Journal of Global Optimization **27:** 375–397, 2003.

Solving Large Scale Molecular Distance Geometry Problems by a Smoothing Technique via the Gaussian Transform and D.C. Programming

LE THI HOAI AN
Laboratory of Modelling, Optimization and Operations Research, National Institute for Applied Sciences-Rouen, BP 8, F-76 131 Mont Saint Aignan Cedex, France (e-mail: lethi@insa-rouen.fr)

(Received 6 November 2001; accepted in revised form 27 February 2003)

Abstract. We study a continuation approach via the Gaussian transform and D.C. programming for solving both exact and general distance geometry problems. This approach relies on a new formulation of the problems and their Gaussian transforms which are both smooth D.C. (difference of convex functions) programs. A D.C. optimization algorithm is investigated for solving the transformed problems. Numerical experiments on the data derived from PDB data bank up to 4189 atoms show the usefulness of the reformulation, the globality of sought solutions, the robustness and the efficiency of the proposed approach.

Key words: continuation method, D.C. algorithm (DCA), distance geometry problems, Gaussian transform, molecular optimization, reformulation

1. Introduction

The determination of a molecular conformation can be tackled by either minimizing a potential energy function (if the molecular structure corresponds to the global minimizer of this function) or solving the distance geometry problem [7, 10] (when the molecular conformation is determined by distances between pairs of atoms in the molecule). Both methods are concerned with global optimization problems. The *molecular exact distance geometry problem* consists in finding positions $x^1, \ldots, x^n$ of n atoms in $\mathbb{R}^3$ such that

$$\|x^i - x^j\| = \delta_{ij}, (i, j) \in \mathcal{S}, \tag{1}$$

where $\mathcal{S}$ is a subset of the atom pairs, δ_{ij} with $(i,j) \in \mathcal{S}$ is the given distance between atoms i and j, and $\|\cdot\|$ denotes the Euclidean norm. Usually, a small subset of pairwise distances is known, i.e., $\mathcal{S}$ is small, and in practice, lower and upper bounds of the distances are given instead of the exact values. Further, it should be noted that, by the error in the theoretical or experimental data, there may not exist a solution to this problem, then an ε-optimal solution of (1), namely a configuration

$x^1, \ldots, x^n$ satisfying

$$| \|x^i - x^j\| - \delta_{ij} | \leq \varepsilon, (i, j) \in \mathcal{S}, \tag{2}$$

is useful. When only the lower and upper bounds of δ_{ij} are given, we have the so-called *general distance geometry problem* which consists of finding a set of positions $x^1, \ldots, x^n$ in $\mathbb{R}^3$ such that

$$l_{ij} \leq \|x^i - x^j\| \leq u_{ij}, (i, j) \in \mathcal{S}, \tag{3}$$

where l_{ij} and u_{ij} are lower and upper bounds of the distance constraints, respectively.

The data of (3) can be succinctly represented by a graph $G(N, \mathcal{S})$. The vertices $N = \{1, ..., n\}$ correspond to the atoms and an edge $(i, j) \in \mathcal{S}$ connects vertices i and j if the bounds l_{ij} and u_{ij} of the distance between the corresponding atoms are known. Throughout this paper, we assume that the graph $G(N, \mathcal{S})$ is connected. This assumption is not restrictive for Problem (3) since it can be decomposed into a number of smaller problems otherwise.

The standard formulation of (3), due to Crippen and Havel [7], is in terms of globally solving the nonconvex program

$$0 = \min \left\{ f(x^1, \ldots, x^n) = \sum_{(i,j)\in S} p_{ij}\theta_{ij}(x^i - x^j) : x^1, ..., x^n \in \mathbb{R}^3 \right\}, \tag{4}$$

where the pairwise function $\theta_{ij} : \mathbb{R}^n \mapsto \mathbb{R}$ is defined by

$$\theta_{ij}(x) = \min^2 \left\{ \frac{\|x\|^2 - l_{ij}^2}{l_{ij}^2}, 0 \right\} + \max^2 \left\{ \frac{\|x\|^2 - u_{ij}^2}{u_{ij}^2}, 0 \right\}. \tag{5}$$

It is clear that $f \in C^1$ but $f \notin C^2$.

The distance geometry problems are very important in molecular biology, and have recently attracted a fair amount of attention within the optimization community. Several methods have been proposed for solving the distance geometry problems (1) and/or (3), among them it is worth citing the interesting approaches such as the ABBIE algorithm [11], the continuation methods based on the Gaussian transform [13, 15], the stochastic/perturbation algorithm [21], the αBB algorithm [8], the semi-definite programming approach [1], the matrix completion approach [12], and the D.C. (different of convex functions) optimization algorithms called DCA [3, 5]. For a rather complete lists of references the reader is referred to [5, 14]. Practically, one is often faced with very large scale problems for which global methods like branch and bound are expensive. So it is worth investigating local approaches conjointly with global techniques such as smoothing. An advantage of local approaches for these problems is that the optimal value is a priori known, therefore it will be easy to recognize whether the algorithm provides a global minimizer.

Very recently, the DCA have been extensively studied for solving the exact distance geometry problem [5] and the general distance geometry problem ([3]). It has been shown in [3, 5] that the DCA can be well exploited to obtain efficient algorithms that solve both exact and general large scale distance geometry problems. Note that the standard optimization problem (4)–(5) is a D.C. program but the objective D.C. function is too complex and inconvenient for DCA. To simplify matters, the objective function in [3] to the general distance geometry problem (3) has been chosen in an *elegant* way (a nonsmooth nonconvex function)

$$0 = \min \left\{ \sum_{(i,j)\in\mathcal{S}} p_{ij}(\|x^i - x^j\| - t_{ij})^2 : x^1, ..., x^n \in \mathbb{R}^3, l_{ij} \leqslant t_{ij} \leqslant u_{ij}, (i, j) \in \mathcal{S} \right\} \tag{6}$$

which makes it possible to express DCA in a simple form; it actually requires matrix-vector products and only one Cholesky factorization, and allows one to exploit sparsity in the large scale setting. The algorithms proposed in [3] are in fact composed of two phases: the first phase consists of finding a good starting point for Phase 2. It begins by completing the missing distance matrix (using the shortest paths between the pairs of atoms) and then solves the exact distance geometry problem in which all "distances" are known by the DCA. The second phase is the DCA applied to the original problem. The algorithms work well for the artificial test problems where a protein contains at most 4096 atoms and for the real world models derived from the PDB data bank (http:www.rcsb.org/pdb/) with up to 4189 atoms. A disadvantage of these methods is that, although the strategy of Phase 1 is quite suitable for DCA to reach global solutions to the distance geometry problem, it is quite expensive (the running time of Phase 1 is equal to that of Phase 2).

To get around this drawback, more precisely, to find a good starting point of DCA without using Phase 1, we propose in the present paper a combined DCA-smoothing technique. Instead of (6) we consider a new D.C. formulation with a smooth (actually infinitely differentiable) objective function:

$$0 = \min \left\{ \sum_{(i,j)\in\mathcal{S}} p_{ij}(\|x^i - x^j\|^2 - t_{ij}^2)^2 : x^1, ..., : x^n \in \mathbb{R}^3, l_{ij} \leqslant t_{ij} \leqslant u_{ij}, (i, j) \in \mathcal{S} \right\}. \tag{7}$$

Like (6), this formulation corresponds to both exact and general distance geometry problems, because in the exact distance geometry problem one has $l_{ij} = u_{ij} = \delta_{ij}$, then we take $t_{ij} = \delta_{ij}$ and (7) becomes (8). The new formulation has several advantages which are favourable to the use of DCA as well as smoothing techniques via the Gaussian transform. First, the objective function is infinitely differentiable, and its Gaussian transform can be computed explicitly. Second, the

transformed problem is a *nice* D.C. program for which DCA is fast because its works only on vector products and does not require the Cholesky factorization. Third, from the numerical point of view, although the additional variables t_{ij} increase the dimension of the problem, they do not really cause any trouble, since the problem is solved separately in variables x and t. However, the introduction of t_{ij} in (7) is crucial to get an explicit Gaussian transformed function of the original objective function.

The idea of the use of the continuation approach via the Gaussian transform for distance geometry problems is not new: Moré and Wu [13] proposed an algorithm for solving the exact distance geometry problem (1) in the form

$$0 = \min \left\{ \sum_{(i,j)\in\mathcal{S}} p_{ij} \left(\delta_{i,j}^2 - \|x^i - x^j\|^2\right)^2 : x^1, ..., x^n \in \mathbb{R}^3 \right\} \tag{8}$$

by the Gaussian smoothing technique via the trust region method. Computational experiments with up to 648 variables ($n = 216$) in [13] proved that the continuation method is more reliable and efficient than the multistart approach, a standard procedure for finding the global minimizer to (8). However, we observe that the trust region method applied to the sequence of subproblems in the continuation approach is expensive and thus may not be efficient in the large scale setting. For the general distance geometry problem (3), using the formulation (4)-(5) Moré and Wu [15] introduced the dgsol algorithm based on the Gauss-Hermite transform and the variable-metric limited-memory code MINPACK-2. Computational experiments on protein fragments with 100 and 200 atoms from the PDB data bank showed that the dgsol code is also more efficient than the multistart algorithm.

It is worth noting that our continuation approach for the distance geometry problems is completely different from that of Moré-Wu's work: with the formulation (7) we get exactly and explicitly the Gaussian transformed function while with the formulation (4)–(5) Moré and Wu [15] were able only to get an approximation to the Gaussian transformed function, say the Gauss–Hermite approximation which is quite complicated and inconvenient to use DCA. On the other hand, our optimization method for the transformed problem is based on D.C. programming and DCA while Moré and Wu used the trust region method in [13] (for the exact distance geometry problem) and the limited-memory code in [15]. Featured as a *descent method without line-search*, DCA is at present one of a few algorithms in the local approach which has been successfully applied to many large-scale D.C. optimization problems and proved to be more robust and efficient than the related standard methods (see [2, 4, 5, 18, 19] and references therein).

Our present method has several advantages. First, by using the continuation approach which traces the minimizers of the smooth function back to the original function, the Phase 1 in [3, 5] is no longer needed. Second, since the DCA applied to Gaussian transformed problems works only on vector products and does not

require the Cholesky factorization, it is efficient in large scale problems and faster than the trust region approach. Third, we can exploit sparsity of the given distance matrix. This is important because only a small subset of constraints is known in practice.

We have tested our algorithm on a set of data derived from PDB data and compared it with the two phase algorithm **GDCA** in [3]. Numerical experiments show that the proposed algorithm solves these problems more efficiently and reliably than **GDCA** [3].

The paper is organized as follows. In Section 2 we compute the Gaussian transform of the objective function. The DCA for solving the smoothing distance geometry problem is presented in Section 3. Our main algorithm in the continuation approach is described in Section 4 and the numerical experiments as well as the conclusion are reported in Section 5.

2. The Gaussian Transform of the Objective Function of (7)

In general the distance geometry problems are defined in arbitrary dimensions p. Here, for molecular conformations, the problems are considered in $\mathbb{R}^3$. However the results in this paper hold for any dimension p.

In Euclidean distance geometry problems, we must take into consideration the symmetry of both the subset $\mathcal{S}$ (i.e., $(i, j) \in \mathcal{S}$ implies $(j, i) \in \mathcal{S}$) and the weight matrix $P = p_{ij}$). Let $\mathcal{S}^w$ be the set defined by

$$\mathcal{S}^w := \{(i, j) \in \{1, \ldots, n\} \times \{1, \ldots, n\} \quad \text{s.t.} \quad (i, j) \in \mathcal{S}, i < j\}. \tag{9}$$

To alleviate calculations in the sequel, we consider problem (7) in the form:

$$(GDGP) \quad \inf \left\{ \frac{1}{4} \sum_{(i,j)\in\mathcal{S}^w} p_{ij}(\|x^i - x^j\|^2 - t_{ij}^2)^2 : x^1, ..., x^n \in \mathbb{R}^3, l_{ij} \right. \tag{10}$$
$$\left. \leq t_{ij} \leq u_{ij}, (i, j) \in \mathcal{S}^w \right\}.$$

Since only the pairs $(i, j) \in \mathcal{S}^w$ are involved in the constraints, we denote by $ind(i, j)$ the indices of the pairs (i, j) in the set $\mathcal{S}^w$ and consider only the elements t_{ij} with $(i, j) \in \mathcal{S}^w$. Let T be the column-vector in $\mathbb{R}^d$ ($d = |\mathcal{S}^w|$) defined by these elements: $T = (T_1, \ldots, T_{ind(i,j)}, \ldots, T_d)^T \in \mathbb{R}^d$ with $T_{ind(i,j)} = t_{ij}$ for $(i, j) \in \mathcal{S}^w$. Let also X be the column-vector in $\mathbb{R}^{p.n}$ that contains n elements $x^j \in R^p$, $j = 1, \ldots, n$, say $X = \begin{pmatrix} x^1 \\ . \\ . \\ . \\ x^n \end{pmatrix}$. Denote by $c(i, j)$ the indices in X of $i-th$ component of the vector x^j in R^p. Then $c(i, j) = p(j - 1) + i$ for $i = 1, ...p$, $j = 1, ..., n$,

and $X_{c(i,j)} = (x^j)_i$. Define now the pairwise functions $d_{ij} : \mathbb{R}^{pn} \longmapsto \mathbb{R}$ and $\varphi_{ij} : \mathbb{R}^{\mathrm{d}} \longmapsto \mathbb{R}$ by $d_{ij}(X) := \|x^i - x^j\|^2$, $\varphi_{ij}(T) := T^2_{ind(i,j)} = t^2_{ij}$. Then (GDGP) can be rewritten in the form

$$(GDGP) \quad 0 = \inf \left\{ F(X,T) := \frac{1}{4} \sum_{(i,j)\in\mathcal{S}^w} p_{ij}(d_{ij}(X) - \varphi_{ij}(T))^2 : \right. \tag{11}$$
$$\left. (X,T) \in \mathbb{R}^{\mathrm{pn}} \times \mathcal{C} \right\},$$

where

$$\mathcal{C} := \{T \in \mathbb{R}^d : l_{ij} \le t_{ij} \le u_{ij}, (i,j) \in \mathcal{S}^w\}. \tag{12}$$

We have

PROPOSITION 1 *(i) If a set of positions* $(x^1, ..., x^n)$ *is a solution to problem (3), then the corresponding* (X,T) *verifying* $t_{ij} = \|x^i - x^j\|$ *for* $(i,j) \in \mathcal{S}$ *is a solution to (GDGP).*
(ii) If (X,T) *is a solution to (GDGP), then the set of positions* $(x^1, ..., x^n)$ *is a solution to (3) and* $t_{ij} = \|x^i - x^j\|$ *for* $(i,j) \in \mathcal{S}$.

The Gaussian transform of a function $f : \mathbb{R}^n \to \mathbb{R}$, denoted by $\langle f \rangle_\lambda$, is defined as

$$\langle f \rangle_\lambda(x) = \frac{1}{\pi^{n/2}\lambda^n} \int_{\mathbb{R}^n} f(y) \exp\left(\frac{-\| y - x \|^2}{\lambda^2}\right) \mathrm{dy}$$

or again, by the change of variable $y \mapsto x + \lambda u$:

$$\langle f \rangle_\lambda(x) = \frac{1}{\pi^{n/2}} \int_{\mathbb{R}^n} f(x + \lambda u) \exp\left(-\| u \|^2\right) \mathrm{du}.$$

In this section we compute the Gaussian transform of $F(X,T)$, the objective function of (GDGP), by using the basic results on the computational Gaussian transform of functions given in [13]. Let $h_{ij} : \mathbb{R}^p \times \mathbb{R} \mapsto \mathbb{R}$ be the function defined by

$$h_{ij}(x,t) := (\|x\|^2 - \frac{1}{2}t^2)^2. \tag{13}$$

Then we can express $F(X,T)$ in the form

$$F(X,T) := \frac{1}{4} \sum_{(i,j)\in\mathcal{S}^w} p_{ij} h_{ij}(x^i - x^j, \sqrt{2}t_{ij}), \tag{14}$$

or

$$F(X,T) := \frac{1}{4}\sum_{(i,j)\in\mathcal{S}^w} p_{ij}h_{ij}(\mathcal{P}_{ij}^T(X,T)), \tag{15}$$

where $\mathcal{P}_{ij}$ is the $(pn+d)\times(p+1)$-matrix with $\mathcal{P}_{ij}^T = \begin{pmatrix} Q_{ij} & 0_{d\times 1} \\ 0_{1\times np} & q_{ij} \end{pmatrix}$. Here Q_{ij} is the $(pn \times p)-$ matrix defined by

$$Q_{ij} = (e_{c(1,i)} - e_{c(1,j)}, ..., e_{c(p,i)} - e_{c(p,j)}), \tag{16}$$

q_{ij} is the row-vector in $\mathbb{R}^d$ such that $[q_{ij}]_{ind(i,j)} = \sqrt{2}, [q_{ij}]_k = 0, \forall k \neq ind(i,j)$, and $e_k \in \mathbb{R}^{p.n}$ is the unit vector with one in the k^{th} component and zero otherwise.

Let $F_{ij} : \mathbb{R}^{pn} \times \mathbb{R}^d \mapsto \mathbb{R}$ be such that $F_{ij}(X,T) = h_{ij}(\mathcal{P}_{ij}^T(X,T))$. Then we have

$$\langle F\rangle_\lambda(X,T) = \frac{1}{4}\sum_{(i,j)\in\mathcal{S}^w} p_{ij}\langle F_{ij}\rangle_\lambda(X,T) = \frac{1}{4}\sum_{(i,j)\in\mathcal{S}^w} p_{ij}\langle h_{ij}\rangle_{\sqrt{2}\lambda}(x^i - x^j, \sqrt{2}t_{ij}), \tag{17}$$

where the first equality is obtained from the linear property of the operator $\langle F\rangle_\lambda$ (see [13], p. 819) and the second by applying Theorem 4.1 of [13].

So computing the Gaussian transform of F amounts to computing that of h_{ij} which is quite simple. Since $h(x,t) = \|x\|^4 - \|x\|^2t^2 + \frac{1}{4}t^4$, the linear property of the operator $\langle h\rangle_\lambda$ implies that

$$\langle h\rangle_\lambda(x,t) = \langle h_1\rangle_\lambda(x) + \langle h_2\rangle_\lambda(x,t) + \langle h_3\rangle_\lambda(t), \tag{18}$$

where $h_1(x) := \|x\|^4, h_2(x,t) := -\|x\|^2t^2, h_3(t) := \frac{1}{4}t^4$. An application of Theorem 3.4 of [13] gives

$$\langle h_1\rangle_\lambda(x) = \|x\|^4 + (2+p)\lambda^2\|x\|^2 + \frac{1}{4}p(p+2)\lambda^4, \tag{19}$$

$$\langle h_3\rangle_\lambda(t) = t^4 + 3\lambda^2t^2 + \frac{3}{4}\lambda^4. \tag{20}$$

On the other hand, noting that $h_2(x,t) := -\|x\|^2t^2 = -\sum_{i=1}^p x_i^2t^2$ and using the formulation of the Gaussian transform of a decomposable function given in [13], (p. 820), we get

$$\langle h_2\rangle_\lambda(x,t) = -\sum_{i=1}^p\left(x_i^2 + \frac{1}{2}\lambda^2\right)\left(t^2 + \frac{1}{2}\lambda^2\right) = -\left(t^2 + \frac{1}{2}\lambda^2\right)\left(\|x\|^2 + \frac{p}{2}\lambda^2\right). \tag{21}$$

So

$$\langle h\rangle_\lambda(x,t) = h(x,t) + \frac{1}{2}(3+2p)\lambda^2\|x\|^2 + \frac{1}{4}(\,3-2p)\lambda^2 t^2 + \frac{1}{16}(3+4p+4p^2)\lambda^4. \tag{22}$$

Finally, from (17) and (22) we get the expression of the Gaussian transform of F:

PROPOSITION 2 *If F: $\mathbb{R}^{pn} \times \mathbb{R}^{nn} \mapsto \mathbb{R}$ is defined by (14) and (13), then*

$$\langle F\rangle_\lambda(X,T) = F(X,T) + \left(\frac{1}{4}\sum_{(i,j)\in\mathcal{S}^w} p_{ij}\,[\,(3+2p)\lambda^2\|x^i - x^j\|^2 + (3-2p)\lambda^2 t_{ij}^2\,]\right) + \gamma, \tag{23}$$

where $\gamma = \frac{1}{16}(3+4p+4p^2)\lambda^4 \sum_{(i,j)\in\mathcal{S}^w} p_{ij}$.

The main subproblem in our continuation approach to solve (GDGP) is

$$(\text{GDGP})_\lambda \qquad \min\left\{\langle F\rangle_\lambda(X,T) : (X,T) \in \mathbb{R}^{p.n} \times \mathcal{C}\right\}.$$

3. DCA for Solving Problem $(\text{GDGP})_\lambda$

The DCA was introduced by Pham Dinh in 1986 [16, 17] as an extension of his subgradient algorithms (for convex maximization programming) to D.C. programming. However, this field has been really developed since 1994 by the joint work of Le Thi and Pham Dinh [3–5, 18, 19] and references therein). It is a primal-dual subgradient method for solving a general D.C. program of the form

$$(P_{dc}) \qquad \alpha = \inf\{f(x) := g(x) - h(x) : x \in \mathbb{R}^n\}$$

with $g,\ h \in \Gamma_o(\mathbb{R}^n)$ and its dual program

$$(D_{dc}) \qquad \alpha = \inf\{h^*(y) - g^*(y) : y \in \mathbb{R}^n\}.$$

Here $\Gamma_o(\mathbb{R}^n)$ denotes the set of all proper lower semi-continuous convex functions on $\mathbb{R}^n$ which is equipped with the canonical inner product $\langle\cdot,\cdot\rangle$. The dual space of $\mathbb{R}^n$ can be identified with $\mathbb{R}^n$ itself. The conjugate function g^* of $g \in \Gamma_o(\mathbb{R}^n)$ is defined by

$$g^*(y) = \sup\{\langle x, y\rangle - g(x) : x \in \mathbb{R}^n\}$$

and it belongs again to $\Gamma_o(\mathbb{R}^n)$. One says that $g - h$ is a D.C. decomposition of f, and g, h are its D.C. components. We note that any convex constrained D.C. program can be written in the standard form (P_{dc}) by adding the indicator function of the convex set of constraints in the objective function.

Based on the D.C. duality and the local optimality, the DCA consists of the construction of two sequences $\{x^k\}$ and $\{y^k\}$ such that x^{k+1} (resp. y^k) is a solution to the convex program (P_k) (resp. (D_k)), where (P_k) (resp. (D_k)) is obtained from (P_{dc}) (resp. (D_{dc})) by replacing h (resp. g^*) with its affine minorization defined by $y^k \in \partial h(x^k)$ (resp. $x^k \in \partial g^*(y^{k-1})$):

$$(P_k) \qquad \left\{\inf\{g(x) - [h(x^k) + \langle x - x^k, y^k\rangle] : x \in \mathbb{R}^n\right\},$$

$$(D_k) \qquad \left\{\inf\{h^*(y) - [g^*(y^{k-1}) + \langle x^k, y - y^{k-1}\rangle] : y \in \mathbb{R}^n\right\}.$$

In other words, the DCA yields the scheme:

$$y^k \in \partial h(x^k); \quad x^{k+1} \in \partial g^*(y^k). \tag{24}$$

By this way, the sequences $\{g(x^k) - h(x^k)\}$ and $\{h^*(y^k) - g^*(y^k)\}$ are decreasing in an appropriate way and the corresponding limit points x^∞ and y^∞ of $\{x^k\}$ and $\{y^k\}$ (there exist such x^∞'s and y^∞'s if both the sequences $\{x^k\}$ and $\{y^k\}$ are bounded) satisfy the local optimality condition

$$\partial h(x^\infty) \subset \partial g(x^\infty) \quad \textit{and} \quad \partial g^*(y^\infty) \subset \partial h^*(y^\infty), \tag{25}$$

or they are critical points of $g - h$ and $h^* - g^*$, respectively.

The crucial feature of the DCA is the *choice* of a *good* D.C. decomposition and a *good* initial point. These are still open questions to be studied. Of course, this depends strongly on the very specific structure of the problem being considered. In practice, for solving a given D.C. program, we try to choose g and h such that sequences $\{x^k\}$ and $\{y^k\}$ can be easily calculated, i.e. either they are in explicit form or their computations are inexpensive. For a detailed study of D.C. programming and DCA we refer the readers to [2, 4, 19]. DCA was successfully applied to a lot of different and various nonconvex optimization problems (see [2, 3, 4, 5, 18, 19] and references therein). In particular, for the exact and/or general distance geometry problems the two phase algorithms based on DCA [3, 5] are efficient: they globally solved large scale problems (the problem of 12567 variables in x corresponds to the protein contains 4189 atoms). This motivates us to use DCA for solving the main subproblem $(GDGP)_\lambda$ in this work.

In this section we develop a special DCA scheme for solving Problem $(GDGP)_\lambda$. To simplify the presentation we consider the following equivalent (by omitting the constant γ) problem which is also denoted by $(GDGP)_\lambda$:

$$(GDGP)_\lambda \quad \min\{F_\lambda(X, T) : (X, T) \in \mathbb{R}^{pn} \times \mathcal{C}\},$$

with

$$\begin{aligned} F_\lambda(X, T) :=& \frac{1}{4} \sum_{(i,j)\in\mathcal{S}^w} p_{ij}(d_{ij}(X) - \varphi_{ij}(T))^2 \\ &+ \frac{1}{4} \sum_{(i,j)\in\mathcal{S}^w} p_{ij}\left(9\lambda^2 d_{ij}(X) - 3\lambda^2\varphi_{ij}(T)\right). \end{aligned}$$

3.1. D.C. FORMULATION OF $(\mathrm{GDGP})_\lambda$

We first find a D.C. decomposition of $F_\lambda(X, T)$. By some simple operations we get

$$\begin{aligned} F_\lambda(X,T) \;=\; & \frac{1}{2}\sum_{(i,j)\in\mathcal{J}^w} p_{ij}\left(d_{ij}^2(X)+\varphi_{ij}^2(T)\right)+\frac{9}{4}\lambda^2\sum_{(i,j)\in\mathcal{J}^w} p_{ij}d_{ij}(X) \\ & -\frac{1}{4}\sum_{(i,j)\in\mathcal{J}^w} p_{ij}\left[\left(d_{ij}(X)+\varphi_{ij}(T)\right)^2+3\lambda^2\varphi_{ij}(T)\right]. \end{aligned}$$

Let L and H be the functions on $\mathbb{R}^{pn}\times\mathcal{C}$ defined by

$$L(X,T) := \frac{1}{2}\sum_{(i,j)\in\mathcal{J}^w} p_{ij}\left(d_{ij}^2(X)+\frac{9}{4}\lambda^2 d_{ij}(X)\right)+\frac{1}{2}\sum_{(i,j)\in\mathcal{J}^w} p_{ij}\varphi_{ij}^2(T), \tag{26}$$

and

$$H(X,T) := \frac{1}{4}\sum_{(i,j)\in\mathcal{J}^w} p_{ij}\left[\,d_{ij}(X)+\varphi_{ij}(T)\right]^2+3\lambda^2\varphi_{ij}(T)\big]. \tag{27}$$

Clearly, the function L is finite and convex on $\mathbb{R}^{pn}\times\mathbb{R}^d$. Since for every $(i,\ j)\in\mathcal{J}^w$, the function: $(X,T)\to d_{ij}(X)+\varphi_{ij}(T)$ is finite, convex and nonnegative, the function H is convex, too. Then the following D.C. decomposition of F_λ seems to be natural:

$$F_\lambda(X,T) := L(X,T)-H(X,T).$$

Let now $\chi_{\mathcal{C}}$ be the indicator function of $\mathcal{C}$ defined by $\chi_{\mathcal{C}}(T)=0$ if $T\in\mathcal{C}$, $+\infty$ otherwise. Then Problem $(\mathrm{GDGP})_\lambda$ can be expressed in the standard form of D.C. programs:

$$\min\left\{F_\lambda(X,T) := G(X,T)-H(X,T) : (X,T)\in\mathbb{R}^{pn}\times\mathbb{R}^d\right\}, \tag{28}$$

where G is a function defined on the whole space $\mathbb{R}^{pn}\times\mathbb{R}^d$: $G(X,T)=L(X,T)+\chi_{\mathcal{C}}(T)$.

Performing DCA scheme for solving (28) is reduced to calculating subdifferentials of the functions H and G^*:

$$(Y^{(k)},W^{(k)})\in\partial H(X^{(k)},T^{(k)}),\ (X^{(k+1)},T^{(k+1)})\in\partial G^*(Y^{(k)},W^{(k)}). \tag{29}$$

We shall now present the crucial results on the computation of ∂H and ∂G^*.

3.2. COMPUTATION OF ∂H

By the very definition of H we have

$$\partial H(X,T) = \frac{1}{2}\sum_{(i,j)\in \mathcal{S}^w} p_{ij}[d_{ij}(X)+\varphi_{ij}(T)][\partial d_{ij}(X)\times\{0\}+\{0\}\times\partial\varphi_{ij}(T)]$$

$$+\frac{1}{4}\sum_{(i,j)\in \mathcal{S}^w} p_{ij}[\{0\}\times 3\lambda^2\partial\varphi_{ij}(T)]. \tag{30}$$

So computing ∂H amounts to compute ∂d_{ij} and $\partial\varphi_{ij}$.

Clearly, φ_{ij} is differentiable: $\varphi_{ij}(T) = T^2_{ind(i,j)} = \langle T, E_{ind(i,j)}\rangle^2_{\mathbb{R}^d}$, with $E_{ind(i,j)} = (0,...,1_{ind(i,j)},...0)^T \in \mathbb{R}^d$, ($E_k \in \mathbb{R}^d$ is the unit vector with one in the k^{th} component and zero otherwise). Hence $\nabla\varphi_{ij}(T) = 2T_{ind(i,j)}E_{ind(i,j)}$.

On the other hand, the relation $Q_{ij}^T X = x^i - x^j$ (Q_{ij} is the $(pn\times p)-$ matrix defined in (16)) implies that ([20])

$$\partial d_{ij}(X) = Q_{ij}\partial(\|.\|^2)(Q_{ij}^T X) = 2Q_{ij}(x^i - x^j).$$

Hence d_{ij} is also differentiable and $\nabla d_{ij}(X) = 2Q_{ij}(x^i - x^j)$.

Since both φ_{ij} and d_{ij} are differentiable, H is differentiable too, and a gradient of H is explicitly defined by $\nabla H(X,T) = (Y,W)$ with

$$Y = \sum_{(i,j)\in \mathcal{S}^w} p_{ij}[d_{ij}(X)+\varphi_{ij}(T)]Q_{ij}(x^i - x^j), \tag{31}$$

$$W = \sum_{(i,j)\in \mathcal{S}^w} p_{ij}\left[(d_{ij}(X)+\varphi_{ij}(T)\right]T_{ind(i,j)}E_{ind(i,j)} + \frac{3}{2}\lambda^2\sum_{(i,j)\in \mathcal{S}^w} p_{ij}T_{ind(i,j)}E_{ind(i,j)}. \tag{32}$$

3.3. COMPUTATION OF ∂G^*

As aforementioned, a subgradient of G^* at $(Y^{(k)}, W^{(k)})$ is an optimal solution to the convex problem of the form (P_k):

$$\min\{G(X,T) - \langle (Y^{(k)}, W^{(k)}), (X,T)\rangle : (X,T) \in \mathbb{R}^{pn}\times\mathbb{R}^d\}.$$

The function G is separable in its variables, say $G(X,T) = \eta(X) + \zeta(T) + \chi_{\mathcal{C}}(T)$, where

$$\eta(X) := \frac{1}{2}\sum_{(i,j)\in \mathcal{S}^w} p_{ij}\left(d_{ij}^2(X) + \frac{9}{4}\lambda^2 d_{ij}(X)\right), \zeta(T) := \frac{1}{2}\sum_{(i,j)\in \mathcal{S}^w} p_{ij}\varphi_{ij}^2(T).$$

Hence the last problem can be decomposed into the following two ones:

$$\min\left\{\eta(X) - \langle Y^{(k)}, X\rangle : X \in \mathbb{R}^{pn}\right\}, \tag{33}$$

and

$$\min\left\{\zeta(T) - \langle W^{(k)}, T\rangle : T \in \mathcal{C}\right\}. \tag{34}$$

Since

$$\zeta(T) - \langle W^{(k)}, T\rangle = \frac{1}{2}\sum_{(i,j)\in\mathcal{S}^w} p_{ij}T^4_{ind(i,j)} - \sum_{(i,j)\in\mathcal{S}^w} W^{(k)}_{ind(i,j)}T_{ind(i,j)},$$

Problem (34) is again separable, and then $T^* = (T^*_{ind(i,j)})_{(i,j)\in\mathcal{S}^w}$ is a solution to (34) if and only if, for each $(i, j) \in \mathcal{S}^w$, $T^*_{ind(i,j)}$ is a solution to the one-dimensional convex optimization problem

$$\min\left\{\frac{1}{2}p_{ij}T^4_{ind(i,j)} - W^{(k)}_{ind(i,j)}T_{ind(i,j)} : l_{ij} \le T_{ind(i,j)} \le u_{ij}\right\}.$$

Hence, the optimal solution $T^* = (T^*_{ind(i,j)})$ of (34) is explicitly given by

$$T^*_{ind(i,j)} = \begin{cases} \sqrt[3]{\frac{W^{(k)}_{ind(i,j)}}{2p_{ij}}} & \text{if } l_{ij} \le \sqrt[3]{\frac{W^{(k)}_{ind(i,j)}}{2p_{ij}}} \le u_{ij}, \\ l_{ij} & \text{if } \sqrt[3]{\frac{W^{(k)}_{ind(i,j)}}{2p_{ij}}} < l_{ij}, \\ u_{ij} & \text{if } \sqrt[3]{\frac{W^{(k)}_{ind(i,j)}}{2p_{ij}}} > u_{ij}. \end{cases} \tag{35}$$

It remains to solve (33) for constructing the sequence $\{X^{(k)}\}$. Remark that since d_{ij} is differentiable, η is differentiable too, and X^* is a solution of (33) if and only if $Y^{(k)} = \nabla\eta(X^*)$. Hence we can obtain X^* by solving the nonlinear system $Y^{(k)} = \nabla\eta(X^*)$. Clearly, solving (33) by this approach is not efficient because it requires the solution of a nonlinear system.

The efficiency of the DCA suggests us to use it again in this work to solve Problem (33). As will be seen later, with a suitable choice of D.C. decompositions for the objective function of (33) the corresponding DCA is simple, inexpensive since it requires (at each iteration) only the projection of a point onto an Euclidean ball which is explicitly computed.

First of all, to guarantee the existence of a solution to (33), the boundedness of the sequence $\{X^{(k)}\}$ we will recast this problem in an appropriate space. Let $\mathcal{A}$ be the vector subspace of R^{pn} defined by

$$\mathcal{A} := \{X = (x^1, x^2, \ldots x^n)^T \in R^{pn} : x^1 = x^2 = \ldots = x^n\}. \tag{36}$$

Then the orthogonal space of $\mathcal{A}$ is

$$\mathcal{A}^{\perp} := \{X \in R^{pn} : \sum_{i=1}^{n} x^i = 0\}. \tag{37}$$

LEMMA 1 *Problem (33) is equivalent to*

$$\min\left\{\eta(X) - \langle Y^{(k)}, X\rangle : X \in \mathcal{A}^{\perp}\right\} \tag{38}$$

in the sense that if X^* is an optimal solution to (38), then $X^* + U$ is an optimal solution to (33) for all $U \in \mathcal{A}$.

Proof. For any $V \in R^{pn}$ we can write $V = U + X$ with $U \in \mathcal{A}$, $X \in \mathcal{A}^{\perp}$. Clearly, $d_{ij}(U) := \|u^i - u^j\|^2 = 0$ for all $U \in \mathcal{A}$. Then $\eta^{-1}(0) \supset \mathcal{A}$. In fact we have $\eta^{-1}(0) = \mathcal{A}$ in virtue of the connectedness of the graph $G(N, \mathcal{S})$ (see Introduction). On the other hand, from the definition of $Y^{(k)}$ in (31), we have $Y^{(k)} \in \mathcal{A}^{\perp}$, so $\langle U, Y^{(k)}\rangle = 0$ for all $U \in \mathcal{A}$. Hence, for any $V \in R^{pn}$, we have

$$\eta(V) - \langle V, Y^{(k)}\rangle = \eta(X) - \langle X, Y^{(k)}\rangle,$$

with $X \in \mathcal{A}^{\perp}$. The proof is then completed.

It is not difficult to verify that the feasible set $\mathcal{A}^{\perp}$ of Problem (38) enjoys some interesting properties (using $\eta^{-1}(0) = \mathcal{A}$ and the positive homogeneity of η):

(i) The objective function of (38), namely $\eta(X) - \langle Y^{(k)}, X\rangle$ is coercive on $\mathcal{A}^{\perp}$. Consequently, (38) has always a solution.

(ii) $\sum_{i<j} \|x^i - x^j\|^2 = n\|X\|^2$ for all $X = (x^1, x^2, \ldots x^n)^T \in \mathcal{A}^{\perp}$.

Define now the set Ω by $\Omega := \left\{X \in \mathcal{A}^{\perp} : \sum_{i<j} \|x^i - x^j\|^2 \le \sum_{i<j} u_{ij}^2\right\}$, where the upper bounds u_{ij} of distances $\|x^i - x^j\|^2$ for $(i, j) \notin \mathcal{S}^w$ is computed by using the relationships $u_{ij} = \min(u_{ij}, u_{ik} + u_{kj})$. Due to property (ii) Ω is in fact the Euclidean ball in $\mathcal{A}^{\perp}$: $\Omega = \left\{X \in \mathcal{A}^{\perp} : \|X\| \le r := \sqrt{\frac{1}{n}\sum_{i<j} u_{ij}^2}\right\}$. Then we can reformulate Problem (38) in the form

$$\min\left\{\eta(X) - \langle Y^{(k)}, X\rangle : X \in \Omega\right\}. \tag{39}$$

Hence the sequence $\{X^{(k)}\}$ defined from (39) is well defined and bounded.

To solve (39) by DCA, we compute a positive number ρ such that the function $\Psi(X) := \frac{1}{2}\rho\|X\|^2 - \eta(X)$ is convex (a lower bound of such a ρ can be easily determined) and consider the following D.C. decomposition

$$\eta(X) - \langle Y^{(k)}, X\rangle = \Phi(X) - \Psi(X) := \left[\frac{1}{2}\rho\|X\|^2 - \langle Y^{(k)}, X\rangle\right] - \left[\frac{1}{2}\rho\|X\|^2 - \eta(X)\right] \tag{40}$$

Then (39) can be written in the standard form of D.C. programs:

$$\min\left\{\chi_\Omega(X)+\left[\frac{1}{2}\rho\|X\|^2-\langle Y^{(k)},X\rangle\right]-\left[\frac{1}{2}\rho\|X\|^2-\eta(X)\right]: X\in\mathbb{R}^{pn}\right\}. \quad (41)$$

The DCA applied to (41) consists of generating the sequence $\{X^{(l)}\}$ such that $X^{(l+1)}$ solves the following problem

$$\min\left\{\left[\frac{1}{2}\rho\|X\|^2-\langle Y^{(k)},X\rangle\right]-\langle\rho X^{(l)}-\nabla\eta(X^{(l)}),X\rangle : X\in\Omega\right\},$$

$$i.e.\quad X^{(l+1)}=Proj_\Omega\left(X^{(l)}-\frac{1}{\rho}(\nabla\eta(X^{(l)})-Y^{(k)})\right).$$

Here $Proj_\Omega$ stands for the projection onto Ω. Since Ω is an Euclidean ball in $\mathcal{A}^\perp$, the projection onto Ω is explicitly computed, and DCA applied to (39) can be described as follows:

Algorithm 1 *DCA for solving (39)*

Let $\varepsilon>0$, and $X^{(0)}\in\mathcal{A}^\perp$ be given.

For $l=0,1,\ldots$ until $\|X^{(l+1)}-X^{(l)}\|\le\varepsilon$ set $\xi=X^{(l)}-\frac{1}{\rho}\left(\nabla\eta(X^{(l)})-Y^{(k)}\right)$.

If $\|\xi\|\le r$ then set $X^{(l+1)}=\xi$, otherwise set $X^{l+1}=(r/\|\xi\|)\xi$.

REMARK 1 (i) Since (33) is a convex program, Algorithm 1 provides an optimal solution to (33).
(ii) Algorithm 1 is inexpensive, and thus works very well in practice for large scale problems.
(iii) A good starting point of Algorithm 1 may be $X^{(k)}$, the current solution at iteration k of the DCA scheme applied to $(GDGP)_\lambda$.

We now are in a position to describe our special DCA applied to $(GDGP)_\lambda$.

3.4. DESCRIPTION OF THE DCA FOR SOLVING $(GDGP)_\lambda$

Algorithm 2 *(DCA applied to $(GDGP)_\lambda$).*

Let $\varepsilon,\varepsilon_1,\varepsilon_2>0$, and $X^{(0)}\in\mathcal{A}^\perp, T^{(0)}\in\mathcal{C}$ be given, k=0.

Step k1: set

$$Y^{(k)}=\sum_{(i,j)\in\mathcal{S}^w}p_{ij}[\left\|x^{i^{(k)}}-x^{j^{(k)}}\right\|^2+T^{2^{(k)}}_{ind(i,j)})]Q_{ij}(x^{i^{(k)}}-x^{j^{(k)}}). \quad (42)$$

Step k2: set $T^{(k+1)} = (T_{ind(i,j)})_{(i,j)\in\mathcal{S}^w}^{(k+1)}$ as follows:

$$T_{ind(i,j)}^{(k+1)} = \begin{cases} \mu := \sqrt[3]{\frac{1}{2}\left(\left\|x^{i(k)} - x^{j(k)}\right\|^2 + T_{ind(i,j)}^{2(k)}\right) T_{ind(i,j)}^{(k)}} & \text{if} \quad l_{ij} \le \mu \le u_{ij}, \\ l_{ij} & \text{if} \quad \mu < l_{ij}, \\ u_{ij} & \text{if} \quad \mu > u_{ij}. \end{cases} \tag{43}$$

Step k3: set $l := 0$, $X^{(k,l)} := X^{(k)}$

Repeat

$$X^{(k,l+1)} = \begin{cases} \xi := X^{(k,l)} - \frac{1}{\rho}\left(\nabla\eta(X^{(k,l)} - Y^{(k)}\right) & \text{if } \|\xi\| \le r \\ (r/\|\xi\|)\xi & \text{if } \|\xi\| > r \end{cases} \tag{44}$$

until $\|X^{(k,l+1)} - X^{(k,l)}\| \le \varepsilon_1$.

Set $\|\mathbb{X}^{(k+1)} := \mathbb{X}^{(k,l+1)}\|$.

If either $|F_\lambda(X^{(k)}, T^{(k)}) - F_\lambda(X^{(k+1)}, T^{(k+1)})| \le \varepsilon_2 F_\lambda(X^{(k+1)}, T^{(k+1)})$ or

$$(1-\varepsilon)l_{ij} \le \left\|x^{i(k+1)} - x^{j(k+1)}\right\| \le u_{ij}(1+\varepsilon), \quad \text{for all } (i,j) \in \mathcal{S}^w \tag{45}$$

then STOP, $(\mathbb{X}^{(k+1)}, T^{(k+1)})$ is an solution to $(\text{GDGP})_\lambda$

else set $k = k+1$ and go to step k1.

The main results on the convergence of DCA for general D.C. programs (see [2, 4, 19] can be refined as follows:

PROPOSITION 3 *The sequences $\{X^{(k)}, T^{(k)}\}$ and $\{Y^{(k)}, W^{(k)}\}$ generated by Algorithm 2 are bounded and make respectively the primal and dual objective functions in (28) decrease. Their limit points satisfy the local optimality conditions (25).*

Proof. According to [2, 4, 19] it suffices to prove the boundedness of both these sequences. By the very construction, the sequence $\{X^{(k)}, T^{(k)}\}$ is bounded: $\{X^{(k)}, T^{(k)}\} \in \Omega \times \mathcal{C}$. It follows that the sequence $\{Y^{(k)}, W^{(k)}\}$ is bounded too, because the set $\nabla H(\Omega \times \mathcal{C})$ is bounded.

4. A continuation algorithm for the distance geometry problems

We now present our main algorithm for solving the distance geometry problem (GDGP), say, a continuation algorithm based on the Gaussian transform and DCA. Given a sequence of smoothing parameters

$$\lambda_0 > \lambda_1 > \ldots > \lambda_{step} = 0,$$

our continuation algorithm uses the DCA to compute a minimizer (X^{q+1}, T^{q+1}) of $\langle F\rangle_{\lambda_q}$ with the previous one (X^q, T^q) as the starting point. The algorithm generates then a sequence $(\{X^q, T^q\})$, and (X^{step+1}, T^{step+1}) is a candidate for a global minimizer of (GDGP) (see Theorem 5.2 in [13] concerning the convergence of continuation algorithms).

Since the objective function of $(\text{GDGP})_{\lambda_0}$ in our method is not necessarily convex, we investigate a technique for finding a *good* starting point $(X^{(0)}, T^{(0)})$ of the DCA at the first step. We take $T^{(0)}$ by $T^{(0)}_{ind(i,j)} = \frac{u_{ij}+l_{ij}}{2}$ for all $(i, j) \in \mathcal{S}^w$ and then choose $X^{(0)}$ so that $\left\|x^{i(0)} - x^{j(0)}\right\| = T^{(0)}_{ind(i,j)}$ for at least $n-1$ pairs $(i, j) \in \mathcal{S}^w$ by the procedure **Starting** described as follows:

Procedure Starting:

Let $(i_0, j_0) \in \mathcal{S}^w$ such that $T^{(0)}_{ind(i_0,j_0)} = \max\left\{T^{(0)}_{ind(i,j)} : (i, j) \in \mathcal{S}^w\right\}$.
Let $x^{i_0} = (0, 0, 0)^T$ and generate x^{j_0} such that $\left\|x^{i_0} - x^{j_0}\right\| = T^{(0)}_{ind(i,j)}$.
Set $\mathcal{M} := \{i_0, j_0\}$, $k := j_0$.
Do while $|\mathcal{M}| < n$
Choose $(k, j_k) \in \mathcal{S}$ such that $T^{(0)}_{ind(k,j_k)} = \max_j\left\{T^{(0)}_{ind(k,j)} : (k, j) \in \mathcal{S}\right\}$.
Generate x^{j_k} such that $\left\|x^k - x^{j_k}\right\| = T^{(0)}_{ind(k,j_k)}$.
Set $\mathcal{M} := \mathcal{M} \cup \{j_k\}$, $k := j_k$
end do

This procedure is an amelioration of Algorithm struct given in [15]: it provides a point satisfying the *largest* distance constraint in $\mathcal{S}$ and the *largest* constraint (k, j_k) among the pairs $(k, j) \in \mathcal{S}$ for a given k, while the point generated by Algorithm struct satisfies some $n-1$ distance constraints. In our computational experiments this procedure is better than Algorithm Struct for finding the starting point to the DCA.

Finally, the main algorithm can be described as follows:

The main algorithm CGDCA:

Choose T^0 by $T^0_{ind(i,j)} = \frac{u_{ij}+l_{ij}}{2}$ for $(i, j) \in \mathcal{S}^w$ and determine X^0 by Procedure Starting. Set $q := 0$.

Do while $(q \le step)$

Compute (X^{q+1}, T^{q+1}), a solution to Problem $(\text{GDGP})_{\lambda_q}$ by applying Algorithm 2 from the starting point (X^q, T^q). Increase q by 1.

end do

REMARK 2 To solve the exact geometry problem (1) the variable T is not needed. So in Algorithm 2 the step k2 is eliminated and the vector $Y^{(k)}$ at the step k1 is

defined by

$$Y^{(k)} = \sum_{(i,j)\in\mathcal{S}^w} p_{ij}\left[\left\|x^{i^{(k)}} - x^{j^{(k)}}\right\|^2 + \delta_{ij}^2\right] Q_{ij}(x^{i^{(k)}} - x^{j^{(k)}}).$$

5. Computational Experiments and Conclusions

Our algorithms are coded in FORTRAN 77 and run on an SGI Origin 2000 multiprocessor with IRIX system. We considered 16 problems whose data are derived from 16 structures of proteins given in PDB data bank. Table 1 gives the summarized information about these structures (in this table, "Exp."is the abbreviation of "Exploitation" and "MAS" is that of "Minimized Average Structure"). For each structure we generated a set of distances by using distances between the atoms in the same residue as well as those in the neighboring residues (this way has been used in [3, 15]). More precisely, if $\mathcal{R}_k$ is the k^{th} residue, then $\mathcal{S}$ is specified by

$$\mathcal{S} = \{(i, j) : x^i \in \mathcal{R}_k, x^j \in \mathcal{R}_k \cup \mathcal{R}_{k+1}\}$$

and the given distances for the exact geometry problem are $\delta_{ij} = \|x^i - x^j\|$ for all $(i, j) \in \mathcal{S}$. To generate the given bounds in the general distance geometry problem we set

$$l_{ij} = (1-\epsilon)\|x^i - x^j\|, u_{ij} = (1+\epsilon)\|x^i - x^j\|, (i, j) \in \mathcal{S} \tag{46}$$

for a given $\epsilon \geq 0$.

The main aim of the computational experiments is to show that the algorithm **CGDCA** based on the smoothing technique and via DCA is efficient to both exact and general distance geometry problems in large scale real world molecular conformations. We are interested not only in the quality of solutions, but also in the size of molecules to be conformated. We consider molecules containing up to 4189 atoms (the corresponding (GDGP) has 32400 variables).

We set $p_{ij} = 1$ for all $i \neq j$ in (7).

In Algorithm 2 we took $\varepsilon_1 = 10^{-7}, \varepsilon_2 = 10^{-8}$ and $\varepsilon = 0.01$. If (45) is satisfied when stopping algorithm, we say that an ε-global minimizer of the geometry distance problem (3) is obtained.

The parameters λ_0 and $step$ in the continuation algorithm are chosen as in [15], say λ_0 is the median of all $\lambda_{ij} = \left(\frac{l_{ij}}{\sqrt{5}u_{ij}} + \sqrt{2}(1 - \frac{l_{ij}}{u_{ij}}\right) u_{ij}$ with $(i, j) \in \mathcal{S}$ and $step = \lceil 20\lambda_0 \rceil$.

Our experiments are composed of three parts. First, we study in detail the performance of **CGDCA** on 16 test problems with two values of ϵ in (46) while generating given bounds: $\epsilon = 0.001$ and $\epsilon = 0.08$. The results are presented in Tables 2 and 3 where we indicate the following values:

Table 1. Summarized information about test problems from PDB data bank

ID code	Exp. method	Classification	Atoms (n)	Residues
304D	X-ray diffraction	Deoxyribonucleic Acid	237	52
8DRH	NMR (MAS)	Deoxyribonucleic Acid/Ribonucleic Acid	329	16
1AMD	NMR (MAS)	Deoxyribonucleic Acid	380	12
2MSJ	X-ray diffraction	Antifreeze Protein	480	66
124D	NMR	Deoxyribonucleic/Ribonucleic Acid	508	16
141D	NMR	Deoxyribonucleic Acid	527	26
132D	NMR	Deoxyribonucleic Acid	750	24
1A84	NMR	Deoxyribonucleic Acid	758	24
104D	NMR	DNA/RNA Chimeric Hybrid Duplex	766	24
103D	NMR (MAS)	Deoxyribonucleic Acid	772	24
2EQL	X-ray diffraction	Hydrolase (O-Glycosyl)	1023	129
1QS5	X-ray diffraction	Hydrolase	1429	162
1QSB	X-ray diffraction	Hydrolase	1431	162
6GAT	NMR	Complex (Transcription Regulation/DNA)	1853	92
7HSC	NMR	Molecular Chaperone	2482	159
2CLJ	Theoretical model	Hydrolase	4189	543

- numva: the number of variables in (GDGP), say, numva $= 3n + |\mathcal{S}^w|$.
- t0: CPU time of Procedure Starting
- ttotal: the total CPU time of the main **CGDCA** (all CPU times were computed in seconds).
- step: the number of the optimization steps in the continuation algorithm **CGDCA**.
- F^*: the approximate optimal value of the distance geometry problem (GDGP):

$$F^* := \sum_{(i,j)\in\mathcal{S}^w} p_{ij}(\|x^{*^i} - x^{*^j}\|^2 - {t^*}_{ij}^{\,2})^2$$

- aveer: the average relative error defined by

$$\frac{1}{|\mathcal{S}^w|}\left(\sum_{(i,j)\in\mathcal{S}^w} \max\left\{\frac{\left\|x^{*^i} - x^{*^j}\right\| - u_{ij}}{u_{ij}}, \frac{l_{ij} - \left\|x^{*^i} - x^{*^j}\right\|}{l_{ij}}, 0\right\}\right) \quad (47)$$

• maxer: the maximal relative error defined by

$$\max\left\{\max\left\{\frac{\left\|x^{*^i}-x^{*^j}\right\|-u_{ij}}{u_{ij}},\frac{l_{ij}-\left\|x^{*^i}-x^{*^j}\right\|}{l_{ij}},0\right\}:(i,j)\in\mathcal{S}^w\right\}. \quad (48)$$

In the second experiment we are interested in the influence of the length of bounds on the behavior of **CGDCA** on four test problems. We generate the data in the same way as in the first experiment with different values of ϵ: 0.0, 0.001, 0.01, 0.02, 0.04, 0.08. The results are reported in Table 4. Remember that when $\epsilon = 0.0$ we are faced with the *exact distance geometry problem* (1).

In the third experiment we compare the performance of our approach **CGDCA** and the two phase algorithm **GDCA** in [3] with $\epsilon = 0.08$ while constructing given bounds. The results for this experiment are reported in Table 5.

Table 2. The performance of **CGDCA** for PDB data in case $\epsilon = 0.001$

ID code	t0	step	ttotal	F^*	aver	maxer
304D	0.15	4	160.17	4.88E-03	1.24E-05	3.99E-03
8DRH	0.32	4	67.76	1.10E-04	1.01E-06	1.04E-03
1AMD	0.98	4	47.68	9.97E-02	7.23E-05	9.99E-03
2MSJ	0.02	2	28.63	1.47E-04	2.23E-06	9.32E-03
124D	2.02	4	153.20	4.08E-02	2.03E-05	9.99E-03
141D	0.93	4	313.43	7.72E-04	1.07E-05	2.17E-03
132D	4.02	4	433.80	2.17E-03	5.42E-06	2.96E-03
1A84	4.30	4	382.64	3.94E-03	1.62E-06	3.45E-03
104D	4.53	4	151.35	1.87E-02	3.20E-05	9.99E-03
103D	4.86	4	263.19	6.38E-02	2.87E-05	9.99E-03
2EQL	0.80	4	232.47	1.71E-03	2.34E-06	9.99E-03
1QS5	1.37	4	673.05	6.25E-03	4.56E-05	1.00E-02
1QSB	1.35	4	774.48	2.74E-03	3.54E-05	9.99E-03
6GAT	15.58	4	541.07	8.44E-02	4.52E-05	9.98E-03
7HSC	15.40	2	387.53	2.83E-03	1.31E-05	9.99E-03
2CLJ	14.08	2	1079.59	1.12E-02	2.12E-05	1.03E-02

The results in Tables 2 and 3 show that Algorithm **CGDCA** is efficient to both cases when l_{ij} and u_{ij} are close or not so: it finds a global minimizer in all cases when $\epsilon = 0.001$ and in 70% cases when $\epsilon = 0.08$. The algorithm is very fast: it solves problems with 32400 variables in 18 minutes.

Surprisingly enough, the results in Table 4 show that in general the reliability and the convergence rate of GDCA increase when ϵ decreases (the length of bounds decreases). Moreover, the algorithm is very efficient to the exact distance geometry problem (1).

Table 3. The performance of **CGDCA** for PDB data in case $\epsilon = 0.08$

ID code	numva	t0	step	ttotal	F^*	aver	maxer
304D	3030	0.15	6	146.06	0.23E+00	7.80E-05	2.02E-03
8DRH	4536	0.32	6	35.03	1.32E-03	1.93E-05	9.98E-05
1AMD	7326	0.98	6	102.85	9.50E-04	1.25E-05	9.99E-03
2MSJ	1440	0.02	2	32.69	1.83E-01	1.00E-04	3.53E-02
124D	9831	2.02	6	387.06	2.20E-03	1.78E-05	9.99E-03
141D	7196	0.93	6	241.23	1.47E+00	1.22E-04	8.77E-02
132D	14344	4.17	6	320.04	4.50E-03	01.35E-05	9.99E-03
1A84	14619	4.30	6	484.84	5.12E-04	9.50E-06	9.99E-03
104D	14907	4.53	6	371.74	2.54E-04	5.63E-06	1.30E-02
103D	15093	4.86	6	411.48	2.68E-03	1.00E-05	9.99E-03
2EQL	7957	0.80	4	183.27	3.69E-03	2.97E-05	9.99E-03
1QS5	10642	1.37	4	520.72	0.21E+00	8.10E-05	7.58E-02
1QSB	10637	1.41	4	578.31	3.00E-02	4.10E-05	6.33E-02
6GAT	28288	15.67	6	1229.52	2.00E-02	3.22E-05	9.99E-03
7HSC	29962	15.80	4	1129.08	3.66E-03	8.19E-06	9.99E-03
2CLJ	32400	14.08	4	919.04	1.37E+00	5.00E-05	1.12E-01

Table 4. The performance of **CGDCA** as the length of bounds varies

ID code	1AMD		124D		104D		2EQL	
ϵ	ttotal	maxer	ttotal	maxer	ttotal	maxer	ttotal	maxer
0.08	102.85	9.99E-03	387.06	9.99E-03	371.74	1.30E-02	183.27	9.99E-03
0.04	71.03	5.84E-02	388.08	1.02E-02	265.28	9.99E-03	339.41	9.75E-02
0.02	49.18	7.15E-02	372.16	1.03E-02	241.63	9.99E-03	204.86	3.39E-01
0.01	79.20	1.04E-02	272.04	9.99E-03	219.26	9.99E-03	228.89	6.47E-02
0.001	47.68	9.99E-03	153.20	9.99E-03	151.35	9.99E-03	237.47	9.99E-03
0.0	48.06	6.42E-03	129.85	8.30E-03	134.30	9.99E-03	263.07	9.99E-03

According to the results in Table 5, **CGDCA** is faster and more reliable than the two phase algorithm **GDCA**. In fact, the main difference between **CGDCA** and **GDCA** is that the first phase of **GDCA** is replaced by the smoothing technique in **CGDCA**, we then need not complete an approximate distance matrix (this is expensive when the dimension is large and the number of constraints is small) and work only with given distances (sparse distance matrices).

Table 5. Comparison with the two phase algorithm **GDCA**

ID code	**CGDCA**			**GDCA**		
	ttotal	F^*	maxer	ttotal	F^*	maxer
8DRH	35.03	1.32E-03	9.98E-05	198.80	8.66E-02	6.68E-02
2MSJ	32.69	1.86E-01	3.53E-02	360.39	4.65E-04	9.95E-03
132D	320.04	4.50E-03	9.99E-03	1794.52	3.79E-03	9.98E-02
104D	371.84	2.54E-04	1.30E-02	6198.2	1.32E-02	1.00E-02
103D	411.48	2.68E-03	9.99E-03	1107.0	1.69E+00	2.70E-01
2EQL	183.27	3.69E-03	9.99E-03	4318.9	4.00E-03	9.99E-03

Conclusions

We have presented a continuation approach based on D.C. optimization and DCA for solving large scale molecular optimization problems from distance matrices. The main points in this approach are:

(i) Reformulation of the distance geometry problems (1) and (3) as a bound constrained smooth optimization problem;
(ii) Smoothing technique via the Gaussian transform of the objective function, and a D.C. formulation to the transformed problems;
(iii) D.C. optimization algorithm with suitable D.C. decompositions to the resulting problems.

Computational experiments show that our method is successful in locating the large configurations satisfying given distance constraints: the DCA globally solved distance geometry problems with up to 4189 atoms (32 400 variables).

Several interesting issues arise from the present work. The first deals with the *reformulation* of the distance geometry problems. As indicated above, our new formulation has several advantages not only to DCA, but also for existing methods for bound constrained smooth problems. Nevertheless, from the numerical experiments we observe that, the maximal error of distance constraints occurs on the same pair of atoms when the current point is *near* a solution. So it is interesting to consider an objective function dealing simultaneously with the sum of all errors and the maximal error of distance constraints. In other word, we can investigate a predictor-corrector algorithm to exploit the efficiency of DCA.

The second is the amelioration of our code by using a parallel solver.

The third concerns the data. In this paper we generated the distance data from the complete protein given in PDB data bank which are real models (see Table 1). We follow the way of Moré and Wu [15] for constructing the data because our idea

to use the continuation approach is suggested by their work. We wish to expand our testing to distance data generated by more realistic ways and to distance data derived from NMR experiments. These issues are currently in progress.

Acknowledgments

The author would like to express the gratitude to the two referees for their valuable remarks and proposals which led to an improvement of the paper. This work was partially supported by the computer resources financed by “Contrat de Plan Interrégional du Bassin Parisien - Pôle interrégional de modélisation en Sciences pour Ingénieurs”.

References

1. Alfakih A., Khandani A. and Wolkowicz H. (1999) Solving Euclidean Distance Matrix Completion Problems via Semidefinite Programming, *Computational Optimization and Applications*, 12, 13–30.
2. Le Thi Hoai An (1997), Contribution à l’optimisation non convexe et l’optimisation globale: Théorie, Algorithmes et Applications , Habilitation à Diriger des Recherches, Université de Rouen.
3. Le Thi Hoai An, Pham Dinh Tao (2000), D.C. programming approach for large scale molecular optimization via the general distance geometry problem, in the Special Issue *‘Optimization in Computational Chemistry and Molecular Biology: Local and Global Approaches’ Nonconvex Optimization and Its Applications*, Kluwer Academic Publishers, 301–339.
4. Le Thi Hoai An and Pham Dinh Tao, (1999) DCA revisited and D.C. models of real world nonconvex optimization problems, To appear in *Annals of Operations Research* 2003.
5. Le Thi Hoai An, Pham Dinh Tao, Large Scale Molecular Optimization from distance matrices by a D.C. optimization approach, To appear in *SIAM J. Optimization* 2003.
6. Byrd R.H., Lu P., Nocedal J. And Zhu C. (1994) A limited memory algorithm for bound constrained optimization, Technical Report NAM-08, Departement of Electrical Engineering and Computer Science, Northwestern University.
7. Crippen G.M. & Havel T.F., *Distance Geometry and Molecular Conformation*, John Wiley & Sons, (1988), New York.
8. Floudas C., Adjiman C.S., Dallwig S. and Neumaier A., A Global Optimization Method, αBB, for General Twice Differentiable Constrained NLPs - I: Theoretical Advances, *Computational Chemical Engineering* 22 (1998), 11–37.
9. Glunt W., Hayden T.L. & Raydan M., Molecular Conformation from distance matrices, *J. Comp. Chem.* 14 (1993), 114–120.
10. Havel T.F., An evaluation of computational strategies for use in the determination of protein structure from distance geometry constraints obtained by nuclear magnetic resonance, *Prog. Biophys. Mol. Biol.*, 56 (1991), 43–78.
11. Hendrickson B.A., The molecule problem: Exploiting structure in global optimization, *SIAM J. Optimization*, 5(1995), 835–857.
12. Huang H.X., Liang Z.A. and Pardalos P., Some properties for the Euclidean Distance Matrix and Positive Semi-Definite Matrix Completion Problems, Department of Industrial and Systems Enegineering, University Florida, 2001.

13. More J.J. & Wu Z., Global continuation for distance geometry problems, *SIAM J. Optimization*, 8(1997), 814–836.
14. More J.J. & Wu Z., Issues in large-scale Molecular Optimization, preprint MCS-P539-1095, Argonne National Laboratory, Argonne, Illinois 60439, Mars 1996.
15. More J.J. & Wu Z., Distance geometry optimization for protein structures, Journal of Global Optimization, Vol. 15 (1999), 219–234.
16. Pham Dinh Tao, Algorithms for solving a class of non convex optimization problems. Methods of subgradients, Fermat days 85. Mathematics for Optimization, Elsevier Science Publishers B.V. North-Holland, (1986), 249–271.
17. Pham Dinh Tao and Bernoussi S.E., Duality in D.C. (difference of convex functions) optimization. Subgradient methods, Trends in Mathematical Optimization, International Series of Numer Math. Vol 84 (1988), Birkhauser, 277–293.
18. Pham Dinh Tao and Le Thi Hoai An, D.C. optimization algorithms for the trust region problem. *SIAM J. Optimization*, 8(2), (1998), 476–505.
19. Pham Dinh Tao and Le Thi Hoai An, Convex analysis approach to D.C. programming: Theory, Algorithms and Applications (dedicated to Professor Hoang Tuy on the occasion of his 70th birthday), *Acta Mathematica Vietnamica*, 22 (1), 1997, 289–355.
20. Rockafellar R.T., Convex Analysis, Princeton University, Princeton, 1970.
21. Zou Z., Richard. H. Bird, & Robert B. Schnabel, A Stochastic/Pertubation Global Optimization Algorithm for Distance Geometry Problems, *J. of Global Optimization*, 11(1997), 91–105.