Computational Comparison Studies of Quadratic Assignment Like Formulations for the In Silico Sequence Selection Problem in De Novo Protein Design

H. K. Fung[1], S. Rao[1], C. A. Floudas[1*], O. Prokopyev[2], P. M. Pardalos[2], and F. Rendl[3]

[1] Department of Chemical Engineering, Princeton University, Princeton, NJ 08544-5263

[2] Department of Industrial and Systems Engineering, University of Florida, Gainesville, FL 32611-6595

[3] Institut für Mathematik, Universität Klagenfurt, A-9020 Klagenfurt, Austria

Abstract

In this paper an $O(n^2)$ mathematical formulation for *in silico* sequence selection in de novo protein design proposed by Klepeis *et al.* (2003)(2004), in which the number of additional variables and linear constraints scales with the square of the number of binary variables, is compared to three $O(n)$ formulations. It is found that the $O(n^2)$ formulation is superior to the $O(n)$ formulations on most sequence search spaces. The superiority of the $O(n^2)$ formulation is due to the reformulation linearization techniques (RLTs), since the $O(n^2)$ formulation without RLTs is found to be computationally less efficient than the $O(n)$ formulations. In addition, new algorithmic enhancing components of RLTs with inequality constraints, triangle inequalities, and Dead-End Elimination (DEE) type preprocessing are added to the $O(n^2)$ formulation. The current best $O(n^2)$ formulation, which is the original formulation from Klepeis *et al.* (2003)(2004) plus DEE type preprocessing, is proposed for *in silico* sequence search. For a test problem with a search space of 3.4×10^{45} sequences, this new improved model is able to reduce the required CPU time by 67%.

Keywords

Peptide and protein design and discovery; Drug design; In silico sequence selection; Structure prediction; De novo protein design; Optimization

1 Introduction

De novo peptide or protein design starts with a flexible 3-dimensional protein structure and involves the search for all amino acid sequences that fold into such a template. The motivation behind computational protein design is usually a quest for improved activity (e.g., higher inhibitory activity for an inhibitor) (Klepeis *et al.*, 2003)(2004), but it is definitely not where the applications are limited to. De novo protein design has been successfully employed for modulating protein-protein interactions (Kortemme and Baker, 2004), promoting stability of the target protein (Malakauskas and Mayo, 1998) (Kuhlman and Baker, 2004), conferring novel binding sites or properties onto the template (Richards and Hellinga, 1991) (Richards *et al.*, 1991), and locking proteins into certain useful conformations (Shimaoka *et al.*, 2000) (Kraemer-Pecore *et al.*, 2001). To a large extent it enhances our understanding of proteins' sequence-structure relationship and protein molecular and structural biology, a key research area in the post-Human-Genome-Project era.

With an enormous potential that has only been minimally harnessed, computational protein design does, however, possess some inherent limitation. The limitation stems from the fact that de novo protein design is

*Author to whom all correspondence should be addressed; Tel: (609) 258-4595; Fax: (609) 258-0211; E-mail: floudas@titan.princeton.edu.

an NP-hard problem (Pierce and Winfree, 2002), and hence computational time required scales exponentially with the number of design positions on the protein template. This makes full-sequence-full-combinatorial design on proteins of practical size (i.e., 100 - 200 residues) very challenging. The maximum sequence search space de novo protein design can handle varies drastically from one approach to another, with the main determinant being the algorithms behind or the mathematical formulation employed.

Klepeis *et al.* (2003)(2004) proposed a novel two-stage protein design framework. In the first stage *in silico* sequence selection is executed based on the minimization of the sum of energy interactions between each amino acid pair in the protein. In the second stage of fold specificity calculation, protein structure prediction is performed by solving a nonconvex constrainted global optimization formulation with an objective function of an atomistic energy force field over the set of independent dihedral angles which can be used to describe any possible configuration of the system. Klepeis *et al.* (2004) solved the formulation with the αBB deterministic global optimization approach, a branch-and-bound method applicable to the identification of the global minimum in nonlinear optimization problems with twice-differentiable functions (Klepeis *et al.*, 2002) (Klepeis *et al.*, 1999) (Adjiman *et al.*, 1998a,b, 2000) (Klepeis and Floudas, 1999) (Floudas, 2000). In addition, structure prediction is done under two different circumstances. Under the first circumstance, the structure is constrained to vary, with some imposed fluctuations, about the template. Under the second circumstance, a free-folding calculation is carried out with only a limited number of constraints, like the disulfide bridge constraint but not the underlying template structure enforced. A consistent ensemble of low-energy conformations produced by the global optimization algorithm provides a means for quantifying the fold specificity of each low-lying energy sequence obtained from the first stage. The use of a relative probability for folding into the template structure avoids the complications inherent in the specification of an appropriate reference state. The relative folding probability can be found by summing the statistical weights for those conformers from the free folding simulation that resemble the template structure, and dividing this sum by the summation of statistical weights for all conformers from the free folding simulation (Klepeis *et al.*, 2004).

In this article, the focus is on the mathematical formulation for the first stage of *in silico* sequence selection. First, an overview on the mathematical formulation proposed by Klepeis *et al.* (2003)(2004) for computational sequence search will be presented, along with discussion on computational complexity of the considered formulation. Then, three equivalent $O(n)$ formulations, as well as a new improved $O(n^2)$ formulation empowered with algorithmic enhancing components will be introduced. Finally, the computational efficiency of all proposed formulations will be investigated using a selected set of test problems, which search for the sequence with global minimum in energy for the template of human beta defensin 2.

2 Overview of In Silico Sequence Selection in De Novo Protein Design

The novel formulation for the *in silico* sequence selection stage of the de novo protein design framework proposed by Klepeis *et al.* (2003) (2004) is of the following original form:

$$\begin{aligned} \min_{y_i^j, y_k^l} \quad & \textstyle\sum_{i=1}^{n} \sum_{j=1}^{m_i} \sum_{k=i+1}^{n} \sum_{l=1}^{m_k} E_{ik}^{jl}(x_i, x_k) y_i^j y_k^l \\ \text{subject to} \quad & \textstyle\sum_{j=1}^{m_i} y_i^j \;=\; 1 \;\; \forall \;\; i \\ & y_i^j \;,\; y_k^l \;=\; 0-1 \;\; \forall \;\; i,j,k,l \end{aligned} \tag{1}$$

Note that this formulation corresponds to a quadratic assignment like model. It differs, however, in the set of constraints. Set $i = 1, \ldots, n$ defines the number of residue positions along the backbone. At each position i there can be a set of mutations represented by $j\{i\} = 1, \ldots, m_i$, where, for the general case $m_i = 20 \forall i$. The equivalent sets $k \equiv i$ and $l \equiv j$ are defined, and $k > i$ is required to represent all unique pairwise interactions. Binary variables y_i^j and y_k^l are introduced to indicate the possible mutations at a given position. That is, the y_i^j variable will indicate which type of amino acid is active at a position in the sequence by taking the value

of one for that specification. The composition constraints in the formulation require that there is exactly one type of amino acid at each position.

The objective function to be minimized represents the sum of pairwise amino acid energy interactions in the template. Parameter $E_{ik}^{jl}(x_i, x_k)$, which is the energy interaction between position i occupied by amino acid j and position k occupied by amino acid l, depends on the distance between the alpha-carbons at the two backbone positions (x_i, x_k) as well as the type of amino acids j and l. These energy parameters were empirically derived based on solving a linear programming parameter estimation problem subject to constraints which were in turn constructed by requiring the energies of a large number of low-energy decoys to be larger than the corresponding native protein conformation for each member of a set of proteins (Loose *et al.*, 2004). The resulting potential, which contains $1,680$ energy parameters for different amino acid pairs and distance bins, was shown to rank the native fold as the lowest in energy in more proteins tested than other potentials and also yield higher Z-score (Loose *et al.*, 2004) (Tobi and Elber, 2000) (Tobi *et al.*, 2000). The fact that the energy potential is discretized into bins rather than being a continuous function is highly desirable as it inherently incorporates backbone flexibility for the protein.

As indicated in the formulation, bilinear terms appear in the objective function. The objective can be reformulated as a strictly linear (integer linear programming) problem using a standard linearization approach:

$$\begin{array}{rc} \min\limits_{y_i^j, y_k^l} & \sum_{i=1}^{n}\sum_{j=1}^{m_i}\sum_{k=i+1}^{n}\sum_{l=1}^{m_k} E_{ik}^{jl}(x_i,x_k) w_{ik}^{jl} \\ \text{subject to} & \sum_{j=1}^{m_i} y_i^j \;=\; 1 \;\forall\; i \\ & y_i^j \;+\; y_k^l \;-\; 1 \;\leq\; w_{ik}^{jl} \;\leq\; y_i^j \;\forall\; i,j,k,l \\ & 0 \;\leq\; w_{ik}^{jl} \;\leq\; y_k^l \;\forall\; \; i,j,k,l \\ & y_i^j \;,\; y_k^l \;=\; 0-1 \;\;\forall\;\; i,j,k,l \end{array} \qquad \text{(F1)}$$

Formulation (F1) is derived from the transformation of the bilinear combinations into a new set of linear variables, w_{ik}^{jl}, while the addition of the four sets of constraints serves to reproduce the characteristics of the original formulation. For example, for a given i, j, k, l combination, the four constraints require w_{ik}^{jl} to be zero when either y_i^j or y_k^l is equal (or when both are equal to zero). If both y_i^j and y_k^l are equal to one then w_{ik}^{jl} is also enforced to be one. The solution of the integer linear programming problem (ILP) can be accomplished rigorously using branch and bound techniques (CPLEX, 1997) (Floudas, 1995) making convergence to the global minimum energy sequence consistent and reliable.

Formulation (F1) is an $O(n^2)$ formulation, meaning that the number of linear constraints (excluding composition constraints) scales with n^2, where n is the number of binary variables. For instance, if all 20 amino acids are considered for each position in a 40-residue protein, then n equals $40 \times 20 = 800$. The number of variables w_{ik}^{jl} will be $400 \times 820 = 328,000$, and hence number of linear constraints is simply $4 \times 328,000 = 1,312,000$, which is roughly on the order of $|n|^2$.

Klepeis *et al.* (2004) reported that the performance of the branch and bound algorithm could be significantly enhanced through the introduction of reformulation linearization techniques (RLT). The basic strategy is to multiply appropriate constraints by bounded non-negative factors (such as the reformulated variables) and introduce the products of the original variables by new variables in order to derive higher-dimensional lower bounding linear programming (LP) relaxations for the original problem (Sherali and Adams, 1999). These LP relaxations are solved during the course of the overall branch and bound algorithm, and thus speed convergence to the global minimum. In the case of the formulation for *in silico* sequence selection, RLT is introduced by multiplying the composition constraints by the binary variables y_k^l to produce the following additional set of constraints $\forall \;\; j, k, l$:

$$y_k^l \sum_{j=1}^{m_i} y_i^j \;=\; y_k^l \;\;\forall\;\; i,k,l \qquad (2)$$

This equation is linearized using the same variable substitution as introduced for the objective. The set of RLT constraints then become:

$$\sum_{j=1}^{m_i} w_{ik}^{jl} \;=\; y_k^l \;\;\forall\;\; i,k,l \tag{3}$$

In summary, the RLT-empowered $O(n^2)$ formulation of Klepeis *et al.* (2003)(2004) is as follows:

$$\begin{array}{rc} \min\limits_{y_i^j, y_k^l} & \sum_{i=1}^{n}\sum_{j=1}^{m_i}\sum_{k=i+1}^{n}\sum_{l=1}^{m_k} E_{ik}^{jl}(x_i,x_k) w_{ik}^{jl} \\ \text{subject to} & \sum_{j=1}^{m_i} y_i^j \;=\; 1 \;\forall\; i \\ & y_i^j \;+\; y_k^l \;-\; 1 \;\leq\; w_{ik}^{jl} \;\leq\; y_i^j \;\forall\; i,j,k,l \\ & 0 \;\leq\; w_{ik}^{jl} \;\leq\; y_k^l \;\forall\;\; i,j,k,l \\ & \sum_{j=1}^{m_i} w_{ik}^{jl} \;=\; y_k^l \;\;\forall\;\; i,k,l \\ & y_i^j \;,\; y_k^l \;=\; 0-1 \;\;\forall\;\; i,j,k,l \end{array} \tag{F2}$$

In the mathematical formulation comparison studies outlined in this article, both $O(n^2)$ formulations (F1) and (F2), along with other $O(n)$ formulations and new $O(n^2)$ formulations with algorithmic enhancement techniques are employed to compute the sequence with the global energy minimum for the same set of test problems, and the respective computational times required are compared.

3 Complexity Issues

It is known that de novo protein design is an NP-hard problem (Pierce and Winfree, 2002). Next we present another proof of this result. There are two advantages of the presented proof. First, the proposed reduction suggests that unconstrained quadratic 0–1 programming problem (UQ01) is a specific subclass of problem (1). Therefore, some of the complexity results proved for UQ01 are also valid for problem (1) (for more details on complexity of UQ01 see Pardalos and Jha (1992)). The second argument is that problem (1) remains NP-hard even if the number of possible mutations for all residue positions along the backbone is equal to 2. Although this complexity result characterizes worst-case instances, it provides some insight into the problem difficulty and indicates that de novo protein design is a hard combinatorial optimization problem.

Theorem 3.1 *Problem (1) is NP-hard. This result remains valid if for all i the number of possible mutations $m_i = 2$.*

Proof. Consider an unconstrained quadratic 0–1 programming problem, which is defined as follows:

$$\min_{x\in\{0,1\}^p} x^T Q x,$$

where Q is an $p \times p$ symmetric real matrix. This problem is known to be NP-hard. In order to prove the needed statement we reduce UQ01 to formulation (1).

Let $n = 2p$ and for all i we have that $m_i = 2$. Next assign the following energies:

- for $i = 1,\ldots,p$ and corresponding $k = i+1,\ldots,p$ set $E_{ik}^{11} = q_{ik} + q_{ki}$, where q_{ki} and q_{ik} are elements of the matrix Q;
- for $i = 1,\ldots,p$ set $E_{i,i+p}^{11} = q_{ii}$ and $E_{i,i+p}^{12} = q_{ii}$;
- for all other i and corresponding $k = i+1,\ldots,n$ set $E_{ik}^{12} = E_{ik}^{21} = E_{ik}^{11} = E_{ik}^{22} = 0$.

Using the aforementioned values of m_i and energies the objective function in (1) can be rewritten as follows:

$$\sum_{i=1}^{n}\sum_{j=1}^{m_i}\sum_{k=i+1}^{n}\sum_{l=1}^{m_k} E_{ik}^{jl} y_i^j y_k^l = \sum_{i=1}^{n}\sum_{j=1}^{2}\sum_{k=i+1}^{n}\sum_{l=1}^{2} E_{ik}^{jl} y_i^j y_k^l =$$

$$= \sum_{i=1}^{p}\sum_{k=i+1}^{p} E_{ik}^{11} y_i^1 y_k^1 + \sum_{i=1}^{p} E_{i,i+p}^{11} y_i^1 y_{i+p}^1 + \sum_{i=1}^{p} E_{i,i+p}^{12} y_i^1 y_{i+p}^2 =$$

$$= \sum_{i=1}^{p}\sum_{k=i+1}^{p} (q_{ik}+q_{ki}) y_i^1 y_k^1 + \sum_{i=1}^{p} q_{ii} y_i^1 y_{i+p}^1 + \sum_{i=1}^{p} q_{ii} y_i^1 y_{i+p}^2 =$$

$$= \sum_{i=1}^{p}\sum_{k=i+1}^{p} (q_{ik}+q_{ki}) y_i^1 y_k^1 + \sum_{i=1}^{p} q_{ii} y_i^1 (y_{i+p}^1 + y_{i+p}^2) =$$

$$= \sum_{i=1}^{p}\sum_{k=i+1}^{p} (q_{ik}+q_{ki}) y_i^1 y_k^1 + \sum_{i=1}^{p} q_{ii} y_i^1$$

Let $x_i \equiv y_i^1$. All assignment constraints of the type $y_i^1 + y_i^2 = 1$ are automatically satisfied and can be removed since variables y_i^2 do not appear in the objective function. The described reduction is obviously polynomial. Therefore, problem (1) is NP-hard.

4 O(n) Formulations

In this section we describe briefly three $O(n)$ formulations which were derived for quadratic assignment problems and have been proved to be totally equivalent to $O(n^2)$ formulations (F1) and (F2).

First two formulations were proposed by Oral and Kettani (1990) (1992) (and are modifications of the initial technique by Glover (1975)) for a general linearly constrained quadratic 0–1 programming problem. Applications of these general approaches to problem (1), which is a specific class of linearly constrained quadratic 0–1 programming problem, results in the following two formulations:

$$\begin{array}{rc}
\min\limits_{y_i^j,\zeta_i^j} & \sum_{i=1}^{n}\sum_{j=1}^{m_i} {D_i^j}^- y_i^j \;+\; \zeta_i^j \\
\textit{subject to} & \zeta_i^j \;\geq\; \sum_{k=i+1}^{n}\sum_{l=1}^{m_k} E_{ik}^{jl}(x_i,x_k) y_k^l \;-\; {D_i^j}^- y_i^j \;-\; {D_i^j}^+ (1 \;-\; y_i^j) \;\; \forall i,j \\
& \sum_{j=1}^{m_i} y_i^j = 1 \;\; \forall i \\
& \zeta_i^j \;\geq\; 0 \;\; \forall i,j \\
& y_i^j, y_k^l = 0-1 \;\; \forall i,j,k,l \\
& D_i^{j-} \;= -\sum_{k=i+1}^{n} \max_{1\leq l\leq m_k} |\min\{0, E_{ik}^{jl}(x_i,x_k)\}| \\
& D_i^{j+} \;= \sum_{k=i+1}^{n} \max_{1\leq l\leq m_k} \max\{0, E_{ik}^{jl}(x_i,x_k)\}
\end{array} \qquad \text{(F3)}$$

and

$$\begin{array}{rc}
\min\limits_{y_i^j,\zeta_i^j} & \sum_{i=1}^{n}\sum_{j=1}^{m_i} \left(\sum_{k=i+1}^{n}\sum_{l=1}^{m_k} E_{ik}^{jl}(x_i,x_k) y_k^l \;-\; {B_i^j}^+ (1 \;-\; y_i^j) \;+\; \zeta_i^j \right) \\
\textit{subject to} & \zeta_i^j \;\geq\; -\sum_{k=i+1}^{n}\sum_{l=1}^{m_k} E_{ik}^{jl}(x_i,x_k) y_k^l \;+\; {B_i^j}^- y_i^j \;+\; {B_i^j}^+ (1 \;-\; y_i^j) \;\; \forall i,j \\
& \sum_{j=1}^{m_i} y_i^j = 1 \;\; \forall i \\
& \zeta_i^j \;\geq\; 0 \;\; \forall i,j
\end{array} \qquad \text{(F4)}$$

$$
\begin{aligned}
y_i^j, y_k^l &= 0-1 \ \ \forall i,j,k,l \\
B_i^{j-} &= -\sum_{k=i+1}^{n} \max_{1\le l\le m_k} |\min\{0, E_{ik}^{jl}(x_i,x_k)\}| \\
B_i^{j+} &= \sum_{k=i+1}^{n} \max_{1\le l\le m_k} \max\{0, E_{ik}^{jl}(x_i,x_k)\}
\end{aligned}
$$

Both formulations (F3) and (F4) are promising in terms of their times to convergence because as compared to the original binary quadratic integer problem of n variables, the number of auxiliary linear constraints is reduced to n, whereas the number of new continuous variables ζ_i^j introduced is n versus the n^2 binary variables w_{ik}^{jl} in the $O(n^2)$ formulations.

The third formulation is another modification of the previous ones, but in this formulation the number of new variables is increased to $2n$ and we keep some of the upper bounding linear constraints (Oral and Kettani (1990), Pardalos *et al.* (2004)):

$$
\begin{aligned}
\min_{s_i^j, y_i^j, \zeta_i^j} \quad & \sum_{i=1}^{n}\sum_{j=1}^{m_i} s_i^j \ - \ M_i^{j-} y_i^j \\
\text{subject to} \quad & [\sum_{k=i+1}^{n}\sum_{l=1}^{m_k} E_{ik}^{jl}(x_i,x_k) y_k^l] \ - \ \zeta_i^j \ - \ s_i^j \ + \ M_i^{j-} \ \le \ 0 \ \ \forall \ \ i,j \\
& \zeta_i^j \ \le \ M_i^j (1 \ - \ y_i^j) \ \ \forall \ \ i,j \\
& \sum_{j=1}^{m_i} y_i^j \ = \ 1 \ \forall \ i \qquad \text{(F5)} \\
& M_i^{j-} = \sum_{k=i+1}^{n} \max_{1\le l\le m_k} |\min\{0, E_{ik}^{jl}(x_i,x_k)\}| \\
& M_i^{j+} = \sum_{k=i+1}^{n} \max_{1\le l\le m_k} \max\{0, E_{ik}^{jl}(x_i,x_k)\} \\
& M_i^j \ = \ M_i^{j-} \ + \ M_i^{j+} \ \ \forall \ \ i,j \\
& \zeta_i^j \ \ge \ 0, s_i^j \ \ge \ 0, y_i^j \ = \ 0-1 \ \ \forall \ \ i,j
\end{aligned}
$$

where set i, set j, and energy parameters $E_{ik}^{jl}(x_i,x_k)$ are exactly the same as they were in the $O(n^2)$ formulations.

It is easy to show that the above formulations (F3)-(F5) are equivalent to the initial quadratic 0–1 programming problem (1), and hence they should give the same global energy minimum value when it is applied on the same test problem for sequence selection. The computational efficiency of $O(n)$ formulations is of major interest because of the significant reduction in the number of variables and linear constraints.

5 New Improved Class of O(n^2) Formulations

In an effort to generate better O(n^2) formulations, a number of new ideas were investigated so as to speed up the sequence search algorithm. The resulting novel models were included in the computational comparison. Different combinations of the new elements were investigated. The new components to be tested are (a) conversion of the equality RLT constraints into inequality constraints, (b) addition of triangle inequalities, and (c) execution of a preprocessing step using one iteration of the Dead-End Elimination theorem before solving the *in silico* sequence selection model.

Since RLTs and triangle inequalities both lead to superfluous equations which do not affect the feasibility region of the original formulation (F1), and preprocessing simplifies the formulation by eliminating the binary variables that might otherwise be unable to be recognized as fixable, implementation of any combination of the three will certainly not affect the objective function value.

5.1 RLT with inequalities

The rationale for this technique is to relax the RLT constraints, which are supposed to be crucial in speeding up the branch and bound algorithm in the original $O(n^2)$ formulation that Klepeis *et al.* (2004) proposed, by

changing the equality in the equation to "less than or equal to," making the RLT equation look like:

$$\sum_{j=1}^{m_i} w_{ik}^{jl} \;\leq\; y_k^l \;\;\forall\;\; i,k,l \tag{4}$$

Considering that equality is equivalent to both "larger than or equal to" and "less than or equal to," implementing only the latter will probably lead to a problem that is easier and faster to solve.

5.2 Addition of triangle inequalities

Valid triangle inequalities as shown below were added to the $O(n^2)$ formulation in an attempt to hasten convergence to the global energy minimum solution. Similar to RLTs, they are supposed to enhance the algorithm by providing tighter lower bounds to the original problem.

$$(y_i^j \;-\; y_k^l)(y_i^j \;-\; y_m^p) \;\geq\; 0 \;\;\forall i < k < m, j, l, p \tag{5}$$

$$2 \;-\; (y_i^j \;-\; y_k^l)^2 \;-\; (y_i^j \;-\; y_m^p)^2 \;-\; (y_k^l \;-\; y_m^p)^2 \;\geq\; 0 \;\;\forall i < k < m, j, l, p \tag{6}$$

The equations can be expanded to obtain the following final form of linear constraints which have been included into the quadratic integer problem:

$$y_i^j \;-\; w_{im}^{jp} \;-\; w_{ik}^{jl} \;+\; w_{km}^{lp} \;\geq\; 0 \;\;\forall i < k < m, j, l, p \tag{7}$$

$$w_{ik}^{jl} \;+\; w_{im}^{jp} \;+\; w_{km}^{lp} \;-\; y_i^j \;-\; y_k^l \;-\; y_m^p \;+\; 1 \;\geq\; 0 \;\;\forall i < k < m, j, l, p \tag{8}$$

where linear binary variables y_i^j and w_{ik}^{jl} were defined in the same way as before, whereas indices m and p were aliases of position sets i and k and amino acid sets j and l respectively. Moreover, position triplets i, k, and m is subject to the constraint of $i \;<\; k \;<\; m$.

An additional subtlety to consider in applying triangle inequalities is the total number of inequalities to impose, which supposedly has an optimal value giving the best computational efficiency. In view of this, triangle inequalities were to be applied only if the sum of the pairwise energy triplets, namely $S_{ikm}^{jlp} \;=\; E_{ik}^{jl} \;+\; E_{im}^{jp} \;+\; E_{km}^{lp}$ was less than a certain cutoff value. Both cases of no cutoff and cutoff value of -40 were tried in the formulation comparison studies.

5.3 Preprocessing

The way preprocessing delivers improvement in computational efficiency is usually by means of reducing the problem size by eliminating some of the variables. In mathematical terms the preprocessing step can be stated as follows:

$$\begin{gathered}\text{If } \exists\, \widetilde{j} \neq j \text{ s. t. } \textstyle\sum_{k,k>i} \min_l [E_{ik}^{jl} \;-\; E_{ik}^{\widetilde{j}l}] \;>\; 0 \\ \text{then } y_i^j \;=\; 0\end{gathered} \tag{9}$$

The original idea of the above came from the Dead-End Elimination (DEE) criterion (Goldstein, 1994) (Pierce *et al.*, 2000) (Voigt *et al.*, 2000) (Gordon *et al.*, 2003):

$$E(i_a) \; - \; E(i_b) \; + \; \sum_{k \neq i} \min_c [E(i_a, k_c) - E(i_b, k_c)] \; > \; 0 \tag{10}$$

which states that rotamer i_a at position i can be pruned if its energy contributation is always lowered by substituting with an alternative rotamer i_b. In the de novo protein design framework that Klepeis *et al.* (2003) (2004) developed, different conformations for each amino acid mutation were not considered. In other words, the number of rotamers at each position is only one for each amino acid to consider. Nevertheless, the DEE criterion is still applicable independent of the number of rotamers. Since in Klepeis *et al.* (2003) (2004)'s model the total energy only takes into account pairwise amino acid interactions but not each amino acid itself, the energies of the rotamers i_a and i_b themselves (i.e., $E(i_a)$ and $E(i_b)$) immediately go to zero, yielding equation (9) which is in a form incorporable into the $O(n^2)$ formulation.

5.4 The formulations

With the three aforementioned algorithmic enhancing operations, a list of novel $O(n^2)$ formulations were generated:

$$\begin{array}{rc}
\min\limits_{y_i^j, y_k^l} & \sum_{i=1}^{n} \sum_{j=1}^{m_i} \sum_{k=i+1}^{n} \sum_{l=1}^{m_k} E_{ik}^{jl}(x_i, x_k) w_{ik}^{jl} \\
\text{subject to} & \sum_{j=1}^{m_i} y_i^j \; = \; 1 \; \forall \; i \\
& y_i^j \; + \; y_k^l \; - \; 1 \; \leq \; w_{ik}^{jl} \; \leq \; y_i^j \; \forall \; i, j, k, l \\
& 0 \; \leq \; w_{ik}^{jl} \; \leq \; y_k^l \; \forall \; \; i, j, k, l \\
& \sum_{j=1}^{m_i} w_{ik}^{jl} \; \leq \; y_k^l \; \; \forall \; \; i, k, l \\
& y_i^j \; , \; y_k^l \; = \; 0 - 1 \; \; \forall \; \; i, j, k, l
\end{array} \tag{F6}$$

$$\begin{array}{rc}
\min\limits_{y_i^j, y_k^l} & \sum_{i=1}^{n} \sum_{j=1}^{m_i} \sum_{k=i+1}^{n} \sum_{l=1}^{m_k} E_{ik}^{jl}(x_i, x_k) w_{ik}^{jl} \\
\text{subject to} & \sum_{j=1}^{m_i} y_i^j \; = \; 1 \; \forall \; i \\
& y_i^j \; + \; y_k^l \; - \; 1 \; \leq \; w_{ik}^{jl} \; \leq \; y_i^j \; \forall \; i, j, k, l \\
& 0 \; \leq \; w_{ik}^{jl} \; \leq \; y_k^l \; \forall \; \; i, j, k, l \\
& \sum_{j=1}^{m_i} w_{ik}^{jl} \; \leq \; y_k^l \; \; \forall \; \; i, k, l \\
& y_i^j \; - \; w_{im}^{jp} \; - \; w_{ik}^{jl} \; + \; w_{km}^{lp} \; \geq \; 0 \\
\forall i < k < m, j, l, p \; \text{ s. t. } \; & S_{ikm}^{jlp} \; = \; E_{ik}^{jl} \; + \; E_{im}^{jp} \; + \; E_{km}^{lp} \leq \text{cutoff value} \\
& w_{ik}^{jl} \; + \; w_{im}^{jp} \; + \; w_{km}^{lp} \; - \; y_i^j \; - \; y_k^l \; - \; y_m^p \; + \; 1 \; \geq \; 0 \\
\forall i < k < m, j, l, p \; \text{ s. t. } \; & S_{ikm}^{jlp} \; = \; E_{ik}^{jl} \; + \; E_{im}^{jp} \; + \; E_{km}^{lp} \leq \text{cutoff value} \\
& y_i^j \; , \; y_k^l \; = \; 0 - 1 \; \; \forall \; \; i, j, k, l
\end{array} \tag{F7}$$

$$\begin{array}{rc}
\text{Preprocessing:} & \text{If } \exists \; \widetilde{j} \neq j \text{ s. t. } \sum_{k, k>i} \min_l [E_{ik}^{jl} \; - \; E_{ik}^{\widetilde{j}l}] \; > \; 0 \\
& \text{then } y_i^j \; = \; 0 \\
\min\limits_{y_i^j, y_k^l} & \sum_{i=1}^{n} \sum_{j=1}^{m_i} \sum_{k=i+1}^{n} \sum_{l=1}^{m_k} E_{ik}^{jl}(x_i, x_k) w_{ik}^{jl} \\
\text{subject to} & \sum_{j=1}^{m_i} y_i^j \; = \; 1 \; \forall \; i
\end{array}$$

$$
\begin{aligned}
& y_i^j \ + \ y_k^l \ - \ 1 \ \leq \ w_{ik}^{jl} \ \leq \ y_i^j \ \forall \ i,j,k,l \\
& 0 \ \leq \ w_{ik}^{jl} \ \leq \ y_k^l \ \forall \ \ i,j,k,l \\
& \textstyle\sum_{j=1}^{m_i} w_{ik}^{jl} \ \leq \ y_k^l \ \ \forall \ \ i,k,l \\
& y_i^j \ , \ y_k^l \ = \ 0-1 \ \ \forall \ \ i,j,k,l
\end{aligned}
\tag{F8}
$$

Preprocessing:

$$
\begin{aligned}
& \text{If } \exists \ \widetilde{j} \neq j \text{ s. t. } \textstyle\sum_{k,k>i} \min_l [E_{ik}^{jl} \ - \ E_{ik}^{\widetilde{j}l}] \ > \ 0 \\
& \text{then } y_i^j \ = \ 0
\end{aligned}
$$

$$
\begin{aligned}
\min_{y_i^j, y_k^l} \quad & \textstyle\sum_{i=1}^{n} \sum_{j=1}^{m_i} \sum_{k=i+1}^{n} \sum_{l=1}^{m_k} E_{ik}^{jl}(x_i, x_k) w_{ik}^{jl} \\
\text{subject to} \quad & \textstyle\sum_{j=1}^{m_i} y_i^j \ = \ 1 \ \forall \ i \\
& y_i^j \ + \ y_k^l \ - \ 1 \ \leq \ w_{ik}^{jl} \ \leq \ y_i^j \ \forall \ i,j,k,l \\
& 0 \ \leq \ w_{ik}^{jl} \ \leq \ y_k^l \ \forall \ \ i,j,k,l \\
& \textstyle\sum_{j=1}^{m_i} w_{ik}^{jl} \ \leq \ y_k^l \ \ \forall \ \ i,k,l \\
& y_i^j \ - \ w_{im}^{jp} \ - \ w_{ik}^{jl} \ + \ w_{km}^{lp} \ \geq \ 0 \\
& \forall i < k < m, j, l, p \ \text{ s. t. } \ S_{ikm}^{jlp} \ = \ E_{ik}^{jl} \ + \ E_{im}^{jp} \ + \ E_{km}^{lp} \leq \text{cutoff value} \\
& w_{ik}^{jl} \ + \ w_{im}^{jp} \ + \ w_{km}^{lp} \ - \ y_i^j \ - \ y_k^l \ - \ y_m^p \ + \ 1 \ \geq \ 0 \\
& \forall i < k < m, j, l, p \ \text{ s. t. } \ S_{ikm}^{jlp} \ = \ E_{ik}^{jl} \ + \ E_{im}^{jp} \ + \ E_{km}^{lp} \leq \text{cutoff value} \\
& y_i^j \ , \ y_k^l \ = \ 0-1 \ \ \forall \ \ i,j,k,l
\end{aligned}
\tag{F9}
$$

To summarize, formulation (F6) is just the original formulation Klepeis *et al.* (2003)(2004) proposed (i.e., formulation (F2)) with the equality in the RLT constraints changed to "less than or equal to." Formulation (F7) is formulation (F6) with the addition of triangle inequalities. Formulation (F8) is formulation (F6) with preprocessing, whereas formulation (F9) is formulation (F6) with both triangle inequalities and preprocessing. With the use of the forcefield developed by Loose *et al.* (2004) for the pairwise energy parameters, both cases with no cutoff and with cutoff value of -40 were attempted in imposing the triangle inequalities for formulation (F7). This is to confirm our speculation that a small subset rather than all applicable triangle inequalities are needed to speed up the algorithm. For all the other formulations that possess triangle inequalities, only the case of cutoff value of -40 was attempted.

Finally, the counterparts of formulations (F7), (F8), and (F9) with the equality RLT constraints were also included in the formulation comparison studies. They are namely:

$$
\begin{array}{rc}
\min\limits_{y_i^j, y_k^l} & \sum_{i=1}^{n}\sum_{j=1}^{m_i}\sum_{k=i+1}^{n}\sum_{l=1}^{m_k} E_{ik}^{jl}(x_i,x_k)w_{ik}^{jl} \\
\text{subject to} & \sum_{j=1}^{m_i} y_i^j \;=\; 1\;\forall\; i \\
& y_i^j \;+\; y_k^l \;-\; 1 \;\leq\; w_{ik}^{jl} \;\leq\; y_i^j \;\forall\; i,j,k,l \\
& 0 \;\leq\; w_{ik}^{jl} \;\leq\; y_k^l \;\forall\;\; i,j,k,l \\
& \sum_{j=1}^{m_i} w_{ik}^{jl} \;=\; y_k^l \;\;\forall\;\; i,k,l \\
& y_i^j \;-\; w_{im}^{jp} \;-\; w_{ik}^{jl} \;+\; w_{km}^{lp} \;\geq\; 0 \\
\forall i<k<m,j,l,p & \text{ s. t. } \; S_{ikm}^{jlp} \;=\; E_{ik}^{jl} \;+\; E_{im}^{jp} \;+\; E_{km}^{lp} \leq \text{cutoff value} \\
& w_{ik}^{jl} \;+\; w_{im}^{jp} \;+\; w_{km}^{lp} \;-\; y_i^j \;-\; y_k^l \;-\; y_m^p \;+\; 1 \;\geq\; 0 \\
\forall i<k<m,j,l,p & \text{ s. t. } \; S_{ikm}^{jlp} \;=\; E_{ik}^{jl} \;+\; E_{im}^{jp} \;+\; E_{km}^{lp} \leq \text{cutoff value} \\
& y_i^j \;,\; y_k^l \;=\; 0-1 \;\;\forall\;\; i,j,k,l
\end{array}
\tag{F10}
$$

$$
\begin{array}{rc}
\text{Preprocessing:} & \text{If } \exists\, \widetilde{j} \neq j \text{ s. t. } \sum_{k,k>i} \min_l [E_{ik}^{jl} \;-\; E_{ik}^{\widetilde{j}l}] \;>\; 0 \\
& \text{then } y_i^j \;=\; 0 \\
\min\limits_{y_i^j, y_k^l} & \sum_{i=1}^{n}\sum_{j=1}^{m_i}\sum_{k=i+1}^{n}\sum_{l=1}^{m_k} E_{ik}^{jl}(x_i,x_k)w_{ik}^{jl} \\
\text{subject to} & \sum_{j=1}^{m_i} y_i^j \;=\; 1\;\forall\; i \\
& y_i^j \;+\; y_k^l \;-\; 1 \;\leq\; w_{ik}^{jl} \;\leq\; y_i^j \;\forall\; i,j,k,l \\
& 0 \;\leq\; w_{ik}^{jl} \;\leq\; y_k^l \;\forall\;\; i,j,k,l \\
& \sum_{j=1}^{m_i} w_{ik}^{jl} \;=\; y_k^l \;\;\forall\;\; i,k,l \\
& y_i^j \;,\; y_k^l \;=\; 0-1 \;\;\forall\;\; i,j,k,l
\end{array}
\tag{F11}
$$

$$
\begin{array}{rc}
\text{Preprocessing:} & \text{If } \exists\, \widetilde{j} \neq j \text{ s. t. } \sum_{k,k>i} \min_l [E_{ik}^{jl} \;-\; E_{ik}^{\widetilde{j}l}] \;>\; 0 \\
& \text{then } y_i^j \;=\; 0 \\
\min\limits_{y_i^j, y_k^l} & \sum_{i=1}^{n}\sum_{j=1}^{m_i}\sum_{k=i+1}^{n}\sum_{l=1}^{m_k} E_{ik}^{jl}(x_i,x_k)w_{ik}^{jl} \\
\text{subject to} & \sum_{j=1}^{m_i} y_i^j \;=\; 1\;\forall\; i \\
& y_i^j \;+\; y_k^l \;-\; 1 \;\leq\; w_{ik}^{jl} \;\leq\; y_i^j \;\forall\; i,j,k,l \\
& 0 \;\leq\; w_{ik}^{jl} \;\leq\; y_k^l \;\forall\;\; i,j,k,l \\
& \sum_{j=1}^{m_i} w_{ik}^{jl} \;=\; y_k^l \;\;\forall\;\; i,k,l \\
& y_i^j \;-\; w_{im}^{jp} \;-\; w_{ik}^{jl} \;+\; w_{km}^{lp} \;\geq\; 0 \\
\forall i<k<m,j,l,p & \text{ s. t. } \; S_{ikm}^{jlp} \;=\; E_{ik}^{jl} \;+\; E_{im}^{jp} \;+\; E_{km}^{lp} \leq \text{cutoff value} \\
& w_{ik}^{jl} \;+\; w_{im}^{jp} \;+\; w_{km}^{lp} \;-\; y_i^j \;-\; y_k^l \;-\; y_m^p \;+\; 1 \;\geq\; 0 \\
\forall i<k<m,j,l,p & \text{ s. t. } \; S_{ikm}^{jlp} \;=\; E_{ik}^{jl} \;+\; E_{im}^{jp} \;+\; E_{km}^{lp} \leq \text{cutoff value} \\
& y_i^j \;,\; y_k^l \;=\; 0-1 \;\;\forall\;\; i,j,k,l
\end{array}
\tag{F12}
$$

Hence, in total the computational performance of 12 formulations were tested for *in silico* sequence search. They were used to compute the sequence with the global energy minimum for a set of test problems which are listed in the following section.

6 Comparison of Proposed Formulations

6.1 Human beta defensin 2

The template that was employed in the *in silico* sequence selection for formulation comparison in terms of convergence time is the 3-D structure of human beta defensin 2, or hβD-2, which has a PDB code of $1fd3$ in the Protein Data Bank. hβD-2's structure was elucidated using X-ray crystallography at a resolution of $1.35\mathring{A}$ by Hoover *et al.* (2000).

hβD-2 is a small cationic peptide found in the human immune system. It is crucial to innate immunity (Hoover *et al.*, 2000). It possesses antimicrobial property derived from the electrostatic force between the positive charge on the defensin molecule and the negative charge of the anionic head group of the microbe's membrane lipids. This electrostatic force essentially disrupts the microbe's cell membrane and thus kills the cell (Hoover *et al.*, 2000).

It is desirable to gain knowledge about the structure of the protein to be redesigned so as to develop a better amino acid mutation set for each position. As for the structure of hβD-2, hβD-2 possesses an octameric tertiary structure which is largely determined by its primary structure (Hoover *et al.*, 2001). Its tertiary structure is formed by a mix of hydrophobic and hydrogen bonding between the residues Gly^1, Asp^4, Thr^7, Lys^{10}, Gly^{31}, Leu^{32}, Pro^{33}, and Lys^{39}. The monomer units of hβD-2 are grouped into units of four that are oriented in such a way that their N-termini are in the core of the octamer. The core is sealed off from solvent by hydrogen bonds between Gly^1, Gly^3, Asp^4, and Thr^7. The surface of hβD-2 is mostly amphiphilic.

Although the PDB file for hβD-2 has precise structural information about monomer chains A, B, C, and D, only chain A was redesigned in the test problems for formulation comparison. Chain A is a 41-residue peptide with the following natural sequence: GIGDPVTCLKSGAICHPVFCPRRYKQIGTCGLPGTKCCKKP (García *et al.*, 2001). Like other human β-defensins, it has an N-terminus α-helix located at Pro^5-Lys^{10} which is held against the β-sheet by a S-S bond between Cys^8 and Cys^{37}. Two other S-S bonds that stabilize the β-sheet are located at Cys^{15}-Cys^{30} and Cys^{20}-Cys^{38}. The structural properties of chain A of hβD-2 are summarized in Table (1).

6.2 Test problems

A total of five test problems of *in silico* sequence selection for hβD-2 are chosen for comparing performance of the proposed formulations.

Test problem 1: This test problem is from one of the case studies performed by Rao (2004) on hβD-2 which has the smallest sequence search space of 1.3×10^8 sequences. The mutation set is shown in Table (2). It is derived by eliminating the amino acids that appeared less than 10% of the time in the top 100 minimum energy solutions of a bigger *in silico* sequence selection problem on hβD-2 using formulation (F2) (Rao, 2004).

Test problem 2: In this test problem the glycines at positions 1, 3, 12, 28, 31, and 34, the prolines at position 5, 17, 21, 33, and 41, and the cysteines at positions 8, 15, 20, 30, 37, and 38 in the wild type sequence are fixed. The glycines are fixed because of their characteristic flexibility which is deemed as an important property to maintain in the loops. On the other hand, prolines are on the other extreme as their cyclic nature causes significant steric hindrance. Being highly inflexible, prolines at the native positions are likely to exert great influence on the overall protein structure, and hence they should be fixed too. The cysteines are fixed because disulfide bridges usually play an essential role in maintaining the proper fold. Full combinatorial optimization is allowed on the first ten varied positions (i.e., positions 2, 4, 6, 7, 9, 10, 11, 13, 14, and 16), while the remaining positions are also fixed at their native residues. The sequence search space thus amounts to $20^{10} = 1.0 \times 10^{13}$.

Test problem 3: While keeping the glycines, prolines, and cysteines fixed in the natural sequence, the scope of full combinatorial optimization is expanded from the first ten varied positions to the first fifteen

varied positions (i.e., positions 18, 19, 22, 23, and 24 in addition to the ten varied positions in test problem 2). The remaining positions are also kept at their native residues. The corresponding sequence search space is $20^{15} = 3.3 \times 10^{19}$.

Test problem 4: In this test problem the glycines, prolines, and cysteines in the wild type sequence are fixed, while all the other 24 positions along the chain are allowed to pick any one from the full set of 20 amino acids. This leads to a sequence search space of $20^{24} = 1.7 \times 10^{31}$.

Test problem 5: In this final test problem that has the largest sequence search space, the only positions that are fixed at the native residues are those that have cysteines in the wild type sequence. In addition to the varied positions in test problem 4, positions that have glycines or prolines as their native residues are also included in the mutation set and subject to full combinatorial optimization. The size of the respective sequence search space is $20^{35} = 3.4 \times 10^{45}$.

6.3 Results and discussion

The CPU times required by the 12 proposed formulations to compute the sequence with the global energy minimum for the five different test problems are tabulated in Table (3). Performance of the original formulation proposed by Klepeis *et al.* (2003)(2004) (i.e., formulation (F2)) can be used as the base case for comparison. By comparing the CPU times of the $O(n)$ formulations (F3), (F4), and (F5) with those of formulation (F2), it is apparent that $O(n)$ formulations are inferior to $O(n^2)$ formulation in terms of computational efficiency despite the fact that they have significantly fewer variables and linear constraints. We propose that the superiority of formulation (F2) compared to $O(n)$ formulations is due to the RLT constraints which enhance the branch and bound algorithm, since the CPU time of formulation (F2) without the RLTs (i.e., formulation (F1)) for test problem 2 is actually around 3 orders of magnitude of that for the $O(n)$ formulations, as shown in Table (3). For test problem 3, formulation (F1)) fails to converge, whereas the $O(n)$ formulations converge within reasonable timeframes.

The three new components that are supposed to improve computation: RLT constraints with inequality, triangle inequalities, and preprocessing indicate different degrees of success. By comparing each of formulations (F6), (F10), and (F11) with formulation (F2) for test problems 4 and 5 which are of relatively big size, preprocessing is the most powerful in reducing CPU times, followed by RLTs with inequality and then by triangle inequalities. In fact, for test problem 5 which has the largest sequence search space of $20^{35} = 3.4 \times 10^{45}$, formulation (F11) provides the shortest required computation time among all 12 proposed formulations. It is able to reduce the CPU time for computing the same problem by the original formulation proposed by Klepeis *et al.* (2003)(2004) by 67%.

Combination of two or more of the new algorithmic enhancement factors does not necessarily yield a better CPU time than the use of only one single factor. This can be seen by comparing performances on test problem 5 between formulation (F8), which is original formulation plus RLTs with inequality and preprocessing, and formulation (F6), which is original formulation plus RLTs with inequality only. The same phenomenon is indicated by comparing formulation (F9), which is original formulation plus all three new components, and formulation (F11), which is original formulation plus preprocessing only for test problems 4 and 5.

In addition, it is highly more desirable to apply a small subset using a certain cutoff value than to impose the whole set of triangle inequalities in an attempt to speed up the *in silico* sequence search algorithm. This is observed based on the CPU times for test problems 1, 2, and 3 for formulation (F7) in the cases with no cutoff and with an arbitrary cutoff value of -40. The CPU times in the latter case is an order of magnitude shorter.

7 Conclusions

In this paper, a detailed computational comparison of twelve mathematical formulations for the *in silico* sequence selection problem in de novo protein design is reported. A new improved $O(n^2)$ formulation (F11) for performing the first stage of *in silico* sequence selection in the de novo protein design framework developed by Klepeis *et al.* (2003)(2004) is provided. This novel formulation is the old $O(n^2)$ formulation proposed by Klepeis *et al.* (2003)(2004) plus DEE-type preprocessing, and it is shown that it significantly reduces the required computation time to solve the same quadratic assignment like sequence search problem. For instance, to choose the global energy minimum solution from $20^{35} = 3.4 \times 10^{45}$ sequences, the required CPU time is reduced by 67%. This current best formulation we have proposed is obtained based on comparison between performances of $O(n^2)$ formulation and $O(n)$ formulations, as well as incorporation of different combinations of the algorithmic enhancing components of RLTs with inequality, triangle inequalities, and DEE-type processing.

Acknowledgments

CAF gratefully acknowledges financial support from the National Science Foundation and the National Institutes of Health (R01 GM52032, R01 GM069736).

References

C. Adjiman, I. Androulakis, and C. A. Floudas. A global optimization method, αbb, for general twice-differential constrained npls - i. theoretical advances. *Computers Chem. Engng.*, 22:1137–1158, 1998a.

C. Adjiman, I. Androulakis, and C. A. Floudas. A global optimization method, αbb, for general twice-differentiable constrained nlps - ii. implementation and computational results. *Computers Chem. Engng.*, 22:1159–1179, 1998b.

C. Adjiman, I. Androulakis, and C. A. Floudas. Global optimization of mixed-integer nonlinear problems. *AiChE Journal*, 46:1769–1797, 2000.

CPLEX. *Using the CPLEX Callable Library.* ILOG, Inc., 1997.

C. A. Floudas. *Nonlinear and Mixed-Integer Optimization: Fundamentals and Applications.* Oxford University Press, 1995.

C. A. Floudas. *Deterministic Global Optimization : Theory, Methods and Applications.* Nonconvex Optimization and its Applications. Kluwer Academic Publishers, 2000.

J.R.C. García, J. Florian, S. Schulz, A. Krause, F.J. Rodríguez-Jiménez, U. Forssmann, K. Adermann, E. Klüver, C. Vogelmeier, D. Becker, R. Hedrich, W.G. Forssmann, and R. Bals. Identification of a novel, multifunctional β-defensin (human β-defensin 3) with specific antimicrobial activity. *Cell and Tissue Research*, 306:257–264, 2001.

F. Glover. Improved linear integer programming formulations of nonlinear integer problems. *Management Science*, 22:455–460, 1975.

R.F. Goldstein. Efficient rotamer elimination applied to protein side-chains and related spin glasses. *Biophysics Journal*, 66:1335–1340, 1994.

B.B. Gordon, G.K. Hom, S.L. Mayo, and N.A. Pierce. Exact rotamer optimization for protein design. *J. Computational Chemistry*, 24:232–243, 2003.

D.M. Hoover, K.R. Rajashankar, R. Blumenthal, A. Puri, J.J. Oppenheim, O. Chertov, and J. Lubkowski. The structure of human β-defensin-2 shows evidence of higher order oligomeration. *J. Biol. Chem.*, 275:32911–32918, 2000.

D.M. Hoover, O. Chertov, and J. Lubkowski. The structure of human β-defensin-1. *J. Biol. Chem.*, 276:39021–39026, 2001.

J. L. Klepeis and C. A. Floudas. Free energy calculations for peptides via deterministic global optimization. *J. Chem. Phys.*, 110:7491–7512, 1999.

J. L. Klepeis, C. A. Floudas, D. Morikis, and J. Lambris. Predicting peptide structures using nmr data and deterministic global optimization. *J. Comput. Chem.*, 20:1354–1370, 1999.

J. L. Klepeis, H. D. Schafroth, K. M. Westerberg, and C. A. Floudas. Deterministic global optimization and ab initio approaches for the structure prediction of polypeptides, dynamics of protein folding and protein-protein interaction. In R. A. Friesner, editor, *Advances in Chemical Physics*, volume 120, pages 254–457. Wiley, 2002.

J.L. Klepeis, C.A. Floudas, D. Morikis, C.G. Tsokos, E. Argyropoulos, L. Spruce, and J.D. Lambris. Integrated computational and experimental approach for lead optimization and design of compstatin variants with improved activity. *J. Am. Chem. Soc.*, 125:8422–8423, 2003.

J.L. Klepeis, C.A. Floudas, D. Morikis, C.G. Tsokos, and J.D. Lambris. Design of peptide analogs with improved activity using a novel de novo protein design approach. *Industrial & Engineering Chemistry Research*, 43:3817, 2004.

T. Kortemme and D. Baker. Computational design of protein-protein interactions. *Current Opinion in Chemical Biology*, 8:91–97, 2004.

C.M. Kraemer-Pecore, A.M. Wollacott, and J.R. Desjarlais. Computational protein design. *Current Opinion in Chemical Biology.*, 5:690–695, 2001.

B. Kuhlman and D. Baker. Exploring folding free energy landscapes using computational protein design. *Current Opinion in Structural Biology.*, 14:89–95, 2004.

C. Loose, J. Klepeis, and C. Floudas. A new pairwise folding potential based on improved decoy generation and side chain packing. *Proteins*, 54:303–314, 2004.

S.M. Malakauskas and S.L. Mayo. Design, structure, and stability of a hyperthermophilic protein variant. *Nat. Struct. Biol.*, 5:470–475, 1998.

M. Oral and O. Kettani. A linearization procedure for quadratic and cubic mixed-integer problems. *Operations Research*, 40:S109–S116, 1990.

M. Oral and O. Kettani. Reformulating nonlinear combinatorial optimization problems for higher computational efficiency. *European Journal of Operational Research*, 58:236–249, 1992.

P.M. Pardalos and S. Jha. Complexity of uniqueness and local search in quadratic 0-1 programming. *Operations Research Letters*, 11:119–123, 1992.

P.M. Pardalos, H.X. Huang, and O. Prokopyev. Multi-quadratic binary programming. University of Florida, Research Report, 2004.

N.A. Pierce and E. Winfree. Protein design is np-hard. *Protein Engineering.*, 15:779–782, 2002.

N.L. Pierce, J.A. Spriet, J. Desmet, and S.L. Mayo. Conformational splitting: A more powerful criterion for dead-end elimination. *J. Computational Chemistry*, 21:999–1009, 2000.

S. Rao. A novel *In Silico* sequence selection approach to the identidication of hβd-2 analogs with improved specificity. Princeton University, Department of Chemical Engineering, Senior Thesis., 2004.

F.M. Richards and H.W. Hellinga. Construction of new ligand binding sites in proteins of known structure. i. computer-aided modeling of sites with pre-defined geometry. *J. Mol. Biol.*, 222:763–785, 1991.

F.M. Richards, J.P. Caradonna, and H.W. Hellinga. Construction of new ligand binding sites in proteins of known structure. ii. grafting of a buried transition metal binding site into *Escherichia coli* thioredoxin. *J. Mol. Biol.*, 222:787–803, 1991.

H.D. Sherali and W.P. Adams. *A reformulation linearization technique for solving discrete and continuous nonconvex problems*. Kluwer Academic Publishing, Boston, MA, 1999.

M. Shimaoka, J.M. Shifman, H. Jing, L. Takagi, S.L. Mayo, and T.A. Springer. Computational design of an intergrin i domain stabilized in the open high affinity conformation. *Nat. Struct. Biol.*, 7:674–678, 2000.

D. Tobi and R. Elber. Distance-dependent pair potential for protein folding: results from linear optimization. *Proteins*, 41:40–46, 2000.

D. Tobi, G. Shafran, N. Linial, and R. Elber. On the design and analysis of protein folding potentials. *Proteins*, 40:71–85, 2000.

C.A. Voigt, D.B. Gordon, and S.L. Mayo. Trading accuracy for speed: a quantitative comparison of search algorithms in protein sequence design. *J. Mol. Biol.*, 299:789–803, 2000.

Structural Features	Positions
β strands	14 - 16 25 - 28 36 - 39
α helix	5 - 10
S-S bonds	8 - 37 15 - 30 20 - 38
β turns	16 - 19 21 - 24 32 - 35
Hairpins	25 - 29
Bulges	27, 28, 37

Table 1: Structural features of human beta defensin 2.

Position	Amino acids allowed	Position	Amino acids allowed
1	Gly	22	Arg,Asn
2	Gln,Leu,Ser,Val	23	Phe,His,Asn
3	Gly	24	Phe,Met,Arg,Thr
4	Gln,Asn,Lys,Ser	25	Phe,Ile
5	Pro	26	Phe,Thr
6	Arg,Asn,Lys	27	Arg,Gln,Ile,Ser
7	Asn,His,Ile,Thr	28	Gly
8	Cys	29	Gln,Met
9	Asn	30	Cys
10	His,Lys,Ser	31	Gly
11	Arg,Trp,Met	32	His,Ser
12	Gly	33	Pro
13	Tyr	34	Gly
14	Tyr,Lys	35	Ala,Thr
15	Cys	36	Tyr
16	Tyr	37	Cys
17	Pro	38	Cys
18	Arg,Gly,His,Thr	39	Ala
19	Arg, Phe, Ala	40	Met
20	Cys	41	Pro
21	Pro		

Table 2: Mutation set for test problem 1.

Table 3: Comparison of CPU times in seconds to obtain one global energy minimum solution among the proposed formulations. Solutions were obtained with CPLEX 8.0 solver enabled with branch and bound algorithm on a single Intel Pentium IV 3.2GHz processor.

Test problem	Sequence search space	Formulations						
		(F1)[a]	(F2)[b]	(F3)[c]	(F4)[d]	(F5)[e]	(F6)[f]	(F7)[g]
1	1.3×10^8	0.30	0.14	0.05	0.04	0.05	0.15	0.23*, 0.21⋆
2	1.0×10^{13}	34874	1.93	12.80	65.04	13.23	2.16	44.02*, 3.01⋆
3	3.3×10^{19}	70.14% gap†	3.01	137.85	2052.2	278.0	3.22	64.39*, 2.87⋆
4	1.7×10^{31}	-	38.14	-	-	-	31.67	- , 29.06⋆
5	3.4×10^{45}	-	74713	-	-	-	30006	- , 65575⋆

Test problem	Sequence search space	Formulations				
		(F8)[h]	(F9)[i] cutoff for tri. ineq.=-40	(F10)[j] cutoff for tri. ineq.=-40	(F11)[k]	(F12)[l] cutoff for tri. ineq.=-40
1	1.3×10^8	0.16	0.11	0.16	0.17	0.11
2	1.0×10^{13}	2.15	2.26	2.01	2.52	2.10
3	3.3×10^{19}	2.94	3.31	3.03	3.43	3.04
4	1.7×10^{31}	31.08	35.48	35.92	25.00	36.15
5	3.4×10^{45}	32657	52276	61872	24388	57569

[a] Original $O(n^2)$ formulation proposed by Klepeis *et al.* (2003)(2004) without RLT constraints.
[b] Base case: original $O(n^2)$ formulation proposed by Klepeis *et al.* (2003)(2004).
[c,d,e] $O(n)$ formulations.
[f] Original $O(n^2)$ formulation with inequality RLT constraints.
[g] Original $O(n^2)$ formulation with inequality RLT constraints and triangle inequalities.
[h] Original $O(n^2)$ formulation with inequality RLT constraints and preprocessing.
[i] Original $O(n^2)$ formulation with inequality RLT constraints and triangle inequalities and preprocessing.
[j] Original $O(n^2)$ formulation with triangle inequalities.
[k] Original $O(n^2)$ formulation with preprocessing.
[l] Original $O(n^2)$ formulation with triangle inequalities and preprocessing.
† Integrality gap obtained after $100,000$ sec. CPU time.
* No cutoff for triangle inequalities.
⋆ Cutoff $= -40$ for triangle inequalities.