Double Variable Neighbourhood Search with Smoothing for the Molecular Distance Geometry Problem

Leo Liberti[1], Carlile Lavor[2], Nelson Maculan[3], Fabrizio Marinelli[1]

[1] *LIX, École Polytechnique, F-91128 Palaiseau, France*
E-mail:{liberti,marinelli}@lix.polytechnique.fr

[2] *Department of Applied Mathematics (IMECC-UNICAMP), State University of Campinas,*
CP 6065, 13081-970, Campinas-SP, Brazil
E-mail:clavor@ime.unicamp.br

[3] *COPPE – Systems Engineering, Federal University of Rio de Janeiro,*
P.O. Box 68511, 21941-972 Rio de Janeiro, Brazil.
E-mail:maculan@cos.ufrj.br

April 30, 2007

Abstract

We discuss the geometrical interpretation of a well-known smoothing operator applied to the Molecular Distance Geometry Problem, and we then describe a heuristic approach based on Variable Neighbourhood Search on the smoothed and original problem. This algorithm often manages to find solutions having higher accuracy than other methods. This is important as small differences in the objective function value may point to completely different 3D molecular structures.

Keywords: molecular conformation, distance geometry, global optimization, global continuation, variable neighbourhood search, smoothing.

1 Introduction

The Molecular Distance Geometry Problem (MDGP) is the problem of finding an embedding in $\mathbb{R}^3$ of a weighted graph G such that all Euclidean distances between points in the embedding are the same as the corresponding edge weights in the graph. The main application is to find the three-dimensional structure of a molecule given a subset of the atomic distances (these are usually found using NMR techniques) [1, 15]. There are other applications in network localization [5] and graph drawing [2]. It has been shown that the MDGP is NP-hard [17]; if G is a complete graph, however, the MDGP can be solved in linear time [3]. The MDGP applied to protein backbones is susceptible of a discrete formulation which leads to improved accuracy and efficiency [10].

Consider an undirected graph $G = (V, E)$ with weights $d : E \to \mathbb{R}_+$ where V is the set of vertices (also called *atoms*) and E is the set of weighted edges (also called *inter-atomic distances*). Let $N = |V|$ and $d_{ij} = d(\{i, j\})$ for $\{i, j\} \in E$. A solution of the MDGP is a set of points $x_1, \ldots, x_N \in \mathbb{R}^3$ satisfying

$$\forall \{i, j\} \in E \quad ||x_i - x_j|| = d_{ij}. \tag{1}$$

Notationally, each 3-vector x_i has components (x_{i1}, x_{i2}, x_{i3}), and we indicate the vector sequence $(x_1, \ldots, x_N)$ by x. The MDGP can be naturally cast as a continuous nonconvex optimization problem $\min_x f(x)$ with the following objective function:

$$f(x) = \sum_{\{i,j\} \in E} (||x_i - x_j||^2 - d_{ij}^2)^2. \tag{2}$$

Since each equation (1) must be satisfied, a candidate point x is a solution of the MDGP if and only if $f(x) = 0$. Note that (2) has a large number of local minima, so this is a practically hard global optimization problem. See [18] for an overview on the main solution methods used to tackle the MDGP.

One of the most promising methods for solving the MDGP is the Global Continuation Algorithm (GCA) [13, 14]. The GCA relies on a class of smoothed objective functions derived from (2) by means of a Gaussian transform. This class is indexed by a parameter λ; for $\lambda = 0$ we recover the original function (2), and for large enough λ the smoothed function becomes convex. The GCA locally solves a sequence of smoothed problems for decreasing values of λ. To assess solution accuracy, the authors of [13, 14] rely on a different quality measure called the Largest Distance Error (LDE) defined as:

$$\mathrm{LDE}(x) = \frac{1}{|E|} \sum_{\{i,j\} \in E} \frac{|\,||x_i - x_j|| - d_{ij}|}{d_{ij}}. \tag{3}$$

It is clear that x is a solution of the MDGP if and only if $\mathrm{LDE}(x) = 0$. The GCA has been implemented in the `dgsol` code, available from `http://www.mcs.anl.gov/~more/dgsol/`.

One of the striking features of `dgsol` is its speed and the fact that the time taken to solve the problem seems to grow rather slowly as a function of the number of atoms in the molecule. On the other hand, `dgsol` usually finds solutions whose LDE is relatively large (in the order of 0.01 or even 0.1). Since there are many totally different 3D structures having small LDE (see for example Fig. 1), it is of paramount importance to obtain solutions whose LDEs are very close to zero.

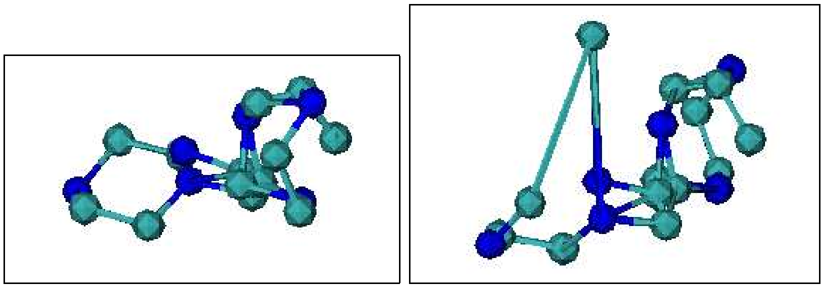

Figure 1: Different 3D graph embeddings with very similar objective function values (both values are in the order of 10^{-11}).

In a previous paper [8], we tested three different general-purpose global optimization methods on several MDGP instances, concluding that Variable Neighbourhood Search (VNS) was the best. Notice that VNS was also used to find the 3D structure of molecules by minimizing a potential energy objective function [4]. In this paper, we combine ideas from both GCA and VNS algorithms to obtain a method which we call Double VNS with Smoothing (DVS). Although considerably slower than `dgsol`, the solutions obtained by DVS generally have a much smaller LDE. DVS is a classical two-phase global optimization algorithm with diversification in the first phase and intensification in the second phase. More precisely, DVS solves a smoothed version of the problem using VNS, then tightens the bounds around the smoothed solution, and finally applies VNS to the original problem with restricted bounds. The bounds are restricted following a geometrical interpretation of the smoothed problem, which makes it possible to relate the smoothing parameter to the bounds where the solution to the original MDGP should lie.

The rest of this paper is organized as follows. In Section 2 we discuss the geometrical interpretation of the MDGP smoothing proposed by [13], which leads to some indications as to the bound restriction at each DVS iteration. In Section 3 we present the DVS algorithm. In Section 4 we report on the computational results. Section 5 concludes the paper.

2 Geometrical interpretation of smoothing

As (2) is a nonconvex function with a considerable number of local minima, it makes sense to look for a *smoothing* (i.e. a function in some sense "close" to (2) with fewer local minima). In [13], a family of smoothings for (2) is proposed which depends on a unique smoothing parameter $\bar\lambda \in [\bar\lambda^L, \bar\lambda^U]$: for $\bar\lambda = \bar\lambda^L$ we recover the original objective (2), and for $\bar\lambda = \bar\lambda^U$ we get a convex smoothing. The smoothing family $\bar f_{\bar\lambda}(x)$ given by [13] (and employed in the GCA) is derived as the closed analytical form of a Gaussian integral:

$$\bar f_{\bar\lambda}(x) = f(x) + \sum_{\{i,j\}\in E} (10\bar\lambda^2 ||x_i - x_j||^2) + \bar\gamma, \tag{4}$$

where $\bar\gamma$ is a constant and f is given by (2). We shall call this smoothing the *Moré-Wu smoothing.* It should be clear that, since the nonconvexities in (2) arise from the negative cross product terms $-2\sum_{\{i,j\}\in E} d_{ij}^2||x_i - x_j||^2$, the smoothing $\bar f_{\bar\lambda}(x)$ tends to eliminate these nonconvexities by adding the positive term $\sum_{\{i,j\}\in E}(10\bar\lambda^2||x_i - x_j||^2)$ to the original objective function $f(x)$. This implies that the quartic terms $\sum_{\{i,j\}\in E} ||x_i - x_j||^4$ have a bigger effect in the minimization process. Intuitively speaking, $\bar f_{\bar\lambda}$ describes a configuration where atoms whose distance d_{ij} is small tend to be positioned even closer together, whilst the relative position of far away atoms remains roughly unchanged; in other words, close atoms tend to be clustered together in the smoothed function. We shall now formalize this geometric intuition.

We introduce a function F which associates to each vector of squared NMR distances (denoted by $d^2 = (d_{ij}^2 \mid \{i,j\} \in E)$) the MDGP objective function (2):

$$F(d^2) = \sum_{\{i,j\}\in E} (||x_i - x_j||^4 - 2d_{ij}^2||x_i - x_j||^2 + d_{ij}^4).$$

Notice we obviously have $(F(d^2))(x) \equiv f(x)$. We re-state the Moré-Wu smoothing as follows, introducing the smoothing parameter $\lambda = \bar\lambda\sqrt{5}$ and ignoring the constant $\bar\gamma$:

$$f_\lambda(x) = f(x) + 2\sum_{\{i,j\}\in E} (\lambda^2||x_i - x_j||^2). \tag{5}$$

Now, we define the *Moré-Wu smoothing operator* (denoted by $\odot$) as follows:

$$\lambda^2 \odot F(d^2) = f_\lambda(x) - \gamma, \tag{6}$$

where $\gamma = \lambda^2(\lambda^2|E| - 2\sum_{\{i,j\}\in E} d_{ij}^2)$. It turns out that the Moré-Wu smoothing operator acts by shortening the square of the distances:

2.1 Theorem
$\lambda^2 \odot F(d^2) = F(d^2 - \lambda^2)$, where $d^2 - \lambda^2$ is the vector $(d_{ij}^2 - \lambda^2 \mid \{i,j\} \in E)$.

Proof.

$$\begin{aligned}
\lambda^2 \odot F(d^2) &= \sum_{\{i,j\}\in E} (||x_i - x_j||^4 - 2(d_{ij}^2 - \lambda^2)||x_i - x_j||^2 + d_{ij}^4) - \gamma \\
&= \sum_{\{i,j\}\in E} (||x_i - x_j||^4 - 2(d_{ij}^2 - \lambda^2)||x_i - x_j||^2 + (d_{ij}^2 - \lambda^2)^2) \\
&= \sum_{\{i,j\}\in E} (||x_i - x_j||^2 - (d_{ij}^2 - \lambda^2))^2 \\
&= F(d^2 - \lambda^2),
\end{aligned}$$

where γ in the first equation was replaced by its definition in terms of λ and d_{ij} and carried into the sum. $\square$

By Theorem 2.1, for all pairs of atoms $\{i,j\} \in E$ such that $\lambda \geq d_{ij}$, the corresponding term contributing to the objective function is $(||x_i - x_j||^2 + \lambda^2 - d_{ij}^2)^2$. This is a convex term with minima attained at $x_i = x_j$; in other words, atoms whose distances are shorter than λ will tend, if possible, to collapse into the same spatial positions. Pairs of atoms which are further away than λ will, if possible, be positioned at a shorter distance $\sqrt{d_{ij}^2 - \lambda^2}$. This observation is based on a term-by-term analysis of a non-separable function, so it is of course only true if λ is large enough to make $\bar{f}_\lambda$ convex (in which case we obtain the trivial solution $x_i = 0$ for all $i \in V$). Its usefulness, however, is that it formalizes an insight as to the general behaviour of a molecule in space in function of a smooth transformation of its associated objective function.

It is interesting to observe what happens if the smoothing terms $2\lambda||x_i - x_j||^2$ are proportional to the distances d_{ij}. Let

$$\tilde{f}_\lambda(x) = f(x) + 2 \sum_{\{i,j\}\in E} (\lambda^2 d_{ij}^2 ||x_i - x_j||^2),$$

and define the *scaling operator* (denoted by $\otimes$) as follows:

$$\lambda^2 \otimes F(d^2) = \tilde{f}_\lambda(x) - \tilde{\gamma}, \tag{7}$$

where $\tilde{\gamma} = (\lambda^4 - 1)\sum_{\{i,j\}\in E} d_{ij}^4$. It turns out that this type of smoothing is in fact a scaling (thus the operator name).

2.2 Proposition
$\lambda^2 \otimes F(d^2) = F(\lambda^2 d^2)$, where $\lambda^2 d^2$ is the vector $(\lambda^2 d_{ij}^2 \mid \{i,j\} \in E)$.

Proof. The proof is by direct computation, similar to that of Theorem 2.1. $\square$

The geometrical interpretation of Prop. 2.2 is simply that all inter-atomic distances are scaled by a factor λ. It shows that the weights of the smoothing terms should not be proportional to the distances, for otherwise the problem is simply scaled rather than actually smoothed.

2.1 Bounds tightening

The aim of this section is to show how to use the geometrical interpretation of smoothing to derive some indications as to how to tighten the bounds at each iteration of the DVS algorithm (i.e. after the solution of each smoothed problem and before the solution of the original problem).

Suppose $\bar{x}$ is a global optimum of the smoothing $\lambda^2 \odot F(d^2)$ for some $\lambda > 0$. According to Theorem 2.1 and Prop. 2.2, this is likely to represent a molecule where the closer atoms are clustered together (possibly even at the same position), whereas the more distant atoms are at a smaller distance. With this information, we can try to enforce meaningful lower and upper bounds x^L, x^U on the atomic position variables x to restrict the search in a subsequent iteration. We recall that as $\lambda \to 0$, $\bar{f}_\lambda \to f$; in other words, the smoothings become more and more nonconvex and more difficult to minimize. In order for intensification to occur in the second phase, the desirable behaviour is that the bounds should get tigther as λ decreases.

For a given $\lambda > 0$, the smoothed function $\lambda^2 \odot F(d^2) = \bar{f}_\lambda$ roughly describes a configuration where the inter-atomic distance between atoms i and j is $\bar{d}_{ij} = \sqrt{d_{ij}^2 - \lambda^2}$. Let $\xi_{ij} = d_{ij} - \bar{d}_{ij}$ be the difference between the smoothed and the original distance, let $\bar{x}$ be the global minimizer of $\bar{f}_\lambda(x)$ and x^* the globally optimal solution of the original objective function $f(x)$. For any atom pairs $\{i,j\} \in E$, if we assume that $x^*_j = \bar{x}_j$ (i.e. atom j is fixed after the solution of the smoothed problem), then x^*_i is on the surface of a sphere S_{ij} centered at x^*_j with radius d_{ij}. Similarly, for any atom $i \in V$, if we assume that $x^*_j = \bar{x}_j$ for all $j \in V : \{i,j\} \in E$, x^*_i is on the intersection of the surfaces of the spheres S_{ij}. Naturally, since the smoothing may not actually describe any feasible molecule, this intersection may be empty (see Fig. 2; the figure shows the pairwise intersections of the spheres, which give a meaningful boundary for the position of the atom once the neighbourhing atoms are perturbed).

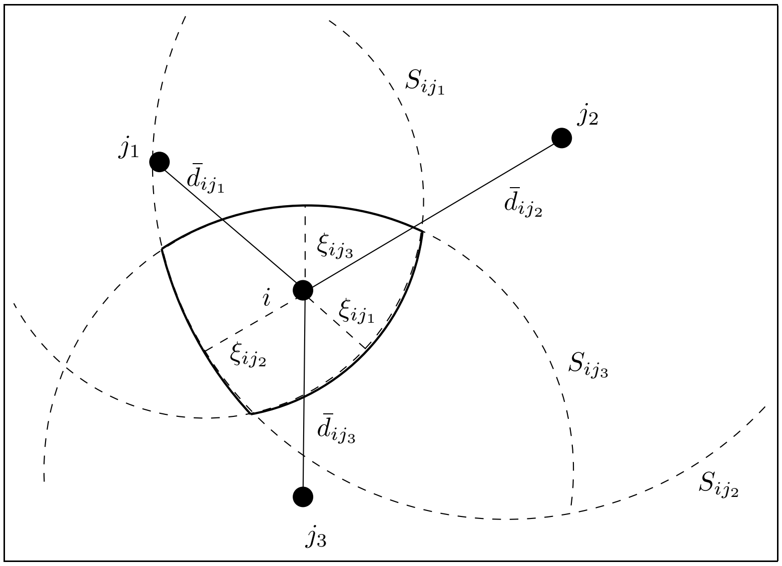

Figure 2: Likely positions of atom i when the adjacent atoms are fixed (viewed in 2D).

If we now allow the previously fixed atoms $j \in V : \{i,j\} \in E$ to be perturbed, the position of the i-th atom is likely to be in $S_i = \bigcap_j S_{ij}$. For algorithmic reasons (see Sect. 3.1) we are interested in a box-shaped neighbourhood of S_i. More precisely, for each $i \in V$ we wish to find the tightest interval $[x^L_{ik}, x^U_{ik}]$ containing x_{ik} (for $1 \le k \le 3$) such that the box $[x^L_{i1}, x^U_{i1}] \times [x^L_{i2}, x^U_{i2}] \times [x^L_{i3}, x^U_{i3}]$ contains S_i. For each $i \in V$ we can therefore solve $6N$ convex problems defined for different $1 \le k \le 3$ and $b \in \{0,1\}$:

$$\begin{array}{rcll} (-1)^b \min(-1)^b x_{ik} & & & \\ \sum\limits_{h \le 3} (x_{ih} - \bar{x}_{jh})^2 & \le & d^2_{ij} & \forall\, j \in V : \{i,j\} \in E \end{array} \tag{8}$$

where the optimum yields x^L_{ik} for $b = 0$ and x^U_{ik} for $b = 1$. Because of the (relative) computational cost of solving $6N$ convex problems (8) at each step of the DVS algorithm (see Sect. 3), we content ourselves with an approximation.

Since each point $\tilde{x}_i \in S_i$ must fall in each sphere S_{ij}, one has $\bar{x}_{jk} - d_{ij} \le \tilde{x}_{ik} \le \bar{x}_{jk} + d_{ij}$ for any k, j. Motivated by this fact, for all $i \in V, k \le 3$ we define:

$$x^L_{ik} = \max\{-d_{ij} + \bar{x}_{jk} \mid j \in V : \{i,j\} \in E\} \tag{9}$$

$$x^U_{ik} = \min\{d_{ij} + \bar{x}_{jk} \mid j \in V : \{i,j\} \in E\}. \tag{10}$$

It is easy to show that the cartesian product across all $k \le 3$ of $[x^L_{ik}, x^U_{ik}]$ contains the set S_i (for all $i \in V$), although it may not in general be the tightest such box.

A more accurate approximation can be derived by iteratively determining a sequence of box-shaped domains the first of which corresponds to the tightest box containing a given sphere S_{i1} (in an arbitrary sequence), the last of which is a box-shaped neighbourhood of S_i, and the h-th of which is the tightest box containing the exact intersection of the h-th sphere with the $(h-1)$-th box-shaped domain. At each iteration, the box-shaped domain can be determined by finding the points on the frontier of (11) that minimize/maximize the coordinate axes. More precisely, let $N_i = |\delta(i)|$ for all $i \in V$, $B_0 = \mathbb{R}^3$ and B_h be the tightest box $[L_{h1}, U_{h1}] \times [L_{h2}, U_{h2}] \times [L_{h3}, U_{h3}]$ containing $S_{ih} \cap B_{h-1}$, $1 \le h \le |\delta(i)|$. It is easy to see that, for $1 \le k \le 3$, $[L_{1k}, U_{1k}] = [\bar{x}_{1k} - d_{i1}, \bar{x}_{1k} + d_{i1}]$ and $[x^L_{ik}, x^U_{ik}] = [L_{N_i k}, U_{N_i k}]$.

The approximation of the box-shaped neighbourhood of S_i can be therefore obtained starting from B_0 and iteratively determining B_h, until $h = N_i$. By definition B_h is the smallest box containing all the points y satisfying

$$\left\{ \begin{array}{lll} \sum_{k=1}^{3} (y_k - \bar{x}_{hk})^2 & \le & d^2_{hk} \\ L_{h-1,k} \le y_k & \le & U_{h-1,k} \qquad 1 \le k \le 3 \end{array} \right. \tag{11}$$

Without loss of generality suppose $B_h \ne \emptyset$. Let C^l, $1 \le l \le q \le 6$, be the intersection arc between S_{ih} and the l-th facet of B_{h-1}, and $C = \bigcup_{l \le q, k \le 3} \{\bar{y} \in C^l | \bar{y}_k \le y_k, \forall y \in C^l\}$. Moreover, let Y be the set of intersection points between S_{ih} and the edges of B_{h-1}. Therefore, for $1 \le k \le 3$,

$$L_{hk} = \min\{\min_{y \in C \cup Y} \{y_k\}, \Gamma^L_k\} \qquad \text{and} \qquad U_{hk} = \max\{\max_{y \in C \cup Y} \{y_k\}, \Gamma^U_k\},$$

where

$$\Gamma^L_k = \left\{ \begin{array}{lll} \bar{x}_{hk} - d_{ih} & \text{if} & \bar{x}_{hk} - d_{ih} \text{ satisfies (11)} \\ +\infty & \text{otherwise} & \end{array} \right.$$

and

$$\Gamma^U_k = \left\{ \begin{array}{lll} \bar{x}_{hk} + d_{ih} & \text{if} & \bar{x}_{hk} + d_{ih} \text{ satisfies (11)} \\ -\infty & \text{otherwise.} & \end{array} \right.$$

3 The algorithm

The DVS algorithm is basically a two-level VNS search within a GCA-type framework. The GCA works by locally solving a sequence of Moré-Wu smoothings $\min \bar{f}_{\bar{\lambda}_k}(x)$. At step k, the GCA obtains the next iterate x_{k+1} by locally solving the current smoothing $\min \bar{f}_{\bar{\lambda}_k}(x)$ from the initial starting point x_k. The problem of this approach is that its search scope is not global enough. There is no guarantee that the global optimum of the initial smoothing will be traced homotopically to the global optimum of the original problem. Moreover, the local solution algorithm may fail to solve even the initial "mildly nonconvex" smoothing to global optimality.

In the DVS, the local search phase is replaced by a two-level VNS. The global solution $\bar{x}$ to the smoothed problem $\min f_{\lambda_k}(x)$ (see Eq. (5)) is found by means of a formulation-based VNS for NLPs [8, 9]. Next, an attempt to find the global solution x^* of $\min f(x)$ is performed by restricting the original problem to a neighbourhood of $\bar{x}$ (as detailed in Section 2) and solving this restriction with the same formulation-based VNS.

3.1 VNS Solver

We employed the VNS solver described in [9]. The search space is defined as the hyper-parallelepiped given by the set of variable ranges $x^L \le x \le x^U$. At first we pick a random point $\tilde{x}$ in the search space, we start one (or optionally, more) local searches and we store the local optimum x^*. Then, until k does not exceed a pre-set $k_{\max}$, we iteratively select new starting points $\tilde{x}$ in an increasingly larger neighbourhood $N_k(x^*)$ and start new local searches from $\tilde{x}$ leading to local optima x'. As soon as we find a local optimum x' better than x^*, we update $x^* = x'$, re-set $k = 1$ and repeat. Otherwise the algorithm terminates.

For each $k \le k_{\max}$ we consider hyper-parallelepipeds $H_k(x^*)$ proportional to $x^L \le x \le x^U$, centered at x^*, whose sides have been scaled by $\frac{k}{k_{\max}}$. More formally, let $H_k(x^*)$ be the hyper-parallelepiped

$y^L \leq x \leq y^U$ where, for all $i \leq n$,

$$\begin{aligned} y_i^L &= x_i^* - \frac{k}{k_{\max}}(x_i^* - x_i^L) \\ y_i^U &= x_i^* + \frac{k}{k_{\max}}(x_i^U - x_i^*). \end{aligned}$$

This construction forms a set of hyper-parallelepiped-shaped shells centered at x^* and proportional to $x^L \leq x \leq x^U$. As has been mentioned above, we define each neighbourhood $N_k(x^*)$ as $H_k(x^*) \backslash H_{k-1}(x^*)$.

3.2 The implementation

The conceptual simplicity of the proposed algorithm does not reflect a corresponding simplicity in the implementation. Our algorithm uses an VNS global optimization solver [9] for continuous nonconvex programming problems as a black-box. In turn, the VNS solver calls a local NLP solver (SNOPT [6]) as a black-box. Careful software architecture and code re-entrancy is required. Our implementation uses the *oo$\mathcal{OPS}$* optimization framework library [12], which allows to easily formulate nonlinear programming problems and solve them with a variety of global and local optimization algorithms. The main adjustable parameters of our algorithm are: the $k_{\max}$ terminating parameter for VNS and the number s of local searches in each VNS neighbourhood.

4 Computational Results

In this section we report on the computational results, obtained on an Intel 2.66GHz Pentium IV CPU with 1GB RAM running Linux. The algorithmic parameters have been set to the following default values: $k_{\max} = 10$, $s = 1$. The initial value λ_0 for the smoothing parameter is chosen so that a reasonable number of close atoms are clustered together:

$$\lambda_0 = \frac{1}{|V|} \sum_{i \in V} \frac{1}{|\delta(i)|} \sum_{j \in \delta(i)} d_{ji}, \tag{12}$$

where $\delta(i)$ is the neighbourhood of vertices adjacent to vertex i in the graph G.

4.1 Instances

Our computational tests refer to two sets of instances: the "Moré-Wu" instances [13] and the "Lavor" instances [7].

The "Moré-Wu" instances are simply a cubic lattice with s^3 atoms ($s = 1, 2, 3, ...$) defined by

$$\{(i_1, i_2, i_3) \in \mathbb{R}^3 : 0 \leq i_k \leq s - 1,\ k = 1, 2, 3\}. \tag{13}$$

See Fig. 3 for an example with $s = 3$. An order is defined for the atoms of the lattice by letting atom i be the atom at position (i_1, i_2, i_3), where

$$i = 1 + i_1 + s i_2 + s^2 i_3, \tag{14}$$

and the set E, is defined by

$$E = \{\{i, j\} : |i - j| \leq s^2\}. \tag{15}$$

For example, for a molecule with 8 atoms ($s = 2$), the sequence of atoms is

$$\begin{aligned} &x_1 = (0,0,0), \quad x_2 = (1,0,0), \quad x_3 = (0,1,0), x_4 = (1,1,0), \\ &x_5 = (0,0,1), \quad x_6 = (1,0,1), \quad x_7 = (0,1,1), x_8 = (1,1,1), \end{aligned}$$

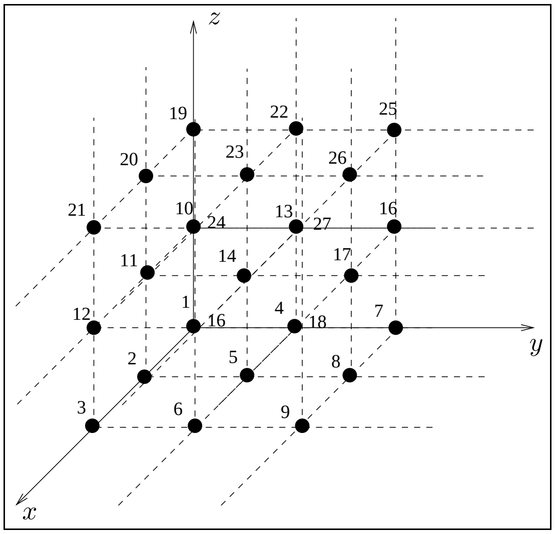

Figure 3: The $s = 3$ Moré-Wu instance with 27 atoms.

and the set E is given by

$$\begin{aligned} E \quad = \quad & \{\{1,2\},\{1,3\},\{1,4\},\{1,5\},\{2,1\},\{2,3\},\{2,4\},\{2,5\},\{2,6\},\{3,1\},\{3,2\},\{3,4\},\{3,5\}, \\ & \{3,6\},\{3,7\},\{4,1\},\{4,2\},\{4,3\},\{4,5\},\{4,6\},\{4,7\},\{4,8\},\{5,1\},\{5,2\},\{5,3\},\{5,4\},\{5,6\}, \\ & \{5,7\},\{5,8\},\{6,2\},\{6,3\},\{6,4\},\{6,5\},\{6,7\},\{6,8\},\{7,3\},\{7,4\},\{7,5\},\{7,6\},\{7,8\}\}. \end{aligned}$$

The "Lavor" instances, described in [7], are based on the model proposed by [16], whereby a molecule is represented as a linear chain of atoms. Bond lengths and angles are kept fixed, and a set of likely torsion angles is generated randomly based on an energy function including some Lennard-Jones potential terms. Depending on the initial choice of bond lengths and angles, the Lavor instances give rather more realistic models of proteins than the Moré-Wu instances do. We generated 10 different Lavor instances for each size $N = 10, \ldots, 70$. These are called `lavor`N-m, where N is the number of atoms in the molecule and m is an instance ID (since there is a random element of choice in the generation of the Lavor instances, many different instances can be generated having the same atomic size). See Fig. 4 for an example.

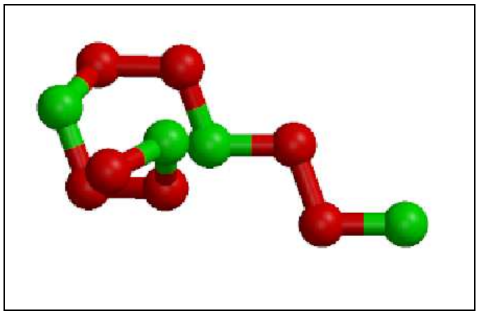

Figure 4: The `lavor11_7` instance.

4.2 Comparative results

In Table 1, we report on the comparative computational results (seconds of user CPU time and Largest Distance Errors) obtained by running the GCA and DVS algorithms on a selection of Moré-Wu and Lavor instances of molecular sizes in the range 8-70 atoms. For molecules with fewer than 50 atoms, the DVS algorithm improved on the straight VNS in only around half of the instances. For molecules with more than 40 atoms, the DVS improves on a straight VNS for the large majority of the instances.

Instance			GCA		**DVS**	
Name	N	$\lvert E\rvert$	CPU	LDE	**CPU**	**LDE**
mmorewu-2	8	28	0.02	2.63E+5	1.09	2.51E-8
mmorewu-3	27	331	0.23	6.99	27.05	2.25E-9
mmorewu-4	64	1882	0.67	7.79E-6	642.97	2.75E-10
lavor10_0	10	33	0.02	1.57E-5	2.76	1.55E-9
lavor15_0	15	57	0.10	4.04E-5	10.01	3.77E-9
lavor20_0	20	105	0.14	2.77E-5	18.14	2.68E-9
lavor25_0	25	131	0.84	1.18E-4	60.41	2.24E-9*
lavor30_0	30	169	0.40	1.75E-5	231.02	6.51E-9*
lavor35_0	35	171	0.81	9.33E-5	624.72	4.54E-3*
lavor40_0	40	295	2.84	0.096	770.78	2.46E-5
lavor45_0	45	239	3.33	0.170	538.25	3.13E-4
lavor50_0	50	271	3.45	0.696	971.79	8.66E-6
lavor55_0	55	551	5.80	0.257	870.50	1.13E-8*
lavor60_0	60	377	5.15	0.049	1800.35	2.85E-4
lavor65_0	65	267	2.61	0.065	1119.82	3.94E-3
lavor70_0	70	431	8.73	0.107	2165.81	4.97E-4

Table 1: Computational results for a sample of Moré-Wu, and Lavor instances. LDE values marked with * have been found by the pre-processing VNS run on the unrestricted original problem (i.e. the smoothing run did not improve the value).

In Table 2, we give arithmetic average values for user CPU time and Largest Distance Error for each sample of 10 Lavor instances. The average LDE is calculated considering only those instances in the sample for which DVS calculated a suitable solution (we consider a solution clearly unsuitable if its associated LDE exceeds 0.01, as such a large LDE usually indicates the wrong 3D structure). We therefore included two columns in Table 2, labelled "LDE-st" and "Unsuitable" that report: the arithmetic average of the LDE for instances with LDE < 0.01, and the number of unsuitable instances in each 10-sample.

Instance	GCA / avg.		**DVS / avg.**		
N	CPU	LDE	**CPU**	**LDE-st**	**Unsuitable**
10	0.03	4.40E-01	2.81	3.04E-9	0
15	0.08	1.96E-02	10.07	3.34E-9	0
20	0.23	3.20E-03	22.01	3.59E-9	1
25	0.56	1.58E-02	46.19	3.64E-5	1
30	0.65	1.03E-02	276.40	2.94E-5	2
35	1.10	5.43E-02	465.72	2.37E-3	1
40	1.41	2.61E-02	486.64	4.03E-4	1
45	2.13	5.80E-02	752.93	1.69E-3	1
50	2.54	1.65E-01	863.14	1.06E-3	0
55	4.10	7.29E-02	762.78	4.03E-4	0
60	4.47	1.59E-01	2172.23	8.06E-4	0
65	4.64	1.16E-01	1404.57	1.97E-3	0
70	7.63	9.28E-02	1912.97	4.12E-4	0

Table 2: Average statistics for Lavor instances (taken over 10 instances for each molecular size).

As can be easily seen from the results, DVS outperforms the GCA in accuracy, whilst the GCA is superior to the DVS in terms of computation times. Since usually looking for the 3D structure of a molecule is a task where accuracy is more important than short CPU times, we feel these computational results validate the soundness of the proposed approach.

5 Conclusion

In this paper we presented an algorithm called Double VNS with Smoothing (DVS) used to solve the Molecular Distance Geometry Problem. The DVS is based on VNS for global optimization problems and a smoothed version of the problem. We tested this algorithm on two classes of problems from the literature. It turns out that the DVS algorithm finds solutions with high accuracy, compared to the Global Continuation Algorithm. This is important insofar as a small error in the objective function may lead to a completely different molecular structure.

Acknowledgements

Some of the authors (CL, NM, LL) would like to thank FAPESP and CNPq for their financial support.

References

[1] G.M. Crippen and T.F. Havel. *Distance Geometry and Molecular Conformation.* Wiley, New York, 1988.

[2] I.F. Cruz and J.P. Twarog. 3d graph drawing with simulated annealing. In F.-J. Brandenburg, editor, *3D Graph Drawing – GD95 Proceedings, LNCS*, volume 1027, pages 162–165, Berlin, 1996. Springer.

[3] Q. Dong and Z. Wu. A linear-time algorithm for solving the molecular distance geometry problem with exact inter-atomic distances. *Journal of Global Optimization*, 22:365–375, 2002.

[4] M. Dražić, C. Lavor, N. Maculan, and N. Mladenović. A continuous variable neighbourhood search heuristic for finding the tridimensional structure of a molecule. *European Journal of Operations Research*, to appear.

[5] T. Eren, D.K. Goldenberg, W. Whiteley, Y.R. Yang, A.S. Morse, B.D.O. Anderson, and P.N. Belhumeur. Rigidity, computation, and randomization in network localization. *IEEE Infocom Proceedings*, pages 2673–2684, 2004.

[6] P.E. Gill. *User's Guide for SNOPT 5.3.* Systems Optimization Laboratory, Department of EESOR, Stanford University, California, February 1999.

[7] C. Lavor. On generating instances for the molecular distance geometry problem. In Liberti and Maculan [11], pages 405–414.

[8] C. Lavor, L. Liberti, and N. Maculan. Computational experience with the molecular distance geometry problem. In J. Pintér, editor, *Global Optimization: Scientific and Engineering Case Studies*, pages 213–225. Springer, Berlin, 2006.

[9] L. Liberti and M. Dražic. Variable neighbourhood search for the global optimization of constrained NLPs. In *Proceedings of GO Workshop, Almeria, Spain*, 2005.

[10] L. Liberti, C. Lavor, and N. Maculan. A branch-and-prune algorithm for the molecular distance geometry problem. *RAIRO-RO*, (submitted).

[11] L. Liberti and N. Maculan, editors. *Global Optimization: from Theory to Implementation*. Springer, Berlin, 2006.

[12] L. Liberti, P. Tsiakis, B. Keeping, and C.C. Pantelides. *oo$\mathcal{OPS}$*. Centre for Process Systems Engineering, Chemical Engineering Department, Imperial College, London, UK, 2001.

[13] J.J. Moré and Z. Wu. Global continuation for distance geometry problems. *Siam Journal of Optimization*, 7(3):814–846, 1997.

[14] J.J. Moré and Z. Wu. Distance geometry optimization for protein structures. *Journal of Global Optimization*, 15:219–234, 1999.

[15] A. Neumaier. Molecular modeling of proteins and mathematical prediction of protein structure. *SIAM Reviews*, 39:407–460, 1997.

[16] A.T. Phillips, J.B. Rosen, and V.H. Walke. Molecular structure determination by convex underestimation of local energy minima. In P.M. Pardalos, D. Shalloway, and G. Xue, editors, *Global Minimization of Nonconvex Energy Functions: Molecular Conformation and Protein Folding*, volume 23, pages 181–198, Providence, 1996. American Mathematical Society.

[17] J.B. Saxe. Embeddability of weighted graphs in k-space is strongly np-hard. *Proceedings of 17th Allerton Conference in Communications, Control and Computing*, pages 480–489, 1979.

[18] J.-M. Yoon, Y. Gad, and Z. Wu. Mathematical modeling of protein structure using distance geometry. Technical Report TR00-24, Dept. Comput. Applied Maths, Rice University, Houston, 2000.