Double VNS for the Molecular Distance Geometry Problem

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

[1]CNRS LIX, Ecole Polytechnique, F-91128 Palaiseau, France.
`liberti@lix.polytechnique.fr`

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

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

Abstract

In this paper we propose a VNS-based algorithm for the solution of the Molecular Distance Geometry Problem. First, we use VNS to solve a smoothed version of the problem to identify the most promising zone in the solution space. We then use VNS again to solve the original problem restricted to the promising zone. This algorithm often manages to find a solutions having higher accuracy than other methods. This is important as small differences in the objective function value may mean completely different 3D molecular structures.

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

1 Introduction

The Molecular Distance Geometry Problem (MDGP) is the problem of finding a weighted graph embedding in $\mathbb{R}^3$ 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, 13]. There are other applications in network localization [5] and graph drawing [2].

It has been shown that the MDGP is NP-hard [15]; if G is a complete graph, however, the MDGP can be solved in linear time [3].

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 *interatomic 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 [16] 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) [11, 12]. The GCA relies on a class of smoothed objective functions derived from (2) by means of a Gaussian transforms. 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 the solution accuracy, the authors 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.

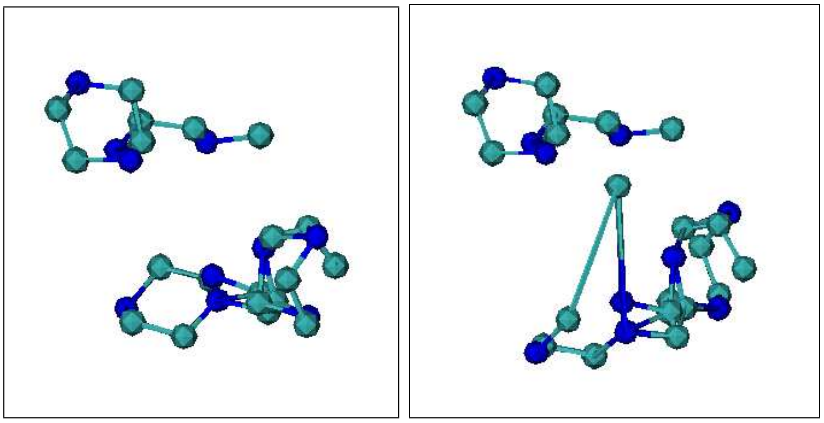

Figure 1: Different 3D graph embeddings with very similar objective function values (both values are in the order of of 10^{-11}).

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 also that VNS was used to find the 3D structure of molecules using a potential energy related objective, rather than distances [4]. In this paper, we combine ideas from smoothing and VNS to obtain a method which we call Double VNS with Smoothing (DVS). Although it is slower than `dgsol`, the obtained solutions generally have much smaller LDE with respect to `dgsol`.

The rest of this paper is organized as follows. In Section 2 we present the algorithm. In Section 3 we report on the computational results. Section 4 concludes the paper.

2 The algorithm

The Double VNS with Smoothing algorithm is conceptually very simple. As a pre-processing step, we solve the original problem via VNS to obtain a solution $\tilde{x}$ (this is used to verify whether the following steps actually improve the solution). Next, we solve the smoothed problem via VNS to obtain a solution $\bar{x}$, then solve the original problem restricted to a hypercubic neighbourhood R around $\bar{x}$ to obtain the solution x^*. If $f(\tilde{x}) < f(x^*)$ we replace x^* with $\tilde{x}$, and the algorithm terminates. Notice that this algorithm is actually just a sequence of two VNS solutions on related problem formulations; no iteration is involved. This algorithm rests on two fundamental building blocks: the smoothed objective function and the VNS algorithm.

2.1 Smoothing

The smoothed objective function, parametrized by λ, is derived in [11] as follows:

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

where γ is a constant. For the choice of λ, we followed the recommendations given in the computational results sections of [11] and [12].

2.2 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 $x^L \leq x \leq x^U$. 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 $k \leq k_{\max}$ we consider hyper-parallelepipeds $H_k(x^*)$ proportional to $x^L \leq x \leq x^U$, 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 $H_k(x^*) \backslash H_{k-1}(x^*)$.

2.3 Parameters

The main adjustable parameters of our algorithm are:

- the smoothing parameter λ;
- the $k_{\max}$ terminating parameter for VNS;
- the number s of local searches in each VNS neighbourhood;
- the size of the restricted neighbourhood R (i.e. R is a hypercube with side $2b$ centered at $\bar{x}$).

2.4 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 [10], which allows to easily formulate nonlinear programming problems and solve them with a variety of global and local optimization algorithms.

3 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: $\lambda = \frac{1}{2}(\min_{\{i,j\}\in E} d_{ij} + \max_{\{i,j\}\in E} d_{ij})$, $k_{\max} = 10$, $s = 1$, $b = 1$.

3.1 Instances

Our computational tests refer to two sets of instances: the “Moré-Wu” instances [11] and the “Lavor” instances [7].

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

$$\{(i_1, i_2, i_3) \in \mathbb{R}^3 : 0 \leq i_k \leq s - 1,\ k = 1, 2, 3\}. \tag{5}$$

See Fig. 2 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

$$i = 1 + i_1 + si_2 + s^2 i_3, \tag{6}$$

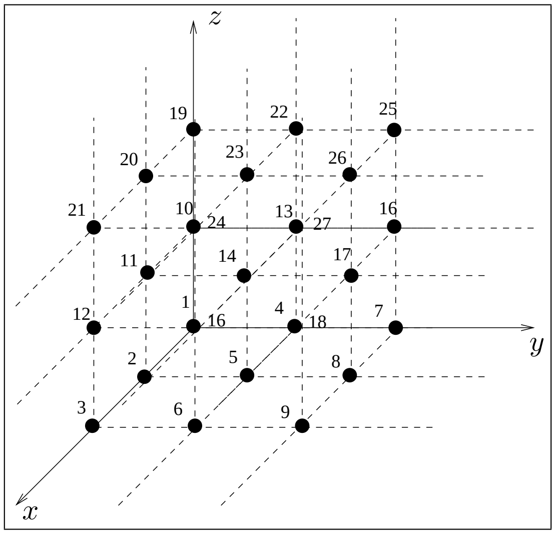

Figure 2: The $s = 3$ Moré-Wu instance with 27 atoms.

and the set E, is defined by

$$E = \{\{i, j\} : |i - j| \leq s^2\}. \tag{7}$$

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

$$x_1 = (0, 0, 0), \quad x_2 = (1, 0, 0), \quad x_3 = (0, 1, 0), x_4 = (1, 1, 0),$$
$$x_5 = (0, 0, 1), \quad x_6 = (1, 0, 1), \quad x_7 = (0, 1, 1), x_8 = (1, 1, 1),$$

and the set E is given by

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

The “Lavor” instances, described in [7], are based on the model proposed by [14], 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 by minimization of an energy function including some Lennard-Jones potential terms. Depending on the initial choice of bond lengths and atoms, 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. 3 for an example.

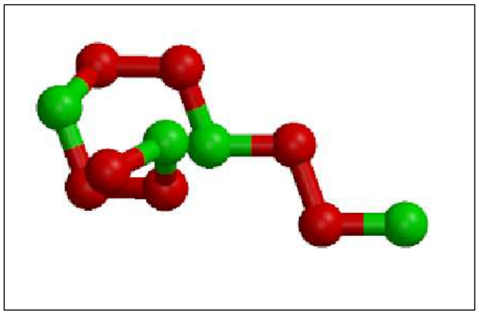

Figure 3: The `lavor11_7` instance.

3.2 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.

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.

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.

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).

4 Conclusion and Future Work

In this paper we presented an algorithm called Double VNS with Smoothing 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 existing methods. This is important insofar as a small error in the objective function may lead to a completely different molecular structure.

Future research in this direction will focus on employing instance information dynamically to adjust some of the DVS parameters in a smart way. Furthermore, we intend to test a variant of the DVS where the smoothed problem is replaced by a relaxed problem where only some of the interatomic distances are considered.

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).

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, *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. Drazić, C. Lavor, N. Maculan, and N. Mladenović. A continuous VNS heuristic for finding the tridimensional structure of a molecule. *Le Cahiers du GERAD*, G-2004-22, 2004.

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

[8] C. Lavor, L. Liberti, and N. Maculan. Computational experience with the molecular distance geometry problem. In J. Pintér, editor, *Global Optimization: Selected Case Studies*, Springer, Berlin, to appear.

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

[10] 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.

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

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

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

[14] 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.

[15] 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.

[16] 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.