of Markov chains

L = length of Markov chain

T = start temperature

E_s = sum of elements of initial DDM

DO 20 I = 1, M

 DO 10 J = 1, L

$s' = s$ (perturbed by configuration change)

$E_{s'} =$ sum of elements in new DDM for state s'

 IF ($\Delta E \leq 0$.OR. $\exp(-\Delta E/T) > \text{random}[0,1]$) THEN

$E_s = E_{s'}$

$s = s'$

 ENDIF

10 CONTINUE

$T = T - \Delta T$

20 CONTINUE

END

The algorithm is controlled by the following factors: the number and length of the Markov chains; the initial temperature and the temperature decrement; the method of perturbation, and the stop criterion. The value for the Boltzmann exponent, $-\Delta E/T$, must lie within the range 0 to -170 for $e^{-\Delta E/T}$ to be computable. If the performance of different algorithms is to be analysed with respect to the behaviour of the temperature decrement for different sized problems, then E , and consequently ΔE , should be appropriately scaled according to the size of the molecular matching problem.

Scaling of E and ΔE

A vital distinction has to be drawn between algorithms written to solve the TSP and those outlined here for molecular matching. Although both types of problem employ distance matrix calculations within the objective function, there is an important difference which precludes the direct use of TSP algorithms here. In the TSP the objective function is the path length through N cities but only one element of each row and column of the distance matrix is summed to generate E . If the distances are scaled to give a constant average distance between the two cities, E is then directly proportional to N . When two cities are exchanged randomly, the standard deviation of ΔE is independent of the number of cities. No scaling is needed for ΔE in the TSP. In sharp contrast with the molecular matching problem, the standard deviation of E and ΔE is increased as the number of atoms in A or B is increased. However, E can be made comparable for different sized matching problems by co-scaling A and B within a unit cube. Figure 1 shows the frequency distribution for the DDM elements taken from a 3-dimensional random distribution of points ($N_A =$

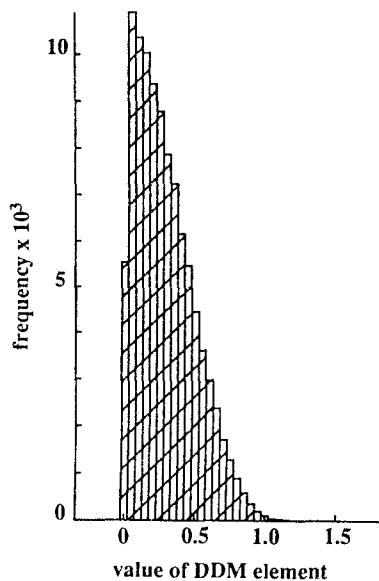


Fig. 1. Frequency distribution for DDM elements taken from a 3-dimensional random distribution of points, with identical coordinates for sets A and B. $N_A = N_B = 150$; mean = 0.28; standard deviation = 0.21; sample size = 100 000 elements.

150); the mean and standard deviations of all distributions for 20, 70, and 150 points are the same (0.28 ± 0.21). Thus the relationship between E and N_A is

$$E \propto N_A (N_A - 1)/2 \quad (8)$$

The value of the objective function that determines the course of annealing is ΔE . Data exploration revealed that the distribution of ΔE is normal with a mean of approximately zero (Figs. 2a, b and c). The relationship between the standard deviation of ΔE and N_A is linear. However, the equations for the regression lines for different data sets are different. This suggests that different shapes for A and B affect the distribution of ΔE . This finding eliminates the possibility of using a predictive equation for the standard deviation of ΔE . Therefore, $\sigma_{\Delta E}$, the standard deviation of ΔE , has to be obtained algorithmically for each problem pair of structures A and B. ΔE , taken from co-scaled data, can then be scaled according to

$$\Delta E_{\text{scaled}} = \Delta E_{\text{unscaled}} C / 3\sigma_{\Delta E} \quad (9)$$

where C is a constant to be determined. This equation assumes that the average ΔE is zero. The denominator ensures that 99.7% of the population of positive ΔEs are included in the scaling so that problems of different size can be handled. A numerical value for C can be obtained by considering the initial acceptance of positive values for ΔE . A positive value is accepted only if

$$\text{random}[0,1] < e^{-\Delta E_{\text{scaled}}/T} \quad (10)$$

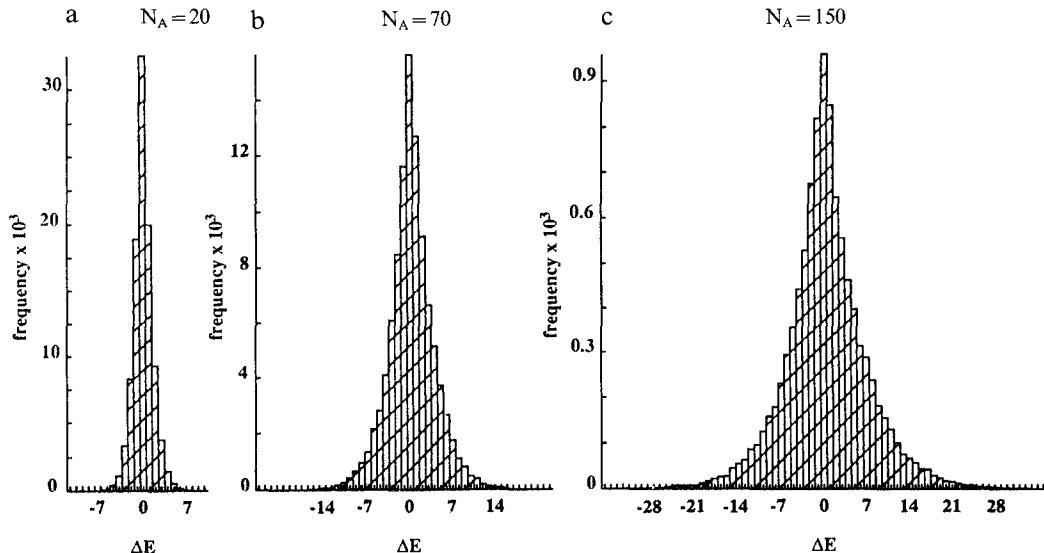


Fig. 2. Frequency distributions for values of ΔE taken from a 3-dimensional random distribution of points, with identical coordinates for sets A and B. Sample size = 100 000 ΔE values per histogram. a: $N_A = N_B = 20$; mean = 0.07; standard deviation = 1.59; skew = 0.08; kurtosis = 1.78; b: $N_A = N_B = 70$; mean = 0.26; standard deviation = 3.86; skew = 0.10; kurtosis = 2.24; c: $N_A = N_B = 150$; mean = 0.42; standard deviation = 6.80; skew = 0.10; kurtosis = 2.06.

Figure 3a shows the theoretical distribution of positive ΔE values along with the cumulative population; Fig. 3b illustrates the Boltzmann probability distribution for the acceptance of scaled positive ΔE at different temperatures and at $C=1$. As the temperature is lowered, the acceptance probability diminishes for any particular value of ΔE . The maximal acceptance for positive ΔE occurs when ΔE approaches zero and the acceptance probability becomes 1. Since the exponent is a ratio of $-\Delta E_{\text{scaled}}/T$, an appropriately determined scaling constant may circumvent the need for different initial starting temperatures which otherwise would be an added variable in unscaled problems. For example, suppose that the initial temperature is set to $T=2$, then the acceptance can be calculated if $\Delta E = 3 \sigma_{\Delta E}$; by substitution in Eq. 10, $C=1$ gives an acceptance probability for positive ΔE of 0.605, $C=4$ gives 0.135 and $C=8$ gives 0.018. In an annealing run with $C=8$ and $T=2$, 99.7% of values with a positive ΔE will be accepted with a probability ranging from 1 to 0.018; when T is reduced to 1, the acceptance ranges from 1 to 0.000335 and the minimization now approximates to iterative improvement, where only negative values of ΔE are accepted. The importance of scaling can now be appreciated: the same initial temperature can always be used for different sized molecules, and for structural comparisons with different configuration landscapes as well since the standard deviation reflects the variation in the landscape.

The exact specification of the objective function and the scaling of that function are the main important differences between the algorithm used here for molecular matching and those used in other applications. Further important variables which are common to other applications are reviewed extensively by van Laarhoven and Aarts [24]. The variables are mentioned briefly here since they are examined in the different algorithms assessed in this paper.

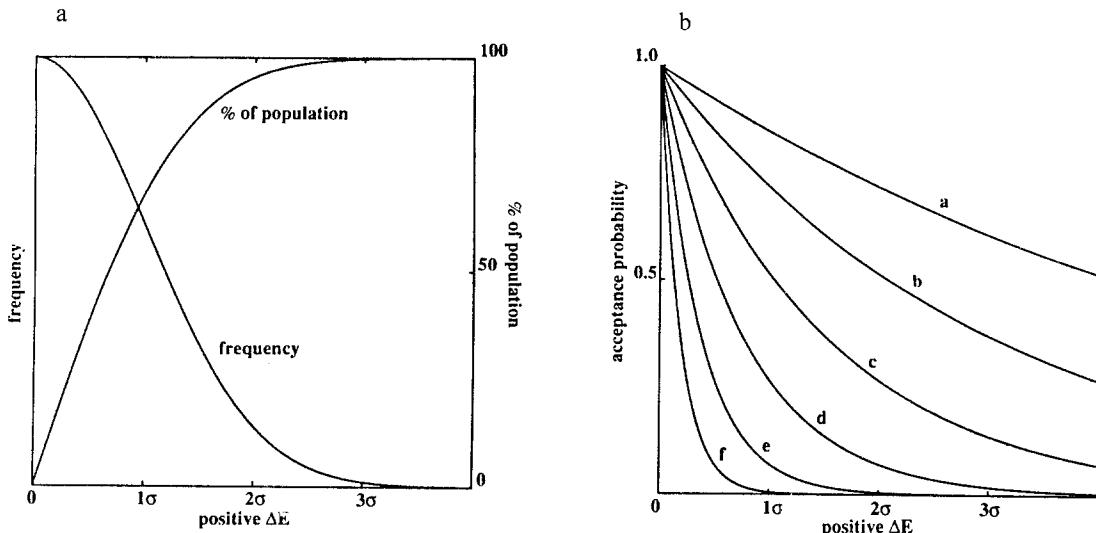


Fig. 3. a: Theoretical frequency distribution of positive ΔE values along with the cumulative population. b: The Boltzmann probability distribution for the acceptance of scaled positive ΔE values at different temperatures and at $C=1$. As the temperature is lowered, the acceptance probability diminishes for any particular value of ΔE . Line *a* is at $T=2.0$; *b* is at $T=1.0$; *c* is at $T=0.5$; *d* is at $T=0.25$; *e* is at $T=0.125$; *f* is at $T=0.0625$.

Length of the Markov chains

The Markov chain is the sequence of generated configurations of B obtained at a particular temperature. The length of the Markov chain, l_M , is crucial, since at each temperature, enough configurations of B must be generated to achieve an equilibrium position for E so that a stationary distribution for the homogeneous Markov chain is reached. Indeed, the length of the chain is one of the determinants of the cooling rate ($\Delta T/l_m$), and the slower the cooling rate, the better the final match, but this is at the expense of computing time. In practice two alternative approaches are possible: the length of the Markov chain can be fixed at some value,

$$l_M = 2 N_A(N_B - 1) \quad (11)$$

or the length should be a dynamic variable determined by the data encountered as the algorithm proceeds. The former approach was adopted, but with a minor modification: if the number of accepted moves in a Markov chain exceeded $N_A(N_B - 1)$, then the chain was terminated on the basis that equilibrium had been reached, and the temperature was subsequently lowered; otherwise the Markov chain was continued to a maximum length of $2 N_A(N_B - 1)$.

The initial temperature

The initial acceptance probability of a sequence of moves is dependent on the frequency of occurrence of negative ΔE values and the frequency of positive ΔE which satisfy the condition of Eq. 10. The acceptance ratio, η , is given by the number of accepted moves divided by the number of proposed moves in a particular Markov chain. Ideally the starting temperature should be high enough to give an acceptance ratio close to 1 and a value of $T = 2$ was chosen.

The temperature decrement

A temperature decrement has to be used so that a small Markov chain can reach equilibrium rapidly after a temperature change. This is an important feature of simulated annealing, since too fast a cooling results in trapping in poor local minima, whereas too slow a cooling requires much computer time. Three types of decrement are used in this paper:

(a) a linear decrease in temperature

$$T_{i+1} = T_i - \Delta T \quad (12)$$

where ΔT is a constant small fraction;

(b) an exponential decrease in temperature

$$T_{i+1} = T_i f \quad (13)$$

where f is a constant large fraction;

(c) a dynamic exponential decrease in temperature

$$T_{i+1} = T_i f_{\text{dyn}} \quad (14)$$

where f_{dyn} depends on the current performance of the algorithm. The parameter f_{dyn} is calculated from the equation

$$f_{\text{dyn}} = \{1 + [T_i \langle E \rangle_{T_i} \ln(1 + \delta)] / 3 \sigma_E\}^{-1} \quad (15)$$

where $\langle E \rangle_{T_i}$ is the mean value of E at a particular T_i , σ_E is the standard deviation of E at T_i and δ is a small positive number, set here to $\delta = 0.02$ [12].

The stop criterion

The simulated annealing algorithm is terminated when there is little material change in the value of the objective function. Three methods can be tested for the stop criterion: (a) when $\langle E \rangle$ is unchanged for three consecutive Markov chains; (b) when the difference between the maximum accepted E and the minimum accepted E is equal to the maximum accepted ΔE [25]; (c) when the acceptance ratio η , is smaller than a specified value. The last method was found to be the most efficient and the specified value for η used throughout the work was 0.008. In the linear cooling schedule, the additional stop criterion of T falling to zero was necessary.

Maximum number of Markov chains

The number of Markov chains, M , to be used is related to the number of possible states by the equation

$$M = \ln[N_B!/(N_B - N_A)!] \quad (16)$$

and the permutations are calculated from Stirling's formula.

METHODS

In this paper all the algorithms were tested on different data types. Firstly, random distributions of points were placed in a unit cube to simulate molecule A. The ability of the algorithms to produce a match could be tested exactly by using identical coordinate data for molecule B but presented in a different order from that of A. Point sets of different size were tested with $N_A = 20, 70$ and 150 giving respectively $10^{18}, 10^{100}$ and 10^{263} possible orderings for B. Secondly, a variation was used to assess the match when the structure of B was perturbed by shifting the coordinates of each point randomly in the range of $\pm 16\%$ of the corresponding position in A. This latter strategy was chosen because the intended future use of the program would be to attempt to match molecules with subsets of similar, but not necessarily identical atomic coordinate data. Thirdly, algorithm performance was tested on high resolution X-ray crystallographic coordinate data from the C_α -atoms of dihydrofolate reductase (*Lactobacillus casei* and *Escherichia coli*) obtained from the Brookhaven Protein Data Bank [26]. The algorithm developed by McLachlan [27] was used to calculate the rotation matrix and a translation vector to minimize the rms separation between the two structures for a given ordering. Special cases where coordinate data sets give rise to particular configuration landscape problems are investigated in the next paper [22]. Programs are written in

FORTRAN and run on an IBM 3084Q. Ribbon diagrams for the protein structures were drawn using SYBYL [28] on an Evans and Sutherland PS390.

These algorithms differ principally in the temperature decrement. The value of the constant used in Eq. 9 for scaling ΔE affects the acceptance probability for transitions involving uphill excursions of E . Tests were performed with the Aarts et al. [12] schedule with an initial temperature set to $T = 2$ to determine an appropriate constant.

As a further refinement to the algorithms, after the system 'froze' (i.e. when the stop criterion was fulfilled), the temperature was raised to 1.5 and a second annealing was performed; in this case however, only the worst 25% of point assignments were allowed to be swapped. The principal effect of this was to overcome any difficulties arising from unattainment of equilibrium at the end of a Markov chain, especially prominent with large numbers of points in sets A and B.

RESULTS

Illustration of simulated annealing in molecular matching

The dynamic cooling schedule of the algorithm of Aarts et al. [12] is the most sophisticated of the three algorithms studied and will be used to illustrate the matching process. Figure 4a shows the decrease in the average value of the objective function as the temperature is decreased ($C = 1$, $N_A = N_B = 70$; coordinates of set B differ from set A by $\pm 16\%$). The average value $\langle E \rangle$ is calculated at the end of each Markov chain. At high temperatures the value of $\langle E \rangle$ diminishes slowly until around $T = 0.75$, thereafter further cooling causes a rapid and smooth decrease in $\langle E \rangle$ to the minimum value. A jagged cooling profile is obtained with $C = 8$, $N_A = N_B = 20$ and a Markov chain half the normal length (Fig. 4b). The cooling pattern clearly shows how the algorithm is capable of handling large uphill excursions that are encountered at $T = 1.5$ and 1.0; smaller oscillations occur at lower temperatures.

The standard deviation of E , σ_E , (corresponding to the situation in Fig. 4a) varies with temperature and oscillates between 5 and 6 over the temperature range 0.25–2.0 (Fig. 4c). A peak in σ_E is found at $T = 0.2$; on further cooling, σ_E decreases to zero as the best match is attained. Larger consecutive variations are encountered in σ_E when the transition is reached leading to freezing.

The behaviour of the temperature factor, described by Eqs. 14 and 15, with progressive cooling is shown in Fig. 4d; f_{dyn} is dependent on $\langle E \rangle$ and σ_E . Rapid cooling occurs at high temperatures and f_{dyn} levels off at a value of about 0.95 at low temperatures. This figure should be studied in conjunction with Figs. 4a and c where the constituent parameters of f_{dyn} are illustrated separately. These curves (Figs. 4a, c and d) show that at a particular temperature, in this case $T = 0.25$, the value of $\langle E \rangle$ declines rapidly and σ_E peaks and then starts to decrease markedly; the algorithm responds in this transition phase so that cooling is slowed as f_{dyn} is increased before the frozen condition is reached.

The effect of cooling on the acceptance ratio, η , is shown in Fig. 5a. The acceptance ratio is defined as the ratio of the number of accepted to the number of proposed configurations in one Markov chain. The ratio includes all negative values of ΔE as well as some of the positive values of ΔE which have satisfied the Metropolis condition. The acceptance ratio rapidly decreases to zero below $T = 0.75$. The change in length of the Markov chain reflects the decline in η (Fig. 5b) until the chain length reaches $2 N_A(N_B - 1)$ whereupon l_M is fixed.

Further microscopic quantities in statistical mechanics can be monitored during cooling. They are often taken as useful indicators of convergence of the algorithms. Specific heat may be thought of in terms of the rate of change of the objective function $\langle E \rangle$ with respect to temperature. In thermodynamics entropy, S , is the capacity factor for isothermally unavailable energy

$$dS/dT = (1/T)d\langle E \rangle / T \quad (17)$$

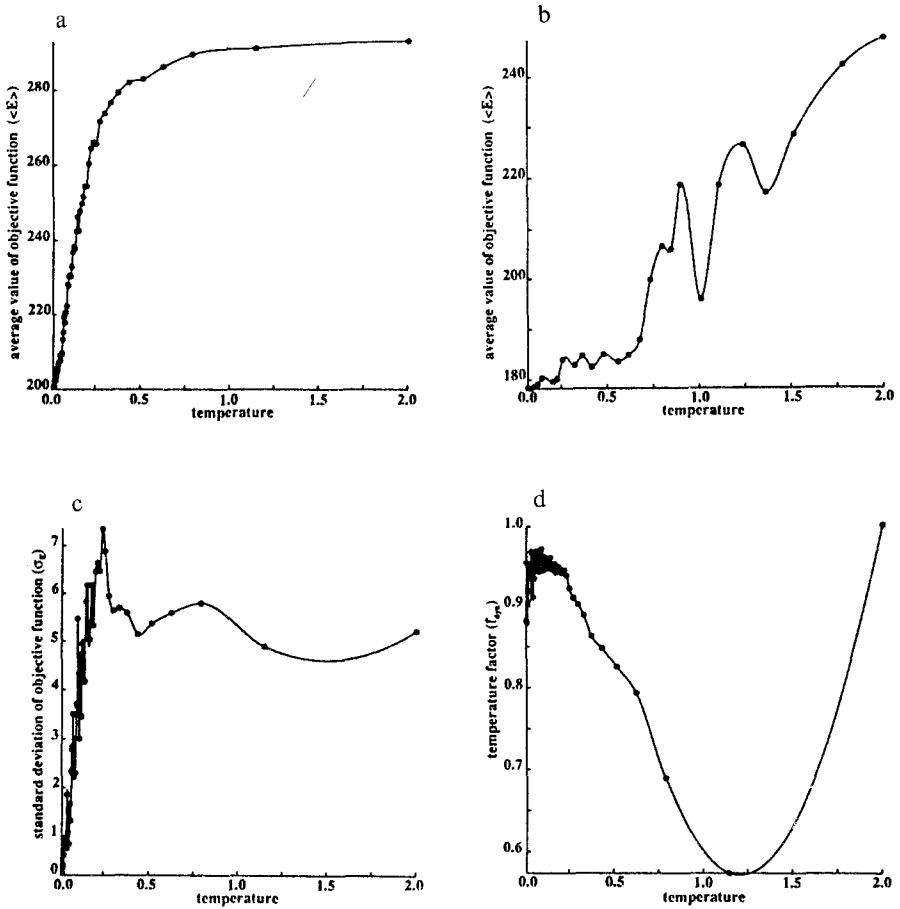


Fig. 4. a: The decrease in average value of the objective function as the temperature is lowered. The Aarts et al. [12] dynamic temperature factor was used. The coordinates of set B differed from set A by $\pm 16\%$ and were generated from a random 3-dimensional distribution of points. $C = 1$; $N_A = N_B = 70$. b: The decrease in average value of the objective function as the temperature is lowered. The Aarts et al. [12] dynamic temperature factor was used. The coordinates of set B differed from set A by $\pm 16\%$ and were generated from a random 3-dimensional distribution of points. $C = 8$; $N_A = N_B = 20$; and the length of the Markov chain was half its normal value. c: The variation in the standard deviation of E , σ_E , with temperature. The Aarts et al. [12] dynamic temperature factor was used. The coordinates of set B differed from set A by $\pm 16\%$ and were generated from a random 3-dimensional distribution of points. $C = 1$; $N_A = N_B = 70$; this corresponds to the situation in Fig. 4a. d: The behaviour of the Aarts et al. [12] dynamic temperature factor, described by Eq. 15, with progressive cooling. The coordinates of set B differed from set A by $\pm 16\%$ and were generated from a random 3-dimensional distribution of points. $C = 1$; $N_A = N_B = 70$; this corresponds to the situation in Fig. 4a.

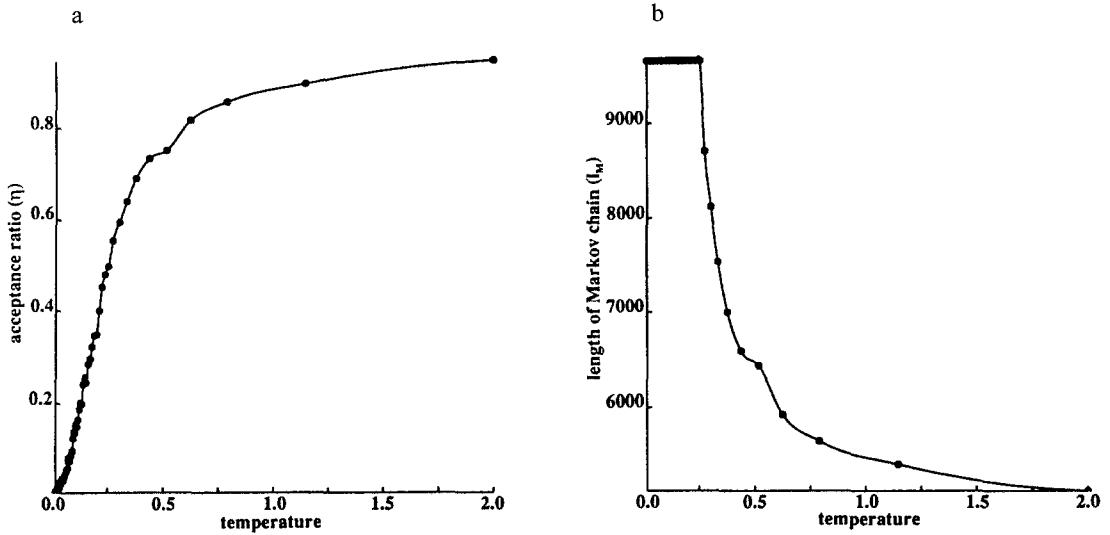


Fig. 5. a: The effect of cooling on the acceptance ratio, η . The Aarts et al. [12] dynamic temperature factor was used. The coordinates of set B differed from set A by $\pm 16\%$ and were generated from a random 3-dimensional distribution of points. $C=1$; $N_A=N_B=70$; this corresponds to the situation in Fig. 4a. b: The variation in the length of the Markov chain with temperature. The Aarts et al. [12] dynamic temperature factor was used. The coordinates of set B differed from set A by $\pm 16\%$ and were generated from a random 3-dimensional distribution of points. $C=1$; $N_A=N_B=70$; this corresponds to the situation in Fig. 4a.

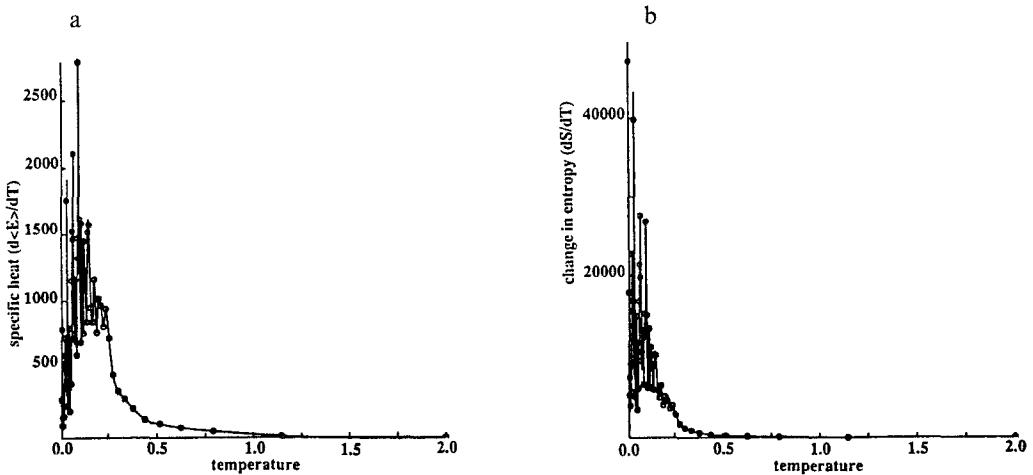


Fig. 6. a: The change in specific heat with temperature. The Aarts et al. [12] dynamic temperature factor was used. The coordinates of set B differed from set A by $\pm 16\%$ and were generated from a random 3-dimensional distribution of points. $C=1$; $N_A=N_B=70$; this corresponds to the situation in Fig. 4a. b: The variation in the rate of change of entropy with temperature. The Aarts et al. [12] dynamic temperature factor was used. The coordinates of set B differed from set A by $\pm 16\%$ and were generated from a random 3-dimensional distribution of points. $C=1$; $N_A=N_B=70$; this corresponds to the situation in Fig. 4a.

The variation in specific heat and the rate of change of entropy are shown in Figs. 6a and b. Specific heat starts to increase as the temperature falls to 0.75; the first peak occurs at $T=0.25$ and may indicate the onset of an important phase transition [29]. Similarly the change in entropy dS/dT indicates a marked transition at $T=0.25$.

Assessment of the performance of the algorithms

Identical sets of A and B, $N_A = N_B$

Tables 1, 2 and 3 illustrate the results from the three annealing algorithms using randomly generated points distributed in a unit cube. Each experiment is repeated 10 times using a non-repeatable start for the random number generator both for the distributions of points and for the Markov chains. The problem sizes are 20, 70 and 150 points for each set.

Table 1 illustrates the annealing data for the Aarts schedule using a dynamic temperature factor and identical coordinate sets for A and B. Scaling constants of $C=1, 4$ and 8 are tested for each problem size. The ability of the algorithm to solve the correspondence problem can be assessed in two ways; firstly by the change in the DDM statistic at the end of the annealing run, and secondly by determining the percentage of correct assignments. The amount of computation time needed is proportional to the number and length of the Markov chains required. The maximum length of the Markov chain is given by Eq. 11. For $N_A=20$, a perfect match is achieved in all 10 runs for each of the three scaling constants. The shortest amount of cpu time, 0.18 s is used with $C=8$ with an average of 3.2 Markov chains. However, with $C=1$ more Markov chains are needed and consequently the cpu time is greater (1.68 s). For $N_A=70$, the average final DDM statistic is similar for the three scaling constants and the number of correct assignments at $C=4$ is 92%, although in the 10 runs 8 produce perfect matches for all 70 points. On average 12.4 Markov chains took 14.3 s. With $C=8$ the assignments were worse, whereas a value of $C=1$ gave assignments no better than $C=4$ but was more expensive on cpu time. For $N_A=150$ the best results were achieved with $C=1$; 81.5% of the points were correctly assigned with 7 out of 10 of the runs producing perfect matches. The time for the annealing schedule averaged 721 s.

TABLE I
ANNEALING DATA USING THE AARTS et al. [12] DYNAMIC TEMPERATURE FACTOR^a

No. of points ($N_A=N_B$)	Scaling constant (C)	Initial DDM statistic	Final DDM statistic	Total No. of Markov chains	Total cpu time (s)	Correct assignments (%)
20	8	0.230	0.000	3.20	0.184	100
20	4	0.226	0.000	9.40	0.496	100
20	1	0.227	0.000	36.9	1.68	100
70	8	0.243	0.028	8.05	10.0	73.6
70	4	0.247	0.009	12.4	14.3	92.0
70	1	0.243	0.009	65.9	50.6	91.7
150	8	0.247	0.050	10.3	132	45.1
150	4	0.249	0.042	20.8	220	56.1
150	1	0.249	0.016	87.5	721	81.5

^a Identical coordinates for sets A and B were used, generated from a random 3-dimensional distribution of points. Scaling constants of $C=1, 4$ and 8 were tested for each problem size. The values represent the averages over 10 runs.

TABLE 2
ANNEALING DATA FOR THE EXPONENTIAL COOLING SCHEDULE^a

No. of points (N _A =N _B)	Temperature factor	Initial DDM statistic	Final DDM statistic	Total No. of Markov chains	Total cpu time (s)	Correct assignments (%)
20	0.85	0.231	0.020	10.9	0.320	86.0
20	0.90	0.224	0.095	10.6	0.357	84.0
20	0.95	0.229	0.000	3.80	0.216	100
70	0.85	0.243	0.019	16.4	17.8	81.4
70	0.90	0.242	0.018	22.3	23.4	83.6
70	0.95	0.245	0.018	34.4	35.4	83.4
150	0.85	0.247	0.034	35.3	522	64.9
150	0.90	0.247	0.036	53.1	822	67.9
150	0.95	0.249	0.016	86.5	1490	82.3

^a Identical coordinates for sets A and B were used, generated from a random 3-dimensional distribution of points. Temperature factors of 0.85, 0.90 and 0.95 were tested for each problem size. The values represent the averages over 10 runs. The following scaling factors were used: C=8 for N_A=20, C=4 for N_A=70, and C=1 for N_A=150.

These results suggest that it should be possible to select an appropriate scaling for the objective function for molecular matching problems of different size. All further experiments were performed with the following scaling constants: N_A=20, C=8; N_A=70, C=4; N_A=150, C=1.

Table 2 illustrates the results for an exponential cooling schedule using identical coordinate sets for A and B. Three temperature factors are used for each problem size. These factors determine how quickly the cooling takes place. With a scaling constant C=8 for N_A=20, only an exponential factor of 0.95 produced 100% correct assignments. For the other factors 8 out of 10 runs produced correct matches. An increase in the number of points to N_A=70 with C=4 produced 8 out of 10 correct matches for the three cooling factors. However, the percentage of correct assignments

TABLE 3
ANNEALING DATA FOR THE LINEAR COOLING SCHEDULE^a

No. of points (N _A =N _B)	Temperature decrement	Initial DDM statistic	Final DDM statistic	Total No. of Markov chains	Total cpu time (s)	Correct assignments (%)
20	0.250	0.231	0.000	3.20	0.190	100
20	0.175	0.232	0.000	3.20	0.176	100
20	0.125	0.226	0.000	4.40	0.243	100
70	0.250	0.247	0.019	11.2	12.4	83.7
70	0.175	0.246	0.009	12.7	16.3	92.9
70	0.125	0.244	0.019	18.7	21.6	83.9
150	0.250	0.248	0.149	14.0	244	8.60
150	0.175	0.248	0.070	21.0	371	35.1
150	0.125	0.246	0.082	28.0	486	25.5

^a Identical coordinates for sets A and B were used, generated from a random 3-dimensional distribution of points. Temperature decrements of 0.25, 0.175 and 0.125 were tested for each problem size. The values represent the averages over 10 runs. The following scaling factors were used: C=8 for N_A=20, C=4 for N_A=70, and C=1 for N_A=150.

was less than the Aarts schedule with the same scaling constant. For a problem size $N_A = 150$, only a temperature factor of 0.95 produced an assignment of 82% compared with a similar result for the Aarts schedule. However, the time taken by the exponential cooling schedule was twice as long as that of Aarts et al. [12].

A linear cooling schedule with $N_A = 20$, $C = 8$ and identical coordinate sets for A and B gave correct assignments for all 10 runs at each of the temperature decrements (Table 3). A comparable performance is found between $N_A = 70$, $\Delta T = 0.175$ and $C = 4$ for the linear cooling and the Aarts schedule; the time taken for both algorithms is similar. However, with the large matching problem, $N_A = 150$, the linear cooling schedule performed badly giving a low percentage of correct assignments.

Where a second attempt at annealing was necessary, significant improvements in the DDM statistic were found compared with the initial annealing procedure. This improvement was found for all three annealing schedules.

Different sets of coordinates for A and B, $N_A = N_B$

Tables 4, 5 and 6 show the performance of the annealing algorithm on matching sets of points where each pair of points differs by a random amount between $\pm 16\%$ of the coordinate position. Exact matches are highly improbable and the success of the algorithm in matching A and B can only be gauged from a decrease in the DDM statistic. In Table 4 the Aarts et al. schedules are compared for $N_A = 20, 70$ and 150 for each of the scaling constants $C = 1, 4$, and 8 . The values are the means of 10 runs. The final DDM statistics are broadly similar at about 0.165. The only significant differences are with the amounts of computing time which are related to the scaling constants for each group of points. The percentage reduction in DDM statistics is not significantly different for either the size of the matching problem or for cases with different scaling constants. Comparison of the computing times given in Tables 1 and 4 shows that where sets of 20 and 70 points are compared, the Aarts algorithm is slower for the cases where the points are not identical.

TABLE 4
ANNEALING DATA USING THE AARTS et al. [12] DYNAMIC TEMPERATURE FACTOR^a

No. of points ($N_A = N_B$)	Scaling constant (C)	Initial DDM statistic	Final DDM statistic	Total cpu time (s)
20	8	0.242	0.171	1.07
20	4	0.243	0.165	1.54
20	1	0.246	0.167	4.88
70	8	0.248	0.166	18.1
70	4	0.250	0.165	27.2
70	1	0.251	0.163	83.5
150	8	0.254	0.166	157
150	4	0.254	0.164	230
150	1	0.255	0.164	719

^a Coordinates for sets A and B differed by a random amount between $\pm 16\%$, generated from a random 3-dimensional distribution of points. Scaling constants of $C = 1, 4$ and 8 were tested for each problem size. The values represent the averages over 10 runs.

TABLE 5
ANNEALING DATA FOR THE EXPONENTIAL COOLING SCHEDULE^a

No. of points ($N_A = N_B$)	Temperature factor	Initial DDM statistic	Final DDM statistic	Total cpu time (s)
20	0.85	0.244	0.165	1.42
20	0.90	0.248	0.164	2.09
20	0.95	0.246	0.164	4.37
70	0.85	0.250	0.168	42.7
70	0.90	0.250	0.166	62.2
70	0.95	0.250	0.166	122
150	0.85	0.254	0.164	745
150	0.90	0.253	0.165	> 1080
150	0.95	0.254	0.164	> 1080

^a Coordinates for sets A and B differed by a random amount between $\pm 16\%$, generated from a random 3-dimensional distribution of points. Temperature factors of 0.85, 0.90 and 0.95 were tested for each problem size. The values represent the averages over 10 runs.

However, with 150 points the computing times are very similar.

Results from the exponential cooling schedule are shown in Table 5. Here the scheme for scaling constants is the same as that for Table 2. The final reduction in DDM statistic is similar for $N_A = 20$ and $N_A = 70$ for the three temperature factors. The exponential cooling schedule is definitely slower than the corresponding Aarts schedule for the same scaling constant. With $N_A = 150$ the exponential algorithm did not converge satisfactorily with temperature factors of 0.9 and 0.95.

Linear cooling is shown in Table 6. Few runs converged satisfactorily probably because the temperature rapidly reached zero.

TABLE 6
ANNEALING DATA FOR THE LINEAR COOLING SCHEDULE^a

No. of points ($N_A = N_B$)	Temperature decrement	Initial DDM statistic	Final DDM statistic	Total cpu time (s)
20	0.250	0.244	0.178	0.53
20	0.175	0.246	0.168	0.75
20	0.125	0.245	0.174	1.04
70	0.250	0.251	0.181	15.9
70	0.175	0.252	0.167	23.1
70	0.125	0.249	0.172	29.3
150	0.250	0.254	0.227	252
150	0.175	0.255	0.183	365
150	0.125	0.254	0.193	493

^a Coordinates for sets A and B differed by a random amount between $\pm 16\%$, generated from a random 3-dimensional distribution of points. Temperature decrements of 0.25, 0.175 and 0.125 were tested for each problem size. The values represent the averages over 10 runs.

Identical sets of A and B; $N_A \neq N_B$

Table 7 shows the results for the Aarts schedule where N_B has twice as many points as N_A although the coordinate data in A is a subset of B. Different scaling factors $C=1, 4$, and 8 are used. For $N_A=10$ and $N_B=20$ the subsets are well assigned; 9 out of 10 runs were exactly matched for $C=8$ and $C=4$. At $C=1$ all assignments were perfectly matched. Computing times were similar to the match for 20 points shown in Table 1. With $N_A=35$ and $N_B=70$ only 70% of the points were correctly assigned at $C=4$; the assignment improved to 81% with $C=1$; however, the differences are not statistically significant. Matching a subset of 75 points from 150 produced a correct assignment of about 70%. As a general statement the amount of cpu time is less for matching problems of a subset compared with matching the whole set.

Matching the C_α -atoms from dihydrofolate reductase (DHFR)

Coordinates taken from DHFR (*L. casei*) contain 162 C_α -atoms. Matching was performed using the Aarts et al. schedule and the results are shown in Table 8. The percentage of correct atom assignments was 71% when *L. casei* DHFR was compared with itself; 7 of the 10 runs gave a perfect match with the cpu time averaging 1160 s. The combinatorial problem of matching 162 atoms is 10^{26} times greater than that for matching 150 atoms.

Matching a subset $N_A=81$ against $N_B=162$ was performed by dividing the molecule into the first or last 81 sequential C_α -atoms. A scaling constant of $C=1$ gave 5 out of 10 perfect matches in the 1–81 subset and 4 out of 10 perfect matches in the 82–162 subset. The bad matches within a group showed a similar alignment to each other but were different from the perfect match alignment. This observation suggests the presence of a configurational landscape problem and is examined in more detail in the following paper [22].

Finally the alignment of DHFR from *E. coli* (DHFR from *E. coli* contains 159 C_α -atoms) and *L. casei* was investigated. Only 58 residues in the two enzymes show sequence identity although the general structure of both proteins is similar. All 159 C_α -atoms from *E. coli* are used as a set to

TABLE 7
ANNEALING DATA USING THE AARTS et al. [12] DYNAMIC TEMPERATURE FACTOR^a

No. of points (N_A)	No. of points (N_B)	Scaling constant (C)	Initial DDM statistic	Final DDM statistic	Total cpu time (s)	Correct assignments (%)
10	20	8	0.214	0.008	0.30	90.0
10	20	4	0.217	0.008	0.55	91.0
10	20	1	0.231	0.000	1.90	100
35	70	4	0.249	0.023	9.79	70.3
35	70	1	0.247	0.015	36.1	81.1
75	150	1	0.246	0.027	313	69.7

^a Identical coordinates for sets A and B were used, generated from a random 3-dimensional distribution of points, and $N_B=2 N_A$. Scaling constants of $C=1, 4$ and 8 were tested. The values represent the averages over 10 runs.

be compared with 159 out of 162 of *L. casei*. Atom correspondences are shown in Table 8 for 10 runs. The DDM statistic was reduced from 0.195 to 0.052 in an average of 1240 s using $C=1$ in the Aarts algorithm. The algorithm developed by McLachlan [27] was used to calculate the rotation matrix and a translation vector to minimize the rms separation between the two structures for a given ordering of 159 C_α -atoms from *L. casei* DHFR. The rotation matrix and translation vector were subsequently used to superimpose the whole of the *L. casei* DHFR C_α -backbone onto the *E. coli* backbone. Figure 7a shows the initial random alignment of the two structures and Fig. 8a shows the superimposition of the atoms after annealing. Figures 7b and 8b show the ribbon diagrams of the two superimposed structures. The rms separation for the best fit is 2.49 Å.

DISCUSSION

This paper has explored the application of the new technique of simulated annealing to the problem of matching molecular coordinates. Most coordinate fitting procedures require an explicit assignment of atom correspondences; simulated annealing does not have this constraint but searches for correspondences which give an optimal match. All atom positions are used in the matching procedure; bonding information is not utilised. The assignment of atom correspondences is a combinatorial problem; the task of fitting the 159 C_α -atoms of *E. coli* DHFR to *L. casei* DHFR containing 162 C_α -atoms would require approximately 10^{288} configurations by brute force testing. The two molecules illustrated here are proteins with the C_α -atoms sequentially linked through the peptide backbone; although linkage information is not utilised by the method. If linkage information is taken into account, then heuristic methods can easily be developed for the protein matching problem and atom assignment is reduced to finding appropriate deletions to match the coordinate sequences. This matching can be carried out using a moving frame travelling down the two sequences and comparing the matched overlays. Even so, Zuker and Somorjai [30] quote a cpu time of about 5.5 h on a VAX 11/750 for a smaller protein matching problem. Our paper is not principally concerned with protein matching per se but with finding optimum atom correspondences where bond topologies are unspecified. The algorithms find good, and frequently perfect, matches between 150 randomly generated test points without any linkage information. The methods are therefore general and applicable to the problem of matching any two sets of atoms in discrete space.

TABLE 8
ANNEALING DATA USING THE AARTS et al. [12] DYNAMIC TEMPERATURE FACTOR^a

C_α -atoms in set A	C_α -atoms in set B	Scaling constant (C)	Initial DDM statistic	Final DDM statistic	Total cpu time (s)	Correct assignments if applicable (%)
1–162 <i>L. casei</i>	1–162 <i>L. casei</i>	1	0.192	0.019	1160	71.1
1–81 <i>L. casei</i>	1–162 <i>L. casei</i>	1	0.189	0.021	513	50.0
81–162 <i>L. casei</i>	1–162 <i>L. casei</i>	1	0.197	0.028	491	40.2
1–159 <i>E. coli</i>	1–162 <i>L. casei</i>	1	0.195	0.052	1240	—

^a C_α -atoms of DHFR were matched. A scaling constant of $C=1$ was used. The values represent the averages over 10 runs.

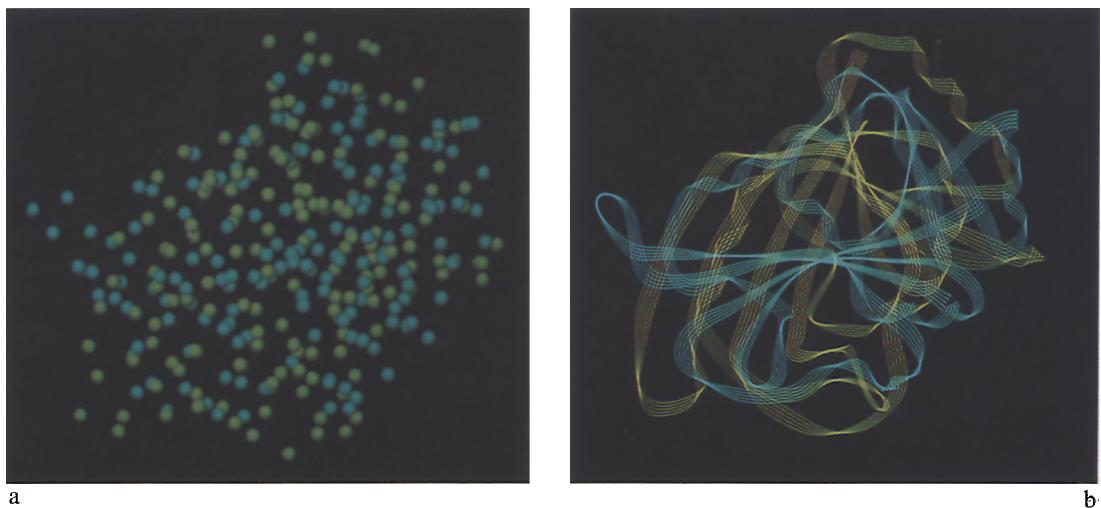


Fig. 7. Initial random orientation for *E. coli* (blue) and *L. casei* (yellow) DHFR. The rms separation for the first 159 C_α -atoms of each is 20.3 Å. a: Spheres represent C_α -atoms; b: Ribbon drawing of the backbones of the two enzymes.

The objective function used here has employed a difference distance matrix for the atomic coordinates. This method has the advantage that the distance matrix is invariant under rotation and translation. The distance matrix for molecules A and B along with the difference distance matrix needs to be computed only once. The difference distance matrix is updated by swapping only a pair of atoms if they satisfy the acceptance condition; this saves considerable amounts of computer time. The change in the value of the objective function, ΔE , needs to be scaled approxi-

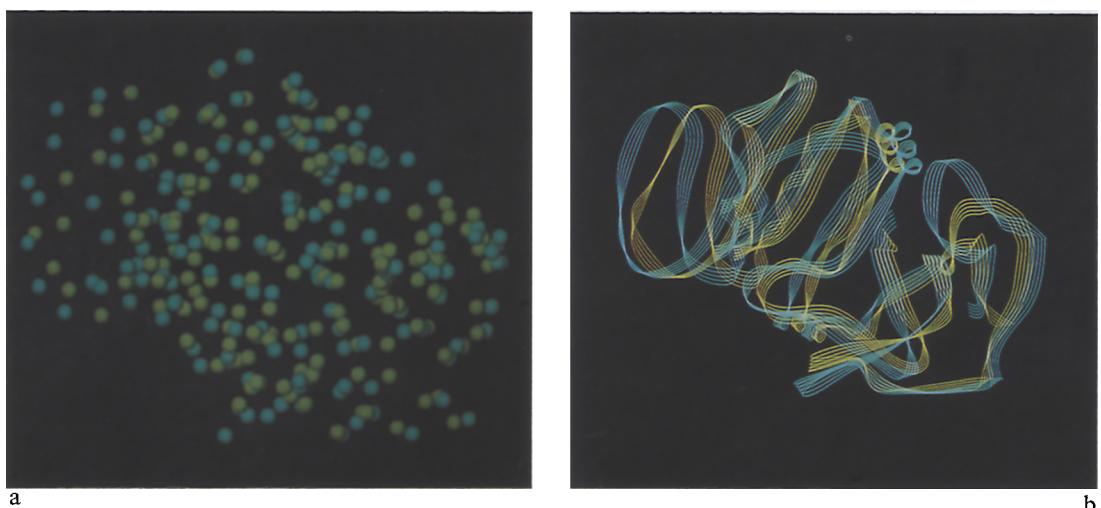


Fig. 8. Orientation for *E. coli* (blue) and *L. casei* (yellow) DHFR after annealing. The rms separation for the matched 159 C_α -atoms is 2.49 Å. a: Spheres represent C_α -atoms; b: Ribbon drawing of the backbones of the two enzymes.

mately to take into account problems of different size and to produce an expected range of values for the probability of accepting transitions to more dissimilar states. Too large a value for the scaling constant would result in trapping the algorithms in poor local minima and would approximate the minimization to iterative improvement, whereas a very low value would require too much cpu time.

Three cooling schedules have been tested: a linear decrement, an exponential decrease and a dynamically varying fractional change in temperature. The efficiency of these cooling schedules on the performance of the algorithm has been extensively tested on randomly distributed coordinates placed within a unit cube. Real molecular data was similarly co-scaled within a unit cube. The atom assignment problem, and hence the simulated annealing method, can be tested exactly by using two identical sets of coordinates but in a different random order; the global minimum is found when the value of the objective function is zero. In general the dynamic temperature factor of the Aarts et al. [12] algorithm proved to be the most effective in producing the most correct assignments and within the shortest time. The great advantage of the dynamic temperature factor is that the cooling rate is slowest at the critical transition phase. Where the two coordinate sets differ by $\pm 16\%$ the correct atom correspondence cannot be precisely established. However, the final values for the DDM statistic suggest that the methods using the dynamic temperature factor and the exponential schedule are consistently better than the linear cooling. Thus it can be inferred that the dynamic temperature factor cooling scheme suggested by Aarts et al. is the best method available for molecular matching by simulated annealing.

Multiple testing of the data using different random sequences gives an indication of the reliability of the method. In general the algorithm either gets it right with high consistency or, in those cases where the correct answer is not produced, a large number of the atom correspondences are wrong. The algorithm thus appears to find either the global minimum, or a local minimum in which it becomes trapped. Although simulated annealing is, in theory, an ergodic process for finding the global minimum in a discrete space, in practice the configurational landscape within a random sequence of moves may trap the configuration along a particular trajectory. The reason for this trapping may be, firstly, that the scaling constant is too large or the initial temperature is too low to give a greater proportion of accepted ΔE values in the initial cooling step; secondly, that the cooling schedule (where cooling rate is $\Delta T/l_M$) is too quick; thirdly, that the Markov chains are too short. However, multiple runs would determine the lowest minimum from the final DDM statistic. Serious landscape problems can be encountered where only a small subset of points are being matched with a large set or where there is pseudo-symmetry in the data. Configuration landscape problems are investigated in the following paper [22]. The dramatic changes in dS/dT and specific heat ($d\langle E \rangle /dT$), signalling the start of the transition phase towards the frozen condition, may provide a useful guide to future work for monitoring the behaviour of the objective function during annealing.

In conclusion, simulated annealing provides a fast and reliable method for determining atom correspondences between two molecules A and B if the set of atoms in A can be specified whereas those in B are not initially known. The combinatorial problems in atom assignment are effectively circumvented by the method. What is now required is an attempt to solve the problem of matching a number n_A of unspecified atoms from A with a subset from B. There are $N_A!/(N_A - n_A)!n_A!$ combinations of n_A atoms in A and $N_B!/(N_B - n_A)!$ permutations for B. The comparison of unspecified sets introduces a strong element of frustration into the matching; both this and the weak er-

godicity of the system, where pseudo symmetry could be encountered, form the crux of the null correspondences problem; its solution would be of great value in drug design.

ACKNOWLEDGEMENTS

We thank Dr. R.A. Lewis for readily providing the molecular graphics. P.M.D. wishes to thank the Wellcome Trust for personal financial support through the Principal Research Fellowship Scheme. Part of the work was supported by the Cambridge SERC Centre for Molecular Recognition.

REFERENCES

- 1 Willett, P., *Similarity and Clustering in Chemical Information Systems*, Research Studies Press. John Wiley & Sons, Chichester, 1987.
- 2 Dean, P.M., In Johnson, M. and Maggiore, G. (Eds.) *Concepts and Applications of Molecular Similarity*, Wiley, New York, 1990, in press.
- 3 Danziger, D.J. and Dean, P.M., *J. Theor. Biol.*, 116 (1985) 215.
- 4 Dean, P.M., Callow, P. and Chau, P.-L., *J. Mol. Graph.* 6 (1988) 28.
- 5 Bowen-Jenkins, P.E., Cooper, D.L. and Richards, W.G., *J. Phys. Chem.*, 89 (1985) 2195.
- 6 Maggiore, G. and Johnson, M., *Concepts and Applications of Molecular Similarity*, Wiley, New York, 1990, in press.
- 7 Papadimitriou, C.H. and Steiglitz, K., *Combinatorial Optimization: Algorithms and Complexity*, Prentice Hall, NJ, 1982.
- 8 Sedgewick, R., *Algorithms*, Addison-Wesley, Reading, MA, 1984.
- 9 Kirkpatrick, S., Gelatt, C.D. and Vecchi, M.P., *Science*, 220 (1983) 671.
- 10 Metropolis, N., Rosenbluth, A., Rosenbluth, M., Teller, A. and Teller, E., *J. Chem. Physics*, 21 (1953) 1087.
- 11 Kirkpatrick, S. and Toulouse, G., *J. Physique*, 46 (1985) 1277.
- 12 Aarts, E.H.L., Korst, J.H.M. and van Laarhoven, P.J.M., *J. Stat. Physics*, 50 (1988) 187.
- 13 Sechen, C. and Sangiovanni-Vincentelli, A.L., In Proc. 23rd Des. Automation Conf., Las Vegas, NV, 1986, pp. 423–439.
- 14 Ettelaie, R. and Moore, M.A., *J. Physique*, 48 (1987) 1255.
- 15 Bonomi, E. and Lutton, J.L., *Eur. J. Oper. Res.*, 26 (1986) 295.
- 16 Bernasconi, J., *J. Physique*, 48 (1987) 559.
- 17 Wille, L.T., *J. Comb. Theory A*, 45 (1987) 171.
- 18 Brunger, A.T., *J. Mol. Biol.*, 203 (1988) 803.
- 19 Brunger, A.T., Karplus, M. and Petsko, G.A., *Acta Crystallogr. A*, 45 (1989) 50.
- 20 Press, W.H., Flannery, B.P., Tenkolsky, S.A. and Vetterling, W.T., *Numerical Recipes: The Art of Scientific Computing*, Cambridge University Press, Cambridge, 1987.
- 21 Bounds, D.G., *Nature*, 329 (1987) 215.
- 22 Barakat, M.T. and Dean, P.M., *J. Comput.-Aided Mol. Design*, 4 (1990) 317.
- 23 Levitt, M., *J. Mol. Biol.*, 104 (1976) 59.
- 24 van Laarhoven, P.J.M. and Aarts, E.H.L., *Simulated Annealing: Theory and Applications*, Reidel, Dordrecht, 1988.
- 25 Huang, M.D., Romeo, F. and Sangiovanni-Vincentelli, A., In Proc. IEEE Int. Conference on Computer-Aided Design, Santa Clara, CA, 1986, pp. 381–384.
- 26 Bolin, J.T., Filman, D.J., Matthews, D.A., Hamlin, R.C. and Kraut, J., *J. Biol. Chem.*, 257 (1982) 13650.
- 27 McLachlan, A.D., *Acta Crystallogr. A*, 38 (1982) 871.
- 28 SYBYL, Molecular Modelling Software, Tripos Associates, St. Louis, MO, 1988.
- 29 Morgenstern, I., In Van Hemmen, J.L. and Morgenstern, I. (Eds.) *Heidelberg Colloquium on Glassy Dynamics and Optimizations*. Springer Lecture Notes in Physics, 275 (1987) pp. 399–427.
- 30 Zuker, M. and Somorjai, R.L., *Bull. Math. Biol.*, 51 (1989) 55.