JOURNAL OF COMPUTATIONAL BIOLOGY
Volume 8, Number 5, 2001
Mary Ann Liebert, Inc.
Pp. 523–547

Constrained Global Optimization for Estimating Molecular Structure from Atomic Distances

GLENN A. WILLIAMS, JONATHAN M. DUGAN, and RUSS B. ALTMAN

ABSTRACT

Finding optimal three-dimensional molecular configurations based on a limited amount of experimental and/or theoretical data requires efficient nonlinear optimization algorithms. Optimization methods must be able to find atomic configurations that are close to the absolute, or global, minimum error and also satisfy known physical constraints such as minimum separation distances between atoms (based on van der Waals interactions). The most difficult obstacles in these types of problems are that 1) using a limited amount of input data leads to many possible local optima and 2) introducing physical constraints, such as minimum separation distances, helps to limit the search space but often makes convergence to a global minimum more difficult. We introduce a constrained global optimization algorithm that is robust and efficient in yielding near-optimal three-dimensional configurations that are guaranteed to satisfy known separation constraints. The algorithm uses an atom-based approach that reduces the dimensionality and allows for tractable enforcement of constraints while maintaining good global convergence properties. We evaluate the new optimization algorithm using synthetic data from the yeast phenylalanine tRNA and several proteins, all with known crystal structure taken from the Protein Data Bank. We compare the results to commonly applied optimization methods, such as distance geometry, simulated annealing, continuation, and smoothing. We show that compared to other optimization approaches, our algorithm is able combine sparse input data with physical constraints in an efficient manner to yield structures with lower root mean squared deviation.

Key words: constrained global optimization, protein, RNA, 3D molecular structure, mathematical modeling.

1. INTRODUCTION

DEVELOPING METHODS THAT WILL YIELD accurate estimates of the three-dimensional structure of biological molecules—such as proteins, nucleic acids, or combinations of both—based on experimental data, sequence data, and theoretical considerations (e.g., secondary structure prediction) is one of the major goals of computational biology and an important part of structural genomics. Experimental methods such as NMR and X-ray crystallography are expensive and time consuming and sometimes provide only partial information. Thus, computational methods for molecular structure estimation can serve as an

Stanford Medical Informatics, Stanford University, Stanford, CA, 94305-5479.

important complement to these costly experimental methods. Ideally, computational methods will take a sparse amount of data from various sources and produce an estimate of the optimal three-dimensional configuration of atoms in the molecule.

The goal of the work presented here is to develop a set of optimization methods and algorithms that will provide improved performance in estimating biomolecular structure from sparse input data. The specific objectives of this work are 1) to develop an efficient constrained global optimization algorithm that will take input data regarding the three-dimensional configuration of atoms in a molecule, add known physical information to serve as constraints in the optimization, and efficiently yield accurate estimates of the optimal values of the x, y, and z coordinates of each atom in the molecule; 2) to evaluate and compare this algorithm to the local and global optimization methods currently used in molecular structure estimation; and 3) to apply and test this algorithm to a wide variety of molecular structures.

The method we introduce is novel in two ways. First, it is designed never to report structures that violate minimum separation distances derived from van der Waals interactions. Second, it is based on a nonstochastic global optimization algorithm that is still able to avoid local minima, in a manner similar to stochastic Monte Carlo methods.

Given the limited amount of available molecular structure data, it is essential to develop computational methods that can complement this data with inviolable physical constraints to efficiently produce optimal three-dimensional atomic configurations. At the core of such computational methods is the need for an effective nonlinear optimization algorithm. Attempting to find optimal x, y, and z coordinates for each atom in a molecule requires optimizing for $3n$ degrees of freedom, where n is the number of atoms in the molecule. When modeling large molecules or ensembles of molecules, the number of atoms and thus the degrees of freedom can become very large and lead to significant optimization challenges. The variables in these optimization problems are generally nonlinear since most of the input data are nonlinear functions of the x, y, and z atomic coordinates.

In molecular structure estimation, the goal of the optimization procedure is to find the minimum value of a given objective function, which is defined in terms of differences between the input data and the corresponding information calculated from the estimated structure. The iterative minimization process involves calculating the first derivatives, and possibly the actual or approximated second derivatives, of the objective function at the current state and using that information to adjust the coordinates of the atoms to find a lower value of the objective function. The most commonly applied minimization techniques are steepest descent (SD), conjugate gradient (CG), and Newton's methods.

SD methods are the simplest to implement but convergence is often slow (Press *et al*., 1992). SD involves repeatedly following the direction given by the negative gradient of the objective function to a new local minimum. CG methods are motivated by the desire to accelerate the slow convergence associated with SD. CG makes use of the previous history of minimization steps as well as the current gradient to ensure that move directions are conjugate to all previous directions traversed (Hestenes, 1980). This approach significantly accelerates convergence compared to the SD method (Luenberger, 1989).

Newton's method operates on local quadratic approximations to the objective function. The essential idea is that the quadratic approximation, unlike the objective function itself, is trivial to minimize. The quadratic approximation is updated and minimized during each iteration of the optimization routine. The routine seeks to find a local minimum of the objective function by repeating this process. The degree to which the objective function actually assumes a quadratic shape is indicative of how successful the results will be.

Newton's method requires the use of not only first derivative information but also a second derivative, or Hessian, matrix. Analytic second derivatives may not be available or may be too costly to evaluate. A difference approximation to the Hessian could be used but the cost of the necessary function evaluations may also be very high. The prohibitive costs, combined with the fact that the performance of Newton's method can degrade if positive definiteness of the Hessian matrix is not maintained (which often occurs), provide the motivation for the quasi-Newton (QN) approach (Luenberger, 1989; Dennis and Schnabel, 1996).

QN methods use approximations to the Hessian matrix that 1) are less costly to construct because they are defined in terms of first-derivative information and 2) perform better because they preserve symmetry and, in most cases, positive definiteness. There are several variations of the QN method, including the Gauss-Newton (GN), symmetric rank one (SR1), and Broyden–Fletcher–Goldfarb–Shanno (BFGS) (Broyden, 1970; Fletcher, 1970; Goldfarb, 1970; Shanno, 1970) algorithms. GN is used to solve nonlinear least squares problems. GN generally produces symmetric, positive definite matrices (Dennis and Schnabel,

1996). The BFGS and SR1 are forms of secant approximations for H and can be used for any general function. The BFGS approach produces symmetric and positive definite H matrices, whereas the SR1 produces H matrices that are symmetric but not necessarily positive definite. Implementational details of the GN, BFGS, and SR1 methods can be found in Dennis and Schnabel (1996).

Many molecular structure estimation codes use SD or CG in the energy minimization routines to avoid the computational requirements associated with Newton's or QN methods, which include the storage and evaluation of the Hessian or approximate Hessian matrix and the solution of a system of linear equations at each update step. YAMMP (Tan and Harvey, 1993) is an example of a code that uses only SD and CG minimization methods. The Modeller code (Sali and Blundell, 1993) uses a variable target function method (Braun and Go, 1985) employing a conjugate gradient-based minimization approach. Newton-type or QN methods have been implemented in molecular modeling algorithms (Head and Zerner, 1985, 1989). The CHARMM molecular modeling package offers SD, CG, and Newton's methods (Brooks *et al*., 1983). The ROAR package (Cheng *et al*., 1999)—used in the AMBER code (Pearlman *et al*., 1995)—offers SD, CG, and a limited memory BFGS approach. The Proteinmorphosis molecular modeling system uses QN methods to minimize potential energy functions in simulating conformational changes of proteins and protein complexes (Meiyappan *et al*., 1999).

SD, CG, and QN methods perform best when a quadratic approximation of the objective function is a relatively good one. Unfortunately, for many of the nonlinear problems encountered in molecular structure estimation, the quadratic approximation is not good enough and the update produced by SD, CG, or QN is unsatisfactory; i.e., the value of the objective function at the updated position is greater than that at the initial position. In this case, one approach for proceeding towards a more favorable position is to perform a line search. The idea of a line-search algorithm is to try the full SD, CG, or QN step first and if that fails, backtrack in a systematic way along the direction defined by that step until an acceptable step length is found.

Common among all local optimization methods is the propensity to become trapped in local minima of the objective function. A local minimum is a state where the gradient is very small, thus indicating to the local minimization routine that an acceptable state has been reached and terminating the minimization procedure. When building a three-dimensional model of a molecule from a sparse amount of data, many low-error conformations exist, each serving as a potential local minimum trap. In order to keep the optimization from becoming stalled in a local minimum state, the local minimization algorithm must be augmented with a technique to force the state out of the resulting local minima and towards a global minimum. Methods for finding the global minimum state are referred to as global optimization methods. There are many different global search methods that have been examined, including simulated annealing (Kirkpatrick *et al*., 1983; Braun and Go, 1985; Ingber, 1989; Wilson and Cui, 1990; Sali and Blundell, 1993; Pearlman *et al*., 1995; Güntert *et al*., 1997), continuation and smoothing (Moré and Wu, 1996, 1997), Monte Carlo (Li and Scheraga, 1987; Chang *et al*., 1989; Ripoll and Thomas, 1990; Shakhnovich *et al*., 1991; Covell, 1992; O'Toole and Panagiotopoulos, 1992; Kolinski and Skolnick, 1994), convex global underestimation (Dill *et al*., 1997), diffusion equation method (Kostrowicki and Piela, 1991; Kostrowicki and Scheraga, 1992), and genetic algorithms (Sun, 1993; Unger and Moult, 1993; Herrmann and Suhai, 1995).

Global optimization methods attempt to perturb the state of the system when trapped in local minima and thereby continue to search for the globally minimum state. These perturbations are usually random in nature and are often performed at prespecified intervals during the optimization process. Although this produces the desired result of moving out of a local minimum and is very effective in searching a wide range of the possible configuration space, it often results in unfavorable movements. That is, perturbations may occur at times when the optimization is moving towards the global minimum, and the random perturbation directions may reconfigure the state variables into difficult starting positions for the subsequent minimization procedure. A domain-specific, nonrandom perturbation method may alleviate these difficulties, yet it would have to be developed in such a way that is searches as much of the configuration space as a random perturbation method.

The ability to move towards a global minimum in a computationally efficient manner during molecular structure computations is further complicated by the addition of physical constraints on atom locations, such as chirality or minimum separation distances between pairs of atoms or groups of atoms. MSD constraints between atoms can be derived from van der Waals theory (Halgren, 1992), Lennard-Jones potentials (Lennard-Jones, 1924), Buckingham potentials (Buckingham *et al*., 1988), and Morse potentials

(Dinur and Hagler, 1991). MSD constraints in molecular structure estimation pose a significant global optimization challenge.

Codes currently used for molecular structure estimation (Sali and Blundell, 1993; Tan and Harvey, 1993; Pearlman *et al*., 1995; Moré and Wu, 1997; Macke and Case, 1998) often have difficulty in maintaining acceptable convergence rates while trying to satisfy MSD constraints. The standard procedure is to add some sort of penalty function to the objective function in order to keep atoms from moving too close to each other. This approach is usually ineffective as it establishes more barriers to the global minimum and makes it more difficult to move out of local minima. Turning MSD constraints on and off during the course of the optimization process is another approach but may result in alternating between low-error, MSD illegal structures and high-error, MSD legal structures, without being able to find the desired low-error, MSD legal structures.

Optimization methods that simultaneously update the positions of all the atoms in a molecule, or some group of atoms, will have difficulty in attempting to simultaneously enforce MSD constraints. Strict MSD constraint enforcement is computationally challenging and becomes intractable when moving a large number of atoms simultaneously during the course of the minimization process. Many of the proteins that we are interested in modeling are on the order of hundreds or thousands of atoms and will require novel approaches for MSD constraint satisfaction.

The demand for effective MSD constraint satisfaction methods is also motivated by the fact that MSD information is freely available from van der Waals theory and the Lennard-Jones, Buckingham, or Morse potentials and can be applied to atoms in any molecule. MSD constraints serve as a valuable complement to sparse molecular structure input data. We have shown previously that the distribution of interatomic distances derived from MSD constraints is a valuable structural constraint (Chen *et al*., 1996). If a relatively small amount of data is used as input to the optimization, satisfying all of the data may not be sufficient to find good quality structures. In these cases, more information is needed and can be found simply by adding physical constraints such as MSD. This added information will reduce the number of allowable conformations and will result in improved resolution of those structures that satisfy the input data.

2. METHODS

The following sections outline our nonlinear optimization approach for estimating biomolecular structure. These methods take sparse input data and produce accurate estimates of three-dimensional molecular structure that are guaranteed to satisfy MSD constraints, while also maintaining good convergence properties.

2.1. Formulation

The first step in the molecular structure estimation process is to define an objective function to serve as a basis for developing the search strategy. We define the objective function to be the sum of squares of the weighted residuals. A residual is defined as the difference between a piece of input data and the corresponding calculated value from the current model. Weights are assigned to each piece of input data according to its given variance. The variance corresponds to the potential inherent error associated with a piece of input data and serves as a measure of the reliability of that data. In this work, we will be considering only distance input data, where each distance represents the pairwise distance between any two atoms in the molecule. In this case, the objective function, f, is given by

$$f = \frac{1}{2}\sum_{i=1}^{m} R_i^2 \tag{1}$$

where m is the total number of input distances and R_i is the weighted residual of the i-th input distance. The weighted residuals are defined as

$$R_i = \frac{d_i - d_c}{\sigma_i} \tag{2}$$

where d_i is the i-th input distance, d_c is the calculated distance corresponding to the i-th input distance, and σ_i is the standard deviation of the i-th input distance.

2.2. Local optimization

The goal of the local optimization algorithm is to update the atomic coordinates to a position that will minimize Equation (1). The update of coordinates from an old position to a new position is represented mathematically as

$$\begin{bmatrix} x_1 \\ y_1 \\ z_1 \\ \vdots \\ x_n \\ y_n \\ z_n \end{bmatrix}_{new} = \begin{bmatrix} x_1 \\ y_1 \\ z_1 \\ \vdots \\ x_n \\ y_n \\ z_n \end{bmatrix}_{old} + \alpha \begin{bmatrix} \Delta x_1 \\ \Delta y_1 \\ \Delta z_1 \\ \vdots \\ \Delta x_n \\ \Delta y_n \\ \Delta z_n \end{bmatrix} \tag{3}$$

where x_i, y_i, and z_i are the x, y, and z coordinates, respectively, of atom i, and α is the step length to move in the direction of the Δ vector. To simplify the notation, Equation (3) can be expressed as

$$\vec{x}_{new} = \vec{x}_{old} + \alpha \Delta \vec{x} \tag{4}$$

where $\vec{x}$ now refers to a state vector consisting of all the atomic coordinates.

There are several different methods for finding $\Delta \vec{x}$ and thus updating the independent variables $\vec{x}$ in Equation (4). We implement several update techniques, including the Polak–Ribiere variant of CG (Polak, 1971) and three QN methods—the GN, SR1, and BFGS methods (Dennis and Schnabel, 1996).

The QN update is given by

$$\Delta \vec{x} = -[\hat{H}(\vec{x}_{old})]^{-1} \nabla f(\vec{x}_{old}) \tag{5}$$

where $\hat{H}$ is an approximation to the Hessian matrix of f and ∇f is the gradient of f with respect to $\vec{x}$. $\hat{H}$ is a $3n \times 3n$ matrix and ∇f is a $3n \times 1$ vector, where n is the number of atoms in the molecule.

Instead of inverting $\hat{H}$ in Equation (5), the more common and computationally efficient approach is to solve the following equivalent system of linear equations for $\Delta \vec{x}$:

$$[\hat{H}(\vec{x}_{old})] \Delta \vec{x} = -\nabla f(\vec{x}_{old}). \tag{6}$$

We enhance the local minimization procedure with a quadratic/cubic backtracking line search (Dennis and Schnabel, 1996). The line search will check to see if using $\alpha = 1$ (full CG or QN step) in Equation (4) yields an acceptable function value; i.e., $f(\vec{x}_{new}) < f(\vec{x}_{old})$ (minus some small correction factor). If the resulting function value is not acceptable, then a new value for α is found by approximating f along the search direction, $\Delta \vec{x}$, with a quadratic or cubic function, and calculating the value of α that minimizes the approximating function. This procedure is repeated until either 1) an α is found that yields an acceptable function value or 2) the number of searches exceeds some prespecified limit.

2.3. Atom-based optimization

Instead of simultaneously updating the positions of all the atoms in the molecule, or some group of atoms, we only update the coordinated locations of one atom at a time. This atom-based approach reduces the dimensionality of the system from $3n$ to 3 and makes MSD constraint enforcement more tractable. The cost is that instead of updating all atomic coordinates at once, the algorithm has to cycle through all the single-atom updates. But the advantage is that the algorithm is guaranteed to produce configurations that satisfy all physical constraints on atom locations after the update of all the atoms within a cycle.

Our algorithm is based on a multilevel optimization process. At the highest level, we attempt to find optimal configurations of subsets of all the atoms in the molecule. We will refer to this level as a *group*. We build groups based on molecular sequence information. For example, in estimating optimal protein

C_α backbone structures, we start by optimizing the C_α atoms from the first two amino acid residues. Subsequent optimization groupings (from 3 atoms up to n atoms) are formed by simply adding the C_α atom from the next amino acid residue in the sequence.

Within each of the optimization groups, we perform a series of *cycles*. The total number of cycles is determined by a user-specified limit or termination criteria. Each cycle consists of a series of *atom moves*, one for each atom in the current group. The order of atoms to move is based on each atom's attributed error, and is determined at the beginning of a cycle. Atoms are moved in decreasing order of attributed error. Moving an atom consists of a series of *iterations*. In each iteration, the movement of the atom is determined by the local optimization procedures described above, i.e., CG, GN, SR1, or BFGS, to determine step direction and line search to determine step length. The iterations continue until some stopping criteria is met.

For each atom a in the molecule, the iterations are based on a local objective function, f_a, given by

$$f_a = \frac{1}{2}\sum_{i=1}^{m_a} R_i^2 \tag{7}$$

where m_a is the total number of input distances associated with atom a. The three-dimensional update equation is then given by

$$\vec{x^a}_{new} = \vec{x^a}_{old} + \alpha \Delta \vec{x^a} \tag{8}$$

where $\vec{x^a}$ now refers to the state vector consisting of the x, y, and z coordinates of atom a. Equation (8) is simply the reduced dimension form of Equation (4). The linear system of equations in the QN update, Equation (6), is also reduced from dimension $3n$ to 3 and is given by

$$[\hat{H}_a(\vec{x^a}_{old})]\Delta \vec{x^a} = -\nabla f_a(\vec{x^a}_{old}) \tag{9}$$

where $\hat{H}_a$ is an approximation to the Hessian matrix of f_a and ∇f_a is the gradient of f_a with respect to $\vec{x^a}$; $\hat{H}_a$ is a 3×3 matrix, and ∇f_a is a 3×1 vector.

2.4. MSD constraint enforcement

One of the major goals of this work is to develop an optimization algorithm that can satisfy input data and physical constraints, particularly MSD constraints, in an efficient manner. The method we have developed for enforcing MSD constraints on atom locations is based on the line search method described in Section 2.2

Because the optimization algorithm operates on one atom at a time, we can restrict the movement of any atom to locations that do not violate MSD constraints. This is done by treating the atom being moved as a point in space, placing spheres around all the other atoms in the molecule, and not allowing the atom being moved to enter into any of the MSD spheres surrounding the other atoms. Restricting atoms to MSD-legal space is achieved computationally in the line search procedure. After a search direction, $\Delta \vec{x_a}$, is calculated, we can determine which segments along that line intersect the spheres surrounding all other atoms. If the step length, α, resulting from the line search moves an atom into a violating segment, the position of the atom is moved to that segment's endpoint with the lowest function value. The violating line segments could be the result of the MSD sphere from a single atom, or the overlap of MSD spheres from several atoms.

By constraining the line search in this way, all atoms can be guaranteed to satisfy MSD constraints after each iteration. But imposing constraints after each iteration is too restrictive and results in atoms becoming stuck and not being able to move through other atoms to find more optimal configurations. Therefore, we allow atoms to move freely, unconstrained by MSD forces, up until the final iteration. After the step direction is determined on the final iteration, we impose the constraints on the line search as described above. This allows for atoms to sample a larger portion of the configuration space during the search, yet their final placement is guaranteed to be in locations that do not violate MSD constraints.

Constraining atoms to move into MSD-legal spaces relative to all other atoms can also be to restrictive. Strict MSD constraint enforcement need only be imposed on those atoms that have already been moved

during the current cycle, since there will be no opportunity subsequent to the atom move to enforce the constraint between those atoms and the atom being moved. But MSD constraint enforcement need not be imposed on atoms that have not been moved yet in the current cycle because those constraints can be satisfied during the movement of those atoms. As long as we terminate the optimization at the end of a cycle and do not allow termination within a cycle, this MSD constraint checking approach is guaranteed to produce structures that satisfy al MSD constraints.

We define a "real" MSD radius, r_u, to be that which is derived from the true MSD constraints; r_u is used as the radii for those atoms where strict MSD constraint enforcement is necessary. The value of r_u is set by the user based on physical knowledge of how close an atom can be to other atoms in the molecule.

For the case where strict MSD constraint enforcement is not necessary, we are free to define the sphere radii arbitrarily. This "artificial" MSD radius, r_v, is defined to vary in magnitude between 0 and the "real" MSD sphere radius, r_u. The value of r_v is set by the following empirically discovered functions:

if an input distance exists between the atom being moved and the atom being checked

$$r_v = \left| \frac{R_i}{d_i} \right| * r_u \tag{10}$$

else

$$r_v = \sqrt{f_m + f_c} \tag{11}$$

where R_i is the weighted residual of the input distance between the atom being moved and the atom being checked, d_i is the input distance between the atom being moved and the atom being checked, f_m is the value of the local objective function of the atom being moved, and f_c is the value of the local objective function of the atom begin checked. Additionally, a maximum value of r_u is imposed; i.e., if r_v calculated from Equations (10) or (11) above is greater than r_u, then r_v is set to be equal to r_u.

2.5. Local perturbation

In addition to MSD spheres, we add a local perturbation (LP) sphere to further improve global convergence properties. This is done by defining a sphere around the atom being moved, centered at its position at the start of the iterations. The LP sphere is added to the MSD spheres as constraints on the resulting position of the atom being moved. The radius, r_m, of the LP sphere around an atom begin moved is defined to be proportional to the atom's attributed error. If the atom has a relatively high error, a relatively large LP sphere will be used; and as the atom's error decreases, the size of the LP sphere will decrease.

Therefore, if the atom has a relatively high error and has not moved beyond r_m from its initial position, then it is perturbed by not allowing it to remain within the sphere of radius r_m centered at its initial position. The criteria imposed for determining whether to perturb an atom is precisely the same criteria for indication of an unacceptable local minimum; i.e., the atom has relatively high error and has not moved very far from its initial position. This procedure will not perform unnecessary perturbations on atoms that are making large movements, regardless of their error; and it will not perform unnecessary perturbations of atoms that have relatively low error, regardless of how small their movements may be.

The decaying nature of the LP radii is analogous to the decaying temperature in a simulated annealing schedule. Yet, in our method the perturbations are nonrandom and domain specific. Also, because these perturbations occur continuously and not at prespecified intervals as in simulated annealing, they are less likely to jeopardize a potentially favorable atomic configuration. The exact function that we found empirically to be most effective for defining local perturbations as a function of attributed error is

$$r_m = \sqrt{0.5 * f_m} \tag{12}$$

where f_m is the value at the end of the iterations of the local objective function of the atom being moved, as defined by Equation (7). In this case, the local objective function serves as the attributed error for the atom. A maximum value is set for r_m equal to the maximum weighted residual from the previous cycle.

2.6. Constrained optimization algorithm

We have implemented the ideas described in Section 2 in a computer program called GNOMAD. The following pseudocode outlines the constrained, global optimization GNOMAD algorithm.

```
for natoms = 2 → total # of atoms in molecule {GROUPS}
  determine starting positions for atoms 1 → natoms
  for cycle = 1 → # of cycles {CYCLES}
    determine order of atoms to move, based on attributed error
      (move atoms in decreasing order or error)
    for a = 1 → natoms (in order determined above) {ATOM MOVES}
      perform nonlinear iterations to move atom a {ITERATIONS}
        determine move direction using BFGS quasi-Newton minimization
        determine move length using quadratic/cubic backtracking line search
      perform MSD constraint checks
      perform local perturbation checks
      merge violating segments along search line into nonoverlapping segments
      if necessary, perturb final position of atom a out of violating segment
```

The GNOMAD algorithm is very general and can be applied to any constrained optimization problem to search for a global minimum although in this work we only test its effectiveness in solving molecular structure estimation problems.

2.7. Initial positions

In any large-scale, nonlinear optimization, starting values are a very important factor in successful convergence. In GNOMAD, the initial position of each atom within a *group* is based on information derived from the previous *group*'s optimization. Specifically, the initial positions of atoms $1 \rightarrow (natoms - 1)$ within the *group* $1 \rightarrow natoms$ are simply the final positions resulting from the preceding optimization of the *group* $1 \rightarrow (natoms - 1)$. The selection of the position of atom *natoms* within the *group* $1 \rightarrow natoms$ is somewhat more arbitrary. In GNOMAD, the initial position of atom *natoms* is determined in one of two ways, depending on what stage the optimization procedure is in.

During the first pass through the entire optimization algorithm described above, atom *natoms* is initially placed at a constant and preset distance and direction from the initial position of atom $(natoms - 1)$ within the same optimization *group*. At the completion of the above algorithm, the final locations of all atoms are stored and the optimization is restarted. During the first *restart* (second pass of the optimization algorithm), the *natoms* atoms are placed initially at their stored locations from the first pass, regardless of the initial positions of all other atoms within the *group*. Multiple restarts can be made, but our experience shows that solutions do not improve significantly beyond the first restart.

3. TEST COMPUTATIONS

3.1. Comparison procedure

To evaluate our algorithm and compare it to other molecular modeling codes, we performed a series of optimization experiments. The data used in these experiments were derived from molecules with known three-dimensional structure. These "crystal" structures were taken from the Protein Data Bank (PDB) (Berman *et al*., 2000). The goal of the experiments was to recreate the known crystal structure using various subsets of all the distances between pairs of atoms in the molecule. We first made performance comparisons among the various local optimization methods implemented in our algorithm. Based on these comparisons, a preferred local optimization method was chosen. We then compared our constrained global optimization algorithm, using the best local optimization technique, against several other molecular modeling codes.

An optimization experiment consisted of first generating data sets containing 20%, 30%, . . . ,70% of all possible "short-range distances" (SRD) between atoms in the molecule. The definition of SRD is

somewhat arbitrary but is based on the range of distances that might be expected as realistic input, which can be derived from knowledge of biochemical theory or experimental measuring techniques such as NMR. For each of the input data sets, we estimated the optimal three-dimensional molecular structure using the various optimization methods. The resulting structures were then analyzed in terms of 1) root mean squared deviation (RMSD) from the crystal structure and 2) satisfaction of input data. Satisfaction of input data is measured in terms of maximum distance residual, where residual is defined as in Equation (2).

The first set of experiments was made using only a "backbone" structure of the molecule. This backbone structure consisted only of C_a atoms for proteins and phosphate atoms for RNA. SRD data sets for the backbone structure experiments were generated so that all "i to $i+1$" distances along the backbone chain were included. As a further test of robustness, we performed optimizations on two molecules using all the nonhydrogen atoms in the molecule. In these full-atomic estimates, we used 30% and 70% SRD data sets, which were constructed to include all distances between atoms that are covalently bonded and all distances between atoms that share covalent bonds with the same atom (constant bond angle assumption).

3.2. Test structures

We used four different test structures in our optimization experiments, three of which are proteins and one is a tRNA. The first test structure is the L7/L12 50S ribosomal protein (C-terminus domain) from *E. coli*, which has the PDB ID 1ctf. This is a 68-residue protein, containing three helices and one beta sheet comprised of three strands. The second test structure is the yeast phenylalanine tRNA, which has the PDB ID 1tra. The phosphate backbone contains 76 atoms which form three double-helices and three loops. The third test structure is the triose phosphate isomerase, or tim barrel as it is commonly known, which has the PDB ID 1tim. This is a 247-residue protein, containing 24 helices and two beta sheets comprised of nine strands each. The fourth test structure is the periplasmic nitrate reductase from desulfovibrio desulfuricans, which has the PDB ID 2nap. This is a single-chain, 720-residue protein, which is one of the largest single-chain entries in the PDB. This protein has 31 helices and six beta sheets comprised of three, five, five, two, two, and seven strands each. Table 1 shows the number of atoms and SRD for the backbone estimates and full-atomic estimates we performed on the various test structures.

3.3. Local optimization comparison

The first set of experiments was designed to compare the performance of the various local optimization techniques that can be used within the GNOMAD algorithm. The methods we tested were the Polak-Ribiere variant of CG, and the GN, SR1, and BFGS variants of the QN method. We performed backbone structure estimations on all four test structures, each with 20%, 30%, ..., 70% of SRD as input, using all of the local optimization methods. Selection of a preferred local optimization method was made based on comparisons of maximum residuals and RMSD from crystal structure for each of the resulting structures.

3.4. Global optimization comparison

We chose three molecular modeling codes to use for comparisons to our algorithm—PROTEAN (Altman, 1995; Chen *et al*., 1998), DGSOL (Moré and Wu, 1997), and NAB (Nucleic Acid Builder) (Macke and Case, 1998). Although there are other codes that could have been used for comparison, our choice was

TABLE 1. TEST STRUCTURES

	Structure (PDB ID)			
	1ctf	*1tra*	*1tim*	*2nap*
# of backbone atoms	68	76	247	720
Definition of "short-range distance" for backbone model	10 Å	25 Å	10 Å	10 Å
# of short-range distances for backbone models	557	1213	2225	7497
Total # of atoms	487	1613	1870	5591
Definition of "short-range distance" for full-atomic models	6 Å	···	6 Å	···
# of short-range distances for full-atomic models	8578	···	35201	···

limited by the ability to set up test computer simulations in a consistent manner. We chose codes that could easily accommodate distance-only data sets and provide flexibility in adjusting parameters in order to make fair comparisons.

The PROTEAN code is a probabilistic least-squares estimator that transforms a collection of potentially noisy observations into a set of optimal (in the least-squares sense) mean atomic positions and a set of uncertainties associated with each atomic coordinate (Altman, 1995). It uses a gradient-based search approach, combined with a hierarchic decomposition of structural computations (Chen *et al*., 1998). For the purpose of this work, we use a version of the PROTEAN code that was modified to include simulated annealing. We use this as a comparison to our global optimization algorithm, as simulated annealing is a very common method for improving global convergence properties in molecular simulations.

DGSOL solves distance geometry problems with a global continuation algorithm, with Gaussian smoothing of an objective function that only depends on the sparse distance data. Several minimizers are traced in the continuation process, and the best minimizer is selected as the global minimizer (Moré and Wu, 1997). Distance geometry is a commonly used method for estimating molecular structure based on distance input data alone. Since we consider only distance data in this work, distance geometry codes provide a good comparison. But our goal is to develop algorithms that are capable of incorporating many different types of data, so if we are successful then GNOMAD will not be as limited as distance geometry in its applicability to molecular structure estimation problems. There are other more mature implementations of distance geometry, and our choice of DGSOL was based on its straightforward implementation and little extra features that are specific to particular experiments or types of molecules. DIANA is an example of a distance geometry algorithm that is closely tied to the iterative interpretation of NMR data (Mertz *et al*., 1991).

NAB was primarily designed to construct models of helical and nonhelical nucleic acids from a few dozen to a few hundred nucleotides in size, but can also be used for other molecular modeling tasks, including proteins (Macke and Case, 1998). NAB uses a combination of rigid body transformations and distance geometry to create candidate structures that match input criteria. It is designed to provide a flexible way to describe molecular structures at an atomic level of resolution and contains built-in connections to the AMBER (Pearlman *et al*., 1995) and YAMMP (Tan and Harvey, 1993) molecular modeling packages.

4. RESULTS

Results from the local optimization methods—CG, GN, SR1, and BFGS—revealed several important issues. First, the CG method is not as effective as the QN methods when used within the framework of our constrained global optimization algorithm. Using the CG method as a local optimizer in our algorithm tends to move atoms into alignments that result in numerical instabilities and thus poor convergence performance. For example, CG sometimes moves more than two atoms into alignment along a single line, which will result in instabilities.

The GN method was somewhat effective when used as a logical optimizer in our algorithm, but it is applied only to objective functions that are expressed in terms of sum of squares functions. Distances fit such a criterion but our goal is to build a set of methods that will be applicable to any type of input data, including those that can only be expressed in terms of nonquadratic functions.

Both the SR1 and BFGS methods are applicable to general objective functions, and both showed relatively good performance. But BFGS was slightly more accurate and robust and thus it is the preferred local optimization method for use in our algorithm. All the following results were generated using the BFGS method as the local optimization procedure within the GNOMAD constrained global optimization algorithm.

Figures 1 and 2 show comparisons of local optimization, simulated annealing (SA), and the full GNOMAD algorithm on the 1ctf and 1tim proteins, respectively. Local optimization runs were made using the GNOMAD algorithm described previously, but with the MSD constraint checking and local perturbation methods turned off. SA runs were made using PROTEAN with the addition of a simulated annealing routine to improve the global convergence.

Figure 3 shows sample convergence profiles from an optimization of the 1ctf protein. The plots in Fig. 3 show 1) how the constraint error is reduced as the entire optimization algorithm is applied through all the

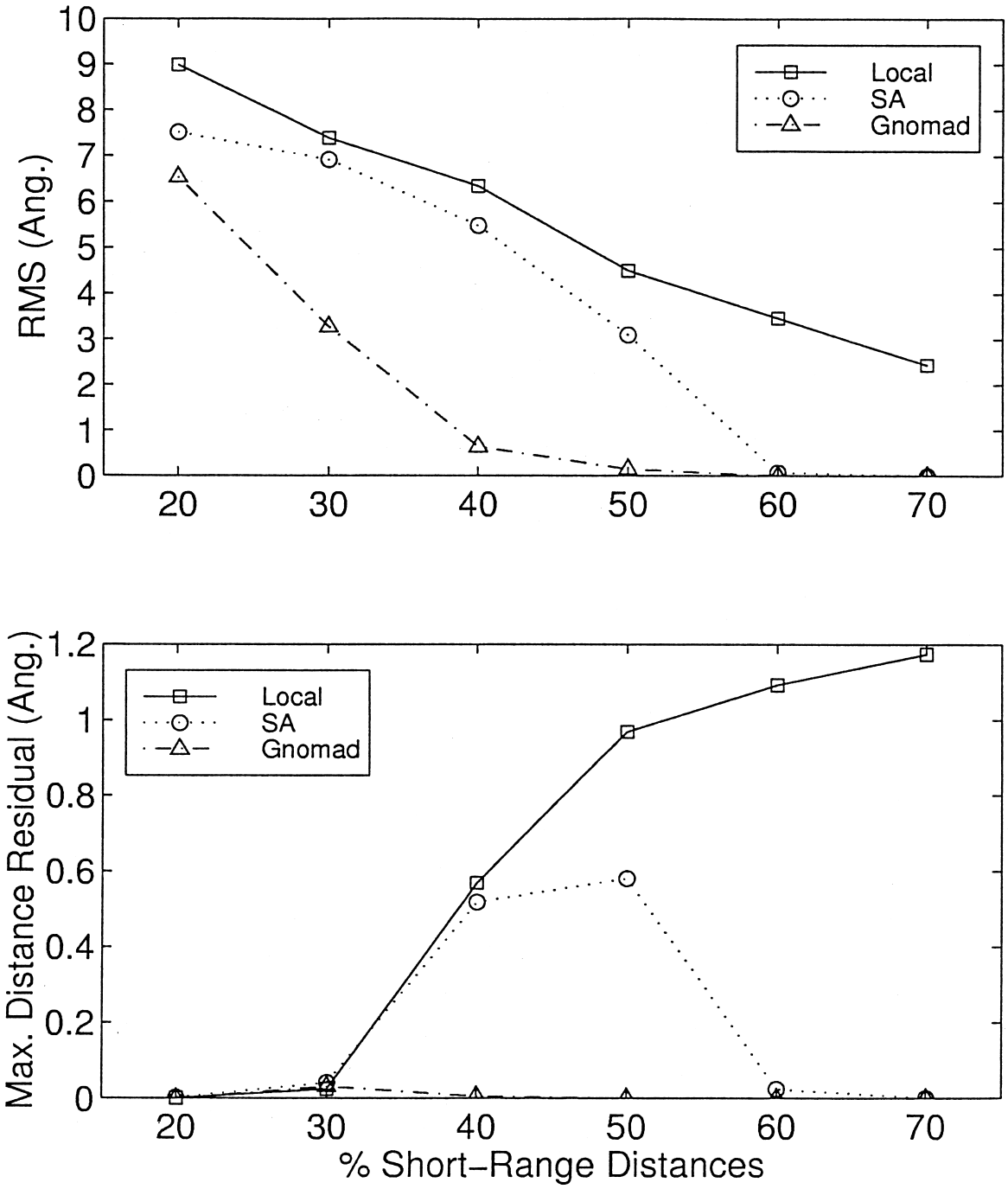


FIG. 1. Comparison of local optimization, simulated annealing, and GNOMAD results on the 1ctf protein, showing both RMSD and maximum residual of all distance constraints.

atoms of the 1ctf protein and again for one restart, and 2) how the constraint error is reduced during one specific suboptimization grouping within the overall optimization process.

To illustrate the effects of adding MSD constraints, Figs. 4 and 5 show qualitative comparisons of structures resulting from non-MSD and MSD GNOMAD runs and their relative proximity to the crystal structure. These results are from input distance data sets consisting of 30% SRD on the 1ctf and 1tim proteins, respectively.

Figures 6–9 present RMSD and error results from DGSOL, NAB, and GNOMAD runs on each of the four test structures. Each plot contains a set of vertical bars, where each bar represents an entire distribution

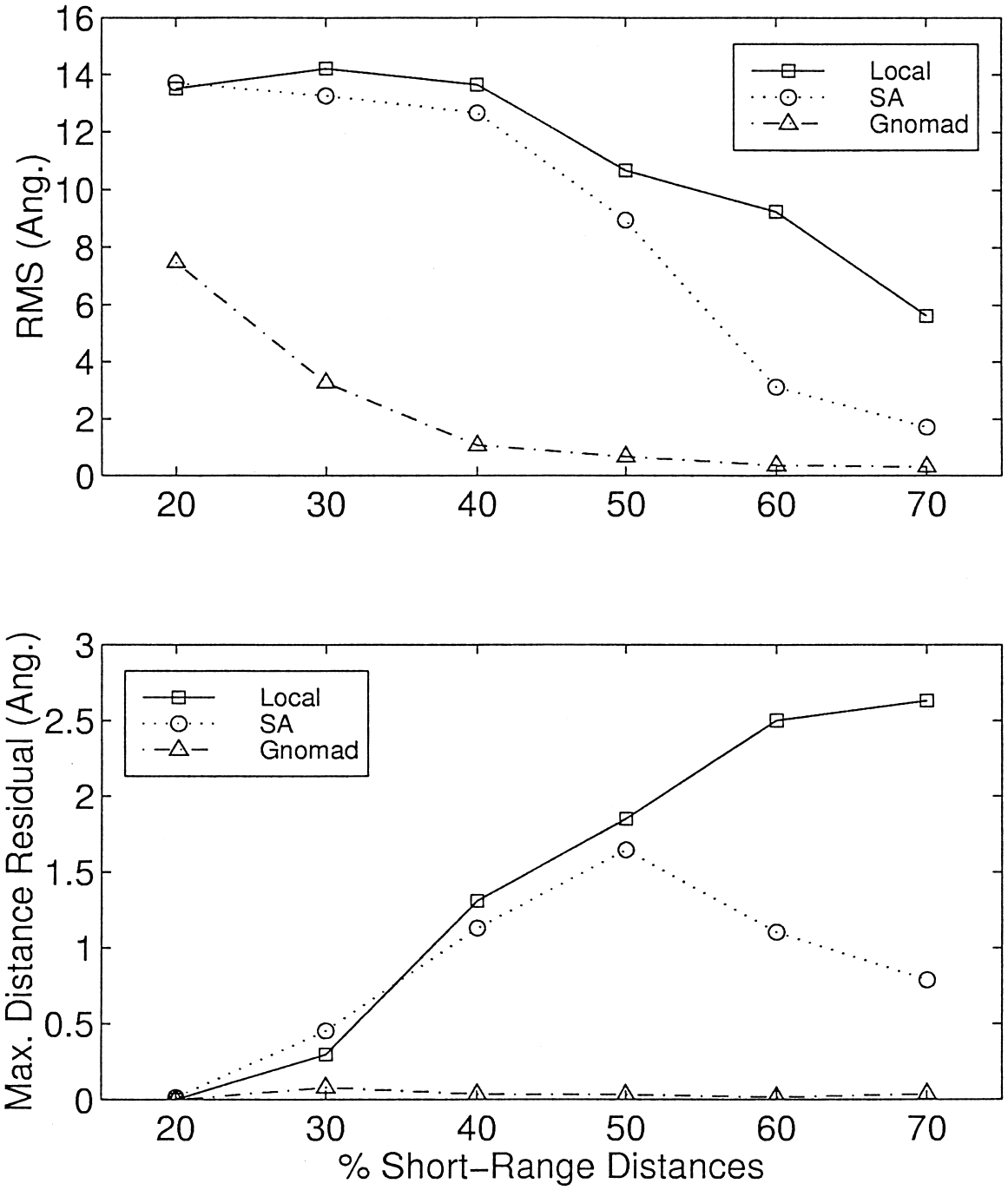


FIG. 2. Comparison of local optimization, simulated annealing, and GNOMAD results on the 1tim protein, showing both RMSD and maximum residual of all distance constraints.

of output data for a set of runs. The horizontal lines within each bar reflect the distribution of results for that set of runs. The bottom and top of each bar represent the minimum and maximum output values. For each % SRD value on the x-axes, ten different distance data sets were used. For the codes that involve random parameters—DGSOL and NAB—ten different runs were made for each of the ten distance data sets. Therefore, each of the bars for the DGSOL and NAB runs contain results for 100 data points, whereas each of the bars for the GNOMAD runs contain results for only ten data points.

Table 2 shows the average CPU times for all of the backbone optimization experiments. As discussed above, for each of the % SRD, ten different data sets were used. For the GNOMAD runs, average CPU

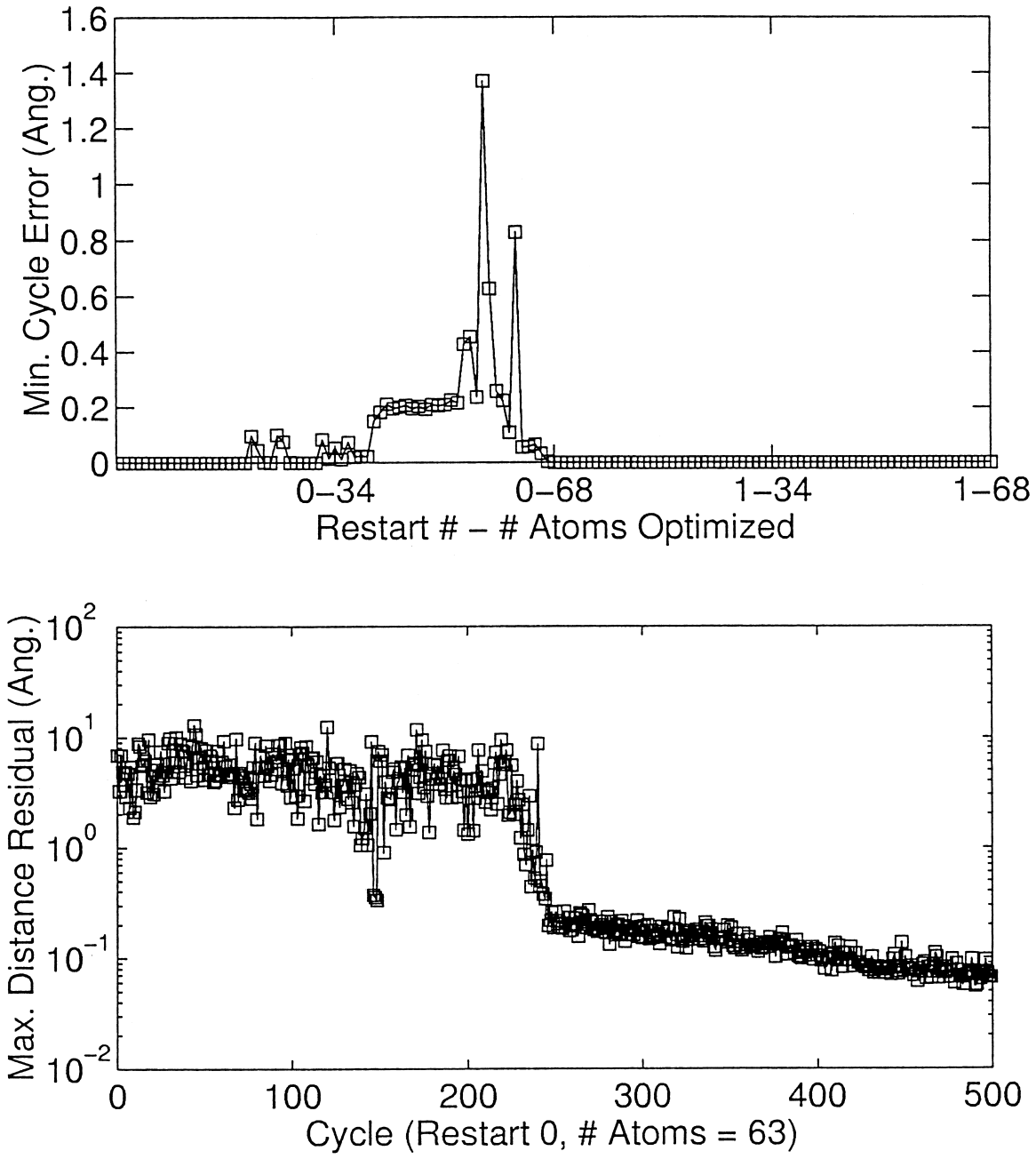
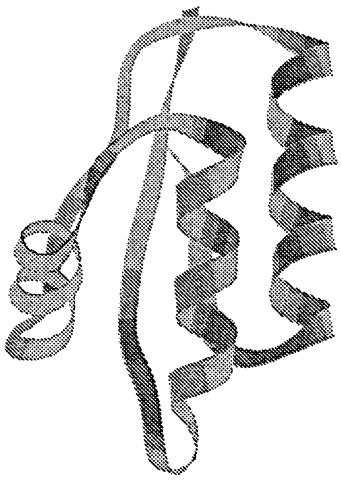


FIG. 3. Sample convergence profiles from the 1ctf protein. The **top** plot shows the minimum average cycle error over for each of the series of suboptimization groupings in a complete GNOMAD optimization. The **bottom** plot shows a more detailed convergence profile of one of the suboptimization groupings, displaying the maximum distance residual for each cycle.

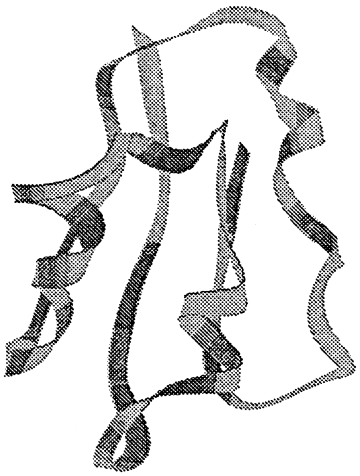
times are simply the averages over those ten runs. But for the DGSOL and NAB runs, which involve random parameters, ten runs were made (varying the random seed) for each of the 10 distance data sets. So CPU times are first summed over the 10 different random seed runs, and those sums are then averaged.

Figures 10 and 11 show qualitative results of the full-atomic optimizations for the 1ctf and 1tim proteins, respectively, using 30% of distance constraints less than 6.0 Å. Ribbon diagrams of the backbone structure are shown for each, as well as the full-atomic models. Runs were also made using 70% of distance constraints less than 6.0 Å but results for these were almost identical to the crystal structure and thus are not displayed separately.



Crystal Structure

Not MSD Constrained
(max. distance residual = 0.01 Å,
RMSD = 7.02 Å)

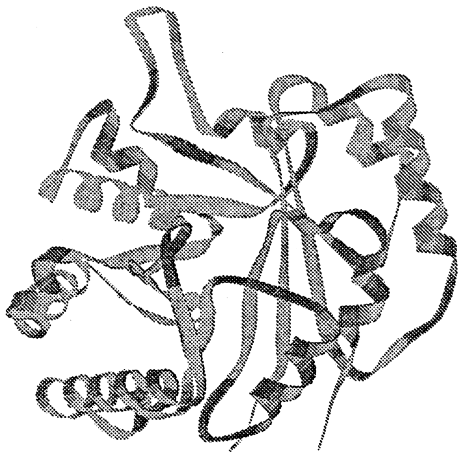


MSD Constrained
(max. distance residual = 0.01 Å,
RMSD = 2.87 Å)

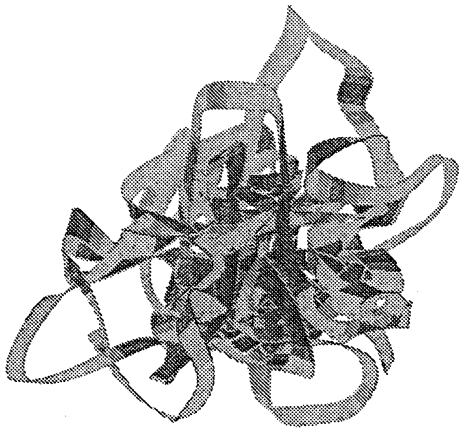
FIG. 4. Crystal structure and structure estimation of 1ctf protein using 30% of distance constraints, illustrating the effects of adding MSD constraints with GNOMAD. Distance satisfaction between MSD-constrained and non-MSD-constrained structural estimates is very similar, yet the MSD-constrained estimate shows significantly improved RMSD.

5. DISCUSSION

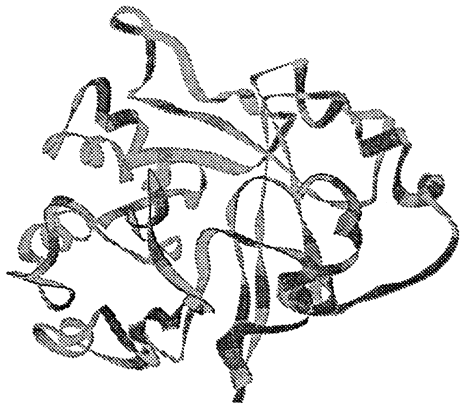
To evaluate and compare the performance of the GNOMAD algorithm to other optimization methods, we will consider two different error metrics. The first is the maximum distance residual, which is a true indication of how well the optimization procedure is able to satisfy its objective of minimizing all of the distance residuals. The second is the RMSD, which measures how close the resulting three-dimensional structure is to the true structure. We are able to use the RMSD error in these test cases because the true



Crystal Structure



Not MSD Constrained
(max. distance constraint error = 0.27 Å,
RMSD = 15.40 Å)



MSD Constrained
(max. distance constraint error = 0.17 Å,
RMSD = 3.10 Å)

FIG. 5. Crystal structure and structure estimation of 1tim protein (tim barrel) using 30% of distance constraints, illustrating the effects of adding MSD constraints with GNOMAD. In this case, the MSD-constrained estimate yields slightly better distance residuals and again shows significantly better RMSD than the non-MSD-constrained estimate.

structure is known. But in cases where the optimization procedure is used to predict unknown structure, only the first error metric would be available. For the purpose of this discussion, we will refer to the first error metric as "error" and the second as RMSD.

In general, the relationship between optimization performance and number of SRD used in estimating molecular structure displays an almost dichotomous pattern. At the low end of the SRD spectrum ($\leq \sim 30\%$ SRD) many low-error local minima exist that are easy to find using any optimization procedure, yet the RMSD for these structures can vary widely with most being relatively high. This is because the level of distance information provided is not sufficient to define a single high-resolution structure. At the higher end of the spectrum ($> \sim 30\%$ SRD) the error landscape becomes rugged with relatively high-error local minima and a low-error global minimum that is very difficult to find yet will yield a structure with

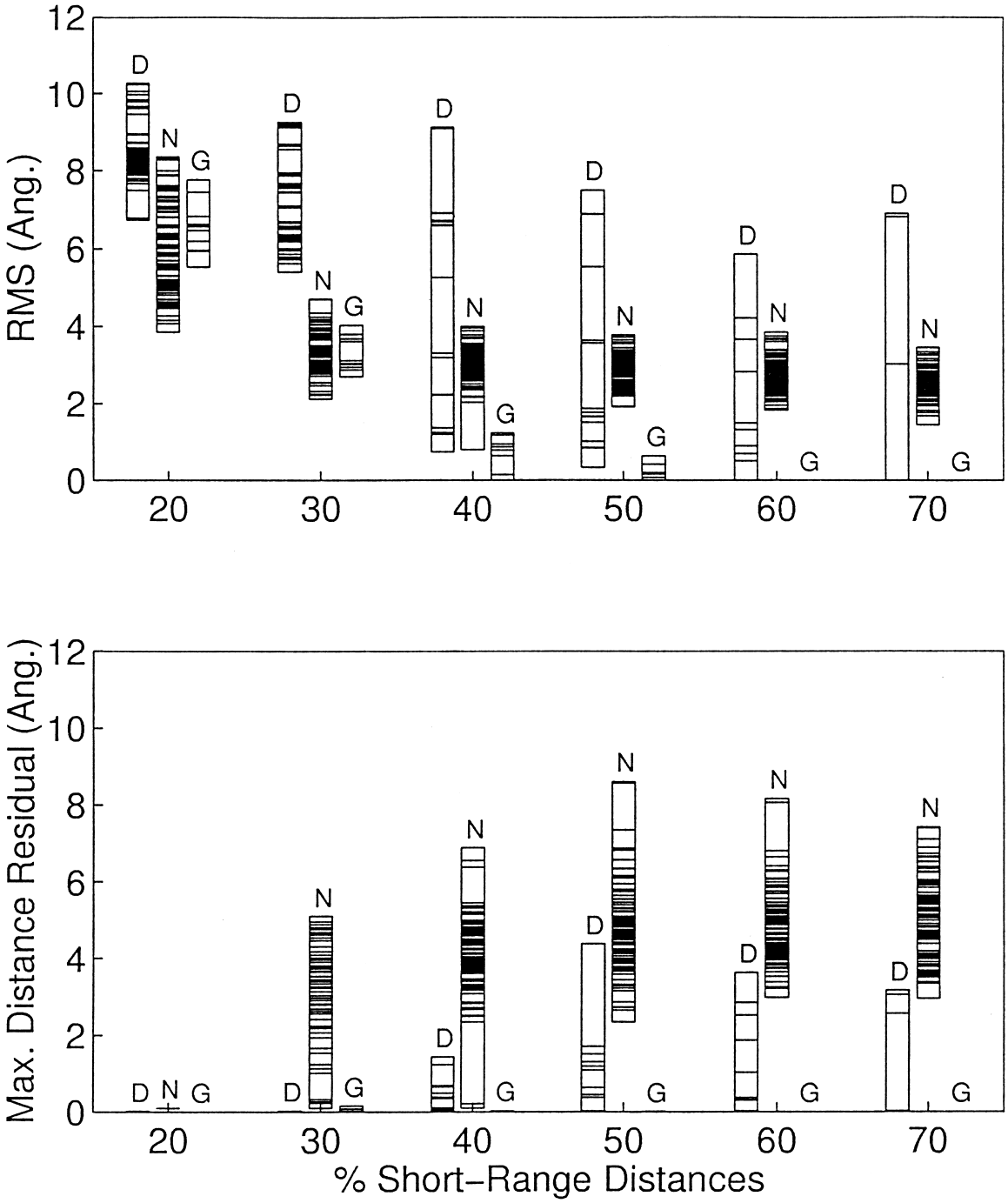


FIG. 6. Comparison of DGSOL, NAB, and GNOMAD results on the 1ctf protein, showing both RMSD and maximum residual of all distance constraints. For this protein, GNOMAD is able to achieve very low distance residuals for all SRD levels and shows improved performance in terms of RMSD, particularly at the higher SRD levels.

relatively good RMSD. At this end of the spectrum there is a more direct relationship between low error and low RMSD.

In the lower SRD range, more information is required to distinguish which of the low-error solutions will yield the better RMSD. In the higher SRD range, global optimization methods are required to move solutions out of local minima and towards the global minimum, which will yield a good RMSD. The MSD constraint enforcement and local perturbation methods used in GNOMAD are effective in addressing both

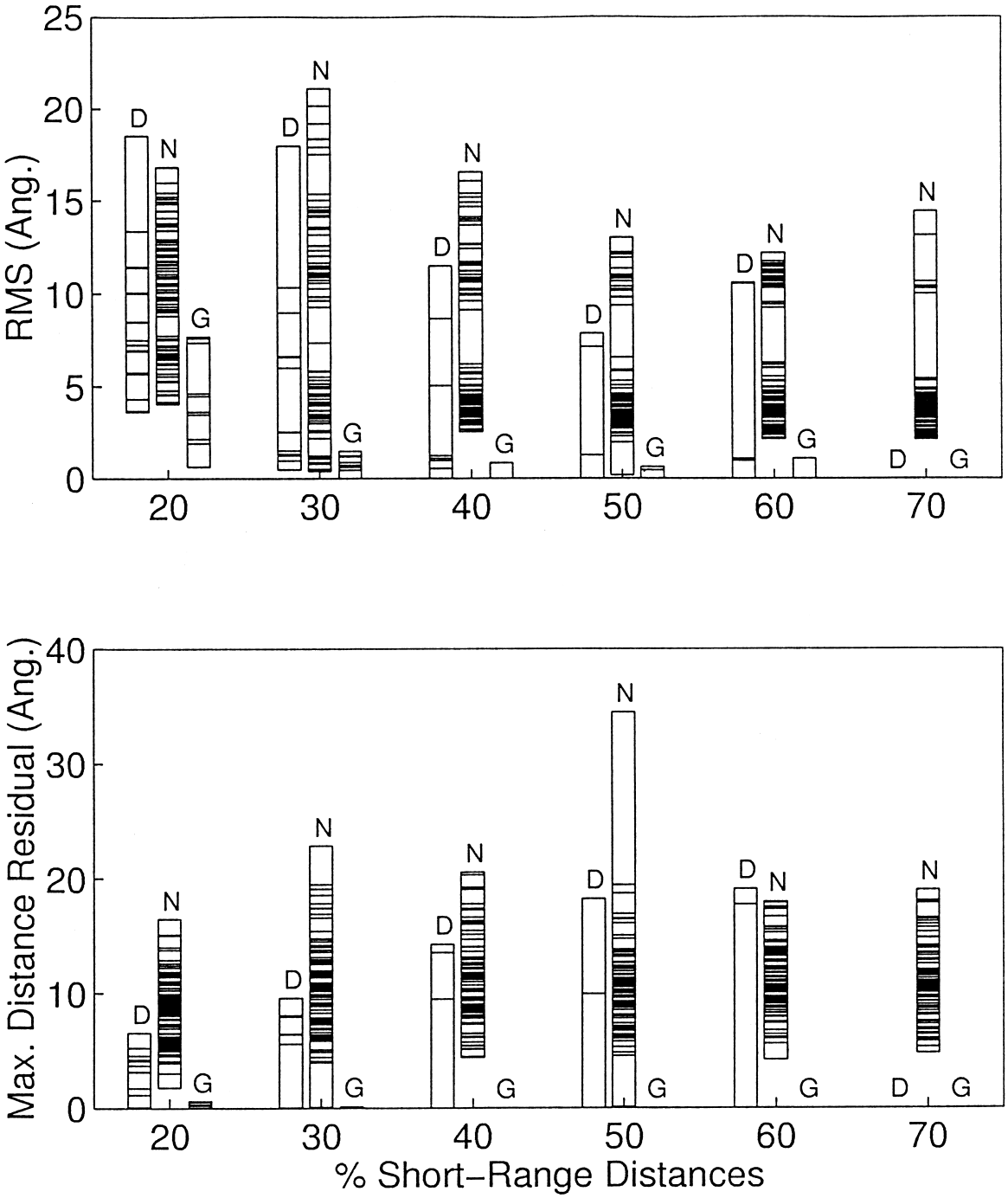


FIG. 7. Comparison of DGSOL, NAB, and GNOMAD results on the yeast phenylalanine tRNA, showing both RMSD and maximum residual of all distance constraints. GNOMAD is able to find very low distance residuals across the entire range of SRD and yields structures with consistently low RMSD.

the low SRD and high SRD concerns. At lower SRD, the MSD constraints serve as added information to reduce the number of potential low-error local minima. At higher SRD, the MSD constraint enforcement combined with the local perturbation provides the global optimization necessary to move solutions out of the local minima and towards the global minimum.

Figures 1 and 2 show how GNOMAD compares to local optimization and SA global optimization methods over the entire range of SRD. We can see from these figures that the local optimization follows

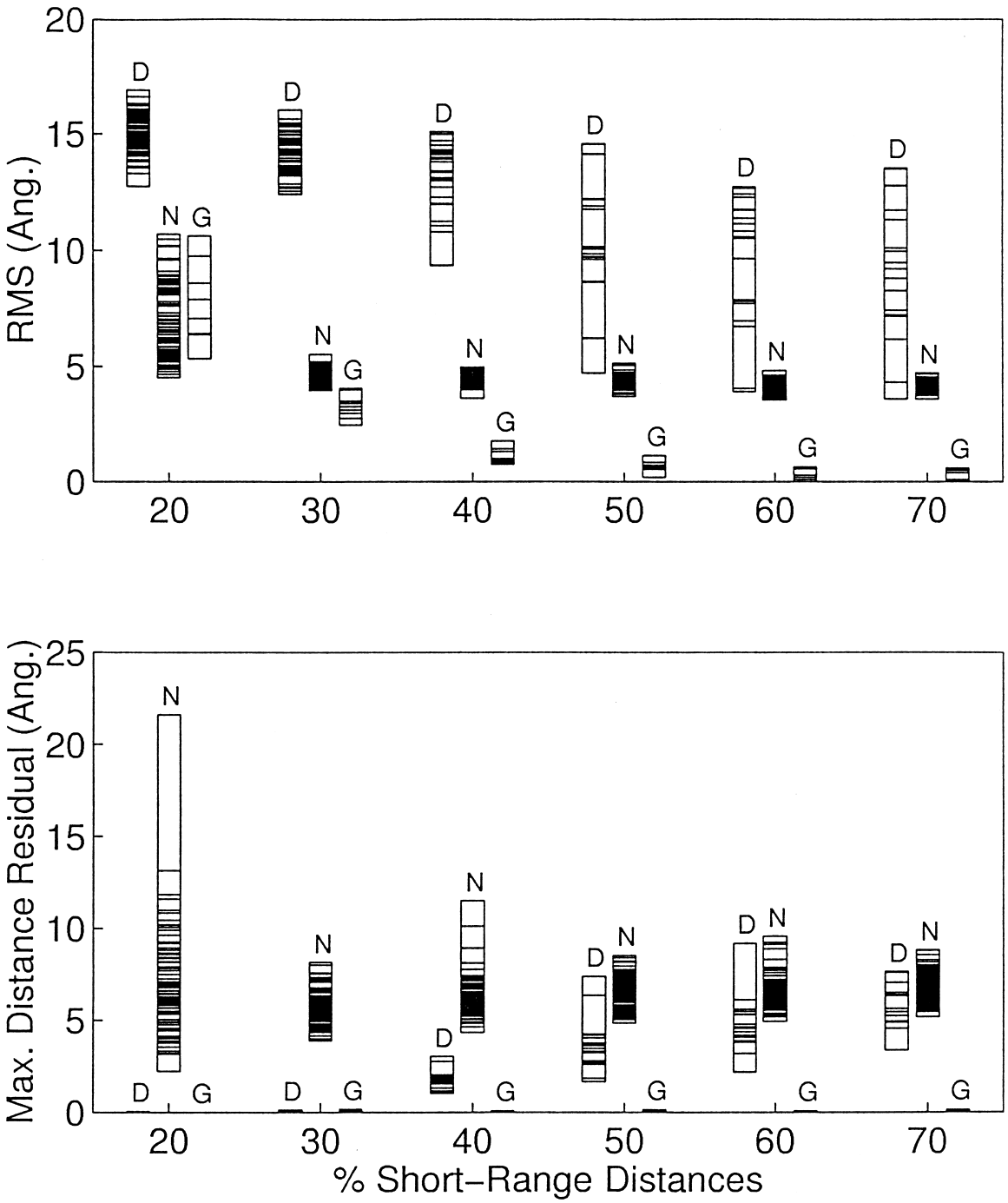


FIG. 8. Comparison of DGSOL, NAB, and GNOMAD results on the 1tim protein, showing both RMSD and maximum residual of all distance constraints. For this larger protein, GNOMAD continues to be able to satisfy distance inputs very well for all SRD levels. These low distance residual structure estimates also translate consistently to low RMSD.

a typical pattern of low error at low SRD and increased error as SRD is increased. At low SRD, the low error translates to poor RMSD. As SRD increases, RMSD improves slightly but not to an acceptable level, due to the increased error from local minima entrapment. The SA global optimization procedure is able to decrease the error in the 60–70% SRD range but is not as effective in the 40–50% SRD range. Also, because neither the local optimization nor the SA method is using MSD constraints, their RMSD is relatively high at lower SRD, where the error is small. GNOMAD, on the other hand, yields lower RMSD

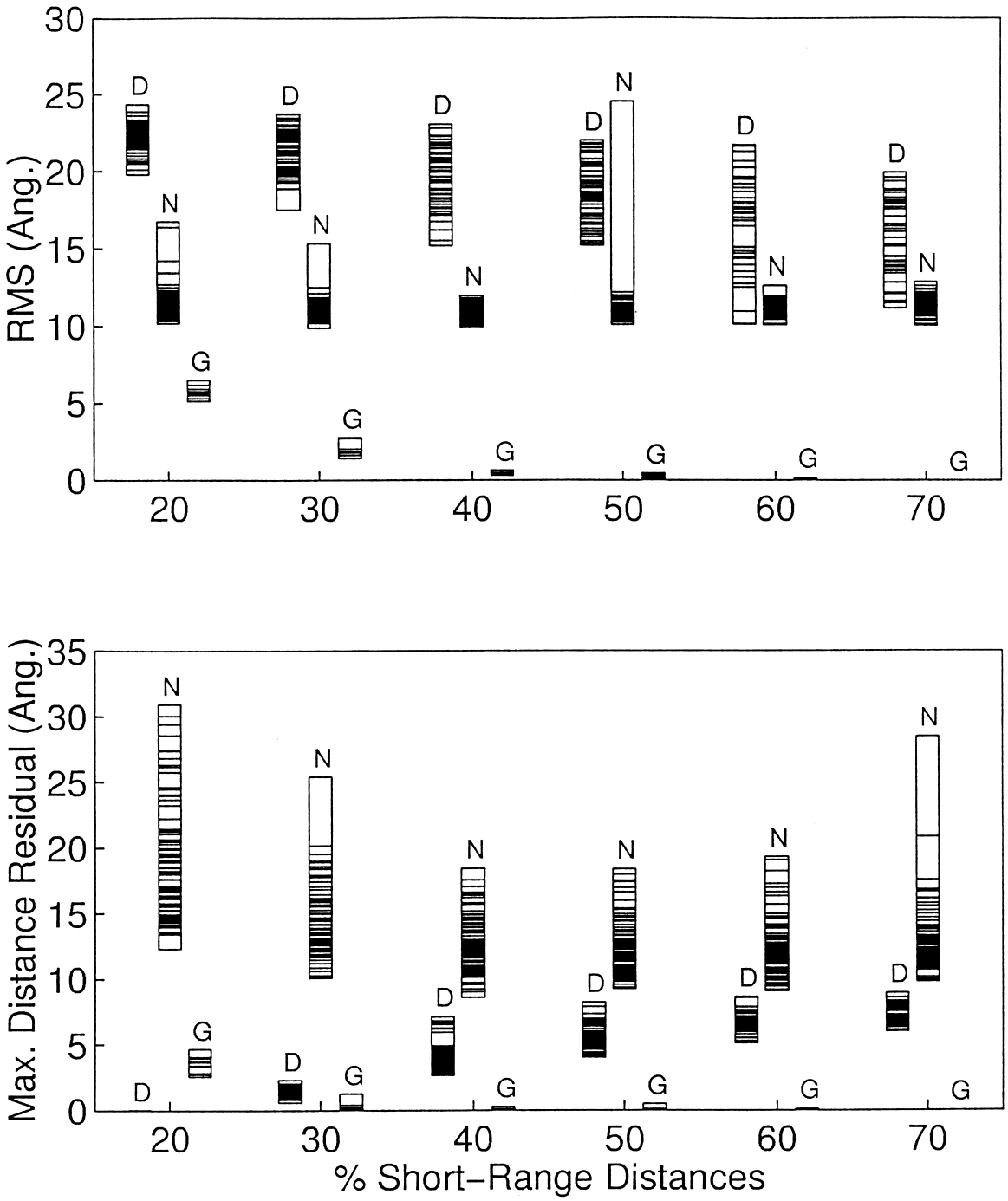


FIG. 9. Comparison of DGSOL, NAB, and GNOMAD results on the 2nap protein, showing both RMSD and maximum residual of all distance constraints. Results on this very large protein demonstrate how well GNOMAD scales. GNOMAD continues to be able to find structures with relatively low distance residuals and RMSD.

structures at lower SRD, given the same level of near-zero error. This is due to the addition of MSD constraints. GNOMAD is also able to find lower error structures at high SRD, which translate directly to lower RMSD. This is due to a combination of the MSD constraint and local perturbation algorithms which yield very good global convergence.

The MSD constraint enforcement/local perturbation algorithm used in GNOMAD is effective in avoiding local minima for two reasons: 1) it significantly reduces the search space and thereby eliminates many

TABLE 2. CPU TIME (SEC) COMPARISONS

Structure (PDB ID)	*Optimization code*	*% short-range distances*					
		20	*30*	*40*	*50*	*60*	*70*
1ctf	DGSOL	34.20	116.90	127.00	101.50	95.90	89.60
	NAB	150.00	153.19	156.00	158.41	161.12	163.05
	GNOMAD	11.94	255.49	344.81	208.83	127.34	111.22
1tra	DGSOL	265.40	267.30	291.60	323.50	372.70	402.50
	NAB	160.98	164.95	171.24	174.08	177.64	181.28
	GNOMAD	581.89	505.00	548.63	774.33	1007.96	1292.72
1tim	DGSOL	253.00	1263.20	838.30	841.50	904.70	1001.40
	NAB	490.84	493.00	496.53	499.55	502.40	506.13
	GNOMAD	1340.59	9849.11	10893.97	10546.32	11633.17	13308.66
2nap	DGSOL	2052.38	4991.28	3924.62	4076.40	4514.01	4912.13
	NAB	10801.80	10831.78	10855.57	10869.53	10915.25	10941.22
	GNOMAD	22350.82	24974.29	25402.08	25757.34	26205.89	27354.62

of the local minima that a greedy, unconstrained optimization algorithm might become trapped in, and 2) it provides for natural, nonrandom perturbation of atoms. The perturbations occur each time an α (Equation 8) is adjusted due to a MSD constraint violation. This keeps the algorithm from always moving atoms to their locally optimal value ($\alpha = 1$) in the QN procedure and getting stuck in local minima. This perturbation acts in a similar fashion to perturbations found in other global optimization approaches, such as simulated annealing, the difference being that our approach is completely nonrandom and is instead based on domain-specific information acquired during the course of the optimization process.

It could be argued that the SA algorithm might also find the same global minimum solutions that GNOMAD finds if it were allowed to run long enough and with the right combination of parameters. But given the random nature of the algorithm and that there are many parameters to set, it could be very time consuming and computationally inefficient to try to find the global minimum for the structures presented here using SA. In the results presented in Figs. 1 and 2, we ran the SA algorithm for the same number of cycles as the GNOMAD algorithm to make for a fair comparison. Global convergence of the GNOMAD algorithm is not guaranteed, but for the types of structures examined in this work, which were mainly globular, we found that global convergence was uniform and relatively fast. For all of the test cases considered in this work, GNOMAD was allowed to run for 3,000 cycles on the final grouping containing all toms. Minimum error configurations, however, were generally found within 1,000–2,000 of those cycles.

Figure 3 gives a more detailed view of a typical GNOMAD convergence profile. From the first plot, we can see that as atoms are added to the suboptimization groupings, distance residuals tend to grow. But during the first restart (second pass) of the algorithm, very low distance residuals are found throughout all groupings. This is due in part to better initial values for atoms that are added during the restart. The second plot shows the specific convergence profile for one of the optimization subgroupings. From this plot, we see how the perturbations provided by the MSD constraint-checking algorithm allow the algorithm to search many configurations before settling into the path toward the globally optimal configuration.

It is interesting to note from the second plot in Fig. 3 that despite an apparent move near cycle 150 towards the global minimum, the GNOMAD algorithm is intelligent in detecting that this is a path towards a potentially local minimum. At this point the nonrandom domain-based perturbations force the optimization out of this path and back into the search for the global minimum. It is also interesting to note the subsequent behavior around cycle 250, where the next low-error configurations are found. At this point, instead of being disturbed by some prespecified perturbation pattern, the algorithm recognizes this as a viable path towards the global minimum, which it follows, and does eventually find the global minimum.

The qualitative results presented in Figs. 4 and 5 show what typically happens when a relatively small number of distance constraints (30% of SRD in these cases) is used with and without the addition of MSD constraints. In Fig. 4, we see that for the 1ctf protein a very low maximum distance error is found for both

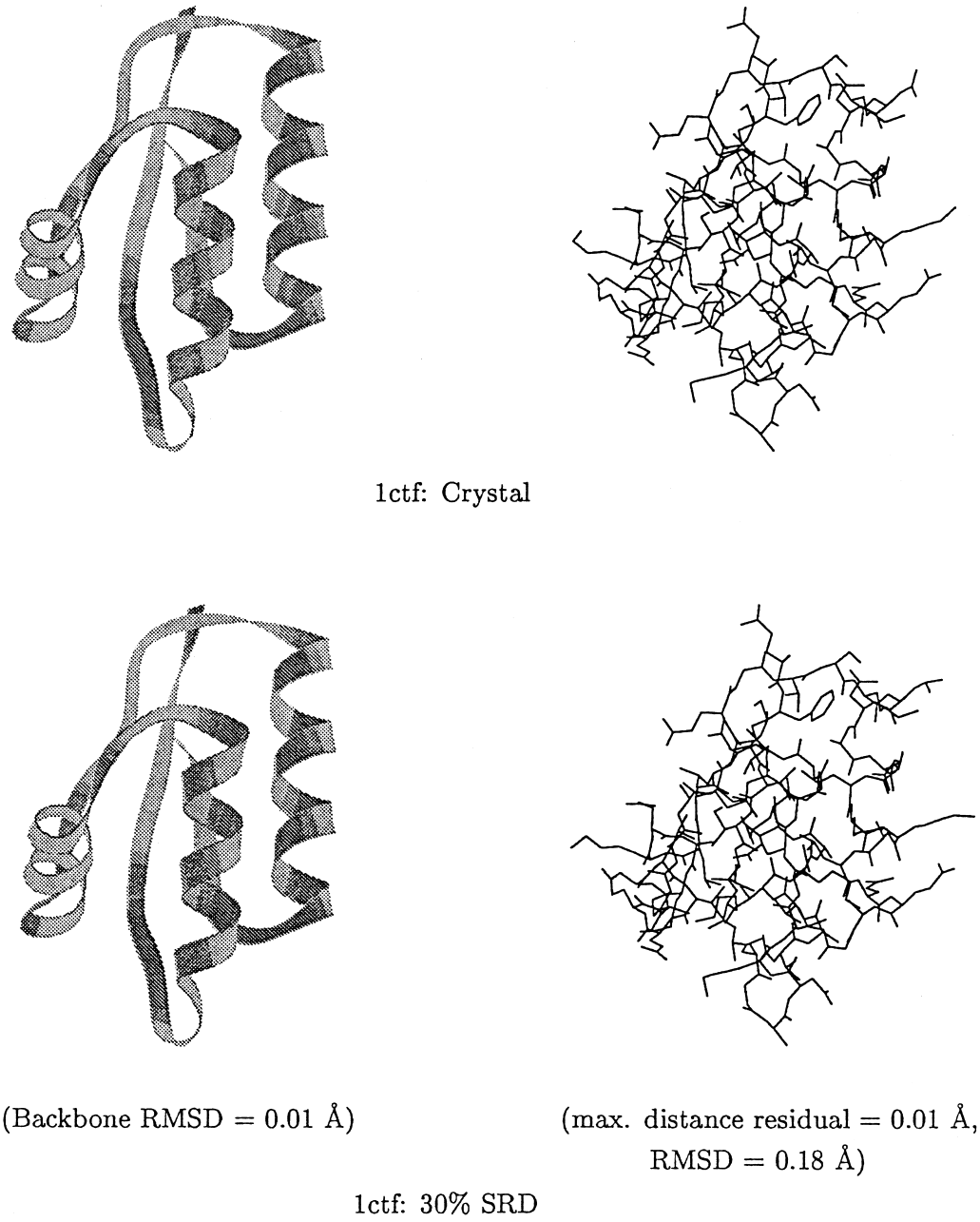


FIG. 10. Results from full-atomic optimization of 1ctf protein using 30% of distance constraints less than 6 Å. On the left are ribbon representations of only the backbone atoms from the resulting structure estimate, and on the right are the full-atomic representations.

non-MSD and MSD runs but qualitatively the results look much better for the case where MSD constraints are used. This illustrates the point that MSD constraints serve to 1) limit the search space, 2) reduce the number of acceptable low-error structures, and 3) yield better resolution low-error structures. For the case of the 1tim protein, Fig. 5, maximum distance errors for the non-MS and MSD cases are again similar, yet again the MSD results show a much better resolution structure. These results show the value that is added by being able to efficiently incorporate MSD constraints into the optimization procedure. In doing so, we are able to get better structural estimates with less distance information.

Results in Figs. 6–9 show that GNOMAD is able to consistently satisfy the distance data to a very low error, over a wide range of problem sizes. Because of the use of MSD constraints, the RMSD

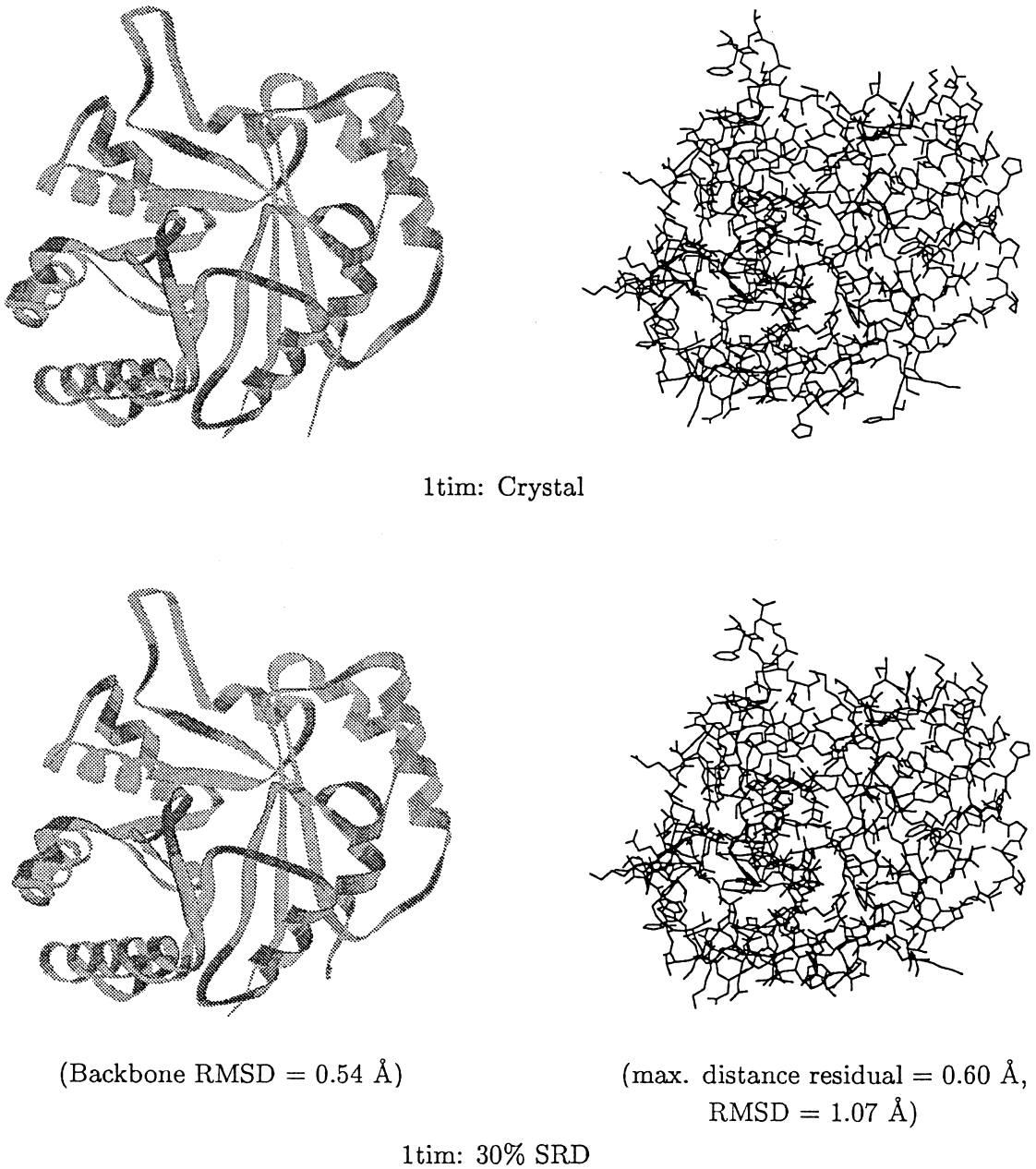


FIG. 11. Results from full-atomic optimization of the 1tim protein using 30% of distance constraints less than 6 Å. On the left are ribbon representations of only the backbone atoms from the resulting structure estimate, and on the right are the full-atomic representations.

from GNOMAD runs is relatively low in the 20–30% SRD level. And because of the improved global convergence properties of GNOMAD, lower errors are attained in the higher % SRD level, which translate directly to lower RMSD structures.

It should be noted that for the results shown in Figs. 6–9, relatively tight bounds on the input distances (± 0.1 Å) were used for the distance geometry-based DGSOL and NAB codes. This was done to make a fair comparison to GNOMAD for the case considered in this work where we know the input distance data are exact and we are trying to evaluate and compare the accuracy of the various codes in satisfying this data. DGSOL seemed to perform best when given tight distance bounds, but NAB seemed to have some "optimal" distance bounds where it performed best. In future work, we will consider the case of inexact data and methods for finding optimal structures in light of residual error.

All of the optimization methods were also slightly sensitive to the default MSD constraint radius value, r_u, as defined in Section 2.4. For smaller proteins, this value can be set closer to the actual minimum distance seen between any atoms in the molecule. Yet, for larger proteins, this constraint must be relaxed slightly. For example, we used a value of 3.7 Å for the backbone C_α estimates of the smaller 1ctf protein and 3.5 Å for the larger 1tim and 2nap proteins. A value of 5.0 Å was used for the backbone phosphate estimates of the 1tra tRNA molecule.

Timing results in Table 2 show that GNOMAD takes approximately the same amount of run time as DGSOL and NAB for the smaller structure, 1ctf, which is not a particularly difficult problem. Then, as the structures become larger and it becomes more difficult to find optimal estimates, GNOMAD spends more CPU time but finds much better solutions. Experiments were performed with DGSOL and NAB to allow them to run longer but this did not seem to correspondingly increase their search capabilities or improve the results of the optimization in terms of error or RMSD. The extra CPU times used by GNOMAD appears to be spent effectively searching the space of possible configurations to find a global or near-global minimum.

Results shown in Figs. 10 and 11 demonstrate that the GNOMAD code is robust and scales well for larger optimization problems that consider all atoms in the molecule. For the 1ctf and 1tim full-atomic structures, GNOMAD is able to yield structural estimates of approximately 1.0 Å and less using only 30% of distances less than 6.0 Å.

Based on the results of this work, we can draw several conclusions:

- Optimization methods play an important role in molecular structure estimation, as they can provide better resolution structures given the same amount or even less input data. This is valuable because the means for collecting input data are often time consuming and expensive.
- Two important aspects of molecular optimization algorithms are the ability to effectively incorporate MSD constraints and to find the global minimum error solution. Each of these will contribute significantly to finding better resolution structures from relatively sparse data.
- A new optimization algorithm is presented that displays very good global convergence properties, while maintaining MSD constraint satisfaction. This algorithm is able to find relatively low RMSD structures over the entire spectrum of SRD information content.
- Tests of this new optimization algorithm on various types of biological molecules show it to be very robust and computationally efficient. The new algorithm compares favorably to other molecular structure estimation codes, in terms of both error and RMSD. The algorithm scales well and remains effective when applied to optimizations involving a large number of atoms.

ACKNOWLEDGMENTS

This work was supported by the National Institutes of Health (NIH) grants LM-05652, LM-06422, LM-07033, and HG-00044, the Burroughs Wellcome Foundation, and a National Partnership for Advanced Computational Infrastructure (NPACI)/National Science Foundation (NSF) grant through the San Diego Supercomputer Center (subcontract UCSD 10152756). We would like to thank David Case and Jorge Moré for allowing us to use their computer codes and for their assistance in the implementation of those codes. We would also like to thank Michelle Carrillo for her assistance during the development and validation of the algorithms and Teri Klein for her guidance in the submission of the manuscript.

REFERENCES

Altman, R.B. 1995. A probabilistic approach to determining biological structure: Integrating uncertain data sources. *Int. J. Human-Computer Studies* 42, 593–616.

Berman, H.M., Westbrook, J., Feng, Z., Gilliland, G., Bhat, T.N., Weissig, H., Shindyalov, I.N., and Bourne, P.E. 2000. The protein data bank. *Nucl. Acids Res.* 28, 235–242.

Braun, W., and Go, N. 1985. Calculation of protein conformations by proton–proton distance constraints: A new efficient algorithm. *J. Mol. Biol.* 186, 611–626.

Brooks, B.R., Bruccoleri, R.E., Olafson, B.D., States, D.J., Swaminathan, S., and Karplus, M. 1983. CHARMM: A program for macromolecular energy, minimization, and dynamics calculations. *J. Comp. Chem.* 4, 187–217.

Broyden, C.G. 1970. The convergence of a class of double-rank minimization algorithms, parts I and II. *JIMA* 6, 76–90, 222–236.

Buckingham, A.D., Fowler, P.W., and Hutson, J.M. 1988. Theoretical studies of van der Waals molecules and intermolecular forces. *Chem. Rev.* 88, 6, 963–988.

Chang, G., Guida, W.C., and Still, W.C. 1989. An internal coordinate Monte Carlo method for searching conformational space. *J. Am Chem. Soc.* 111, 12, 4379–4386.

Chen, C.C., Chen, R.O., and Altman, R.B. 1996. Constraining volume by matching the moments of a distance distribution. *Comp. Appl. Biosci.* 12, 4, 319–326.

Chen, C.C., Singh, J.P., and Altman, R.B. 1998. The hierarchical organization of molecular structure computations. *J. Comp. Biol.* 5, 3, 409–422.

Cheng, A., Stanton, R.S., Vincent, J.J., van der Vaart, A., Damodaran, S.I., Dixon, D., Hartsough, S., Mori, M., Best, S.A., Monard, G., Garcia, M., van Zant, I.C., and Merz, K.M. 1999. ROAR 2.0. The Pennsylvania State University.

Covell, D.G. 1992. Folding protein alpha-carbon chains into compact forms by Monte Carlo methods. *Proteins: Struct. Funct. Genet.* 14, 409–420.

Dennis, J.E., and Schnabel, R.B. 1996. *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*. SIAM, Philadelphia, PA.

Dill, K.A., Phillips, A.T., and Rosen, J.B. 1997. Protein structure and energy landscape dependence on sequence using a continuous energy function. *J. Comp. Biol.* 4, 3, 227–239.

Dinur, U., and Hagler, A.T. 1991. New approaches to empirical force fields. In Lipkowitz, K.B., and Boyd, D.B., eds., *Reviews in Computational Chemistry*. VCH, New York.

Fletcher, R. 1970. A new approach to variable metric algorithms. *Comput. J.* 13, 317–322.

Goldfarb, D. 1970. A family of variable metric methods derived by variational means. *Math. Comp.* 24, 23–26.

Güntert, P., Mumenthaler, C., and Wüthrich, K. 1997. Torsion angle dynamics for NMR structure calculation with the new program DYANA. *J. Mol. Biol.* 273, 283–298.

Halgren, T.A. 1992. Representation of van der Waals (vdW) interactions in molecular mechanics force fields: Potential form, combination rules, and vdW parameters. *J. Am. Chem. Soc.* 114, 20, 7827–7843.

Head, J.D., and Zerner, M.C. 1985. A Broyden–Fletcher–Goldfarb–Shanno optimization procedure for molecular geometries. *Chem. Phys. Lett.* 122, 3, 264–270.

Head, J.D., and Zerner, M.C. 1989. Newton based optimization methods for obtaining molecular-conformation. *Adv. Quantum Chem.* 20, 239–290.

Herrmann, F., and Suhai, S. 1995. Energy minimization of peptide analogues using genetic algorithms. *J. Comput. Chem.* 16, 11, 1434–1444.

Hestenes, M.R. 1980. *Conjugate-Direction Methods in Optimization*. Springer-Verlag, Berlin.

Ingber, L. 1989. Very fast simulated re-annealing. *J. Math. Comput. Modeling* 12, 967–973.

Kirkpatrick, S., Gelatt, C.D., and Vecchi, M.P. 1983. Optimization by simulated annealing. *Science* 220, 4598, 671–680.

Kolinski, A., and Skolnick, J. 1994. Monte Carlo simulations of protein folding. I. Lattice model and interaction scheme. *Proteins: Struct. Funct. Genet.* 18, 338–352.

Kostrowicki, J., and Piela, L. 1991. Diffusion equation method of global minimization: Performance for standard test function. *J. Opt. Theory Appl.* 69, 269–284.

Kostrowicki, J., and Scheraga, H.A. 1992. Application of the diffusion equation method for global optimization to oligopeptides. *J. Phys. Chem.* 96, 7442–7449.

Lennard-Jones, J.E. 1924. *Proc. Roy. Soc. London*, Ser. A, 106, 463.

Li, Z., and Scheraga, H. 1987. Monte Carlo minimization approach to the multiple minima problem in protein folding. *Proc. Natl. Acad. Sci. USA* 84, 6611–6615.

Luenberger, D.G. 1989. *Linear and Nonlinear Programming*. Addison-Wesley, Reading, MA.

Macke, T., and Case, D.A. 1998. Modeling unusual nucleic acid structures. *In* Leontes, N.B., and SantaLucia, J., eds., *Molecular Modeling of Nucleic Acids*, 379–393. American Chemical Society, Washington, DC.

Meiyappan, S., Raghavan, R., Viswanathan, R., and Yu, Y. 1999. Proteinmorphosis: A mechanical model for protein conformational changes. *Proc. Pacific Symposium on Biocomputing*, 341–353.

Mertz, J.E., Güntert, P., Wüthhrich, K., and Braun, W. 1991. Complete relaxation matrix refinement of NMR structures of proteins using analytically calculated dihedral angle derivatives of NOE intensities. *J. Biomol. NMR* 1, 3, 257–269.

Moré, J., and Wu, Z. 1996. Smoothing techniques for macromolecular optimization. *In* Di Pillo, G., and Gianessi, F., eds., *Nonlinear Optimization and Applications*. Plenum Press, New York.

Moré, J., and Wu, Z. 1997. Global continuation for distance geometry problems. *SIAM J. Optim.*7, 3, 814–836.

O'Toole, E.M., and Panagiotopoulos, A.Z. 1992. Monte Carlo simulation of folding transitions of simple model proteins using a chain growth algorithm. *J. Chem. Phys.* 97, 8644-8652.

Pearlman, D.A., Case, D.W., Caldwell, J.W., Ross, W.R., Cheatham, III, T.E., DeBolt, S., Ferguson, D., Seibel, G., and Kollman, P. 1995. AMBER, a computer program for applying molecular mechanics, normal mode analysis,

molecular dynamics and free energy calculations to elucidate the structures and energies of molecules. *Comp. Phys. Comm.* 91, 1–41.

Polak, E. 1971. *Computational Methods in Optimization*. Academic Press, New York.

Press, W.H., Teukolsky, S.A., Vetterling, W.T., and Flannery, B.P. 1992. *Numerical Recipes in C: The Art of Scientific Computing* 2nd ed., 318–323. Cambridge University Press, Cambridge.

Ripoll, D.R., and Thomas, S.J. 1990. A parallel Monte Carlo search algorithm for the conformational analysis of proteins. *Proc. IEEE/ACM Supercomputing '90*, 94–102.

Sali, A., and Blundell, T.L. 1993. Comparative protein modelling by satisfaction of spatial restraints. *J. Mol. Biol.* 234, 779–815.

Shakhnovich, E.I., Farztdinov, G., Gutin, A.M., and Karplus, M. 1991. Protein folding bottlenecks: A lattice Monte Carlo simulation. *Phys. Rev. Lett.* 67, 1665–1668.

Shanno, D.F. 1970. Conditioning of quasi-Newton methods for function minimization. *Math. Comp.* 24, 647–657.

Sun, S. 1993. Reduced representation model of protein structure prediction: Statistical potential and genetic algorithms. *Protein Sci.* 2, 762–785.

Tan, R.K.Z., and Harvey, S.C. 1993. Yammp: Development of a molecular mechanics program using the modular programming method. *J. Comput. Chem.* 14, 455–470.

Unger, R., and Moult, J. 1993. Genetic algorithms for protein folding simulations. *J. Mol. Biol.* 231, 75–81.

Wilson, S.R., and Cui, W. 1990. Applications of simulated annealing to peptides. *Biopolymers* 29, 1, 225–235.

Address correspondence to:
Glenn A. Williams
Stanford University Medical Center
Stanford Medical Informatics
251 Campus Drive, MSOB X-215
Stanford, CA 94305–5479

E-mail: gaw@smi.stanford.edu