Performance of Efficient Minimization Algorithms as Applied to Models of Peptides and Proteins

C. Baysal[‡], H. Meirovitch[‡], and I. M. Navon[†‡]

[†]*Department of Mathematics and* [‡]*Supercomputer Computations Research Institute,*
Florida State University, Tallahassee, Florida 32306, U.S.A.
phone: (850) 644-7032
fax : (850) 644-0098
e-mail: hagai@scri.fsu.edu

Abstract

We test the efficiency of three minimization algorithms as applied to models of peptides and proteins; they are, the limited memory quasi-Newton (L-BFGS) of Liu and Nocedal, the truncated Newton (TN) with automatic preconditioner of Nash, and the nonlinear conjugate gradients (CG) of Shanno and Phua. The molecules are modeled by two energy functions, one is the GROMOS87 united atoms force field (defining the energy E_{GRO}), which takes into account the intramolecular interactions only; the second is defined by the energy $E_{\mathrm{tot}} = E_{\mathrm{GRO}} + E_{\mathrm{solv}}$ where E_{solv} is an implicit solvation free energy term based on the solvent accessible surface area of the atoms. The molecules studied are *cyclo*-(D-Pro^1-Ala^2-Ala^3-Ala^4-Ala^5) (31 atoms), axinastatin 2 [*cyclo*-(Asn^1-Pro^2-Phe^3-Val^4-Leu^5-Pro^6-Val^7), 62 atoms], and the protein bovine pancreatic trypsin inhibitor (58 residues, 568 atoms). With E_{GRO}, the performance of TN with respect to the CPU time is found to be 1.2 - 2 times better than that of both L-BFGS and CG, whereas with E_{tot}, L-BFGS outperforms TN by a factor of 1.5 - 2.5, and CG by a larger factor. Still, the quality of the solution in terms of the value of the minimized energy and the gradient norm, obtained with TN, is always equivalent to or better than those obtained with L-BFGS and CG. The performance is analyzed in terms of criteria outlined by Nash and Nocedal. We find the distribution of the Hessian eigenvalues to be a reliable predictor of efficiency.

Keywords: energy minimization; cyclic peptides and proteins; implicit solvation models; truncated and quasi-Newton, Hessian eigenvalues

Introduction

The interatomic interactions of a protein or other biomolecules, such as a DNA segment, are usually described by an empirical potential energy function (force field), which is structure dependent and typically leads to an energy surface "decorated" by a tremendous number of local minima.[1] Identifying the lowest energy minima, in particular the *global* one, is the goal of protein folding where the energy, rather than the free energy, is accepted as an approximate criterion of stability. A more rigorous criterion is minimum harmonic free energy, F^{har}, where F^{har} around a minimum is obtained from the harmonic entropy, S^{har}. S^{har} is proportional to the determinant of the Hessian, the matrix of second derivatives of the energy with respect to the molecular coordinates.[2–6] Calculation of the Hessian at a minimum is also an essential part of a normal mode analysis.[7]

This short discussion already demonstrates the importance of energy minimization in computational structural biology and the need for the development of efficient minimization algorithms. The common algorithms, such as conjugate gradients or Newton methods, are of a local character, that is, they drive an initial molecular structure to the closest energy minimum rather than to the global one. However, in practice, this does not limit their applicability because most of the *global* optimization procedures, including our LTD method for cyclic molecules,[8,9] are based on a large number of local energy minimizations (see for example Refs. 10–14). Therefore, attempting to optimize LTD, we tested previously several minimization algorithms and found the limited memory BFGS (L-BFGS)[15] to be the most efficient.[9] The main objective of the present paper is to compare the performance of L-BFGS to that of truncated Newton (TN),[16–19] which was not tested in Ref. 9, as applied to peptide and protein models. Such a study is necessary because the performance of minimization algorithms is known to be problem dependent to a large extent.[20]

The experience gained thus far from treating various problems, in particular large-scale unconstrained minimizations,[21–24] is that TN and L-BFGS are powerful optimization methods which are more efficient than other techniques (see also Refs. 25, 26 and 27). TN tends to blend the rapid (quadratic) convergence rate of the classical Newton method with feasible storage and computational requirements. The L-BFGS algorithm is simple to implement since it does not require knowledge of the sparsity structure of the Hessian, or knowledge of the separability of the objective function. Furthermore, the amount of storage needed can be controlled by the user. It has been found that, in general, TN performs better than L-BFGS for functions that are nearly quadratic, while for highly nonlinear functions L-BFGS outperforms TN.[28]

These aspects and others are discussed in an excellent review on minimization methods by

Schlick,[29] who together with Fogelson also programmed their TN algorithm and included it in the package TNPACK.[18,19] This package enables the user to supply a sparse preconditioning matrix that transfers the Hessian into a matrix with more clustered eigenvalues, which in turn enhances convergence. This implementation of TN differs from that of Nash,[17] which uses automatic preconditioning and has been applied to a variety of problems with considerable success; the latter has the advantage of easy portability, because the preconditioner does not have to be tailored to the specific problems at hand. In her review, Schlick presents systematic efficiency comparisons between several algorithms applied to the molecule deoxycytine (87 variables) and to clusters of water molecules. For the former system, TN with preconditioning is found to be the most efficient requiring $\sim$2 times less CPU time than L-BFGS with preconditioning, while for the water clusters the picture is more complex.

Derreumaux et al.[30] tested the efficiency of TN as applied to peptides and proteins modeled by the CHARMM force field[31] using an updated version of TNPACK. In this implementation, the preconditioner is based on the short-range interactions, i.e., the bond stretching and bending, and the torsional potentials. It is shown that for several molecules of sizes $n = 12$ to 1299 atoms, TNPACK with preconditioning outperforms the steepest descent, nonlinear conjugate gradient, adapted basis Newton-Raphson, and Newton-Raphson algorithms installed in the CHARMM package.[30] More recently, Xie and Schlick showed that for molecules of sizes $n = 22$ to 2030 atoms, TNPACK requires less CPU time than both CG and L-BFGS, and reaches very low gradient norms.[32]

In the present paper, we study the relative performance of L-BFGS, TN, and, as a reference, also of conjugate gradient (CG) as applied to two cyclic peptides of 31 and 62 united atoms and to the protein bovine pancreatic trypsin inhibitor (BPTI) of 568 united atoms. These molecules are described by the GROMOS87 force field,[33] rather than by CHARMM; in addition, the two cyclic peptides are modeled by the GROMOS energy together with an implicit free energy solvation term derived previously for a peptide in DMSO.[9] This term, which is based on the solvent accessible surface area of the atoms, is expected to increase the nonlinearity of the potential energy function and thus constitutes an important test case that has not been studied before. We use the TN algorithm of Nash with an automatic preconditioner, the L-BFGS algorithm of Liu and Nocedal,[15,34] and the CG algorithm of Shanno and Phua.[35] The performance of the algorithms is compared with respect to the CPU time, the number of iterations, and the magnitude of the final energy and gradient. The results are analyzed in light of theories developed by Nash and Nocedal[28] and Axelsson and Lindskog[36,37] which rely on the eigenvalues and other quantities derived from the Hessian. To the best of our knowledge, this is the first study where such an analysis has been applied to optimization problems of biomacromolecules.

Theory and Methods

The Molecular Model

The intramolecular interactions are described by the GROMOS 37D4 united atom force field,[33] which defines the molecular energy $E_{\rm GRO}$. $E_{\rm GRO}$ consists of harmonic bond stretching and bending, torsional and improper torsional potentials, and non-bonded 6-12 Lennard-Jones and electrostatic interaction terms,

$$\begin{aligned} E_{\rm GRO} = &\sum_{n=1}^{N_b} \frac{1}{2} K_{b_n} [b_n - b_{0_n}]^2 + \sum_{n=1}^{N_\theta} \frac{1}{2} K_{\theta_n} [\theta_n - \theta_{0_n}]^2 + \sum_{n=1}^{N_\xi} \frac{1}{2} K_{\xi_n} [\xi_n - \xi_{0_n}]^2 \\ &+ \sum_{n=1}^{N_\phi} K_{\phi_n} [1 + \cos(m_n \phi_n - \delta_n)] + \sum_{i,j;j \geq i+4} \frac{A_{ij}}{r_{ij}^{12}} - \frac{B_{ij}}{r_{ij}^6} + \frac{C_{ij}}{r_{ij}} \end{aligned} \tag{1}$$

b_n is the bond length, θ_n, ϕ_n, and ξ_n are the valence, dihedral, and improper dihedral angles, respectively; the subscript 0 denotes equilibrium values. N_b, N_θ, N_ϕ, and N_ξ, are the number of bonds and valence, dihedral, and improper dihedral angles in the molecule, respectively; K_{b_n}, K_{θ_n}, K_{ϕ_n}, and K_{ξ_n}, are the corresponding force constants, m_n is an integer within the interval [1,6], and $\delta_n = 0$ or π. A_{ij} and B_{ij} are Lennard-Jones force constants for atom pair i and j separated by distance r_{ij}. C_{ij} is the corresponding electrostatic interaction constant which is defined for dielectric constant $\epsilon = 1$. In GROMOS 37D4 the hydrogen atoms are treated as collapsed on their first neighboring atoms except for hydrogens bonded to a nitrogen, oxygen or a sulphur.

Solvent effects can be taken into account implicitly by adding to $E_{\rm GRO}$ a free energy solvation term $E_{\rm sol}$,

$$E_{\rm tot} = E_{\rm GRO} + E_{\rm sol} = E_{\rm GRO} + \sum_i \sigma_i A_i \tag{2}$$

where A_i is the (conformational dependent) solvent accessible surface area (SASA) of atom i, and σ_i is the corresponding atomic solvation parameter (ASP). As has been pointed out in the Introduction, we also test the performance of the minimizers as applied to the two peptides described by $E_{\rm tot}$ with the ASPs derived in our previous work[9] for a cyclic peptide in DMSO.

The SASA is defined as the surface traced by the center of a spherical probe as it is rolled over the surface of the molecule; for DMSO the probe radius is 3Å. This area is calculated analytically with the program MSEED[38] which is based on a modification of the analytical equations presented by Connoly[39] and Richmond.[40] Use is made of the global Gauss-Bonet formula that describes the closed boundary of a regular region bounded by simple, piece-wise regular curves. The program provides analytical derivatives of the SASA with respect to the Cartesian coordinates, which are required by the present minimizers. One problem with $E_{\rm tot}$ is the possible occurrence of discontinuities in the gradient of A_i. This might stop the minimization process close to a local minimum, when the contributions to the gradient from all the components are small. In fact, gradient norms of only up to $\sim 10^{-3}$ kcal/(mol·Å) have been found to be attainable with $E_{\rm tot}$. For more information on this problem, see Ref. 38. Notice also that early termination of the minimization process might be a result of flatter minima expected to occur with the inclusion of $E_{\rm solv}$.

Description of the Algorithms

In this work we test implementations of the CG, L-BFGS, and TN optimization algorithms that are well documented in the literature. Thus, we use the nonlinear CG algorithm of Shanno and Phua[35] included in CONMIN, and the L-BFGS version VA15[15,34] in the Harwell library; the TN method is that described by Nash.[17] A brief description of the major components of each algorithm is given below. For a molecule of n atoms, we use the following notations: $f_k = f(\mathbf{x}_k)$ denotes the potential energy function $E_{\rm GRO}$ [eq. (1)] or $E_{\rm tot}$ [eq. (2)], where $\mathbf{x}_k$ is the $3n$ vector of the Cartesian coordinates at the k^{th} iteration. $\mathbf{g}_k = g(\mathbf{x}_k) = \nabla f_k$ is the gradient vector of size $3n$, and $H_k = \nabla^2 f_k$ is the $3n \times 3n$ symmetric Hessian matrix of the second partial derivatives of f with respect to the coordinates. In all the three algorithms, the new iterate is calculated from

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k \mathbf{p}_k \tag{3}$$

where $\mathbf{p}_k$ is the descent direction vector, and α_k is the step length. Iterations are terminated when

$$\|\mathbf{g}_k\|_\infty = \max_i \mid g_k^i \mid < 10^{-6}(1+ \mid f_k \mid) \tag{4}$$

where g_k^i is the $i^{\rm th}$ component of the vector $\mathbf{g}_k$. The necessary changes in the programs were made to ensure that all three algorithms use the same termination criterion. Also, the three methods use the same line search which is based on cubic interpolation, and is subject to the so called *strong Wolfe conditions*,[41]

$$\begin{aligned} f(\mathbf{x}_k) - f(\mathbf{x}_k + \alpha_k \mathbf{p}_k) &\geq -\mu \alpha_k \mathbf{p}_k^T \mathbf{g}_k \\ \mid \mathbf{g}(\mathbf{x}_k + \alpha_k \mathbf{p}_k)^T \mathbf{p}_k \mid &\leq \eta \mid \mathbf{g}_k^T \mathbf{p}_k \mid \end{aligned} \tag{5}$$

where $0 < \mu < \eta < 1$, and the superscript T denotes transpose.

The nonlinear conjugate gradient algorithm. CG uses the analytic derivatives of f, defined by $\mathbf{g}_k$. A step along the current negative gradient vector is taken in the first iteration; successive directions are constructed so that they form a set of mutually conjugate vectors with respect to the Hessian. At each step, the new iterate is calculated from eq. (3) and the search directions are expressed recursively as

$$\mathbf{p}_k = -\mathbf{g}_k + \beta_k \mathbf{p}_{k-1} \tag{6}$$

Calculation of β_k with the algorithm incorporated in CONMIN is described by Shanno.[42] Automatic restarting is used to preserve a linear convergence rate. For restart iterations, the search direction $\alpha_k = 1$. On the other hand, for nonrestart iterations,

$$\alpha_{k+1} = \frac{\alpha_k \mathbf{g}_k^T \mathbf{p}_k}{\mathbf{g}_{k+1}^T \mathbf{p}_{k+1}} \tag{7}$$

The limited memory BFGS algorithm. The L-BFGS method is an adaptation of the BFGS method to large problems, achieved by changing the Hessian update of the latter. Thus, in BFGS,[43,44] eq. (3) is used with an approximation $\tilde{H}_k$ to the inverse Hessian which is updated by

$$\tilde{H}_{k+1} = V_k^T \tilde{H}_k V_k + \rho_k \mathbf{s}_k \mathbf{s}_k^T \tag{8}$$

where $V_k = I - \rho_k \mathbf{y}_k \mathbf{s}_k^T$, $\mathbf{s}_k = \mathbf{x}_{k+1} - \mathbf{x}_k$, $\mathbf{y}_k = \mathbf{g}_{k+1} - \mathbf{g}_k$, $\rho_k = 1/(\mathbf{y}_k^T \mathbf{s}_k)$, and I is the identity matrix. The search direction is given by

$$\mathbf{p}_{k+1} = -\tilde{H}_{k+1} \mathbf{g}_{k+1} \tag{9}$$

In L-BFGS, instead of forming the matrices $\tilde{H}_k$ explicitly (which would require a large memory for a large problem) one only stores the vectors $\mathbf{s}_k$ and $\mathbf{y}_k$ obtained in the last m iterations which define $\tilde{H}_k$ implicitly; a cyclical procedure is used to retain the latest vectors and discard the oldest ones. Thus, after the first m iterations, eq. (8) becomes

$$\begin{aligned}
\tilde{H}_{k+1} = {} & (V_k^T \dots V_{k-m}^T) \tilde{H}_{k+1}^0 (V_{k-m} \dots V_k) \\
& + \rho_{k-m} (V_k^T \dots V_{k-m+1}^T) \mathbf{s}_{k-m} \mathbf{s}_{k-m}^T (V_{k-m+1} \dots V_k) \\
& + \rho_{k-m+1} (V_k^T \dots V_{k-m+2}^T) \mathbf{s}_{k-m+1} \mathbf{s}_{k-m+1}^T (V_{k-m+2} \dots V_k) \\
& \vdots \\
& + \rho_k \mathbf{s}_k \mathbf{s}_k^T
\end{aligned} \tag{10}$$

with the initial guess $\tilde{H}_{k+1}^0$, which is the sparse matrix,

$$\tilde{H}_{k+1}^0 = \frac{\mathbf{y}_{k+1}^T \mathbf{s}_{k+1}}{\mathbf{y}_{k+1}^T \mathbf{y}_{k+1}} I \tag{11}$$

Many previous studies have shown that typically $3 \leq m \leq 7$, where $m > 7$ does not improve the performance of L-BFGS.

The truncated Newton algorithm. In TN, a search direction is computed by finding an *approximate* solution to the Newton equations,

$$H_k \mathbf{p}_k = -\mathbf{g}_k \tag{12}$$

The use of an approximate search direction is justified because an exact solution of the Newton equation at a point far from the minimum is unnecessary and computationally wasteful in the framework of a basic descent method. Thus, for each *outer* iteration [eq. (12)], there is an *inner* iteration loop making use of the conjugate gradient method that computes this approximate direction, $\mathbf{p}_k$.

The conjugate gradient inner algorithm is preconditioned by a scaled two-step limited memory BFGS method with Powell's restarting strategy used to reset the preconditioner periodically. A detailed description of the preconditioner may be found in Ref. 45. The Hessian vector product $H_k\mathbf{v}$ for a given $\mathbf{v}$ required by the inner conjugate gradient algorithm is obtained by a finite difference approximation,

$$H_k\mathbf{v} \approx [\mathbf{g}(\mathbf{x}_k + h\mathbf{v}) - \mathbf{g}(\mathbf{x}_k)]/h \tag{13}$$

A major issue is how to adequately choose h;[24] in this work we use $h = \sqrt{\epsilon}(1 + \|\mathbf{x}_k\|)$, where ϵ is the machine precision and $\|\cdot\|$ denotes the Euclidian norm. The inner algorithm is terminated using the quadratic truncation test which monitors a sufficient decrease of the quadratic model $q_k = \mathbf{p}_k^T H_k \mathbf{p}_k/2 + \mathbf{p}_k^T \mathbf{g}_k$,

$$(1 - q_k^{i-1}/q_k^i) \leq c_q/i \tag{14}$$

where i is the counter for the inner iteration and c_q is a constant, $0 < c_q \leq 1$. The inner algorithm is also terminated if an imposed upper limit on the number of inner iterations, M, is reached, or when a loss of positive-definiteness is detected in the Hessian (i.e. when $\mathbf{v}^T H_k \mathbf{v} < 10^{-12}$). TN methods can be extended to more general problems which are not convex in much the same way as Newton's method (see Ref. 46).

The Initial Tuning of the Algorithms

The initial tuning of the algorithms was obtained from four randomly selected conformations generated for problems 1 and 4 (see Table I and the beginning of the next section) by varying the parameters over a wide range of permissible values. In most cases E_{GRO} and E_{tot} have led to the same optimal parameters. For eq. (5), these are $\mu = 10^{-4}$ for the three algorithms, $\eta = 0.25$ for L-BFGS, and $\eta = 0.9$ for TN and CG; all of them are the default values.

L-BFGS was found to be most efficient with $m = 6$ [eq. (10)], except for BPTI, where the algorithm failed unless $m = 1$ was used. For larger values of m, the line-search was found to fail in early stages of the minimization process and changing the parameters μ and η did not remedy the problem. This suggests that L-BFGS is intrinsically sensitive to instantaneous losses of positive definiteness whereas TN can accomodate them. Typically such a failure is preceded by a sudden increase in the value of $\|\mathbf{g}_k\|/\|\mathbf{g}_0\|$. Thus, we have plotted $\log(\|\mathbf{g}_k\|/\|\mathbf{g}_0\|)$ versus the number of inner iterations for $m = 2$ for 16 different minimizations (see next section). We have always found an increase in the gradient ratio prior to the failure of the line search (data not shown). It should be pointed out that using CHARMM, Xie and Schlick applied L-BFGS successfully to BPTI and the larger protein lysozyme with m larger than 1,[32] which suggests that the present failure of L-BFGS is related to the GROMOS force field. The further investigation for elucidating this point is beyond the scope of the present paper.

For TN, the default value, $c_q = 0.5$ is the best for the quadratic truncation test [eq. (14)]. For this algorithm we also modified M, the upper limit of the number of inner iterations, and found that $M \geq 25$ led approximately to the same best performance which is, however, decreased by up to a factor of 2 for $M < 25$; hence the default value $M = \max[N/2, 50]$ was used. In the finite difference approximation of TN [eq. (13)], we tested several choices of h, such as $h = \sqrt{\epsilon}/\|\mathbf{v}\|$, $h = 2(1 + \|\mathbf{x}_k\|)\sqrt{\epsilon}/\|\mathbf{v}\|$, and $h = 2(1 + \|\mathbf{x}_k\|)\sqrt{\epsilon}/\|\mathbf{v}\|$; none of them led to better results than the adopted default, $h = \sqrt{\epsilon}(1 + \|\mathbf{x}_k\|)$. All the calculations were carried out in double precision on an SGI O_2 workstation with an R10000 processor and 192 Mbytes of memory. The machine precision is $\epsilon = 10^{-15}$.

Results and Discussion

Numerical Tests

Our test systems are listed in Table I together with the corresponding numbers of atoms n. The two cyclic peptides are *cyclo*-(D-Pro^1-Ala^2-Ala^3-Ala^4-Ala^5) (n=31) and axinastatin 2, *cyclo*-(Asn^1-Pro^2-Phe^3-Val^4-Leu^5-Pro^6-Val^7) (n=62) whose structures were investigated with nuclear magnetic resonance by Kessler's group.[47,48] The x-ray atomic coordinates of BPTI (58 residues, n =568) were taken from the Brookhaven Protein Data Bank (PDB) (entry 4pti[49]); the interior water molecules were omitted in the present calculations. Usually, in large biological applications inclusion of all the non-bonded interactions is not feasible. On the other hand, when cut-offs are introduced, the minimization is known to be difficult and can lead to false minima due to function discontinuities. In this study, all the non-bonded interactions are taken into account for the five problems studied; however, for BPTI we also test 8 and 15 Å cut-off distances.

After optimizing the parameters, the algorithms were applied to a variety of initial structures. For each of problems 1, 2, 4, and 5, fifty conformations were generated randomly; some of them conformed with the geometrical requirements of the molecule (i.e., realistic bond lengths and angles, and no excluded volume violations) while the rest were highly distorted. For BPTI, 16 new conformations were generated in addition to the PDB structure by selecting at random $\sim$10% of the dihedral angles of the molecule, and rotating them randomly within the range $\pm 10^\circ$ around their PDB structure values.

The general behavior of each algorithm was found to be similar for most of the conformations. However, to make the efficiency comparisons as reliable as possible, we selected for each problem of Table I only two initial structures for which the minimized energy values, E_f obtained by the three algorithms are approximately the same. Even when the E_f results differ significantly, the minimized structures are verified to be very close, i.e. to have small rms deviations. For example, in problem 5a of Table II, the difference between E_f of TN and L-BFGS is $\sim$2 kcal/mol, whereas the all-atom rms deviation is 0.9Å. Even in problem 3b, where E_f(L-BFGS)$-E_\mathrm{f}$(TN)$\approx$ 50 and E_f(CG)$-E_\mathrm{f}$(TN)$\approx$ 80 kcal/mol, the rms deviations are relatively small, 0.9 and 1.2 Å, respectively. Note that early termination of the minimization process due to the failure of the line search in problems 4a, 4b, 5a, and 5b occurred at a tolerance of $\sim 10^{-2}$ instead of the specified condition 10^{-6} [eq. (4)] for all the algorithms. This occurs either due to the discontinuities in the gradient of E_sol or because of the flatter minima resulting from this potential; these issues are discussed in the section, The Molecular Model. In Problem 3b, the PDB structure of BPTI is minimized.

Let us discuss first the results obtained with the GROMOS energy E_GRO. Table II shows that for problems 1-3, TN is superior to both L-BFGS and CG in terms of CPU time (up to a factor of 2), and the number of function/gradient evaluations. Note that for molecular models based on a force field as the present one, calculation of the energy and its gradient is time consuming and therefore the number of function/gradient evaluations is a crucial factor determining the CPU time.

A close examination of Table II reveals that the most reliable assessment of the efficiency can be obtained from problems 1a, 1b, 2a, and 2b, where the same energy minima E_f are attained by the three algorithms. Here, TN is faster by a factor of $\sim$1.2 - 1.4 than L-BFGS and $\sim$1.8 - 2.1 than CG. On the other hand, for BPTI (without nonbonded energy cut-off), TN usually leads to lower E_f values (see problems 3a and 3b). In 3b (the initial PDB structure), the minimum found by TN is considerably lower than the two other minima, which also differ significantly from each other. This precludes drawing clear-cut conclusions about the efficiency. In this respect, 3a is a better problem since E_{f}(TN) is only 2.7 kcal/mol lower than E_{f} obtained by the other algorithms, and the rms deviations are 0.2 Å. Here, TN is about 1.5 and 1.8 times faster than L-BFGS and CG, respectively. As pointed out previously, we also minimized the energy of BPTI using nonbonded cut-offs of 8 and 15 Å with neighbor updates of every 1-20 steps. We have found that TN is faster than CG by an average factor of 1.2 (data not shown). On the other hand, minimizations with L-BFGS based on the 16 starting conformations for the above updates and $m \geq 1$ have failed. This indicates that handling of function discontinuities with L-BFGS is inferior to that of TN and CG.

The relative efficiency of the three algorithms changes significantly for E_{tot} [eq. (2)], which includes the solvation energy E_{sol}. Thus, as demonstrated by problems 4 and 5 in Table II, L-BFGS becomes the best performer, which is 1.6 - 2.4 and 1.9 - 4.4 times faster than TN and CG, respectively. However, because of the early termination of the minimization process, the most reliable comparison would be with respect to problems 4a and 5b, where the E_{f} values are the closest. In these cases, L-BFGS is almost twice as fast as TN. Problem 5b is probably more suitable than 4a for judging the performance of CG since the corresponding CG minimized energies are 0.04 and 0.51 kcal/mol above the respective true values. For 5b we find that CG is 4 and 2 times slower than L-BFGS and TN, respectively.

The quality of the minimized energy can also be measured by the value of the gradient norm, $||\mathbf{g}||$, at the final iteration. The values of $\log(||\mathbf{g}_k||/||\mathbf{g}_0||)$ for axinastatin 2 with E_{GRO} (2a) and E_{tot} (4a) are displayed in Figure 1. A close examination of this figure shows that the final value of $\log(||\mathbf{g}_k||/||\mathbf{g}_0||)$ is slightly lower for TN than for L-BFGS (-5.93 versus -5.85), which is in turn followed by CG (-5.60). This is a general observation, characterizing our large number of minimizations. The gap between these values is even larger with E_{tot}, as can be seen in Figure 1b [$\log(||\mathbf{g}_k||/||\mathbf{g}_0||) = -5.99, -5.80, -5.32$ for TN, L-BFGS, and CG, respectively]. Note that in problem 4a the minimized values increase in the same order for the three algorithms. The relative efficiency of TN stems from the relatively small number of outer iterations required to reach the final solution. Therefore, any improvement in the inner algorithm to reduce the number of function/gradient evaluations, while maintaining a similar level of accuracy in the line search would further enhance the performance of this method. The other two algorithms tested here lack this potential.[24]

Computational Behavior of the Algorithms

Nash and Nocedal prescribe certain criteria that are relevant to the convergence theory and computational behavior of algorithms.[28] The first criterion is the deviation from quadratic.

One defines the Taylor series approximation of the gradient, DQ,

$$\mathrm{DQ} = \frac{\|\mathbf{g}_0 - \mathbf{g}_\mathrm{f} - H_\mathrm{f}(\mathbf{x}_0 - \mathbf{x}_\mathrm{f})\|_\infty}{\|(\mathbf{x}_0 - \mathbf{x}_\mathrm{f})\|_\infty} \tag{15}$$

where $\mathbf{x}_0$ and $\mathbf{x}_\mathrm{f}$ are the starting and final structures, respectively; DQ provides a measure of the size of the third derivatives.

Another criterion is the Hessian condition number, $K = \lambda_N/\lambda_1$, where λ_N and λ_1 are the largest and smallest eigenvalues of the Hessian, respectively. We calculate the condition numbers of the Hessian at $\mathbf{x}_0$ and $\mathbf{x}_\mathrm{f}$ and denote them by K_0 and K_f, respectively. In general, CG algorithms are expected to converge faster as $K \to 1$. The convergence also depends strongly on the eigenvalue structure, which is known to affect the performance of the inner conjugate-gradient procedure of TN.[50] An additional criterion used by Nash and Nocedal[28] is the function convexity. In our problems, the eigenvalues of the Hessian at $\mathbf{x}_0$ and $\mathbf{x}_\mathrm{f}$ were calculated for each of the test problems and were all found to be positive, which implies that the Hessians are positive definite and the energy functions are strictly convex at the initial structure and near the solution. Hence, it is reasonable to compute condition numbers at these points (K_0 and K_f).

The characteristics of the test problems are presented in Table III. For BPTI the results are calculated only for the PDB structure (problem 3b), because computing the Hessian explicitly without using a cut-off on the non-bonded interactions is time consuming ($\approx$ 4 days of CPU time). Note that the calculation of DQ and K_f is based on the TN results, which are of the highest quality.

The force field E_GRO [eq. (1)] consists of both quadratic terms (i.e., bond stretching and bending potentials of order n) and highly nonlinear non-bonded terms (of order n^2). DQ, which is equal to 0 for a quadratic function, is expected to increase with increasing the molecular size, due to the dominant number of the non-bonded terms. The nonlinear implicit solvation free energy E_sol in eq. (2) is expected to increase DQ as well.

The DQ values fall within the range $10^2 - 10^5$ for all our large number of test cases, where the values of 3b and 4a are significantly higher than the others. The high value of BPTI (3b) probably reflects its relatively large size. The large nonlinearity of 4a may be attributed to the large difference between the starting and minimized energies ($E_0 = 69917.00$ versus $E_\mathrm{f} = -40.69$ kcal/mol, see Table I). However, we also find that 5b has a smaller energy difference than 5a (0.74 versus 8.15 kcal/mol) while the corresponding DQ values are 876 and 317. Therefore, the large DQ value of 4a stems, perhaps, from a highly distorted initial structure, which exposes a relatively large number of atoms to the solvent, contributing thereby to the nonlinearity through E_sol.

Nevertheless, it is not possible to infer which algorithm will perform better by simply comparing the DQ values. Similarly, the condition numbers, K_0 and K_f, which are in the range $\sim 10^6 - 10^9$ do not indicate a particular preference towards the winning algorithm and, unlike with DQ, they do not provide information on the nonlinearity. In general, there is no obvious correlation between the values of DQ, K_0, and K_f and the performance of the algorithms.

We have also investigated differences between the spectra of the eigenvalues, λ_i in order to understand the behavior of the algorithms with E_GRO and E_tot. We examine the effect of minimization on the eigenvalue distribution using the spectrum representation of Axelsson and Lindskog.[36] In Figure 2a, the eigenvalues are displayed for $\mathbf{x}_0$ (upper panel) and $\mathbf{x}_\mathrm{f}$ (lower panel) of problem 1a; in Figure 2b similar information is presented for problem 3b. The figure shows that in all cases a small group of the lowest eigenvalues is separated from the rest, the minimization increases their number and sometimes shifts their location. In fact, in all cases, there are exactly six isolated lowest eigenvalues in the minimized structures.

Minimization with E_tot affects the low end spectra differently. Typical spectra of axinastatin 2 after minimization with E_GRO and E_tot are shown in Figure 3. The spectrum with E_tot of the initial structure is similar to those obtained with E_GRO (top panels of Fig. 2a and 2b) and therefore is not displayed. Figure 3 shows that there are only three isolated eigenvalues with E_tot (problem 5b) as opposed to the six obtained with E_GRO (problem 2b). Also, the lowest eigenvalue of the continuous part of the spectrum is smaller by a factor of $\sim$10 for E_tot than for E_GRO; this factor increases up to $\sim$100 in other test structures.

The picture provided by Figs. 2 and 3 explains the different behavior of the optimization algorithms with the two potential energy functions: The convergence rate of a conjugate gradient algorithm is governed by an *effective* condition number $K' = \lambda_N/\lambda_{l+1}$, where l is the number of small, isolated eigenvalues.[37,50] Therefore, CG and also TN, which uses the conjugate gradient procedure in its inner iterations, are both expected to perform better with E_GRO than with E_tot, since $\lambda_N(E_\mathrm{GRO}) \approx \lambda_N(E_\mathrm{tot})$ but $\lambda_{l+1}(E_\mathrm{GRO}) > \lambda_{l+1}(E_\mathrm{tot})$ at the minimum point; the latter inequality can range from one to two orders of magnitude, resulting in K' values that are systematically smaller with E_GRO. On the other hand, the L-BFGS algorithm seems to be largely unaffected by the eigenvalue distribution.

Notice, however, that this conclusion from Fig. 3 is drawn from two minimizations that are effectively based on different termination criteria [see eq. (4)]. Therefore, to put the comparison on an equal footing, we have also calculated from E_GRO the spectra of several minimized structures of axinastatin 2 using a termination criterion of 10^{-2}; this number is close to the actual values observed with E_tot. The distribution of the eigenvalues has been found to be unaffected by this relatively loose termination criterion, which verifies the validity of our analysis.

Finally, it should be pointed out that λ_i at E_f are proportional to ω_i^2, where ω_i are the frequencies that would be obtained in a normal mode analysis of the present molecular models, assuming atoms of equal mass. In an exact solution, the six frequencies of translation and rotation are zero and they correspond to the six low eigenvalues of the present models that are, however, non-zero due to numerical approximations.

Conclusions

The relative performance of the minimization algorithms, L-BFGS, TN, and CG has been investigated as applied to two cyclic peptides and the protein BPTI described by different potential energy functions. With the GROMOS force field E_{GRO} alone, TN was found to be the clear winner. Similar results were also obtained in a recent study by Xie and Schlick using the CHARMM force field; for example, for the protein lysozyme (2030 atoms), it was found that TNPACK is three times faster than L-BFGS and reaches lower gradient norms.[32] On the other hand, L-BFGS is known to be particularly well-suited to handle highly nonlinear problems.[28] The nonlinearity is expected to increase when a solvation term, which depends on the solvent accessible surface area is added to E_{GRO}. Indeed, with the corresponding total energy function E_{tot}, L-BFGS becomes the winner with respect to computing time and the number of function/gradient calculations; notice, however, that TN has provided solutions of slightly better quality, i.e. with lower minimized energy and gradient values. Thus, in these problems the user is faced with a choice of accuracy versus convergence rate.

The performance of the various algorithms is explained here in terms of parameters that depend on the distribution of the eigenvalues of the Hessian, in particular, the effective condition number K', which is the ratio between the largest eigenvalue and the smallest one after ignoring the small isolated eigenvalues. K' is 1-2 orders of magnitude larger for E_{tot}, where L-BFGS performs the best. This, however, does not indicate an improvement in L-BFGS, but rather a deterioration in the efficiency of the CG procedure, which is heavily used also in TN.

The results obtained here are valid for the GROMOS87 force field, although similar observations are expected from other force fields because of the similarity in the energy terms of the potential energy function [eq. (1)] which usually only differ in the values of the various constants. A future study comparing the behavior of such minimizers on different force fields (e.g. CHARMM and GROMOS) for extreme molecular sizes will clarify these expectations.

Finally, it would be of interest to compare the performance of the present version of TN to that used in TNPACK, as applied to biomolecules described by force fields with and without implicit solvation terms. The TN procedure of Nash tested here[17] is convenient to use due to its automatic preconditioner; on the other hand, TNPACK of Schlick and Fogelson[18] allows the user to optimize the preconditioner for the specific problems. It will be important to check whether the gain in efficiency which is potentially achievable with the latter algorithm justifies the extra amount of work involved.

Acknowledgments

We acknowledge support from the Florida State University (FSU) Supercomputer Computations Research Institute, which is partially funded by the US Department of Energy (DOE) under contract number DE-FC05-85ER250000. This work was also supported by DOE grant number DE-FG02-97ER62490. We would like to acknowledge support by FSU through the allocation of supercomputer resources. We thank Professor Tamar Schlick for her insightful comments during the revision of the manuscript.

References

1. M. Vásquez, G. Némethy, and H. A. Scheraga, *Chem. Rev.* **94**, 2183 (1994).
2. N. Gō and H. A. Scheraga, *J. Chem. Phys.* **51**, 4751 (1969).
3. K. D. Gibson and H. A. Scheraga, *Physiol. Chem. Phys.* **1**, 109 (1969).
4. A. T. Hagler, P. S. Stern, R. Sharon, J. M. Becker, and F. Naider, *J. Am. Chem. Soc.* **101**, 6842 (1979).
5. M. Karplus and J. N. Kushick, *Macromolecules* **14**, 325 (1981).
6. H. Meirovitch, E. Meirovitch, and J. Lee, *J. Phys. Chem.* **99**, 4847 (1995).
7. D. A. Case, *Curr. Opin. Struct. Biol.* **4**, 285 (1994).
8. C. Baysal and H. Meirovitch, *J. Phys. Chem. A* **101**, 2185 (1997).
9. C. Baysal and H. Meirovitch, *J. Am. Chem. Soc.* **120**, 800 (1998).
10. M. Vásquez and H. A. Scheraga, *Biopolymers* **24**, 1437 (1985).
11. Z. Li and H. A. Scheraga, *Proc. Natl. Acad. Sci. U.S.A.* **84**, 6611 (1987).
12. G. Chang, W. C. Guida, and W. C. Still, *J. Am. Chem. Soc.* **111**, 4379 (1989).
13. M. Saunders, K. N. Houk, Y.-D. Wu, W. C. Still, M. Lipton, G. Chang, and W. C. Guida, *J. Am. Chem. Soc.* **112**, 1419 (1990).
14. I. Kolossváry and W. C. Guida, *J. Am. Chem. Soc.* **118**, 5011 (1996).
15. D. C. Liu and J. Nocedal, On the Limited Memory BFGS Method for Large Scale Optimization, Technical Report NAM03, Department of Electrical Engineering and Computer Science, Northwestern University, 1988.
16. R. S. Dembo, S. C. Eisenstat, and T. Steihaug, *SIAM J. Numer. Anal.* **19**, 400 (1982).
17. S. G. Nash, User's Guide for TN/TNBC: Fortran Routines for Nonlinear Optimization, Report 397, Mathematical Sciences Department, The Johns Hopkins University, Baltimore, MD, 1984.
18. T. Schlick and A. Fogelson, *ACM Trans. Math. Softw.* **18**, 46 (1992).
19. T. Schlick and A. Fogelson, *ACM Trans. Math. Softw.* **18**, 71 (1992).

20. J. Nocedal, *Acta Numer.* , 199 (1991).

21. I. M. Navon, X. Zou, M. Berger, P. K. H. Phua, T. Schlick, and F. X. LeDimet, In *Optimization Techniques and Applications, Vol. 1*, P. K. H. Phua, Ed., World Scientific Publishing Co., 1992, p. 33.

22. I. M. Navon, X. Zou, M. Berger, P. K. H. Phua, T. Schlick, and F. X. LeDimet, In *Optimization Techniques and Applications, Vol. 1*, P. K. H. Phua, Ed., World Scientific Publishing Co., 1992, p. 445.

23. X. Zou, I. M. Navon, M. Berger, M. K. Phua, T. Schlick, and F. LeDimet, *SIAM J. Opt.* **3**, 582 (1993).

24. Z. Wang, I. M. Navon, X. Zou, and F. X. LeDimet, *Comput. Opt. App.* **4**, 241 (1995).

25. I. M. Navon, F. Brown, and D. H. Robertson, *Computers Chem.* **14**, 305 (1990).

26. D. H. Robertson, B. F. Brown, and I. M. Navon, *J. Chem. Phys.* **90**, 3221 (1989).

27. Z. Wang and K. Droegemeier, *Comput. Opt. App.* **10**, 283 (1998).

28. S. G. Nash and J. Nocedal, *SIAM J. Opt.* **1**, 358 (1991).

29. T. Schlick, *Rev. Comput. Chem.* **3**, 1 (1992).

30. P. Derreumaux, G. Zhang, T. Schlick, and B. Brooks, *J. Comput. Chem.* **15**, 532 (1994).

31. B. R. Brooks, R. E. Bruccoleri, B. D. Olafson, D. J. States, S. Swaminathan, and M. Karplus, *J. Comput. Chem.* **4**, 187 (1983).

32. D. Xie and T. Schlick, *SIAM J. Opt.* **in press**.

33. W. F. van Gunsteren and H. J. C. Berendsen, *Groningen Molecular Simulation (GROMOS) Library Manual*, Biomos, Nijenborgh 16 9747 AG Groningen NL, 1987.

34. D. C. Liu and J. Nocedal, *Math. Program.* **45**, 503 (1989).

35. D. F. Shanno and K. H. Phua, *ACM Trans. Math. Softw.* **6**, 618 (1980).

36. O. Axelsson and G. Lindskog, *Numer. Math.* **48**, 479 (1986).

37. O. Axelsson and G. Lindskog, *Numer. Math.* **48**, 499 (1986).

38. G. Perrot, B. Cheng, K. D. Gibson, J. Villa, K. A. Palmer, A. Nayeem, B. Maigret, and H. A. Scheraga, *J. Comput. Chem.* **13**, 1 (1992).

39. M. L. Connoly, *J. Appl. Cryst.* **16**, 548 (1983).

40. T. J. Richmond, *J. Mol. Biol.* **178**, 63 (1984).

41. P. E. Gill and W. Murray, Conjugate Gradient Methods for Large-Scale Nonlinear Optimization, Report SOL 79-15, Department of Operations Research, Stanford University, Stanford, CA, 1979.

42. D. F. Shanno, *Math. Oper. Res.* **3**, 244 (1978).

43. J. E. Dennis, Jr. and J. J. Moré, *SIAM Rev.* **19**, 46 (1977).

44. J. E. Dennis, Jr. and R. B. Schnabel, *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*, Prentice-Hall, Englewood Cliffs, NJ, 1983.

45. S. G. Nash, *SIAM J. Sci. Stat. Comput.* **6**, 599 (1985).

46. S. G. Nash, *SIAM J. Numer. Anal.* **21**, 770 (1984).

47. D. L. Mierke, M. Kurz, and H. Kessler, *J. Am. Chem. Soc.* **116**, 1042 (1994).

48. O. Mechnich, G. Hessler, H. Kessler, M. Bernd, and B. Kutscher, *J. Am. Chem. Soc.* **116**, 1042 (1994).

49. J. Deisenhofer and W. Steigemann, *Acta. Cryst. Sect. B* **31**, 238 (1975).

50. Y. Notay, *Numer. Math.* **65**, 301 (1993).

Figure Captions

Figure 1. The variation $\log(||\mathbf{g}_k||/||\mathbf{g}_0||)$ with the number of iterations k for CG, L-BFGS, and inner iterations for TN; $||\mathbf{g}_k||$ is the norm of the gradient. (a) problem 2a; (b) problem 4a.

Figure 2. Log scaled distribution of the Hessian eigenvalues in the initial iteration (upper panel) and final iteration (lower panel) of (a) problem 1a and (b) problem 3b.

Figure 3. Log scaled distribution of the Hessian eigenvalues at the final iteration of problems 2b (upper panel) and 5b (lower panel).

Table I: List of Test Problems

Problem	Molecule	Function [a]	n [b]
1	*cyclo*(D-Pro-Ala$_4$)	E_{GRO}	31
2	Axinastatin 2	E_{GRO}	62
3	BPTI	E_{GRO}	568
4	*cyclo*(D-Pro-Ala$_4$)	E_{tot}	31
5	Axinastatin 2	E_{tot}	62

[a] Algorithms are tested with potential energy functions E_{GRO}[eq. (1)] or E_{tot} [eq. (2)]. [b] Number of atoms in each test problem.

Table III: Problem Characteristics

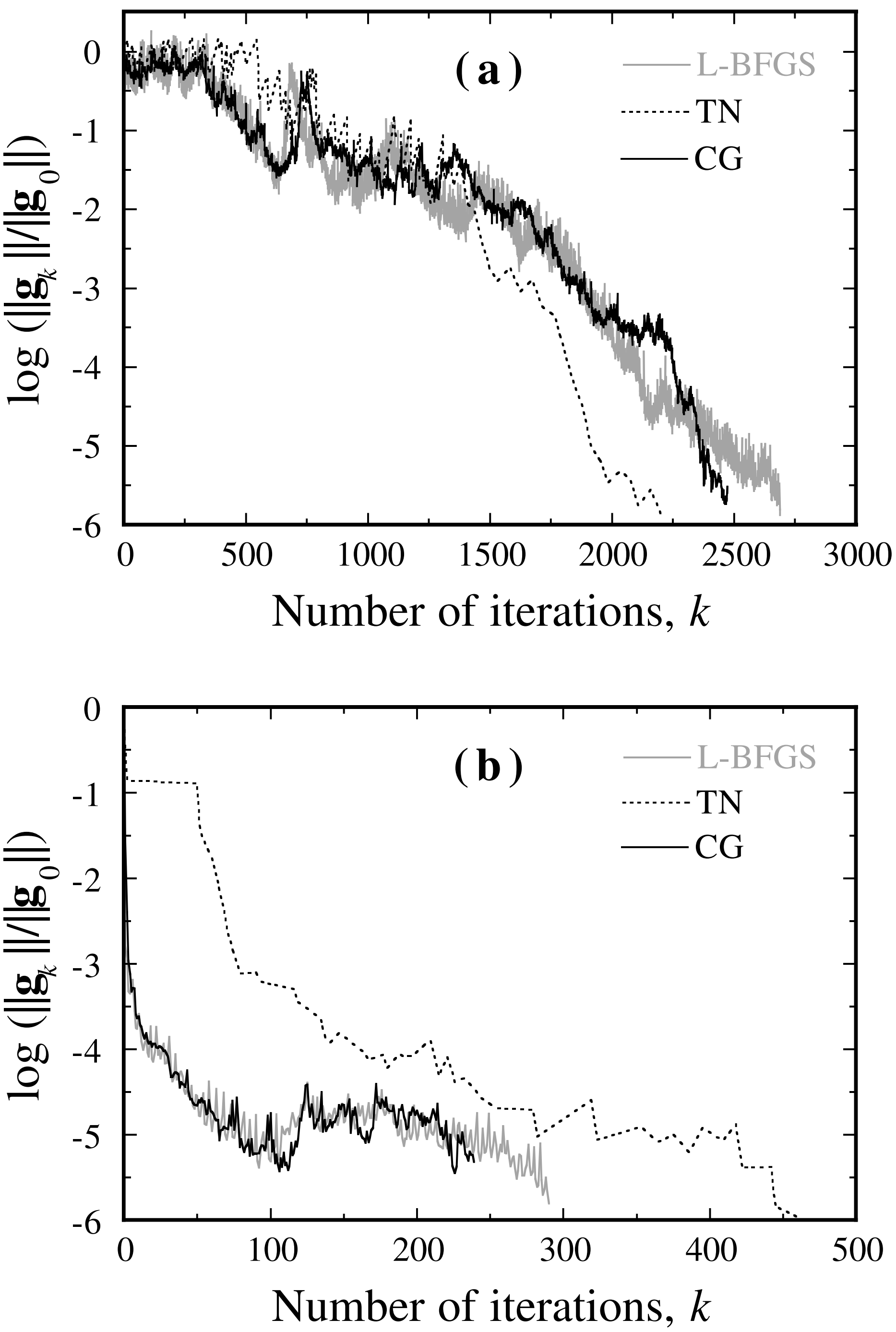
P [a]	DQ [b]	K_0 [c]	K_f [d]	Winner	Percent difference[e]	
					f-g	time
1a	378	6.3×10^7	5.8×10^7	TN	4	18
1b	151	1.4×10^9	7.5×10^8	TN	26	44
2a	226	2.9×10^8	5.8×10^7	TN	10	16
2b	322	4.5×10^7	4.8×10^7	TN	24	33
3a				TN	49	47
3b	8518	9.3×10^5	7.5×10^7	TN	8	7
4a	7460	3.1×10^8	2.1×10^8	L-BFGS	67	89
4b	148	2.2×10^7	4.8×10^7	L-BFGS	162	138
5a	316	7.5×10^7	5.2×10^7	L-BFGS	63	57
5b	876	5.2×10^7	9.7×10^7	L-BFGS	105	100

[a] Problem number corresponding to that listed in Table I; different initial structures are denoted by a or b. [b] A measure of the deviation of a particular minimum from quadratic [eq. (15)] [c] Condition number of the initial structure. [d] Condition number of the final minimized structure. [e] A measure of the amount by which the winner algorithm is better than its closest follower in terms of function/gradient evaluations (f-g) or CPU time.

Table II: Performance comparison of L-BFGS, TN, and CG

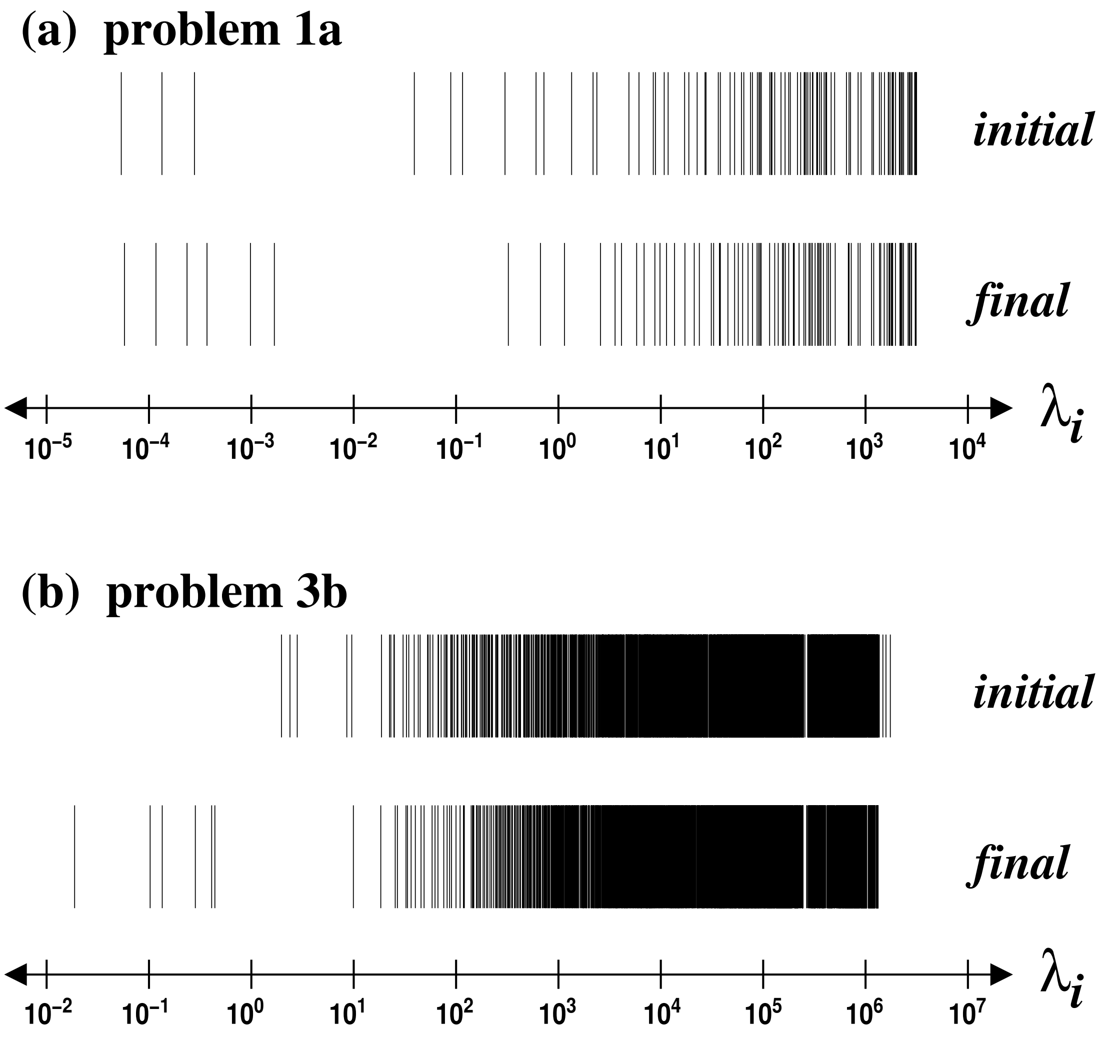
		L-BFGS				TN				CG			
P [a]	E_0 [b]	E_f [c]	It [d]	f-g [e]	time [f]	E_f	It	f-g	time	E_f	It	f-g	time
1a	31.49	9.60	1017	1053	1.36	9.60	76	1009	1.15	9.60	889	1789	2.11
1b	24.05	21.47	597	613	0.79	21.47	33	487	0.55	21.47	491	985	1.14
2a	25.73	−38.86	2688	2765	11.7	−38.86	137	2509	10.1	−38.86	2473	4979	19.9
2b	−37.52	−43.49	1266	1298	5.58	−43.49	47	1043	4.18	−43.49	933	1874	7.51
3a	−824.60	−4032.77	4824	5197	1617.7	−4035.46	125	3481	1100.7	−4032.77	3174	6114	2006.4
3b	−2239.13	−4057.97	13152	14108	4430.6	−4106.45	241	6088	1888.9	−4024.90	3228	6546	2012.2
4a	69917.00	−40.68	290	301	5.12	−40.69	50	594	9.80	−40.18	239	502	9.69
4b	−44.83	−45.12	72	110	1.92	−45.68	20	288	4.56	−45.09	76	158	8.47
5a	−74.04	−80.31	186	211	8.41	−82.19	23	344	13.20	−81.59	244	498	29.68
5b	−120.64	−121.38	166	174	6.92	−121.38	17	356	13.81	−121.34	115	235	27.84

[a] Problem number corresponding to that listed in Table I; results for different initial structures are denoted by a or b. [b] The potential energy of the initial structure in kcal/mol. [c] Final minimized energy in kcal/mol [d] Number of conjugate gradient iterations for CG, and number of Newton iterations for L-BFGS and TN. [e] Total number of function/gradient evaluations. [f] CPU time in seconds



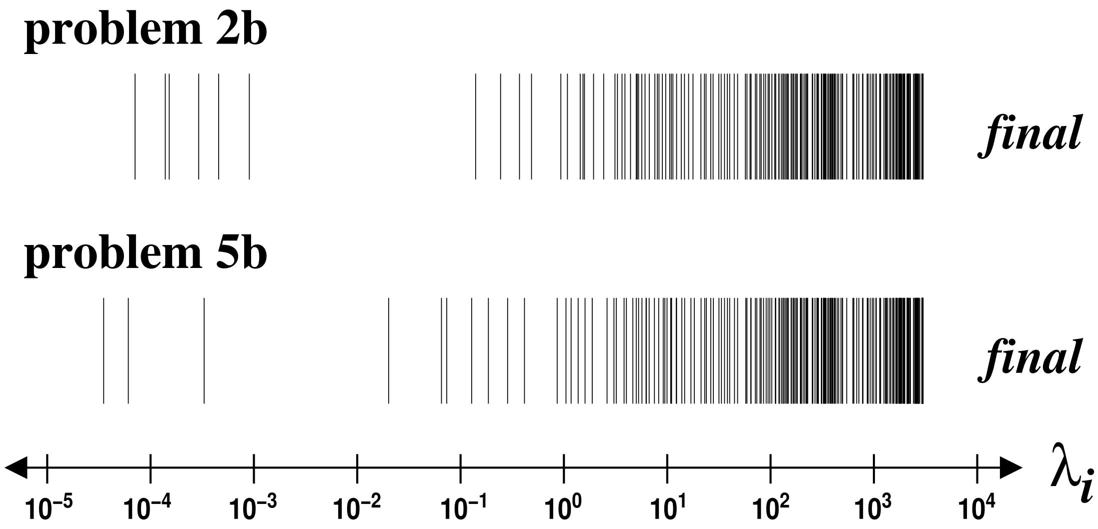
(a)
(b)
L-BFGS
TN
CG
log (||$\mathbf{g}_k$||/||$\mathbf{g}_0$||)
Number of iterations, k
0
-1
-2
-3
-4
-5
-6
0
500
1000
1500
2000
2500
3000
100
200
300
400
500

Figure 1:



(a) problem 1a
initial
final
λ_i
10^{-5} 10^{-4} 10^{-3} 10^{-2} 10^{-1} 10^{0} 10^{1} 10^{2} 10^{3} 10^{4}
(b) problem 3b
initial
final
λ_i
10^{-2} 10^{-1} 10^{0} 10^{1} 10^{2} 10^{3} 10^{4} 10^{5} 10^{6} 10^{7}

Figure 2:



problem 2b
final
problem 5b
final
λ_i
10^{-5}
10^{-4}
10^{-3}
10^{-2}
10^{-1}
10^{0}
10^{1}
10^{2}
10^{3}
10^{4}

Figure 3: