Exact Rotamer Optimization for Protein Design

D. BENJAMIN GORDON,[1] GEOFFREY K. HOM,[2] STEPHEN L. MAYO,[3] NILES A. PIERCE[4]
[1]*Whitehead Institute for Biomedical Research, Cambridge, Massachusetts 02142*
[2]*Biochemistry Option, California Institute of Technology, Pasadena, California 91125*
[3]*Howard Hughes Medical Institute and Division of Biology, California Institute of Technology, Pasadena, California 91125*
[4]*Department of Applied and Computational Mathematics, California Institute of Technology, Pasadena, California 91125*

Received 16 December 2001; Accepted 18 March 2002

Abstract: Computational methods play a central role in the rational design of novel proteins. The present work describes a new hybrid exact rotamer optimization (HERO) method that builds on previous dead-end elimination algorithms to yield dramatic performance enhancements. Measured on experimentally validated physical models, these improvements make it possible to perform previously intractable designs of entire protein core, surface, or boundary regions. Computational demonstrations include a full core design of the variable domains of the light and heavy chains of catalytic antibody 48G7 FAB with 74 residues and 10^{128} conformations, a full core/boundary design of the β1 domain of protein G with 25 residues and 10^{53} conformations, and a full surface design of the β1 domain of protein G with 27 residues and 10^{60} conformations. In addition, a full sequence design of the β1 domain of protein G is used to demonstrate the strong dependence of algorithm performance on the exact form of the potential function and the fidelity of the rotamer library. These results emphasize that search algorithm performance for protein design can only be meaningfully evaluated on physical models that have been subjected to experimental scrutiny. The new algorithm greatly facilitates ongoing efforts to engineer increasingly complex protein features.

 J Comput Chem 24: 232–243, 2003

Key words: dead-end elimination; side-chain placement; protein design; combinatorial optimization; *NP*-hard

Introduction

Advances in computational protein design have largely been paced by two factors: the development of biologically meaningful physical models for describing the design space, and the development of combinatorial optimization algorithms for searching this space over all allowed sequences and conformations. High-performance search algorithms make it possible to perform atomic-resolution side-chain placement calculations for the selection of novel amino acid sequences. The sequences can then be evaluated in the laboratory to validate and/or improve the physical model. Both the discrete rotamer libraries used to represent the possible side-chain conformations and the empirical potential function used to assess the quality of the possible design sequences are critical to the biological validity of the approach.

Several computational models have been experimentally validated for the design of protein cores[1–5] and for the design of boundary and surface residues with varying degrees of solvent exposure.[6–8] However, there are still few examples of experimentally validated computational designs for complete protein domains.[9] In the context of protein design, physical model validation[2] is a challenging endeavor in which experimental assays of designed molecules are used to parameterize or enhance an existing model. A physical model becomes useful if it is able to identify sequences that fulfill the design requirements from amidst the astronomically large number of possible sequences. To improve the prospects for performing more ambitious designs including protein function, it is necessary to increase the efficiency of the computer algorithms while maintaining or even improving the physical models that form the basis for sequence selection.

Correspondence to: N. A. Pierce; e-mail: niles@caltech.edu

Contract/grant sponsor: Helen G. and Arthur McCallum Foundation (D.B.G.)

Contract/grant sponsor: NRSA; contract/grant number: 5T32-GM07616 (G.K.H.)

Contract/grant sponsor: Howard Hughes Medical Institute (S.L.M.)

Contract/grant sponsor: Burroughs-Wellcome Foundation (N.A.P.)

Contract/grant sponsor: Caltech Initiative in Computational Molecular Biology (N.A.P.)

Significant effort has been expended in developing both exact and approximate search algorithms for protein design.[10] Protein design has recently been shown to be *NP*-hard,[11] meaning that it joins a class of challenging combinatorial optimization problems for which no exact polynomial-time algorithms are known. Approximate algorithms that have been applied to protein design include Monte Carlo methods,[12,13] genetic algorithms,[1] and self-consistent mean field approaches.[14,15] These methods are computationally inexpensive, but their accuracy in identifying the global minimum energy conformation (GMEC) is known to degrade as problem size increases.[16] To avoid corrupting the potential function with experimental feedback based on incomplete searches, it is highly desirable to rely on exact search algorithms if effective exponential-time algorithms are available. Exact tree-based algorithms have been successfully applied to protein design.[17,18] For large design problems, methods based on the dead-end elimination (DEE)[19−26] theorem have emerged as the most successful.

It is important to note that search algorithm performance is strongly affected by the physical model on which the optimization is based. For example, DEE has been found to perform best when the number of rotamers per position is small, as evidenced by the relative ease in performing side-chain placement calculations for homology modeling studies on large proteins.[23,26] For design calculations, the number of rotamers at each position increases dramatically because multiple amino acid identities are represented at each position. This change increases both the cost of each iteration and the difficulty in reducing the combinatorial size of the problem. These effects are only exacerbated as the fidelity of the rotamer library is improved or as the number of design positions is increased. The physical context of the design positions influences algorithm performance significantly; confined core residues are generally much faster to design than less-constrained residues on the protein surface. This is because side chains packed into protein cores experience more physical restrictions, facilitating the identification of rotamers that do not belong to the GMEC. We have also observed that the precise implementation of the energy expression can have dramatic effects on the search speed. As is demonstrated later, alterations of the potential function can make seemingly intractable optimization problems trivial.

Because the performance of search algorithms depends strongly on factors in addition to sheer combinatorial complexity, it is critical to evaluate new improvements using potential functions and rotamer libraries that are meaningful in the context of protein design. Thus, optimization benchmarks are best performed on potential functions and rotamer libraries that have been subjected to experimental scrutiny, or alternatively, benchmarks should be closely followed by experimental validation. At this time, there still remains a need to develop search algorithms that can perform large-scale optimizations on experimentally validated physical models.

The development of increasingly powerful dead-end elimination algorithms is specifically targeted at addressing this challenge. The basic idea of DEE is to eliminate rotamers from consideration that can be proven to be incompatible with the GMEC. In the context of side-chain placement for homology modeling, the original algorithm introduced criteria for eliminating individual rotamers and "flagging" dead-ending pairs of rotamers to facilitate the elimination of single rotamers during subsequent iterations.[19,20] The "unification" of rotamers at two or more positions into supperrotamers was subsequently introduced as an effective method for starting new cascades of eliminations.[21,22] Using Goldstein's more powerful elimination criteria,[22] DEE methods were extended to protein design applications.[2] Metrics were subsequently developed to mitigate the added expense of these criteria, particularly for the flagging of dead-ending rotamer pairs.[24] Recently, more sophisticated elimination criteria were introduced based on the concept of conformational splitting.[25] An adaptive implementation of split DEE has since been described that reduces the cost of each iteration in the case of multiple splitting positions.[26] A further approach termed "generalized" DEE has been introduced,[26] although in our hands it does not yield a performance enhancement over existing methods.

The present work reports new ideas that extend the application of DEE algorithms to larger design regimes for all structural contexts: core, boundary, and surface. We have previously obtained large speed enhancements from optimizing dead-ending pairs calculations.[24] These improvements are effective because the majority of the overall calculation time is spent attempting to flag dead-ending pairs. To further reduce this time, we have focused on exploring additional flagging methods. Two complementary approaches have resulted, each providing new and inexpensive ways to find dead-ending pairs. Taken together, and often independently of each other, these methods make previously intractable optimization problems solvable.

The first approach employs bounding criteria that were originally developed for use in tree-based optimization methods.[17,27] These bounding criteria are fundamentally different in nature from the dominance criteria that typify dead-end elimination. A bounding criterion eliminates a rotamer by comparing the lower energy bound of possible sequences containing that rotamer to the total energy of a known reference sequence. On the other hand, DEE criteria are examples of dominance relations[28] that attempt to show that one rotamer is preferred over another rotamer in all circumstances. As with the DEE dominance criteria, bounding criteria may be used both to eliminate individual rotamers and to flag pairs of rotamers. Moreover, because "bound flags" are obtained by measures other than dominance, they have the potential to augment the DEE reductions and enhance the performance of the algorithm. Bounding criteria require the energy of a reference sequence to which bounding energies may be compared. We therefore employ a stochastic Monte Carlo search to rapidly determine a valid reference energy. Interestingly, the algorithm remains exact, but it is no longer deterministic.

The second approach makes it possible to flag many dead-ending pairs at essentially no additional cost. These "split flags" are generated as a by-product of applying the conformational splitting criteria to eliminate single rotamers. By promoting further reduction in the combinatorial size of the problem prior to the application of expensive doubles criteria, these dead-ending pairs provide substantial computational savings.

The algorithm described in the present work combines three completely different search paradigms (dominance, bounding, and stochastic) into a single compatible approach. For ease of description, we term the new method hybrid exact rotamer optimization (HERO). Taken together, the two new strategies for flagging dead-ending pairs have dramatically increased the size of the

design problems that can be attempted on a daily basis in the laboratory of one of the authors (S. L. Mayo). Results in the present work will demonstrate that exact search algorithms based on experimentally validated physical models are now able to tackle design problems that could previously be attempted only with approximate methods. In particular, it is frequently possible to perform full protein core, boundary, or surface designs with surprising efficiency.

Theory

Energy Expression

Using a potential function described in terms of pairwise interactions, the total energy of the protein can be expressed as

$$E_{\text{total}} = E_{\text{template}} + \sum_i E(i_r) + \sum_i \sum_{j,j<i} E(i_r, j_u) \quad (1)$$

where E_{template} represents the self-energy of the backbone, $E(i_r)$ represents the energy of rotamer r at position i interacting with the backbone, and $E(i_r,j_u)$ represents the interaction energy between rotamers r and u at positions i and j, respectively. The objective of dead-end elimination criteria is to eliminate single rotamers that are dominated by other competing rotamers and to flag dead-ending pairs of rotamers that are dominated by other competing rotamer pairs. Either of the rotamers in a dead-ending pair could still belong to the GMEC conformation, but they cannot appear together; this strengthens the possibility of eliminating rotamers during subsequent iterations. For notational convenience, a flagged dead-ending pair is said to belong to the set F.

Goldstein DEE

The Goldstein DEE criterion for single rotamers states that rotamer i_r can be eliminated if there exists a competing rotamer i_t that satisfies

$$E(i_r) - E(i_t) + \sum_{j,j\neq i} \min_{\substack{u \\ (i_r, j_u)\notin F}} [E(i_r, j_u) - E(i_t, j_u)] > 0. \quad (2)$$

In other words, i_r can be eliminated if the contribution to the total energy is always reduced by using an alternative rotamer i_t. Note that the minimum specifically excludes contributions from flagged (i_r, j_u) pairs, as these rotamers cannot coexist in the GMEC. If there are p residue positions and an average of n rotamers per position, the computational complexity of attempting to eliminate each rotamer during a round of Goldstein DEE is $O(n^3p^2)$, corresponding to loops of cost n over r, t, and u as well as loops of cost p over i and j.

The doubles version of this criterion[22] flags a rotamer pair (i_r,k_s) if there exists a competing pair (i_v, k_w) that satisfies

$$[E(i_r) + E(k_s) + E(i_r, k_s)] - [E(i_v) + E(k_w) + E(i_v, k_w)] + \sum_{j,j\neq i\neq k} \min_{\substack{u \\ (i_r, j_u)\notin F \\ (k_s, j_u)\notin F}} \{[E(i_r, j_u) + E(k_s, j_u)] - [E(i_v, j_u) + E(k_w, j_u)]\} > 0 \quad (3)$$

For each of the rotamer pairs between two given positions, $O(n^2)$ comparisons are made with the other rotamer pairs at these positions. This criterion therefore makes $O(n^2)$ dominance checks in attempting to flag each rotamer pair. The computational complexity of Goldstein doubles is $O(n^5p^3)$, representing the most expensive component in most DEE implementations.

To obtain a subset of these flags at a lower cost, a "magic bullet" version of Goldstein doubles[24] was introduced that uses only one competing (i_v,k_w) pair to attempt to flag all other pairs of rotamers between positions i and k. The computational complexity is thus reduced to $O(n^3p^3)$, and only a single dominance check is made in attempting to flag each rotamer pair.

Split DEE

If no i_t rotamer dominates i_r for all possible conformations, then the Goldstein criterion will fail to make an elimination. Conceptually, however, i_r may still be eliminated if at least one (possibly varying) i_t rotamer dominates i_r for each conformation. Split DEE[25] embodies this idea by splitting the conformational space into partitions and checking to see if i_r is dominated by some i_t rotamer within each partition. In the simplest case (called "$s = 1$"), $O(n)$ partitions are created using the rotamers at a single splitting position. The rotamer i_r can then be eliminated if, for each splitting rotamer v at some splitting position k, there exists an i_t rotamer that dominates i_r within that partition:

$$E(i_r) - E(i_t) + \sum_{j,j\neq k\neq i} \{ \min_{\substack{u \\ (i_r, j_u)\notin F}} [E(i_r, j_u) - E(i_t, j_u)]\} + [E(i_r, k_v) - E(i_t, k_v)] > 0. \quad (4)$$

Domination in partition k_v is automatic if (i_r,k_v) is a flagged pair. The split DEE ($s = 1$) criterion is illustrated in Figure 1(a). The computational complexity of this approach remains $O(n^3p^2)$ despite the increase in elimination power.[25]

Increasing the number of splitting positions increases both the elimination power and the computational complexity. For two splitting positions ($s = 2$), there are $O(n^2)$ partitions, and i_r may be eliminated if, for each pair of splitting rotamers k_v and h_w at splitting positions $k \neq h \neq i$, there exists an i_t rotamer that dominates i_r in that partition:

$$E(i_r) - E(i_t) + \sum_{j,j\neq i\neq h\neq k} \{ \min_{\substack{u \\ (i_r, j_u)\notin F}} [E(i_r, j_u) - E(i_t, j_u)]\} + [E(i_r, k_v) - E(i_t, k_v)] + [E(i_r, h_w) - E(i_t, h_w)] > 0. \quad (5)$$

Here, domination follows automatically if either (i_r,k_v) or (i_r,h_w) is a flagged pair. The application of this criterion is illustrated in Figure 1(b), where the rotamers at the second splitting position

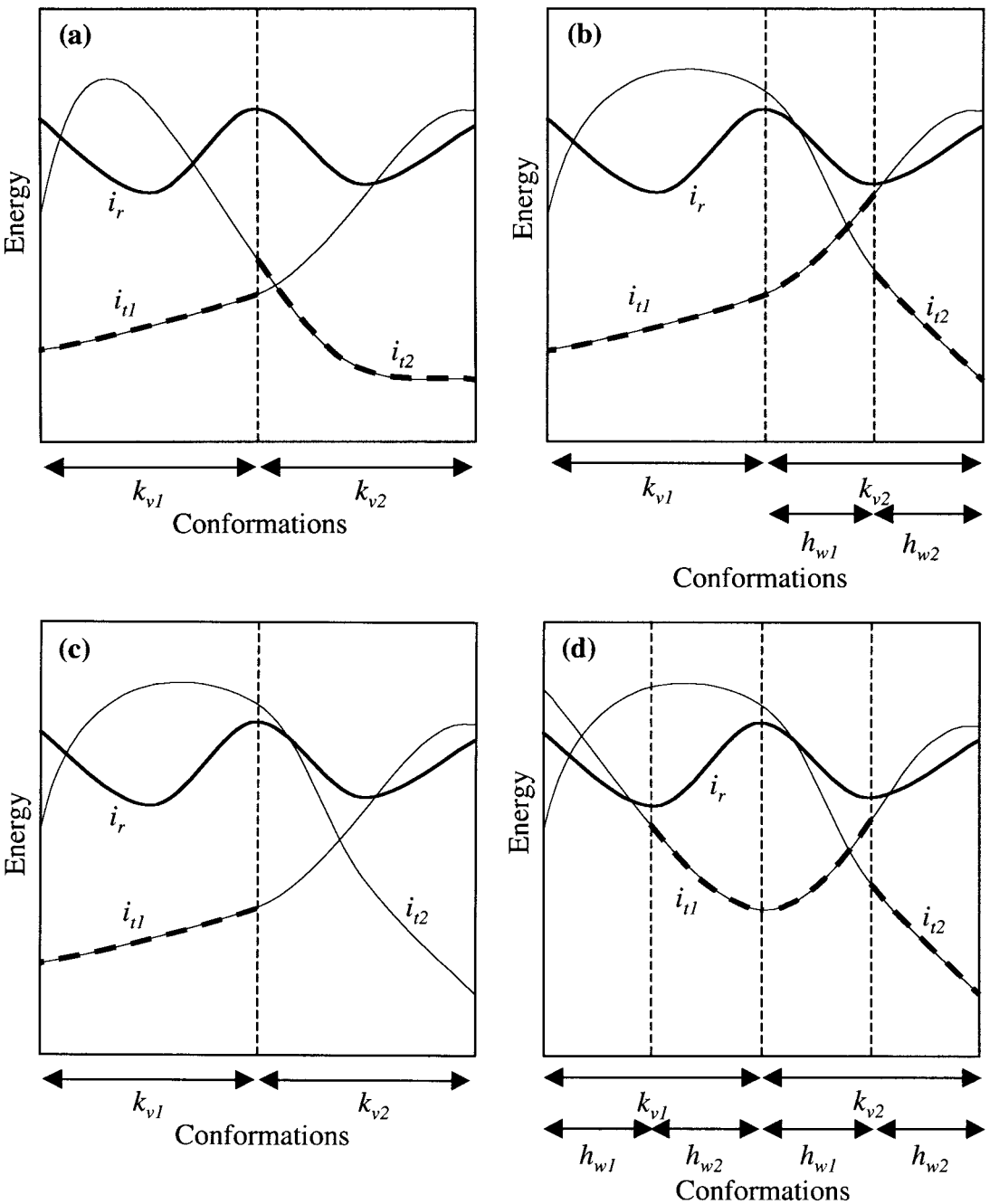

Figure 1. Application of split DEE to sample energy profiles. The abscissa represents all possible conformations of the protein and the ordinate represents the net energy contributions produced by interactions with specific rotamers at position i. (a) Split elimination ($s = 1$): i_r is dominated by i_{t1} and i_{t2} in the partitions corresponding to splitting rotamers k_{v1} and k_{v2}, respectively. Hence, i_r may be eliminated even though it is not dominated by any single rotamer for all of conformational space. (b) Split elimination ($s = 2$): because neither i_{t1} nor i_{t2} dominates i_r in partition k_{v2}, a second splitting position is used to create subpartitions h_{w1} and h_{w2}, where i_r is dominated by i_{t1} and i_{t2}, respectively. Hence, i_r may be eliminated using two splitting positions. (c) Split flagging ($s = 1$): i_r is not dominated in partition k_{v2} so elimination is not possible with only one splitting position. However, i_r is dominated by i_{t1} in partition k_{v1}, so that pair (i_r, k_{v1}) may be flagged. (d) Split flagging ($s = 2$): i_r is no longer dominated for all of conformational space so it cannot be eliminated with only two splitting positions. However, i_r is dominated for partition k_{v2} by i_{t1} and i_{t2} in subpartitions h_{w1} and h_{w2}, respectively. Hence, the pair (i_r, k_{v2}) may be flagged. Likewise, the pair (i_r, h_{w2}) may be flagged, as becomes more readily apparent if the hierarchy of the splitting positions k and h is reversed.

Table 1. Cost Comparison of Different Flagging Approaches.

Method	Iteration complexity	Flag attempts per rotamer pair
Full Goldstein doubles	$O(n^5p^3)$	$O(n^2)$
Magic bullet Goldstein doubles	$O(n^3p^3)$	1
Split flags ($s = 1$)	$O(n^3p^2)$	$O(n)$
Split flags ($s = 2$)	$O(n^4p^3)$	$O(n)$

(h_w) effectively create subpartitions of those created by the first splitting position (k_v). The computational complexity for ($s = 2$) split DEE is $O(n^4p^3)$.[25] Looger and Hellinga[26] present the same approach and provide the same complexity estimates in a later publication. With regard to implementation, Looger and Hellinga make the useful observation that conformational splitting may be coded adaptively so that subpartitions at a new splitting level are explored only within those existing partitions that have failed to achieve dominance of i_r at the current level. This decreases the computational cost of an iteration relative to the worst-case complexity estimates.

Expressions for split DEE criteria and cost bounds for arbitrary numbers of splitting positions have been reported previously.[25] In practice, we rarely find it beneficial to use splitting criteria beyond $s = 2$. Split DEE criteria may be extended to flag pairs of rotamers exactly as for the Goldstein doubles criterion (3), with a corresponding increase in computational overhead relative to the singles implementation. However, we now pursue the following more interesting observation that flags can be generated during split singles calculations with no increase in computational complexity.

Split Flags

Consider the scenario where i_r cannot be eliminated by split DEE ($s = 1$) because there are some partitions in which no i_t rotamer dominates i_r. It may still be possible to identify dead-ending pairs during the process of discovering this negative result. In those partitions k_v where i_r *is* dominated by some i_t, then the rotamer pairs (i_r, k_v) may be flagged as dead-ending. This concept is illustrated in Figure 1(c). The comparisons that are made in an effort to identify flags remain a subset of those made for a full Goldstein doubles calculation. $O(n)$ dominance comparisons are made in attempting to flag each rotamer pair. The complexity of split DEE is unaffected by this modification, remaining $O(n^3p^2)$ for ($s = 1$), so this approach compares very favorably with both full Goldstein doubles and magic bullet Goldstein doubles, as summarized in Table 1.

Split flags may be generated with arbitrary numbers of splitting positions. The concept is illustrated for split DEE ($s = 2$) in Figure 1(d). In this case, the number of flagging comparisons remains $O(n)$ per rotamer pair, but the iteration complexity increases to $O(n^4p^3)$. For a given maximum number of splitting positions, the attempt to eliminate a rotamer i_r has failed as soon as a subpartition at the lowest level is encountered in which i_r is not dominated by some competitor. It is possible to continue checking dominance for other partitions to attempt to identify more flags, but for ($s \geq 2$), the mounting cost motivates our decision to branch out of an elimination attempt as soon as failure is assured. It appears that Looger and Hellinga[26] allude to a special case of this approach corresponding to ($s = 1$) split flags.

Bounding Expressions

Bounding expressions provide an alternative means of determining whether a particular arrangement of rotamers at a subset of the residue positions can exist as part of the GMEC. Rather than eliminating rotamers by comparing them to other competing rotamers at the same positions, bounding expressions seek to produce a sharp lower bound on the total conformational energy given a certain subset of specified rotamers. If this bound is higher than the energy of some known complete reference sequence

$$E_{\text{bound}}(\text{subset}) > E_{\text{total}}(\text{reference}) \tag{6}$$

then the specified rotamers cannot coexist in the GMEC. The reference energy should be as low as possible and may be obtained by a computationally inexpensive approximate search of the same rotamer conformation space.

There are many possible ways of constructing an expression to compute the lower energy bound for an arrangement of rotamers. The expression that yields the best performance in the branch-and-terminate algorithm[17] folds the one-body terms into the two-body terms:

$$E'(i_r, j_u) \equiv \frac{E(i_r) + E(j_u)}{2(p-1)} + \frac{E(i_r, j_u)}{2} \tag{7}$$

and computes the lower bound on the total energy as

$$E_{\text{bound}} = \sum_{i \in C} \sum_{\substack{j \in C \\ j \neq i}} E'(i_r, j_u) + \sum_{i \in V} \min_r \left\{ 2 \sum_{j \in C} E'(i_r, j_u) + \sum_{j \in V} \min_u [E'(i_r, j_u)] \right\}. \tag{8}$$

The set of residue positions C is the subset of "constrained" positions that are occupied by the rotamers under scrutiny, and the set V encompasses all the remaining "variable" residue positions. The more positions that are constrained, the sharper the bound becomes.

To use the bounding expression efficiently in the context of dead-end elimination, the set C may be considered to consist of a single rotamer, so that the lower bound on the energy of all conformations containing rotamer i_r is

$$E_{\text{bound}}(i_r) = \sum_{m, m \neq i} \min_{\substack{t \\ (i_r, m_t) \notin F}} \left\{ 2E'(i_r, m_t) + \sum_{j, j \neq m \neq i} \min_{\substack{u \\ (j_u, m_t) \notin F}} [E'(j_u, m_t)] \right\} \tag{9}$$

Using the implementation described previously,[17] where the innermost summation is precomputed, the complexity of computing the energy bound for each single rotamer is $O(n^2p^3)$. Note that

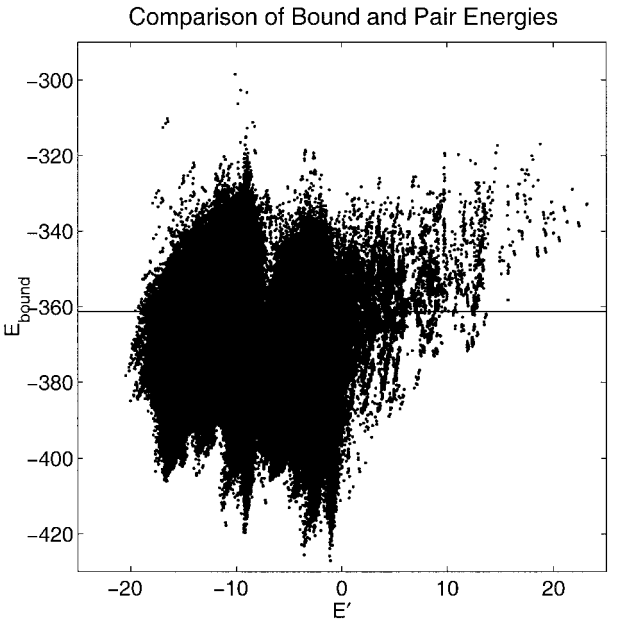

Figure 2. Comparison of bounding and pairs energies during a bound flags iteration of the plastocyanin core calculation of Case 1. The reference energy obtained by a Monte Carlo calculation is shown as a horizontal line. All pairs with a bounding energy above the line may be flagged. In this instance, 400,822 out of 966,656, or 41%, of the remaining unflagged pairs can now be flagged as dead-ending.

flagged dead-ending pairs can be excluded during the "min" operations.

Bounding Flags

Increasing the constrained set C to encompass a pair of rotamers produces the bounding expression

$$E_{\text{bound}}(i_r, k_s) = 2E'(i_r, k_s) + \sum_{m,m\neq i\neq k} \min_{\substack{t \\ (i_r,m_t)\notin F \\ (k_s,m_t)\notin F}} \left\{ 2E'(i_r, m_t) + 2E'(k_s, m_t) + \sum_{j,j\neq m\neq i\neq k} \min_{\substack{u \\ (j_u,m_t)\notin F}} [E'(j_u, m_t)] \right\} \tag{10}$$

The pair (i_r,k_s) can be flagged if $E_{\text{bound}}(i_r,k_s) > E_{\text{total}}(\text{reference})$ even if the pair is not dead-ending according to any known DEE criterion. The innermost summation is invariant with the rotamer indices r and s for a choice of positions i and k. By precomputing this term independent of r and s, the computational complexity of bounding the total energy for each rotamer pair is $O(n^2p^4 + n^3p^3)$. Again, it is possible to take advantage of previously flagged pairs in computing the energy bounds.

The potential benefit of using this bounding expression is illustrated in Figure 2, where E_{bound} is compared to E' for all the remaining unflagged rotamer pairs at one point during the convergence process for the core design of plastocyanin (described as Case 1 in Methods). The performance of the bounds improves as residues are unified together to create super-rotamers representing larger fractions of conformational space.

Monte Carlo Search

The efficacy of using bounding expressions to eliminate candidate rotamers and to flag rotamer pairs depends critically on the availability of a reference energy of a rotameric arrangement close in energy to the GMEC. This reference energy is obtained during the calculation using parallel Monte Carlo[29] searches from the current state of the conformational ensemble. The overall approach is therefore stochastic but exact, in the standard sense that if it converges, it converges to the GMEC.

Because Monte Carlo is repeated periodically as rotamers are eliminated, the searches are performed on a shrinking conformational space and the reference energy typically decreases as the calculation proceeds. By monitoring the top-ranked Monte Carlo sequences, it is possible to gain some insight into the convergence of the algorithm in sequence space prior to reaching full convergence. This can be particularly valuable for very large calculations that converge slowly or do not converge at all.

Unification

Dominance and bounding criteria can often benefit from residue unification, in which a "super-residue" is constructed from the rotamer pairs at two residue positions. The super-residue is treated as a single residue for the remainder of the calculation and may be unified with other residues at a later iteration. Because flagged pairs that are unified can be eliminated, unification is performed on the pair of residues that have the largest fraction of dead-ending rotamer pairs, provided that the resulting super-residue has fewer than some maximum number of super-rotamers [typically $(np)_{\max} = 10^4$].

Algorithm Schedule

The criteria described above may be coupled in many different ways. Our preferred strategy is to develop a standard schedule that performs well for a variety of design problems to minimize the need for user intervention. The entire iterative process is guaranteed to converge given sufficient time and computer memory. In practice, convergence is only possible if the elimination and flagging criteria prune the size of the combinatorial problem sufficiently rapidly to remain within the bounds of a human attention span and available computer memory.

Our preferred HERO implementation is described in Figure 3. The Goldstein singles criterion is applied iteratively until no further rotamers are eliminated. The split ($s = 1$) criterion is then applied iteratively until no further eliminations are found. Split flags are generated during this process with no increase in the computational complexity of the original split implementation. Split criteria are then applied with multiple splitting positions [to the desired partition depth ($s \geq 2$)] once for each rotamer. A magic bullet metric may be employed to select the splitting partitions that are deemed most likely to produce flags or an elimination.[25] Magic bullet Goldstein doubles is then applied once to each rotamer pair

1. Goldstein singles DEE until no further eliminations
2. Split singles DEE with split flags (s=1) until no further eliminations
3. Split singles DEE with split flags ($s \geq 2$) once for each rotamer (with or without magic bullet metric)
4. Singles bounding criterion once for each rotamer
5. Alternate sequentially between the following, applying one during each cycle:
   - Magic bullet Goldstein doubles once for each rotamer pair
   - Monte Carlo search to find $E_{\text{reference}}$ from a valid conformation followed by doubles bounding criterion once for each rotamer pair
   - Full Goldstein doubles once for each rotamer pair using q_{rs} and q_{uv} metrics
   - Unification of residues with the highest fraction of dead-ending pairs
6. Return to 1

Figure 3. The schedule of dominance and bounding criteria used for hybrid exact rotamer optimization (HERO).

to generate flags. The singles-elimination and split-flagging process is then repeated taking advantage of these new flags. On the second time through the cycle, a Monte Carlo search is performed to attempt to reduce the reference energy used to inform the bounding criteria. (Initially, the reference energy is set to be an arbitrarily large number.) The doubles bounding criterion is then applied once to each rotamer pair to identify more flags. After another round of singles eliminations and split flagging, a full round of Goldstein doubles flagging is performed using "q_{rs}" and "q_{uv}" metrics[24] to enhance performance. Following a fourth and final singles-elimination and split-flagging phase, unification is performed in lieu of a doubles calculation and the entire process is repeated.

For purposes of this study, we perform split DEE only up to two splitting positions ($s = 2$). For historical purposes, we include results using a previously published[25] magic bullet ranking metric (DEE $s2_{mb}$) that selects the two splitting positions that appear most likely to facilitate the elimination of rotamer i_r. The current baseline scheme for demonstrating the advancements of the present work is (DEE s2) without split flagging or bound flagging. To demonstrate the role that bound flagging and split flagging play in protein design calculations, these components are introduced separately to produce the schemes (DEE s2 bound flags) and (DEE s2 split flags). The complete hybrid exact rotamer optimization method described above is then termed HERO, which in longhand would be the less wieldy (DEE s2 bound & split flags).

Results and Discussion

Benchmark Design Calculations

The protein design benchmarks described in this work are performed using a potential function and rotamer libraries that have been subjected to extensive laboratory testing.[2,5–7,9,30–37] This is an important consideration when assessing the significance of computational demonstrations. In particular, it is trivial to dramatically improve apparent search algorithm performance either by reducing the size of the rotamer library or by modifying the potential function. Such modifications would require laboratory validation before the resulting increase in algorithm efficiency could be considered to have significance to the field of protein design.

The performance enhancements provided by bound and split flags in the context of an experimentally validated physical model are demonstrated by the five problems described in Table 2. These design cases arose during computational and experimental studies in the lab of one of the authors (S. L. Mayo). The conformational sizes in Table 2 are based on the rotamers that remain after high-energy threshold reduction (HETR)[23] is used to eliminate rotamers that clash with the backbone [for these tests, we removed rotamers with $E(i_r) > 20$ kcal/mol]. This practice reduces the risk of inflating the apparent conformational size of the problem by using a large number of rotamers that are incompatible with the protein fold.

Table 2. Benchmark Design Cases.

Case	Description	Type	Residues	Rotamers	Conformations
1	Plastocyanin	Core	25	1716	1.7×10^{38}
2	Novel backbone	Linked core	34	674	8.4×10^{39}
3	Catalytic antibody	Core	74	4919	4.7×10^{128}
4	β1 of protein G	Core/boundary	25	4295	4.0×10^{53}
5	β1 of protein G	Surface	27	4842	4.9×10^{60}

Table 3. CPU Times for Benchmark Design Cases Running on 16 Processors of an IBM SP3.

Case	Method	Time (min)	Remaining conformations
1	DEE s2mb	334	7×10^{14}
	DEE s2	150	2×10^{11}
	DEE s2 bound flags	22	1
	DEE s2 split flags	46	1
	HERO	13	1
2	DEE s2mb	250	1×10^{18}
	DEE s2	210	1×10^{18}
	DEE s2 bound flags	23	1
	DEE s2 split flags	167	3×10^{16}
	HERO	7	1
3	DEE s2mb	984	3×10^{8}
	DEE s2	687	1
	DEE s2 bound flags	663	1
	DEE s2 split flags	299	1
	HERO	359	1
4	DEE s2mb	1449	2×10^{35}
	DEE s2	1333	1×10^{35}
	DEE s2 bound flags	1688	1×10^{35}
	DEE s2 split flags	875	9×10^{19}
	HERO	476	1
5	DEE s2mb	292	3×10^{16}
	DEE s2	129	1
	DEE s2 bound flags	72	1
	DEE s2 split flags	46	1
	HERO	35	1

Case 1 represents a full core design of plastocyanin.[38] Case 2 is an unusual design problem involving all core positions on a novel repeating backbone based on the leucine-rich-repeat motif;[39] the residues in each of two repeats are restricted to have linked (but unspecified) amino acid identities. Case 3 represents the full core design of the variable domains of the light and heavy chains of catalytic antibody 48G7 FAB.[40] Case 4 is a full core and boundary design of the β1 domain of protein G,[41] and Case 5 is a full surface design of the same domain.

Timing results for the five benchmark design cases are described in Table 3 and displayed graphically in Figure 4. Failure to converge implies that the unification process cannot continue without exceeding the specified maximum number of rotamers [we use $(np)_{\max} = 10^4$ for Cases 1, 2, 4, and 5; we use $(np)_{\max} = 2 \times 10^4$ for the larger conformational space of Case 3]. For the plastocyanin core design of Case 1, the previously published method (DEE $s2_{mb}$) fails to converge, leaving over 10^{14} conformations after 334 min. The current baseline scheme (DEE s2) also fails to converge, requiring 150 min to narrow the search space to 10^{11} conformations. This improvement is due both to the additional eliminations produced by full ($s = 2$) split DEE (as compared to the magic bullet version) and to the time savings yielded by the adaptive implementation of this approach.[26] Introducing bound flags gives full convergence to the GMEC in 22 min, while split flags give full convergence in 46 min. The combined approach (HERO) reaches convergence in 13 min.

Case 2 is unusual because the number of rotamers is not large and yet the case is challenging. This is evidently a product of the linking of amino acid identities across the repeating subunits of the design. The algorithm converges only when using bound flags, requiring 23 min for (DEE s2 bound flags) and 7 min for HERO.

Case 3 is a large core design that converges with all schemes except the previously published method (DEE $s2_{mb}$), requiring 299 min for (DEE s2 split flags) and 359 min for HERO. Evidently, the bound flags do not play a substantial role for this problem and their calculation is effectively a computational overhead that accounts for the increase in time.

Case 4 is a full core/boundary design that fails to converge with any algorithm except HERO, which converges in 476 min. Case 5 is a full surface design of the same protein; it converges with all but (DEE $s2_{mb}$), with both bound and split flags yielding improvements, and HERO converging fastest in 35 min.

Performance of "Generalized" DEE

"Generalized DEE" was introduced[26] as another method for eliminating rotamers that cannot be eliminated by Goldstein DEE. The idea is to reoptimize a portion of the conformational background, taking advantage of flags between the reoptimized positions to increase the disparity in the net energy contributions of the i_r and i_t rotamers with these positions. The method is dominated by conformational splitting in the sense that for the same number of generalized positions g or splitting positions s, the eliminations obtained by generalized DEE are a subset of those obtained by split DEE. However, generalized DEE is amenable to less costly implementations than split DEE, so it is possible that performance enhancements might still be achieved. Unfortunately, in our hands, this has not been observed, as illustrated in Figure 5 for a subset of 14 surface positions from benchmark Case 5. This smaller case was chosen to allow all of the generalized variants to run to completion. Generalized DEE was performed starting from the baseline scheme (DEE s2) with the maximum number of reoptimized positions corresponding to (g = 2,3,4,5). For this example, the algorithm performance decreases monotonically with increasing g.

Physical Model Dependence

As is apparent from eqs. (1) and (2), the performance of any DEE algorithm will depend heavily on the nature of the physical model used to compute the one- and two-body terms [$E(i_r)$ and $E(i_r,j_u)$, respectively in eq. (1)]. Potential functions that emphasize energy terms that contribute to $E(i_r)$ relative to $E(i_r,j_u)$ will result in less coupling and easier optimization. In the limit of $|E(i_r)| >> |E(i_r,j_u)|$, the optimization reduces to the selection of the rotamer with the best one-body energy at each residue position. This observation emphasizes the importance of developing (and comparing) optimization schemes that are based on validated physical models—construction of inappropriate physical models can easily lead to impressive optimization performance.

A demonstration of the dependence of optimization performance on the underlying physical model is shown in Figure 6. This case is a full sequence design of the 56 positions in the β1 domain of protein G. Three of these positions are preset to glycine (posi-

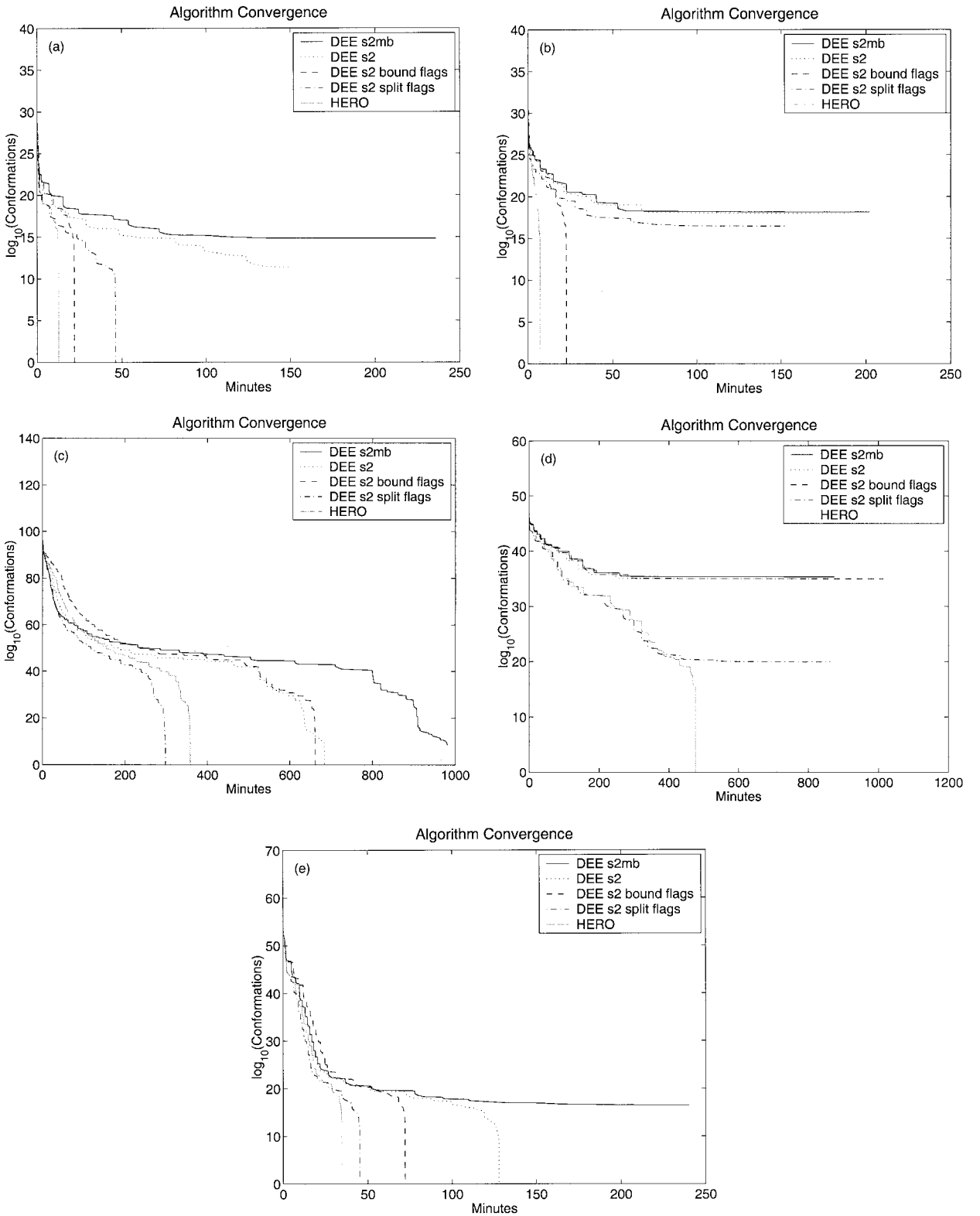

Figure 4. DEE convergence results. (a) Case 1: full core design of plastocyanin, (b) Case 2: full core design of a novel repeating backbone, (c) Case 3: full core design of the variable domains of the light and heavy chains of a catalytic antibody, (d) Case 4: full core and boundary design of the β1 domain of protein G, (e) Case 5: full surface design of the β1 domain of protein G.

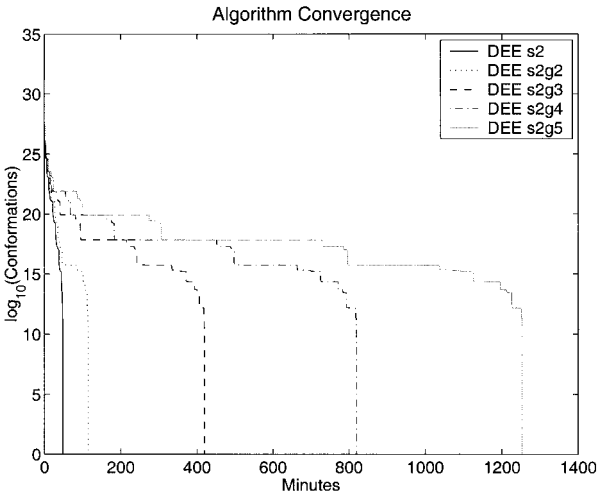

Figure 5. Performance assessment of "generalized DEE" for a partial surface design of the β1 domain of protein G. Comparisons are made relative to the baseline scheme (DEE s2) using reoptimizations at a maximum of (g = 2,3,4,5) positions.

tion 38 has a positive phi angle and functions as a C-cap for the alpha helix; positions 9 and 41 are sterically constrained core positions). The remaining positions are divided into core, boundary, and surface regions with the allowed amino acid identities at each of the 53 positions constrained to preserve the binary pattern of the wild-type sequence.[34] The resulting combinatorial complexity is 10^{112} conformations with 7775 initial rotamers after applying HETR[23] to eliminate rotamers that clash with the backbone. A HERO run with our "standard" potential function and rotamer library fails to converge after more than 1000 min. Optimization with a potential function modified to emphasize one-body terms reaches the GMEC in 20 min. The potential function modifications include (in order of decreasing importance): use of a one-body atomic solvation potential;[42] use of a Coulombic potential with a nondistance-dependent dielectric constant for rotamer/backbone interactions and a distance-dependent dielectric constant for rotamer/rotamer interactions;[43] use of rotamer internal strain energy; use of secondary structure propensities for helical and β-strand positions;[6] and use of normalized van der Waals energies to remove the bias for selection of large amino acids. The validity of these modifications remains to be determined.

In addition to the potential function component of the physical model, great care must be taken with respect to the rotamer library. Previous computational work using surprisingly small rotamer libraries (approximately 67 rotamers per residue position) showed large, full sequence design problems to be tractable.[26] For the full sequence design of protein G described above, the average number of rotamers per residue position is 147. Using an unexpanded rotamer library and aggressive HETR, the average number of rotamers per position can be reduced to 70 (10^{80} conformations for 3705 rotamers). Obtaining the GMEC for the resulting problem using the standard potential function requires 28 min.

These results strikingly illustrate the dependence of algorithm performance on both the potential function and the rotamer library. Clearly, search algorithm performance cannot be meaningfully ascertained on models of uncertain biological validity. On the other hand, the development of biologically valid, one-body-weighted physical models provides an opportunity to tame the combinatorial problem that is at the root of computational protein design.[11]

Approximate Alternatives

It is apparent from Figure 2 that bounding energies are a better indicator than self-energies of the likelihood that certain rotamers are not members of the GMEC. Based on this observation, we have observed that it is sometimes possible to find the GMEC in a few minutes using an approximate version of HERO in which bounding energies are used as a substitute for self-energies when applying HETR[23] to eliminate rotamers (data not shown).

Conclusion

Existing DEE algorithms spend most of their time attempting to flag dead-ending pairs of rotamers to facilitate future eliminations of dead-ending single rotamers. Two new methods have been formulated for efficiently identifying pairs of rotamers that are incompatible with the GMEC. One approach builds on split DEE methods to flag dead-ending pairs during the singles elimination process at essentially no additional expense. The other approach uses bounding criteria to flag pairs of rotamers for which a lower bound on the total conformational energy exceeds the energy of a reference conformation that has been identified by a computationally inexpensive Monte Carlo search. These bound flags would not necessarily be identified as dead-ending by any known DEE criterion. The new hybrid algorithm thus combines dominance crite-

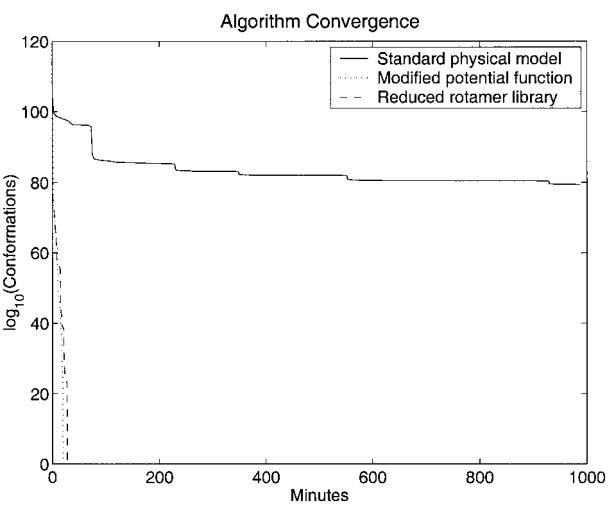

Figure 6. Convergence comparison for HERO on a full sequence design of the β1 domain of protein G using the experimentally validated "standard" potential function and rotamer library, a modified potential function with the standard rotamer library, and the standard potential function with a reduced rotamer library.

ria, bounding criteria, and a stochastic search into a single compatible framework that is exact but no longer deterministic.

The present benchmark calculations and our ongoing experience with these algorithms suggest that the most reliable performance is achieved using the HERO algorithm that combines previous work on dead-end elimination with both new strategies for flagging pairs. This unified approach facilitates the daily optimization of protein design cases that were previously intractable using available computational resources.

As illustrated by our full-sequence design example, care must be taken to ensure that algorithmic performance benchmarks are biologically meaningful. An unbiased evaluation process that mimics the invaluable role that CASP[44] has played for the protein structure-prediction community could similarly aid the development and evaluation of computational protein design algorithms. Comparisons should evaluate two features of protein design methods: search efficiency of test cases based on a validated physical model, and design quality based on new physical models submitted by the contributors.

Methods

Physical Model

The potential function has been previously described,[2,9,45–47] and incorporates terms for van der Waals interactions, hydrogen bonds, electrostatic interactions, and solvation. The van der Waals term is based on a Lennard-Jones 12-6 form with scaled atomic radii to promote overpacking in the protein core;[30] the hydrogen bond potential is a distance-dependent term based on a similar 12-10 form but attenuated by an angle-dependent term to enforce reasonable geometry;[6] electrostatic interactions are modeled using Coulomb's law with a distance-dependent dielectric;[46] solvation effects are modeled using approximate pairwise surface area decompositions to reward and penalize buried and exposed nonpolar surface areas, respectively[45] (this term is not computed for surface positions due to a lack of appropriate experimental data with which to parameterize the scaling factor); an additional solvation term penalizes polar hydrogen burial.[6]

The backbone-dependent rotamer libraries are based on the mean values from the Dunbrack and Karplus library[48] with expansion of the χ_1 and χ_2 angles for the aromatic residues, the χ_1 angle for hydrophobic residues, and no expansion for polar residues. Canonical values of the χ_3 and χ_4 angles are used for amino acids E, Q, K, and R. Residues are classified into core, boundary, or surface positions by an automated algorithm.[47] Core residue identities are selected from among the amino acids A, V, L, I, F, Y, and W, while surface residue identities are selected from among A, S, T, D, N, H, E, Q, K, and R. Boundary residue identities are chosen from the union of these sets.

Benchmark Design Cases

Case 1 represents the design of all 25 nonglycine residues (5, 14, 21, 27, 29, 31, 35, 37, 38, 39, 41, 46, 50, 55, 56, 63, 70, 72, 74, 80, 82, 84, 92, 96, 98) in the core of plastocyanin (PDB code 2pcy).[38] Case 2 involves all 34 core positions on a novel repeating backbone based on the leucine-rich-repeat motif;[39] the 17 residues in each of two repeats have linked (but unspecified) amino acid identities. Case 3 represents the full core design of the variable domains of the light and heavy chains of catalytic antibody 48G7 FAB (PDB code 1gaf).[40] This corresponds to residues (2, 4, 6, 19, 21, 25, 29, 33, 36–38, 44, 46–48, 55, 58, 62, 71, 73, 75, 78, 82, 84–87, 89, 90, 95–98, 102, 104) of chain L and residues (4, 6, 18, 20, 24, 32, 34–39, 45, 47, 48, 50, 51, 53, 61, 64, 68, 70, 72, 77, 79, 81, 83, 86, 90, 92–95, 97, 98, 103, 104, 108, 110) of chain H. Case 4 involves the design of all 10 nonglycine core residues (3, 5, 7, 20, 26, 30, 34, 39, 52, 54) and all 15 boundary residues (1, 11, 12, 16, 18, 23, 25, 27, 29, 33, 37, 43, 45, 50, 56) of the β1 domain of protein G (PDB code 1pga).[41] Case 5 represents the design of all 27 nonglycine surface residues (2, 4, 6, 8, 10, 13, 15, 17, 19, 21, 22, 24, 28, 31, 32, 35, 36, 40, 42, 44, 46, 47, 48, 49, 51, 53, 55) of the β1 domain of protein G. The benchmark calculations were performed on 16 Power3 processors of an IBM SP3 running at 375 MHz.

References

1. Desjarlais, J. R.; Handel, T. M. Protein Sci 1995, 4, 2006.
2. Dahiyat, B. I.; Mayo, S. L. Protein Sci 1996, 5, 895.
3. Lazar, G. A.; Desjarlais, J. R.; Handel, T. M. Protein Sci 1997, 6, 1167.
4. Harbury, P. B.; Plecs, J. J.; Tidor, B.; Alber, T.; Kim, P. S. Science 1998, 282, 1462.
5. Shimaoka, M.; Shifman, J. M.; Jing, H.; Takagi, J.; Mayo, S. L.; Springer, T. A. Nature Struct Biol 2000, 7, 674.
6. Dahiyat, B. I.; Gordon, D. B.; Mayo, S. L. Protein Sci 1997, 6, 1333.
7. Malakauskas, S. M.; Mayo, S. L. Nature Struct Biol 1998, 5, 470.
8. Street, A. G.; Datta, D.; Gordon, D. B.; Mayo, S. L. Phys Rev Lett 2000, 84, 5010.
9. Dahiyat, B. I.; Mayo, S. L. Science 1997, 278, 82.
10. Desjarlais, J. R.; Clarke, N. D. Curr Opin Struct Biol 1998, 8, 471.
11. Pierce, N. A.; Winfree, E. Protein Engineering 2002, 15, 779.
12. Lee, C.; Levitt, M. Nature 1991, 352, 448.
13. Hellinga, H. W.; Richards, F. M. Proc Natl Acad Sci USA 1994, 91, 5803.
14. Koehl, P.; Delarue, M. J Mol Biol 1994, 239, 249.
15. Lee, C. J Mol Biol 1994, 236, 918.
16. Voigt, C. A.; Gordon, D. B.; Mayo, S. L. J Mol Biol 2000, 299, 789.
17. Gordon, D. B.; Mayo, S. L. Structure 1999, 7, 1089.
18. Wernisch, L.; Hery, S.; Wodak, S. J. J Mol Biol 2000, 301, 713.
19. Desmet, J.; De Maeyer, M.; Hazes, B.; Lasters, I. Nature 1992, 356, 539.
20. Lasters, I.; Desmet, J. Protein Eng 1993, 6, 717.
21. Desmet, J.; De Maeyer, M.; Lasters, I. The protein folding problem and tertiary structure prediction; Birkhauser: Boston, 1994.
22. Goldstein, R. F. Biophys J 1994, 66, 1335.
23. DeMaeyer, M.; Desmet, J.; Lasters, I. Fold Des 1997, 2, 53.
24. Gordon, D. B.; Mayo, S. L. J Comput Chem 1998, 19, 1505.
25. Pierce, N. A.; Spriet, J. A.; Desmet, J.; Mayo, S. L. J Comput Chem 2000, 21, 999.
26. Looger, L. L.; Hellinga, H. W. J Mol Biol 2001, 307, 429.
27. Leach, A. R.; Lemon, A. P. Proteins 1998, 33, 227.
28. Papadimitriou, C. H.; Steiglitz, K. Combinatorial optimization: algorithms and complexity; Prentice Hall: New Jersey, 1982; pp 442–444.
29. Metropolis, N.; Rosenbluth, A. W.; Rosenbluth, M. N.; Teller, A. H. J Chem Phys 1953, 21, 1087.

30. Dahiyat, B. I.; Mayo, S. L. Proc Natl Acad Sci USA 1997, 94, 10172.
31. Su, A.; Mayo, S. L. Protein Sci 1997, 6, 1701.
32. Strop, P.; Mayo, S. L. J Am Chem Soc 1999, 121, 2341.
33. Strop, P.; Marinescu, A. M.; Mayo, S. L. Protein Sci 2000, 9, 1391.
34. Marshall, S. A.; Mayo, S. L. J Mol Biol 2001, 305, 619.
35. Sarisky, C. A.; Mayo, S. L. J Mol Biol 2001, 307, 1411.
36. Ross, S. A.; Sarisky, C. A.; Su, A.; Mayo, S. L. Protein Sci 2001, 10, 450.
37. Bolon, D. N.; Mayo, S. L. PNAS 2001, 98, 14274.
38. Garrett, T. P.; Clingeleffer, D. J.; Guss, J. M.; Rogers, S. J.; Freeman, H. C. J Biol Chem 1984, 259, 2822.
39. Kobe, B.; Deisenhofer, J. J Mol Biol 1996, 264, 1028.
40. Patten, P. A.; Gray, N. S.; Yang, P. L.; Marks, C. B.; Wedemayer, G. J.; Boniface, J. J.; Stevens, R. C.; Schultz, P. G. Science 1996, 271, 1086.
41. Gallagher, T.; Alexander, P.; Bryan, P.; Gilliland, G. L. Biochemistry 1994, 33, 4721.
42. Shah, P.; Mayo, S. L. 2001, personal communication.
43. Marshall, S. A.; Mayo, S. L. 2001, personal communication.
44. Venclovas, C.; Zemla, A.; Fidelis, K.; Moult, J. Proteins 1999, Suppl, 231.
45. Street, A. G.; Mayo, S. L. Fold Des 1998, 3, 253.
46. Gordon, D. B.; Marshall, S. A.; Mayo, S. L. Curr Opin Struct Biol 1999, 9, 509.
47. Street, A. G.; Mayo, S. L. Struct Fold Des 1999, 7, R105.
48. Dunbrack Jr, R. L.; Karplus, M. J Mol Biol 1993, 230, 543.