doi:10.1006/jmbi.2000.3758 available online at http://www.idealibrary.com on IDEAL® *J. Mol. Biol.* (2000) **299**, 789–803

JMB

Trading Accuracy for Speed: A Quantitative Comparison of Search Algorithms in Protein Sequence Design

Christopher A. Voigt[1], D. Benjamin Gordon[2] and Stephen L. Mayo[3]*

[1]*Biochemistry Option Divisions of Biology and Chemistry and Chemical Engineering, California Institute of Technology mail code 210-41, Pasadena CA 91125, USA*

[2]*Division of Chemistry and Chemical Engineering California Institute of Technology, Pasadena CA 91125, USA*

[3]*Howard Hughes Medical Institute and Division of Biology, California Institute of Technology, mail code 147-75, Pasadena, CA 91125, USA*

**Corresponding author*

Finding the minimum energy amino acid side-chain conformation is a fundamental problem in both homology modeling and protein design. To address this issue, numerous computational algorithms have been proposed. However, there have been few quantitative comparisons between methods and there is very little general understanding of the types of problems that are appropriate for each algorithm. Here, we study four common search techniques: Monte Carlo (MC) and Monte Carlo plus quench (MCQ); genetic algorithms (GA); self-consistent mean field (SCMF); and dead-end elimination (DEE). Both SCMF and DEE are deterministic, and if DEE converges, it is guaranteed that its solution is the global minimum energy conformation (GMEC). This provides a means to compare the accuracy of SCMF and the stochastic methods. For the side-chain placement calculations, we find that DEE rapidly converges to the GMEC in all the test cases. The other algorithms converge on significantly incorrect solutions; the average fraction of incorrect rotamers for SCMF is 0.12, GA 0.09, and MCQ 0.05. For the protein design calculations, design positions are progressively added to the side-chain placement calculation until the time required for DEE diverges sharply. As the complexity of the problem increases, the accuracy of each method is determined so that the results can be extrapolated into the region where DEE is no longer tractable. We find that both SCMF and MCQ perform reasonably well on core calculations (fraction amino acids incorrect is SCMF 0.07, MCQ 0.04), but fail considerably on the boundary (SCMF 0.28, MCQ 0.32) and surface calculations (SCMF 0.37, MCQ 0.44).

Keywords: protein design; combinatorial optimization

Introduction

A goal of computational protein design is to compute an amino acid sequence *de novo* that will fold into a defined backbone structure (Street & Mayo, 1999). This is a difficult task, as protein stability results from the sum of many subtle and highly coupled interactions. By applying a quantitative, generalized approach, computational protein design has proven successful for cro protein (Desjarlais & Handel, 1995), gcn4 (Dahiyat & Mayo, 1996; Dahiyat *et al.*, 1997a), protein G (Dahiyat & Mayo, 1997b; Su & Mayo, 1997; Malakaukas & Mayo, 1998), ubiquitin (Lazar *et al.*, 1997), zinc finger domain (Dahiyat & Mayo, 1997a), and engrailed homeodomain (Morgan, 1999). Protein design has been successful in designing alpha-helical peptides that form right-handed supercoils (Harbury *et al.*, 1998). The trend towards designing sequences for larger and flexible backbones has been facilitated by a greater understanding of the forces responsible for protein stability as well as improvements in methods to search for the minimum energy conformation.

A combination of important techniques constitutes the protein design algorithm. First, the flexibility of each amino acid is coarse-grained into a discrete set of statistically significant conformations called rotamers (Ponder & Richards, 1987; Dunbrack & Karplus, 1993, 1994). While this drastically reduces the search space and makes the pro-

Abbreviations used: MC, Monte Carlo; MCQ, MC plus quench; GA, genetic algorithms; SCMF, self-consistent mean field; DEE, dead-end elimination; GMEC, global minimum energy conformation.

E-mail address of the corresponding author: steve@mayo.caltech.edu blem computationally tractable, the combinatorial complexity remains enormous. As an illustration, the number of side-chain conformations for a protein of n residues, $a = 20$ amino acids, and r rotamers per amino acid, is $(r \times a)^n$. For even a moderately sized enzyme, n is approximately 200-400, creating an immense set of possible solutions.

Second, the interactions between amino acids pertinent to stability have to be identified and their essence captured by a series of equations that together constitute the force-field. This description represents the balance of forces responsible for protein stability including van der Waals interactions, hydrogen bonds, salt-bridges, and hydrophobic-polar interactions (Gordon *et al.*, 1999). The energy term consists of three contributions; backbone/backbone, rotamer/backbone, and rotamer/rotamer. Because the backbone remains fixed in most protein design algorithms, it is not relevant to the optimization procedure. Therefore, the energy of a sequence folded into a defined structure can be expressed as:

$$E = \sum_{i=1}^{N} E(i_r) + \sum_{i=1}^{N-1} \sum_{j>i}^{N} E(i_r, j_s) \qquad (1)$$

where $E(i_r)$ is the rotamer/backbone energy for rotamer i_r of residue i, $E(i_r, j_s)$ is the rotamer/rotamer energy of rotamers i_r and j_s of residues i and j, respectively, and N is the total number of residues. By assuming that the energy between rotamers is pairwise as in equation (1), certain non-additive energy contributions cannot be treated exactly, such as a surface area-based solvation potential (Street & Mayo, 1998).

When the rotamer description is combined with the force-field, a discrete sequence-rotamer energy landscape is created in which each point represents a rotamer combination and an assigned energy. The final computational task in the protein design algorithm is to search this space for the global minimum energy conformation (GMEC) (Desjarlais & Clarke, 1998). Because the number of points in the landscape increases exponentially with protein size, the time to find the minimum scales unfavorably. In addition, the landscape contains a large number of local minima created by the high degree of side-chain coupling in the system. These effects compound, producing a hard search problem.

As a problem related to protein sequence design, homology modeling aims to align the sequence of an unknown structure with a sequence where the structure is known (Schiffer *et al.*, 1990; Lee & Subbiah, 1991; Tuffery *et al.*, 1991; Laughton, 1994; Lee, 1996; Sánchez & Šali, 1997). As the information in the sequence-structure database grows, it is increasingly observed that newly solved structures share structural motifs with proteins already in the database. Homology modeling consists of three central steps. First, a match is found between the sequence of the unknown structure and a sequence in the database of known structures. Then, the sequence is threaded onto the known backbone. Finally, the side-chains are arranged onto the backbone based on an energy expression (Lee & Subbiah, 1991; Laughton, 1994). There are two issues in positioning the side-chains correctly: the accuracy of the force-field and rotamer descriptions, and the ability to find the minimum energy arrangement. Finding the GMEC of side-chain descriptions has led to the proposal of many search algorithms (Desjarlais & Clarke, 1998). Here, we are interested in comparing the different proposed techniques for energy minimization and are not concerned whether the GMEC of the energy landscape actually coincides with the proper (experimentally determined) side-chain conformation.

There are two general classes of search algorithms: stochastic and deterministic. Stochastic algorithms such as Monte Carlo (MC) (Metropolis *et al.*, 1953) and genetic algorithms (GA) (Holland, 1993) rely on probabilistic trajectories where the outcome is determined by the initial conditions as well as the random number generator seed. Confirming that a solution is the GMEC is impossible, as there is always a degree of uncertainty. In contrast, deterministic methods will repeat the same solutions given the set of parameters used. Both dead-end elimination (DEE) (Desmet *et al.*, 1992) and self-consistent mean-field (SCMF) (Koehl & Delarue, 1994; Lee, 1994) are deterministic; however, they often do not converge to the same solution. If DEE converges, it is the GMEC, whereas this is not necessarily true for SCMF. The issues in comparing search algorithms include weighing the accuracy of the solution with the computational time required. Recently, the common algorithms used in protein sequence design have been reviewed (Desjarlais & Clarke, 1998). However, the tradeoffs of choosing one method over another are not well understood and there has been no comprehensive comparison of methodologies. An understanding of the strengths and weaknesses of each algorithm is required so (1) the algorithm best suited to the design problem can be utilized and (2) if an algorithm is chosen that does not give the GMEC, the expected accuracy of the solution can be estimated. Here, we compare four common approaches; MC, GA, SCMF, and DEE, for both side-chain placement and protein design calculations.

Description of Search Algorithms

Monte Carlo

As one of the simplest stochastic search techniques, Monte Carlo (Metropolis *et al.*, 1953) often performs well on difficult energy landscapes. MC has been applied to problems relating to protein sequence design (Holm & Sander, 1992; Hellinga & Richards, 1994; Godzik, 1995; Sasai, 1995; Dahiyat & Mayo, 1996). Initially, the rotamers for a sequence are picked at random. Then, a rotamer substitution is made at a randomly picked residue in the sequence. Rotamers of different amino acids are treated equally, so a rotamer substitution can be either the same amino acid or a new one. A new energy $E_{\rm new}$ is calculated and if this energy is lower than the previous energy $E_{\rm old}$, the move is accepted. If the energy is higher, the move is accepted with the Boltzman probability $p = \exp(-\beta(E_{\rm new} - E_{\rm old}))$, $\beta = 1/kT$, where k is Boltzman's constant. The role of the temperature T is to overcome the multiple local minima in the energy landscape by allowing the trajectory to surmount energy barriers. To strengthen this effect, an initial temperature is selected and annealed. The temperature is then cyclically raised and lowered over the course of a single run between a designated high and low temperature (for the calculations performed here, the high and low temperatures were set to 4000 and 150 K, respectively). The optimization can be run for any number of cycles, with each cycle containing a number of substitution attempts. Here, the optimization is run for 20 cycles of 10^6 substitution attempts for each test case. The number of cycles is arbitrarily set at 20 to generate computational times comparable to SCMF and DEE. MC can be run for longer periods to theoretically produce better solutions. In our hands, the number of cycles has been typically set at 1000.

At the end of the run, the energy of the stored solutions may be quenched. For each residue, in random order, all possible rotamers of the amino acid in the solution are attempted. If a new rotamer is lower in energy, it is kept; if not, it is rejected. The quench step produces a large improvement in the solution while adding trivially to the time of the run. This step assures that there are no single-rotamer changes that will improve the energy. For the side-chain placement calculations, results are presented for both the MC procedure alone as well as the MC with the quenching step (MCQ).

Genetic algorithm

Genetic algorithms seek to optimize a population of solutions using biologically inspired operators (Holland, 1993). GAs have been applied to a wide range of problems, including protein structure prediction (Pedersen & Moult, 1996) and design (Jones, 1994; Desjarlais & Handel, 1995; Lazar *et al.*, 1997). The advantages of a GA are that the population dynamics can overcome energy barriers by making moves in sequence space that are larger than the moves typically used in MC. In addition, beneficial mutations can be combined onto a single sequence, increasing the number of paths that circumvent local minima. As a disadvantage, GAs tend not to work well on highly coupled systems where crossover disruption is problematic, as is expected for side-chain systems. Further, residues that are close in sequence are not necessarily close structurally, making it difficult for the algorithm to find clean crossover points.

While the specific implementation of GAs varies tremendously in the literature, there are several common characteristics. In order to study the effectiveness of this approach to the protein design problem, we tried several different algorithms and chose a relatively universal description of a GA that produced the best results. The implementation of our algorithm is slightly different from that described by Desjarlais & Handel (1995). They include an inversion operator and utilize a different selection scheme. It is not expected that these differences would significantly change our results.

First, a population of $M = 50$ random sequences is initialized. Then, mutations are applied at rate $p_M = 0.016$, producing a Poisson distribution of mutations with an average of one per sequence. The new energies of the mutants are determined and ranked. The top C of these mutants are chosen for recombination, where C represents the recombination rate. Here, the optimal value is found to be $C = 10$. For each pair, the strings are recombined by comparing each residue and if the rotamers differ among the parents, the offspring will inherit either parent's rotamer with equal probability. The new population is generated using the tournament selection technique, where S sequences are picked randomly from the mutant library and their energies compared. The sequence with the lowest energy is allowed to continue to the next generation. The selection process is repeated M times to produce the pool of sequences that will continue to the next round of mutation, recombination, and selection. This algorithm is repeated, so the average fitness of the population improves after each cycle until equilibrium is reached.

The selection strength, represented by the number of sequences S that undergo competition, is analogous to the temperature in the MC algorithm. By starting at low S and annealing to a high S, the population distribution in sequence space is first very broad and then narrows after each generation until the population consists of a single sequence. This ''heating and cooling'' process is repeated to improve the probability that the population will find lower minima. At the beginning of each cycle, S is initialized at 2 and is incrementally increased to 5. The full optimization procedure consists of ten cycles of 10^4 mutation-recombination-selection steps. Due to its size, the number of cycles was increased to 15 for the 1arb test case. Similar to the Monte Carlo algorithms, the number of cycles was arbitrarily set to produce competitive times against the deterministic algorithms.

Self-consistent mean field

Unlike the MC or GA algorithms that focus on clever search methods to evade local minima, SCMF uses a mean-field description of the rotamer interactions to alter the energy landscape (Lee, 1994; Koehl & Delarue, 1994, 1995, 1996; Vásquez, 1995). SCMF is deterministic, in that given a set of run parameters, the algorithm will always converge to the same solution. Unfortunately, there is no guarantee that the minimum of the mean-field landscape corresponds with the true GMEC. The advantage of SCMF is that the computational time scales linearly with the number of residues, making it possible to obtain solutions for proteins currently unattainable by other methods (Koehl & Delarue, 1996).

As derived by Koehl & Delarue (1994), the mean-field energy for rotamer i_r at residue i is:

$$E_{mf}(i_r) = E(i_r) + \sum_{j \neq i}^{N} \sum_{s=1}^{K_j} E(i_r, j_s) V(j_s) \qquad (2)$$

where K_j is the total number of rotamers at residue j. The weight of each rotamer $V(j_s)$ (the conformational probability vector) is normalized to unity. The first term in equation (2) is the contribution due to the interaction between the rotamer and the backbone, and the second term describes all the inter-rotamer pairwise interactions weighted by the probability of that rotamer existing in the GMEC. The conformational probability vector can be independently calculated by Gibb's ensemble:

$$V(j_s) = \frac{1}{q_j} \mathrm{e}^{-\beta E_{mf}(j_s)} \qquad (3)$$

where q_j is the partition function:

$$q_j = \sum_{s=1}^{K_j} \mathrm{e}^{-\beta E_{mf}(j_s)} \qquad (4)$$

The effect of this procedure is to smooth the landscape and avoid the problem of multiple local minima, making it relatively simple to locate the minimum of the mean-field energy landscape.

The mean-field energy is minimized using an annealing method as described by Lee (1994). A high initial temperature (often >20,000 K) is chosen and the probability vector $V(j_s)$ is initialized at $1/K_j$, thereby assigning equal probability to each rotamer. The purpose of annealing the temperature is to assist the convergence, reducing the total run time. The solution found by SCMF is not dependent on the specific initial temperature used.

A pair-energy threshold is applied that implements a ceiling to which higher energies are set. The success of SCMF is highly dependent on the optimization of this parameter. The optimal threshold is determined individually for each side-chain placement test case and is found to vary widely between 5 and 500 kcal/mol. Qualitatively, smaller backbones tended to correspond with a smaller threshold. The time required for SCMF to converge did not differ significantly with the threshold chosen. For the sequence design calculations, this parameter was set at 500 kcal/mol due to the increase in the problem difficulty.

After initialization, the mean-field potential E_{mf} (i_r) is calculated from equation (2) for each residue and rotamer. The energies are converted into probabilities using Gibb's equations. The algorithm iterates between equations (2) and (3) until the energy converges and self-consistency is achieved (Koehl & Delarue, 1994). A convergence criterion of 0.0001 for $V(j_s)$ was used to define self-consistency. Convergence is significantly improved if the probability vector V is updated with a "memory" of the previous step as expressed by the following:

$$V_{\mathrm{new}}(j_s) = \lambda V_{\mathrm{new}}(j_s) + (1 - \lambda) V_{\mathrm{old}}(j_s) \qquad (5)$$

where the optimum step size was found to be $\lambda = 0.9$ (Koehl & Delarue, 1994). The temperature is then lowered in linear increments of 100 K and the routine repeated. When the final temperature is reached (100 K), the conformational vector represents the probability of finding each rotamer at a given residue position. The best solution is determined as the collection of rotamers that have the highest probability at each position.

Dead-end elimination

As opposed to optimizing a single solution or set of solutions by iterative improvement, as done by the MC procedure or GA, dead-end elimination seeks to systematically eliminate bad rotamers and combinations of rotamers until a single solution remains. Unlike SCMF, the theoretical basis for DEE proves that if DEE converges, the solution is the GMEC with no uncertainty (Desmet *et al.*, 1992). It is a necessary criterion for DEE that the energy description is pairwise as described in equation (1), and the effectiveness of the search is due, in part, to the distribution of interactions that arise in a protein side-chain system (Goldstein, 1994).

DEE is fundamentally based on the following physical concept. Consider two rotamers, i_r and i_t, at residue i and the set of all other rotamer configurations $\{S\}$ at all residues excluding i of which rotamer j_s is a member. If the pairwise energy contributed between i_r and j_s is higher than the pairwise energy between i_t and j_s for all $\{S\}$, then i_r cannot exist in the GMEC and can be eliminated. This notion is expressed mathematically by the inequality:

$$E(i_r) + \sum_{j \neq i}^{N} E(i_r, j_s) > E(i_t) + \sum_{j \neq i}^{N} E(i_t, j_s) \ \forall \{S\} \qquad (6)$$

If the above is true, the single rotamer i_r can be eliminated (Desmet *et al.*, 1992). In this form, inequality (6) is not computationally tractable because, to make an elimination, it is required that the entire sequence/rotamer space be enumerated. To simplify the problem, the bounds implied by inequality (6) can be utilized:

$$E(i_r) + \sum_{j \neq i}^{N} \min_{s} E(i_r, j_s) > E(i_t) + \sum_{j \neq i}^{N} \max_{s} E(i_t, j_s) \qquad (7)$$

Using an analogous argument, equation (7) can be extended to the elimination of pairs of rotamers inconsistent with the GMEC. This is done by determining that a pair of rotamers, i_r at residue i and j_s at residue j, always contribute higher energies than rotamers i_u and j_v with all possible rotamer combinations $\{L\}$. Similar to equation (7), the strict bound of this statement is given by:

$$\varepsilon(i_r, j_s) + \sum_{k \neq i,j}^{N} \min_t \varepsilon(i_r, j_s, k_t) > \varepsilon(i_u, j_v) + \sum_{k \neq i,j}^{N} \max_t \varepsilon(i_u, j_v, k_t)$$

where ε is the combined energies for rotamer pairs:

$$\varepsilon(i_r, j_s) = E(i_r) + E(j_s) + E(i_r, j_s) \tag{9}$$

and:

$$\varepsilon(i_r, j_s, k_t) = E(i_r, k_t) + E(j_s, k_t) \tag{10}$$

This leads to the doubles elimination of the pair of rotamers i_r and j_s, but does not eliminate the individual rotamers completely, as either could exist independently in the GMEC. The doubles elimination step reduces the number of possible pairs (reduces S) that need to be evaluated in the right-hand side of equation (7), henceforth allowing more rotamers to be individually eliminated.

The singles and doubles criteria presented by Desmet *et al.* (1992) fail to discover special conditions that lead to the determination of more dead-ending rotamers. For instance, it is possible that the energy contribution of rotamer i_t is always lower than i_r without the maximum of i_t being below the minimum of i_r. To address this problem, Goldstein (1994) presented a modification of the criteria that determines if the energy-profiles of two rotamers cross. If they do not, the higher-energy rotamer can be determined to be dead-ending. The improved criterion for singles is:

$$E(i_r) - E(i_t) + \sum_{j \neq i}^{N} \min_s [E(i_r, j_s) - E(i_t, j_s)] > 0 \tag{11}$$

and likewise for doubles:

$$\varepsilon(i_r, j_s) - \varepsilon(i_u, j_v) + \sum_{k \neq i,j}^{N} \min_t [\varepsilon(i_r, j_s, k_t) - \varepsilon(i_u, j_v, k_t)] > 0 \tag{12}$$

In computational time, the doubles calculation is significantly more expensive than the singles calculation. To accelerate the process, computationally inexpensive methods have been developed to predict the doubles calculations that will be the most productive (Gordon & Mayo, 1998). These modifications, collectively referred to as fast doubles, significantly improved the speed and effectiveness of DEE.

The probability of successfully finding the GMEC has been shown to improve by utilizing an expanded rotamer library and including an initial energy threshold (De Maeyer *et al.*, 1997). For these calculations, we use an energy cut-off of 1000 kcal/mol. Single or double rotamers that produce energies above this threshold are automatically flagged as dead-ending. These values are considered conservative and ensure that the minimum energy found is the GMEC. Parameters that are more aggressive can be used to improve the speed of DEE, but accuracy is sometimes lost.

Several additional modifications collectively enhance DEE further. Rotamers from multiple residues can be combined into so-called super-rotamers to prompt further eliminations (Desmet *et al.*, 1994; Goldstein, 1994). This has the advantage of eliminating multiple rotamers in a single step. In addition, it was shown that "splitting" the conformational space between rotamers improves the efficiency of DEE (Pierce *et al.*, 2000). Splitting handles the following special case. Consider rotamer i_r. If a rotamer i_{t1} contributes a lower energy than i_r for a portion of the conformational space, and a rotamer i_{t2} has a lower energy than i_r for the remaining fraction, then i_r can be eliminated. This case would not be detected by the less sensitive Desmet or Goldstein criteria. In the implementation used in this study, all of the enhancements described above were combined into a single computational approach. Because of these improvements, convergence to the GMEC in less than 0.5 hour on a single processor can now be expected for side-chain placement calculations on proteins in excess of 300 residues.

Results and Discussion

Side-chain placement

DEE converges to the GMEC rapidly for the entire side-chain placement test set, thereby providing the standard to which the solutions found by other methods can be compared. The results are shown in Table 1 and Figures 1 and 2. We found that MC and SCMF consistently perform the worst, with the average fraction of incorrect rotamers $\langle f \rangle = 0.23$ and 0.12, and the average difference in energy from the GMEC $\langle \Delta E \rangle = 5.6$ and 5.9 kcal/mol, respectively. It is interesting that SCMF consistently gave solutions that have fewer incorrect rotamers, but worse energies than MC, indicating that the methods are failing by different mechanisms. We believe MC does poorly because it becomes easily trapped by rotamer combinations that are relatively low in energy, whereas SCMF has difficulty converging the probability of a single rotamer to unity at certain sequence positions. The GA performed better; $\langle f \rangle = 0.09$ and $\langle \Delta E \rangle = 4.3$ kcal/mol. The MCQ outperformed the other methods with $\langle f \rangle = 0.05$ and

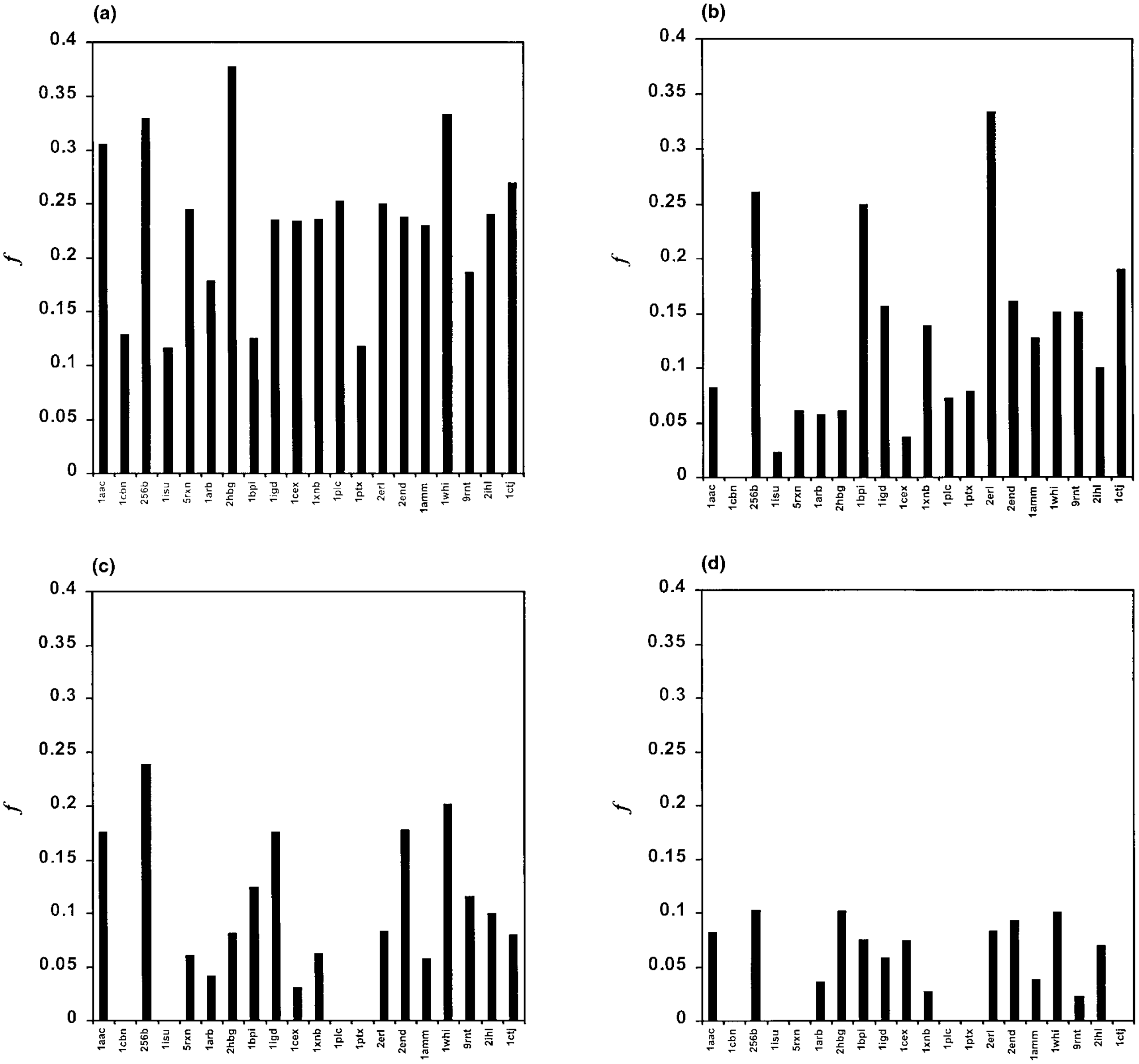

Figure 1. The GMEC was determined by DEE and compared to the result given by the other algorithms for the side-chain placement calculations. The fraction of incorrect rotomers *f* predicted by (a) MC, (b) SCMF, (c) GA, and (d) MCQ is shown for each protein in the test set, as compared to the GMEC found by DEE.

$\langle \Delta E \rangle = 1.3$ kcal/mol. The ability for MCQ to give reasonable solutions indicates that there is no benefit to the more complex GA or SCMF methods. The 2hbg structure was the only case out of 20 where SCMF outperformed MCQ in both *f* and ΔE.

The relationship between the size of sequence/rotamer space and the number of incorrect rotamers predicted by the algorithms was determined (data not shown). As expected, MC showed the greatest correlation ($R^2 = 0.81$) because it is a sampling algorithm and as the size of the search space increases and the number of cycles remain fixed, the fraction of the space searched decreases. Both SCMF and GA do not fit as well ($R^2 = 0.27$ and 0.20), indicating that there are other aspects of the energy landscape that impede their search, such as the strength and distribution of coupling interactions. It has been suggested that the advantage of SCMF is that it provides solutions for problems for which DEE does not converge (Koehl & Delarue, 1996). As shown in Table 2, this is not necessary for side-chain placement calculations, as the recent improvements in DEE have allowed it to converge on solutions in times comparable to SCMF. The times for both the MC/MCQ and GA runs presented here are arbitrary because the algorithms could be run indefinitely and better sol-

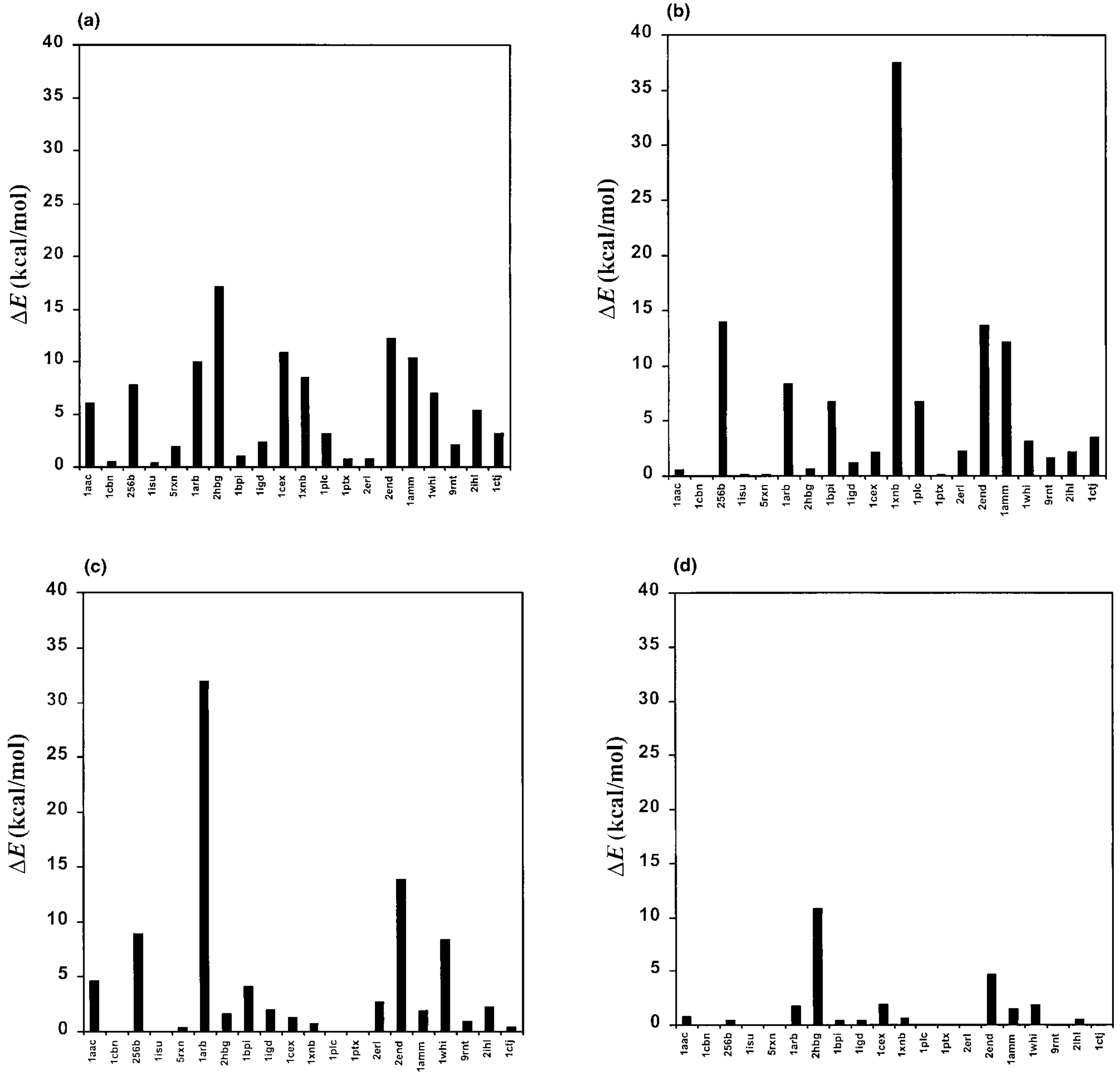

Figure 2. The difference between the GMEC found by DEE and the energy determined by (a) MC, (b) SCMF, (c) GA, and (d) MCQ for the side-chain placement calculations.

utions might be obtained. Here, we ran a fixed number of cycles, making the larger proteins appear to take longer.

The results of the side-chain placement calculations strongly suggest that there is no compelling reason to use an algorithm other than DEE for side-chain placement as it consistently and quickly converges to the GMEC. However, as design calculations become more complex, there is a point beyond which DEE will not converge in a reasonable amount of time. To solve these problems, it is necessary to trade the accuracy of DEE for the speed of SCMF or MCQ.

Sequence design

For the protein sequence design comparisons, amino acid substitutions are allowed at the designed positions, while the side-chains for the remaining residues are floated, as in the side-chain placement calculations. By specifying more positions to be designed, the difficulty of the problem can be tuned from the easier side-chain placement calculation to an intractable full sequence design. Because the GA and MC methods rarely outperformed MCQ, they are not run on the more difficult design problems. While DEE performed

Table 1. Results of side-chain placement calculations

	Number rotamers incorrect[a]				ΔE (kcal/mol)[a]			
	SCMF	MCQ	MC	GA	SCMF	MCQ	MC	GA
1aac	7	7	26	15	0.6	0.8	6.2	4.6
1cbn	0	0	4	0	0.0	0.0	0.5	0.0
256b	23	9	29	21	14.0	0.4	7.8	8.9
1isu	1	0	5	0	0.2	0.0	0.4	0.0
5rxn	3	0	12	3	0.2	0.0	1.9	0.5
1arb	11	7	34	8	8.4	1.8	10.0	32.0
2hbg	6	10	37	8	0.7	10.8	17.2	1.7
1bpi	10	3	5	5	6.8	0.4	1.0	4.1
1igd	8	3	12	9	1.3	0.5	2.4	2.1
1cex	6	12	38	5	2.2	2.0	10.9	1.4
1xnb	20	4	34	9	37.6	0.7	8.5	0.8
1plc	6	0	21	0	6.8	0.0	3.2	0.0
1ptx	4	0	6	0	0.2	0.0	0.8	0.0
2erl	8	2	6	2	2.3	0.1	0.8	2.8
2end	19	11	28	21	13.7	4.7	12.2	13.9
1amm	20	6	36	9	12.2	1.5	10.4	1.9
1whi	15	10	33	20	3.2	1.9	7.0	8.4
9rnt	13	2	16	10	1.7	0.0	2.1	1.0
2ihl	10	7	24	10	2.2	0.6	5.4	2.3
1ctj	12	0	17	5	3.6	0.0	3.2	0.4
Average[b]	0.12	0.05	0.23	0.09	5.9	1.3	5.6	4.3

[a] The solution as compared to the GMEC found by DEE.

[b] The average fraction of rotamers incorrect and the average energy above the GMEC in kcal/mol.

extraordinarily well on the side-chain placement calculations, the time to convergence explodes as the number of designed residues reached some threshold (Table 3 and Figure 3). In contrast, the times required by the other algorithms scale linearly with increasing problem size. This is notably true for SCMF, as the time to solve even large design problems is often less than 30 minutes on a single processor. MCQ is allowed to run for the same number of cycles as the side-chain placement calculations and requires between 60 and 120 minutes to complete. However, we observe that it is highly unlikely that either SCMF or MCQ provides a solution that is the GMEC.

Two important questions arise from this conflict. First, if DEE does not converge, which alternative method will produce the best results? Second, at the point at which DEE explodes, how incorrect

Table 2. Times for side-chain placement calculations

	Time (minutes)			
	DEE	SCMF	MC/Q[a]	GA
1aac	0.2	0.7	53.9	42.0
1cbn	0.0[b]	0.0[b]	7.1	6.1
256b	5.1	0.5	62.8	44.1
1isu	0.0[b]	0.1	16.5	14.5
5rxn	0.1	0.1	12.9	11.1
1arb	26	10.5	402.7	638.2[c]
2hbg	1.0	3.0	120.4	83.3
1bpi	0.0[b]	0.2	12.4	10.6
1igd	0.2	0.3	17.7	14.5
1cex	2.4	6.9	172.9	107.1
1xnb	2.7	3.2	148.9	107.5
1plc	0.2	0.6	48.9	37.3
1ptx	0.0[b]	1.6	12.1	10.2
2erl	0.0[b]	0.1	5.0	4.2
2end	13.8	6.2	118.8	74.3
1amm	1.9	9.6	212.7	158.1
1whi	1.0	3.3	81.3	55.6
9rnt	0.1	0.8	50.6	39.6
2ihl	0.1	1.1	74.7	54.0
1ctj	0.1	0.3	33.2	28.8
Average	2.7	2.5	83.3	77.1

[a] The MC quench step requires insignificant additional time.

[b] Less than 0.05 minute was recorded.

[c] The number of cycles was increased to 15 (see the text).

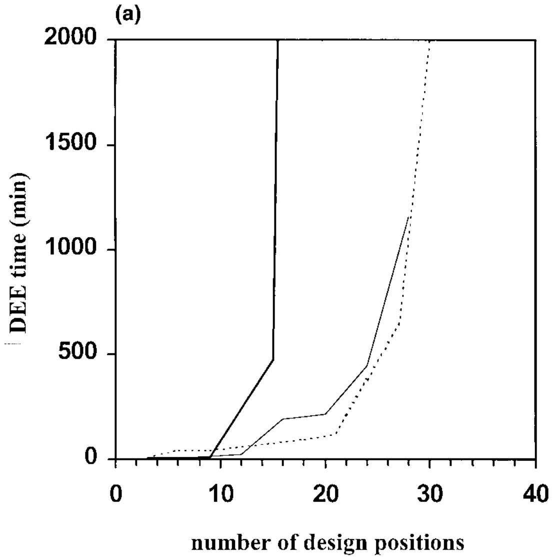

Figure 3. The time for DEE to converge on the 2hbg test case plotted against the number of design positions for the core (thick, continuous), boundary (thin, continuous) and surface (dotted) regions. This graph is representative of the time explosion behavior of the remaining four sequence design test cases. Note that the *y*-axis is truncated at 2000; the time for the hardest core calculation continues to 9999 minutes (Table 4).

are the solutions given by the less accurate algorithms? Because DEE provides the GMEC, the accuracy of SCMF and MCQ can be compared as the design problem increases in complexity. By extrapolating this result into the region where DEE explodes, the accuracy of the other algorithms can be reasonably approximated.

To determine the relationship between these questions and the specifics of the design problem, we ran tests on five structurally different proteins. Cytochrome b_{562} (256b) and hemoglobin (2hbg) are primarily α-helical proteins, amicyanin (1aac) and plastocyanin (1plc) are primarily β-sheet proteins, and ribonuclease (9rnt) is a protein that contains both α-helical and β-sheet structures. To ensure that we studied proteins where the success of the algorithms varied for the side-chain placement calculations, we included 2hbg in the design test set. This represents one of the few proteins where SCMF performed better than MCQ for the side-chain placement calculations.

The residues of each protein are labeled core, boundary or surface, based on solvent-accessibility (Dahiyat & Mayo, 1997a). From the perspective of protein design, this partition is motivated by the need for a hydrophobic core and hydrophilic surface for stable folding. For the design calculations, the hydrophobic amino acids (A, V, L, I, F, Y, W) are considered in the core, the hydrophilic amino acids (A, S, T, D, N, E, Q, H, K, R) at the surface and the combination of both sets in the boundary. The remaining four amino acids (G, C, P, M) are omitted from these calculations. The protein core has been the target of most design efforts, as this region tends to be an easier calculation due to the primary dependence on the sterics of side-chain packing (Lee & Levitt, 1991; Desjarlais & Handel, 1995; Dahiyat & Mayo, 1996, 1997b; Lazar *et al.*, 1997; Su & Mayo, 1997). More recently, computational protein design has expanded successfully into the boundary and surface regions (Malakaukas & Mayo, 1998; Dahiyat *et al.*, 1997b) and to complete sequence design (Dahiyat & Mayo 1997a; Morgan, 1999).

Protein length is not a good indicator of problem difficulty as the number of rotamers allowed at each residue, the specific conformation of the backbone, and the particular choice of the force-field can make a problem difficult. We have found that increasing the number of design positions qualitatively makes the search problem more difficult. For each test protein, we complete three series of runs where design positions are added in sequence order from the core, boundary or surface. To address the concern that the results are dependent on the order in which the design residues are added, we ran a separate series of runs for the core of 1aac. In these runs, we added design residues in structural clusters rather than sequence order. We find that the accuracy of SCMF and MCQ as a function of the number of design positions does not change qualitatively with the order of addition. This result can be generalized to the boundary and surface as, in these regions, residues are separated by greater distances and coupling is less likely to affect the results.

Table 3. DEE explosion behavior for protein sequence designs

	Number of design positions before explosion is observed[a]		
	Core	Boundary	Surface
1aac	24	25	34
9rnt	36[b]	30[b]	38
256b	17	24	8
2hbg	18	28	27
1plc	18	24	25[b]

[a] Defined as the number of design residues at which a time was observed greater than 500 minutes on a single processor.
[b] No time explosion was observed. The largest number of attempted design positions is reported.

DEE tends to converge on problems containing more design residues for the surface and boundary than the core, with the exception of the surface of 256b, which diverges after eight residues are added (Table 3). This result is somewhat dependent on the order in which the design residues are added. There is usually a specific combination of positions that cause DEE to fail. When these residues are designed, the time explosion is observed. The apparent ability to design more positions on the surface than in the core is due to the presence of a higher fraction of deleterious combinations in the core region. This is an expected result, as the core contains more coupled interactions.

For protein design, the relevant measure of accuracy is not the average fraction of incorrect rotamers as for the side-chain placement comparisons. Because the sequence is designed as well as the specific rotamer conformation, it is more interesting to know the fraction of amino acids that are predicted incorrectly, a, as compared to the GMEC. The results for an intermediate and hard design problems are shown for the core (Table 4), boundary (Table 5), and surface (Table 6). The hard design problem represents the case where the time required by DEE diverges. Tables 5, 6, and 7 are representative of only two test runs; the following averages are calculated from the complete trajectories (as in Figure 4) over the entire sequence design test set. For the core, $a = 0.07$ ($\langle \Delta E \rangle = 14.3$ kcal/mol) for SCMF and $a = 0.04$ ($\langle \Delta E \rangle = 1.1$ kcal/mol) for MCQ. For the boundary, $a = 0.28$ ($\langle \Delta E \rangle = 7.1$ kcal/mol) for SCMF and 0.32 ($\langle \Delta E \rangle = 4.6$ kcal/mol) for MCQ. The algorithms performed the worst on surface calculations with $a = 0.37$ ($\langle \Delta E \rangle = 15.1$ kcal/mol) for SCMF and $a = 0.44$ ($\langle \Delta E \rangle = 8.7$ kcal/mol) for MCQ. Similar to the side-chain placement calculations, SCMF obtains a solution that is more accurate in amino acid sequence, but higher in energy than MCQ. It is unclear which is the better answer in practice, as a single bad amino acid substitution can severely disrupt structural integrity, whereas combinations of mutations, which together contribute an energy comparable to the GMEC, may be more acceptable. There is no observed dependence on the secondary structure of the protein, as the results for both the α-helix and β-sheet dominated backbones are qualitatively similar.

We observe that the accuracy of MCQ and SCMF drops rapidly in boundary and surface calculations. This is related to the increase in the number of rotamers allowed at each position. MCQ fails because the size of the search space rapidly increases while the number of cycles remained fixed, thereby allowing less space to be sampled. One explanation for the failure of SCMF is through the compounding of two mechanisms. First, the mean-field description of the energy landscape is approximate, leading to error. Second, SCMF must converge the probability of a single rotamer existing in the GMEC to close to unity. As the number of rotamers is increased at each position, the probability that SCMF cannot converge on a single rotamer also increases, leading to incorrect assignments. It is the second of these effects that causes the loss of accuracy in the surface and boundary regions due to the increase in the number of rotamers allowed at each position.

Through these results, we show that the underlying premise of SCMF is erroneous. The global minimum of the annealed mean-field landscape fails to correspond to the true global minimum. However, to solve problems in the regime where conservative DEE fails, it could be argued that this is a necessary trade-off. While our results demonstrate that mean-field theory does not accurately find the GMEC, this should not be taken as a blanket disqualification of its utility. For example, the calculation of rotomer probabilities is useful in determining the entropy (and free energy) of the sequence (Koehl & Delarue, 1996). However, we have shown it does not accurately find the GMEC of the system. Accurately finding the GMEC is an

Table 4. Core results for sequence design calculations

		No. amino acids incorrect[a]			Time (minutes)		
	Design[b]	DEE	MCQ	SCMF	DEE	MCQ	SCMF
1aac	8	0	0	0	5	57	3
	24	0	0	4	5382	73	13
9rnt	12	0	0	0	2	55	3
	36	0	1	3	71	76	71
256b	10	0	0	1	232	59	4
	17	0	2	4	7271	63	7
2hbg	9	0	2	2	6	120	4
	18	0	2	2	9999	109	7
1plc	9	0	2	1	15	50	3
	18	0	3	3	1704	60	4
Average[c]					52	68	3
					4885	76	20

[a] The solution as compared to the GMEC determined by DEE

[b] The number of sequence design positions. A representative example is shown for a medium and hard calculation. The hard calculation corresponds to the point at which a time explosion was observed for DEE.

[c] The averages are taken for the medium and hard design problems.

Table 5. Boundary results for sequence design calculations

		No. amino acids incorrect[a]			Time (minutes)		
	Design[b]	DEE	MCQ	SCMF	DEE	MCQ	SCMF
1aac	15	0	5	6	68	62	13
	25	0	7	12	917	79	40
9rnt	5	0	0	0	1	53	3
	30	0	11	9	157	84	71
256b	8	0	1	4	51	71	14
	24	0	11	11	1855	90	74
2hbg	12	0	3	0	25	130	21
	28	0	9	12	1153	168	87
1plc	9	0	3	1	12	50	6
	24	0	8	8	599	64	20
Average[c]					31	73	11
					936	97	58

[a] The solution as compared to the GMEC determined by DEE

[b] The number of sequence design positions. A representative example is shown for a medium and hard calculation. The hard calculation corresponds to the point at which a time explosion was observed for DEE.

[c] The averages are taken for the medium and hard design problems.

essential step in the protein design algorithm. Because approximate computational results may be artifactual, it is possible to draw erroneous conclusions about the quality of the design strategy. This is particularly a problem when the combinatorial complexity is high and there is a high density of low-energy configurations. In such a case, it is possible to be close to the true global minimum in energy and have a completely different amino acid sequence.

In this study, we use extremely conservative DEE parameters (1000 kcal/mol threshold for automatic determination of dead-ending rotamers and pairs of rotamers). This was done to ensure that the solution obtained is the GMEC. Most practical design calculations are performed using more aggressive parameters (20 kcal/mol threshold for automatic determination of dead-ending rotamers and 1000 kcal/mol threshold for pairs of rotamers) or highly aggressive parameters (−20 kcal/mol threshold for automatic determination of dead-ending rotamers and −20 kcal/mol threshold for pairs of rotamers). A negative value indicates that the threshold is taken from the minimum rotamer energy at each residue position rather than zero (De Maeyer *et al*., 1997). To test the accuracy of DEE with the moderately and highly aggressive parameters, calculations were performed on the most difficult design problems. In the best case, DEE converged to the same solution up to 15 times more quickly. However, the effect on convergence time was generally unpredictable.

As another option, the quality of solution produced by MCQ can theoretically be improved by running for longer periods. In our hands, MCQ is typically run for 1000 cycles with each cycle consisting of 10^6 moves, requiring 3000 to 10,000 minutes. We ran MCQ on three difficult design problems: 30 surface design positions of 2hbg, 24 boundary positions of 1plc, and 17 core positions of 256b. For each case, the number of incorrect amino acids for 20 cycles *versus* 1000 cycles is

Table 6. Surface results for sequence design calculations

		No. amino acids incorrect[a]			Time (minutes)		
	Design[b]	DEE	MCQ	SCMF	DEE	MCQ	SCMF
1aac	15	0	8	5	3	61	18
	34	0	14	14	918	75	58
9rnt	15	0	8	9	81	58	8
	38	0	21	18	870	79	63
256b	3	0	2	2	18	65	4
	8	0	5	5	2329	66	10
2hbg	9	0	2	4	47	108	8
	30	0	15	15	2006	141	41
1plc	10	0	2	2	13	53	4
	25	0	10	9	151	63	19
Average[c]					162	69	8
					1255	85	38

[a] The solution as compared to the GMEC determined by DEE

[b] The number of sequence design positions. A representative example is shown for a medium and hard calculation. The hard calculation corresponds to the point at which a time explosion was observed for DEE.

[c] The averages are taken for the medium and hard design problems.

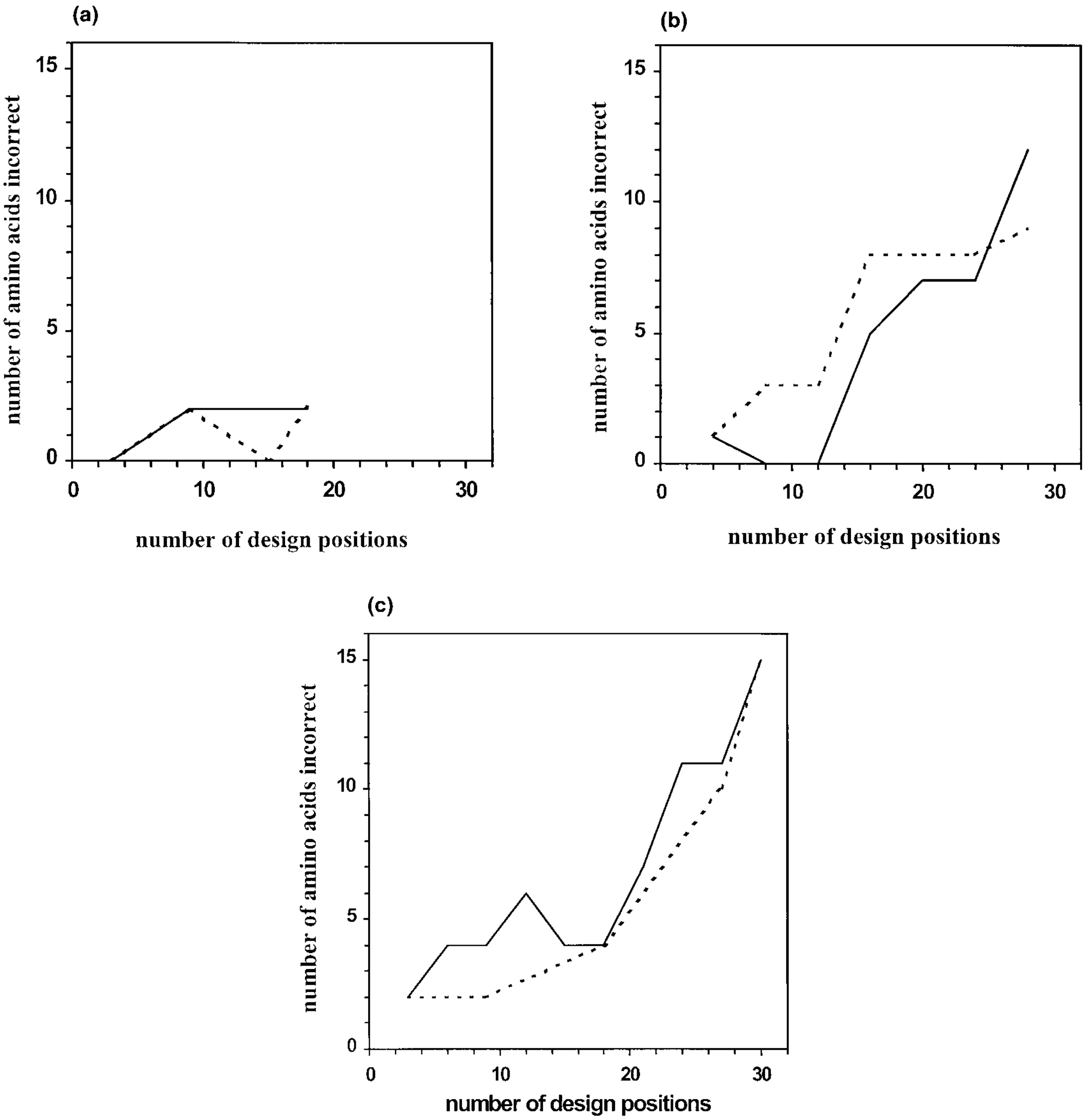

Figure 4. The results of the sequence design test in the (a) core, (b) boundary and (c) surface regions for 2hbg. The number of amino acids incorrect with respect to the GMEC found by DEE is plotted against the number of design positions in the protein for the SCMF (continuous line) and MCQ (dotted line) calculations. The end of each line either corresponds with the DEE explosion as shown in Table 3, or the point at which all of the regional positions have been designed. Note that 2hbg is one of the two cases out of 20 where SCMF outperformed MCQ for the side-chain placement tests.

15/10, 8/2, and 2/0. The additional time clearly improved the results for MCQ. It has been suggested that an improvement in SCMF can be achieved by initializing the rotamer probability vector randomly and running the convergence algorithm for each random initialization (Mendes *et al.*, 1999). This has the effect of extending the run time of SCMF. We implemented this algorithm and found that it never improved the solution. Most likely, the improvement that was observed by Mendes *et al.* was due to their lack of use of temperature annealing. Increasing the run length of SCMF was not found to improve the solution and the aggressiveness had been previously optimized through the energy threshold. Both aggressive DEE and MCQ comprehensively produce better results on the more difficult design problems.

Currently, the SCMF and DEE algorithms can be applied only to pairwise energy functions (equation (1)). Higher-order terms may be important in determining the stabilization energy of a sequence. In particular, buried surface area is a higher-order contribution to energy. In their present form, the stochastic methods can easily incorporate higher-order energy terms. If the incorporation of higher than two-body terms is desired, a new trade-off is created. The reduction of the force-field to the pairwise form that is required for the deterministic methods must be weighed with the inaccuracy of the stochastic search methods.

Of the four search algorithms we studied, there are extensive variations in the literature. Our laboratory uses the MC, MCQ, SCMF, and DEE algorithms, and the specific formulations presented here represent the methods that we have found to be the most successful. The exception is the genetic algorithm, which was programmed solely for this study. We tried many versions and found that the algorithm used here is the most reliable over the entire test set.

Here, we study each algorithm as a stand-alone search technique. An alternative is to create hybrid algorithms that combine elements from different techniques. For instance, the addition of a Monte Carlo quench step to the GA and SCMF algorithms improves the solution. In the case of the GA, the quench step does not improve the algorithm beyond what is attainable with MCQ. In addition, the combination of DEE and MC can potentially improve the search. Finally, a new Branch-and-Terminate technique has been proposed that extends the capabilities of DEE (Gorden & Mayo, 1999).

Conclusions

We have shown that DEE is the most appropriate search algorithm for side-chain placement, as it consistently and rapidly converges to the GMEC for a full range of structure sizes and types. However, for the design calculations, there is a point beyond which DEE fails to converge and to obtain a solution it becomes necessary to utilize a less accurate method. We find that the accuracy and speed of SCMF and MCQ are comparable for the design calculations. Both methods give reasonable solutions in the core, but fail considerably in the boundary and surface regions. The advantage of MCQ relative to SCMF is that, because it is a stochastic method, it can be run for longer periods so better solutions might be obtained. In contrast, the answer provided by SCMF is the only solution that it will provide and therefore does not take advantage of the increasing capability of computer hardware and software. Experimentally, the utility of an imprecise answer is unclear. Because the energy landscape constructed by the force-field is not the actual landscape, an answer that is close to the theoretical GMEC may be adequate to provide a folded, stable structure. Nevertheless, it is clearly important to understand that a solution provided by SCMF or MCQ could be off by more than 20 kcal/mol in computed energy and provide sequences that have 40 % disparate amino acids from the optimum for the given force-field.

Materials and Methods

We use a test set of 20 protein structures (Table 7) from the RCSB Protein Databank (Bernstein *et al*., 1977) that was compiled previously by Carrando and coworkers (Mendes *et al*., 1999). This set was chosen due to the

Table 7. Overview of the 20 protein test set

	No. residues[a]				Secondary structure			Solvent-accessibility[c]		
PDB	Total	Effect	Model	$\log_{10}$ conf[b]	Turn	β	α	Core	Bound	Surf
1aac	105	104	85	86	46	58	0	30	31	43
1cbn	46	40	31	31	13	6	21			
256b	106	106	88	101	16	0	89	28	34	44
1isu	62	62	43	46	45	12	9			
5rxn	54	54	48	48	41	13	0			
1arb	263	250	191	193	119	98	33			
2hbg	147	147	98	110	24	0	123	51	45	51
1bpi	58	52	40	49	25	14	13			
1igd	61	61	51	57	14	30	17			
1cex	213	186	166	158	55	33	98			
1xnb	185	176	144	157	41	125	10			
1plc	99	99	83	86	55	44	0	27	27	45
1ptx	64	56	51	50	31	14	11			
2erl	40	32	24	31	8	0	24			
2end	137	136	118	138	62	4	70			
1amm	174	173	157	174	78	81	14			
1whi	122	120	99	111	49	57	14			
9rnt	104	103	86	87	49	35	19	36	29	38
2ihl	129	119	100	102	42	20	57			
1ctj	89	87	63	65	46	0	51			

[a] The effective and modeled residues are as described in Materials and Methods.

[b] The number of configurations is the total number of points in rotamer space for the homology calculations. The number of configurations increases dramatically for the design calculations (data not shown).

[c] The designation of a residue as core, boundary or surface was done in the following manner. A solvent-accessible surface was generated using the Connolly algorithm with a probe radius of 8.0 Å, a dot density of 10 $Å^{-2}$, and a C^{α} radius of 1.95 Å. A residue was classified as a core position if the distance from its C^{α}, along its C^{α}-C^{β} vector, to the solvent-accessible surface was greater than 5.0 Å, and if the distance from its C^{β} to the nearest surface point was greater than 2.0 Å. The remaining residues were classified as surface positions if the sum of the distances from their C^{α}, along their C^{α}-C^{β} vector, to the solvent-accessible surface plus the distance from their C^{β} to the closest surface point was less than 2.7 Å. All remaining residues were classified as boundary. This classification was necessary only for the test cases chosen for sequence design calculations.

high resolution of the structures and the inclusion of a wide distribution of sizes and structure types (Table 7). For the side-chain placement calculations, there are positions in the fixed backbone where all allowed rotamers cause steric clashes that lead to unrealistically high energies. These positions were not considered in these calculations. The effective number of residues shown in Table 7 is the total number of residues minus cysteine residues involved in disulfide bonds and residues that clash with the backbone. Both alanine and glycine are described by a single rotamer and are therefore not taken into account when comparing the accuracy of the search algorithms for the side-chain placement calculations. The modeled number of residues is determined by the effective number of residues minus wild-type alanine and glycine positions.

We use the DREIDING force-field parameters for the atomic radii and internal coordinate parameters (Mayo *et al.*, 1990). The van der Waals energies are modeled using a 6-12 Leonard-Jones potential with an additional 0.9 scale factor applied to the atomic radii to soften the lack of flexibility implied by the fixed backbone and the rotamer descriptions (Dahiyat & Mayo, 1997b). Solvation terms were not included, in order to accelerate energy matrix calculations. The rotamer library is backbone-dependent, as described by Dunbrack & Karplus (1993). The following modifications were included as described (Dahiyat *et al.*, 1997a). χ_1 and χ_2 angle values of rotamers for all aromatic amino acids, and χ_1 angle values for all other hydrophobic amino acids were expanded ±1 standard deviation about the mean value reported in the Dunbrack and Karplus library. The χ_3 angles that were undetermined from the database statistics were assigned the following values: Arg, -60°, 60°, and 180°; Gln, -120°, -60°, 0°, 60°, 120°, and 180°; Glu, 0°, 60°, and 120°; Lys, -60°, 60°, and 180°. The χ_4 angles that were undetermined from the database statistics were assigned the following values: Arg, -120°, -60°, 60°, 120°, and 180°; Lys, -60°, 60°, and 180°. Rotamers with combinations of χ_3 and χ_4 angles resulting in sequential g^+/g^- or g^-/g^+ angles were eliminated. Uncharged His rotamers were used.

The calculations were performed on an SGI Origin 2000 supercomputer with 32 R10000 processors running at 195 MHz. While the codes for both DEE and SCMF are written to utilize parallel capabilities, the times presented for all algorithms are based on a single processor. The complete energy matrices (all pairwise interactions, $E(i_r,j_s)$ in equation (1)) were computed prior to the optimization procedure. The time required for this step is independent of the search algorithm and was approximately 60 minutes on a single processor for each test case.

Acknowledgments

The work was partially supported by the Howard Hughes Medical Institute, NIH training grant GM 08346, the Helen G. and Arthur McCallum Foundation (DBG), and a National Science Foundation Graduate Fellowship (CAV).

References

Bernstein, F. C., Koetzle, T. F., Williams, G. J. B., Meyer, E. F., Brice, M. D., Rodgers, J. R., Kennard, O., Shimanouchi, T. & Tasumi, M. (1977). The Protein Data Bank: a computer-based archival file for macromolecular structures. *J. Mol. Biol.* **112**, 535-542.

Cordes, M. H. J., Davidson, A. R. & Sauer, R. T. (1996). Sequence space, folding and protein design. *Curr. Opin. Struct. Biol.* **6**, 3-10.

Dahiyat, B. I. & Mayo, S. L. (1996). Protein design automation. *Protein Sci.* **5**, 895-903.

Dahiyat, B. I. & Mayo, S. L. (1997a). *De novo* protein design: fully automated sequence selection. *Science,* **278**, 82-87.

Dahiyat, B. I. & Mayo, S. L. (1997b). Probing the role of packing specificity in protein design. *Proc. Natl Acad. Sci. USA,* **94**, 10172-10177.

Dahiyat, B. I., Gordon, D. B. & Mayo, S. L. (1997a). Automated design of the surface positions of protein helices. *Protein Sci.* **6**, 1333-1337.

Dahiyat, B. I., Sarisky, C. A. & Mayo, S. L. (1997b). *De novo* protein design: towards fully automated sequence selection. *J. Mol. Biol.* **273**, 789-796.

De Maeyer, M., Desmet, J. & Lasters, I. (1997). All in one: a highly detailed rotamer library improves both accuracy and speed in the modelling of side-chains by dead-end elimination. *Fold. Des.* **2**, 53-66.

Desjarlais, J. R. & Clarke, N. D. (1998). Computer search algorithms in protein modification and design. *Curr. Opin. Struct. Biol.* **8**, 471-475.

Desjarlais, J. R. & Handel, T. M. (1995). *De novo* design of the hydrophobic cores of proteins. *Protein Sci.* **4**, 2006-2018.

Desmet, J., De Maeyer, M., Hazes, B. & Lasters, I. (1992). The dead-end elimination theorem and its use in protein side-chain positioning. *Nature,* **356**, 539-542.

Desmet, J., De Maeyer, M. & Lasters, I. (1994). The dead-end elimination theorem: a new approach to the side-chain packing problem. In *The Protein Folding Problem and Tertiary Structure Prediction* (Merz, K., Jr, K. M. & LeGrand, S., eds), pp. 307-337, Birkhauser, Boston.

Dunbrack, R. L. & Karplus, M. (1993). Backbone-dependent rotomer library for proteins - application to side-chain prediction. *J. Mol. Biol.* **230**, 543-574.

Dunbrack, R. L. & Karplus, M. (1994). Conformational analysis of the backbone-dependent rotamer preferences of protein side-chains. *Nature Struct. Biol.* **1**, 335-340.

Godzik, A. (1995). In search of the ideal protein sequence. *Protein Eng.* **8**, 409-416.

Goldstein, R. F. (1994). Efficient rotamer elimination applied to protein side-chains and related spin glasses. *Biophys. J.* **66**, 1335-1340.

Gordon, D. B. & Mayo, S. L. (1998). Radical performance enhancements for combinatorial optimization algorithms based on the dead-end elimination theorem. *J. Comput. Chem.* **19**, 1505-1514.

Gordon, D. B. & Mayo, S. L. (1999). Branch-and-Terminate: a combinatorial optimization algorithm for protein design. *Structure,* **7**, 1089-1098.

Gordon, D. B., Marshall, S. A. & Mayo, S. L. (1999). Energy functions for protein design. *Curr. Opin. Struct. Biol.* **9**, 509-513.

Harbury, P. H., Plecs, J. J., Tidor, B., Alber, T. & Kim, P. S. (1998). High-resolution protein design with backbone freedom. *Science,* **282**, 1462-1467.

Hellinga, H. W. & Richards, F. M. (1994). Optimal sequence selection in proteins of known structure by simulated evolution. *Proc. Natl Acad. Sci. USA,* **91**, 5803-5087.

Holland, J. H. (1993). *Adaptation in Natural and Artificial Systems*, The MIT Press, Boston.

Holm, L. & Sander, C. (1992). Fast and simple Monte Carlo algorithm for side-chain optimization in proteins: application to model building by homology. *Proteins: Struct. Funct. Genet.* **14**, 213-223.

Jones, D. T. (1994). *De novo* protein design using pairwise potentials and a genetic algorithm. *Protein Sci.* **3**, 567-574.

Koehl, P. & Delarue, M. (1994). Application of a self-consistent mean field theory to predict protein side-chains conformation and estimate their conformational entropy. *J. Mol. Biol.* **239**, 249-275.

Koehl, P. & Delarue, M. (1995). A self consistent mean field approach to simultaneous gap closure and side-chain positioning in homology modeling. *Nature Struct. Biol.* **2**, 163-170.

Koehl, P. & Delarue, M. (1996). Mean-field minimization methods for biological macromolecules. *Curr. Opin. Struct. Biol.* **6**, 222-226.

Laughton, C. A. (1994). Prediction of protein side-chain conformations from local three-dimensional homology relationships. *J. Mol. Biol.* **235**, 1088-1097.

Lazar, G. A., Desjarlais, J. R. & Handel, T. M. (1997). *De novo* design of the hydrophobic core of ubiquitin. *Protein Sci.* **6**, 1167-1178.

Lee, C. (1994). Predicting protein mutant energetics by self-consistent ensemble optimization. *J. Mol. Biol.* **236**, 918-939.

Lee, C. (1996). Testing homology modeling on mutant proteins: predicting structural and thermodynamic effects in the Ala98 → Val mutants of T4 lysozyme. *Fold. Des.* **1**, 1-12.

Lee, C. & Levitt, M. (1991). Accurate prediction of the stability and activity effects of site-directed mutagenesis on a protein core. *Nature*, **352**, 448-451.

Lee, C. & Subbiah, S. (1991). Prediction of protein side-chain conformation by packing optimization. *J. Mol. Biol.* **217**, 373-388.

Malakaukas, S. M. & Mayo, S. L. (1998). Design, structure and stability of a hyperthermophilic protein variant. *Nature Struct. Biol.* **5**, 470-475.

Mayo, S. L., Olafson, B. D. & Goddard, W. A., III (1990). DREIDING: a generic force-field for molecular simulations. *J. Phys. Chem.* **94**, 8897-8909.

Mendes, J., Soares, C. M. & Carrondo, M. A. (1999). Improvement of side-chain modeling in proteins with the self-consistent mean field theory method based on an analysis of the factors influencing prediction. *Biopolymers*, **50**, 111-131.

Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. N., Teller, A. H. & Teller, E. (1953). Equations of state calculations by fast computing machines. *J. Chem. Phys.* **21**, 1087-1092.

Morgan, C. S. (1999). Full sequence design of an alpha-helical protein and investigation of the importance of helix dipole and capping effects in helical protein design. PhD thesis, California Institute of Technology.

Pedersen, J. T. & Moult, J. (1996). Genetic algorithms for protein structure prediction. *Curr. Opin. Struct. Biol.* **6**, 227-231.

Pierie, N. A., Spriet, J. A., Desmet, J. & Mayo, S. L. (2000). Conformational splitting: a more powerful criterion for dead-end elimination. *J. Comput. Chem.* In the press.

Ponder, J. W. & Richards, F. M. (1987). Tertiary templates for proteins: use of packing criteria in the enumeration of allowed sequences for different structural classes. *J. Mol. Biol.* **193**, 775-791.

Sánchez, R. & Šali, A. (1997). Comparative protein structure modeling as an optimization problem. *J. Mol. Struct.* **398-399**, 489-496.

Sasai, M. (1995). Conformation, energy, and folding stability of selected amino acid sequences. *Proc. Natl Acad. Sci. USA*, **92**, 8438-8442.

Schiffer, C. A., Caldwell, J. W., Kollman, P. A. & Stroud, R. M. (1990). Prediction of homologous protein structures based on conformational searches and energetics. *Proteins: Struct. Funct. Genet.* **8**, 30-43.

Street, A. G. & Mayo, S. L. (1998). Pairwise calculation of protein solvent-accessible surface areas. *Fold. Des.* **3**, 253-258.

Street, A. G. & Mayo, S. L. (1999). Computational protein design. *Structure*, **7**, R105-R109.

Su, A. & Mayo, S. L. (1997). Coupling backbone flexibility and amino acid sequence selection in protein design. *Protein Sci.* **6**, 1701-1707.

Tuffery, P., Etchebest, C., Hazout, S. & Lavery, R. (1991). A new approach to the rapid determination of protein side-chain conformations. *J. Biomol. Struct. Dynam.* **8**, 1267-1289.

Vásquez, M. (1995). An evaluation of discrete and continuum search techniques for conformational analysis of side-chains in proteins. *Biopolymers*, **36**, 53-70.

Edited by J. Thornton

(Received 27 September 1999; received in revised form 27 January 2000; accepted 23 February 2000)