Unassigned distance geometry and the Buckminsterfullerene*

Leo Liberti

Abstract The Buckminsterfullerene is an inorganic molecule consisting of 60 carbon atoms in the shape of a soccer ball. It was used in [15] to showcase algorithms that find the correct shape of a protein from limited data (length of inter-atomic distances) without any further chemical experiment: in that case, by means of a complicated constructive heuristic based on genetic algorithms. In this paper we show that we can reconstruct the Buckminsterfullerene structure by means of mathematical programming, standard solver software, and little else.

Key words: mathematical programming, molecule, diagonal dominance, matrix cones.

1 Introduction

The authors of [15] state that "*ab initio* structure solution of nanostructured materials [that cannot be solved using crystallographic methods] is feasible using diffraction data in combination with distance geometry methods". They then proceed to find the structure of Buckminsterfullerene (a highly symmetric molecule consisting of 60 carbon atoms [25]) using a heuristic method, called LIGA, that employs distance values in order to construct small rigid polytopes which are then combined into larger structures. The combination is achieved by means of a genetic algorithm approach that involve tournaments for selection of the fittest. Some backtracking also occurs.

Leo Liberti
LIX CNRS Ecole Polytechnique, Institut Polytechnique de Paris, 91128 Palaiseau, France
e-mail: liberti@lix.polytechnique.fr

* This paper was partly supported by the ANR PRCI project "MultiBioStruct".

The input to LIGA (number of atoms and list of all inter-atomic distance values) is obtained from powder diffraction data: we note that the graph underlying the distance values is not a part of the input. The experiment in [15, Fig. 2c-d] suggests that the algorithmic reconstruction process yields some errors; but that it suffices to increase the multiplicities of the distance values in the list to obtain the correct shape.

In this paper we show, using Mathematical Programming (MP) methods, that a correct structure of the Buckminsterfullerene can be obtained from the original distance list, without having to add multiple copies of each distance value, and without having to create a new algorithm. Most of the work is carried out on one of two MP formulations involved in the reconstruction problem, as explained below. The only algorithmic contribution is given by a "vanilla" MultiStart (MS) approach.

Our interest in the Buckminsterfullerene only extends so far as it constitutes a well-known and interesting instance of the Unassigned Distance Geometry Problem (UDGP), defined below. Many people have managed to reconstruct the shape of the Buckminsterfullerene: most of them have used heuristic optimization methods to find "good" minima of an energy formula derived from the partial differential equations that define the atomic force field [22, 10, 6]. We follow a different path, namely that of [15], which takes as input the values of the inter-atomic distances of the molecule.

1.1 Two problems in distance geometry

We now introduce the Distance Geometry Problem (DGP) and its "unassigned" variant, the UDGP. The input of the DGP is an integer K and a simple edge-weighted undirected graph $G = (V, E, d)$ where $d : E \to \mathbb{R}_+$. The DGP asks whether there exists a *realization* $x : V \to \mathbb{R}^K$ of G in $\mathbb{R}^K$ satisfying

$$\forall \{i, j\} \in E \quad \|x_i - x_j\|_2^2 = d_{ij}^2. \tag{1}$$

In other words, the DGP asks us to draw the graph G in K dimensions such that edges are drawn as segments having length equal to the weight [19]. The DGP is **NP**-hard [24], it is in **NP** if $K = 1$ but it is unknown whether it is in **NP** for $K > 1$ [3]. It has applications in designing time synchronization protocols in wireless networks, in the localization of mobile sensors, in the determination of protein structure from Nuclear Magnetic Resonance (NMR) experiments, and in many other fields. See [20, 9] for more information about the DGP.

The UDGP removes the adjacency information from the graph edges: its input is a pair of integers K, n and a list δ of m positive scalars. It asks whether there exists an assignment α mapping $[m] = \{1, \ldots, m\}$ to the set $\bar{E} = \{\{i, j\} \mid i, j \in [n]\}$, and a realization $x : [n] \to \mathbb{R}^K$ such that

$$\forall \ell \leq m \text{ with } \alpha(\ell) = \{i, j\} \quad \|x_i - x_j\|_2^2 = \delta_\ell^2. \tag{2}$$

Although the UDGP is not as well known as the DGP, in many applications it is the problem that is actually closest to the experiments leading to the use of the DGP. For example, the determination of protein structures from distances involves the solution of a DGP, but the actual raw input from the NMR experiments is closer to a list of values δ than to a graph G. In fact, G is inferred from δ because the assignment α is computed first, together with a very imprecise realization [26], see Sect. 3.1 below.

The UDGP has applications in the study of DNA [17, 8], where it is known as the *turnpike* or *beltway* or *partial digest* problem. It is the natural modelling framework for the study of nanostructures [4] such as the Buckminsterfullerene. It also arises in the context of protein structure [26], but in this setting $|\delta|$ (the length of the sequence δ) is much smaller than the number of atom pairs $N = n(n-1)/2$, which yields very imprecise assignment functions α. Conversely, it is often the case that all (or almost all) pairwise distances can be measured by NMR interaction in molecules that are much smaller than proteins (e.g. the Buckminsterfullerene) [15]. In turn, this yields much more precise assignments. The UDGP is **NP**-hard even if $m = N$ [17].

Both DGP and UDGP can be defined for any K, and there exist many applications for $K \in \{1,2,3\}$, as well as some for larger values of K. In this paper we fix $K = 3$, since the Buckminsterfullerene is realized in 3D space.

1.2 Goal of this paper

In this paper we solve the UDGP instance known as "Buckminsterfullerene" [16] by means of MP, a formal language for specifying and solving optimization problems. We recall an existing Mixed-Integer Nonlinear Programming (MINLP) formulation of the UDGP, and propose a new Mixed-Integer Quadratically Constrained Programming (MIQCP) formulation. These can only be solved for tiny inputs much smaller than the Buckminsterfullerene. We therefore decompose the decision on the assignment α and the realization x: we construct an approximate Mixed-Integer Linear Programming (MILP) formulation of the UDGP which can be solved to larger sizes. We obtain an approximate assignment α' and an approximate realization, which we discard. We then use α' to define an edge-weighted graph $G_{\alpha'}$, and obtain a more precise realization by solving the DGP on $G_{\alpha'}$. Finally, we show the perfectly reconstructed shape of the Buckminsterfullerene from a distance list δ of size N.

2 MINLP formulations for the UDGP

The UDGP aims at finding an assignment $\alpha : [m] \to \bar{E}$ and a realization x consistent with each other. We note that realizations are defined as functions from V to $\mathbb{R}^K$ that satisfy the distance equations (1). Therefore x can be represented by an $n \times K$ matrix, where $n = |V|$: every row x_i of x is a vector in $\mathbb{R}^K$ for $i \in V$. We can encode the assignment α by means of binary decision variables $y_{ij\ell}$ such that $y_{ij\ell} = 1$ iff $\alpha(\ell) = \{i, j\}$. The assignment α satisfies the following constraints: each distance value δ_ℓ for $\ell \leq m$ is assigned to exactly one edge $\{i, j\}$, and each pair $\{i, j\}$ is assigned to at most one ℓ (some pairs may remain unassigned).

2.1 Exact MINLP formulations

These considerations yield the following "natural" MP formulation for the UDGP, proposed in [11]:

$$\left.\begin{array}{rc} \min\limits_{x,y} & \sum\limits_{i<j\leq n}\sum\limits_{\ell\leq m} y_{ij\ell}(\|x_i - x_j\|_2^2 - \delta_\ell^2)^2 \\ \forall \ell \leq m & \sum\limits_{i<j\leq n} y_{ij\ell} = 1 \\ \forall i < j \leq n & \sum\limits_{\ell\leq m} y_{ij\ell} \leq 1 \\ & x \in \mathbb{R}^{nK} \quad \wedge \quad y \in \{0,1\}^{Nm} \end{array}\right\} \tag{3}$$

Since Eq. (3) has nonlinear terms and integer variables, it is a MINLP formulation with a quintic polynomial as objective.

We propose a new MIQCP formulation. It decreases the polynomial degree of Eq. (3) from five to two.

$$\left.\begin{array}{rc} \min\limits_{x,y,z} & \sum\limits_{i<j\leq n}\sum\limits_{\ell\leq m} (z^+_{ij\ell} + z^-_{ij\ell}) \\ \forall i < j \leq n, \ell \leq m & -z^-_{ij\ell} - M(1-y_{ij\ell}) \leq \|x_i - x_j\|_2^2 - \delta_\ell^2 \leq z^+_{ij\ell} + M(1-y_{ij\ell}) \\ \forall \ell \leq m & \sum\limits_{i<j\leq n} y_{ij\ell} = 1 \\ \forall i < j \leq n & \sum\limits_{\ell\leq m} y_{ij\ell} \leq 1 \\ & x \in \mathbb{R}^{nK} \ \wedge \ y \in \{0,1\}^{Nm} \ \wedge \ z \geq 0, \end{array}\right\} \tag{4}$$

where $M = (\sum_{\ell\leq m} \delta_\ell)^2$ (see [5, Prop. 2.2]). "Sandwiching" the distance constraints allows constraint activity whenever $\alpha(\ell) = \{i, j\}$, and trivialize the constraint otherwise. The slack variables z^-, z^+ allow the solver to find an approximately feasible solution with imprecise distances.

2.2 An approximate MILP formulation

The only nonlinear part of Eq. (4) is the Euclidean distance term $\|x_i - x_j\|_2$ in the middle member of the distance constraints. We have:

$$\|x_i - x_j\|_2 = \|x_i\|_2 + \|x_j\|_2 - 2\langle x_i, x_j\rangle = \langle x_i, x_i\rangle + \langle x_j, x_j\rangle - \langle x_i, x_j\rangle.$$

Now we linearize the nonlinear (inner product) terms: we create an $n \times n$ variable matrix X, and we replace each term $\langle x_i, x_j\rangle$ by X_{ij} (which, by commutativity, yields a symmetric matrix X). For all $i < j$ and ℓ, we obtain the linearized distance constraint

$$-z^-_{ij\ell} - M(1 - y_{ij\ell}) \le X_{ii} + X_{jj} - 2X_{ij} - \delta^2_\ell \le z^+_{ij\ell} + M(1 - y_{ij\ell}).$$

So far, this is the usual first step in the construction of the Semidefinite Programming (SDP) relaxation for MP formulations with quadratic terms. However, instead of imposing the positive semidefinite (PSD) constraint $X \succeq 0$, we restrict our attention to a polyhedral inner approximation thereof, that of *Diagonally Dominant* (DD) matrices. An $n \times n$ matrix is DD if it satisfies:

$$\forall i \le n \quad X_{ii} \ge \sum_{j \ne i} |X_{ij}|. \tag{5}$$

By Gershgorin's Circle Theorem [13], every eigenvalue λ_i of a matrix A is contained in an interval $[A_{ii} - \sum_{j\ne i} |A_{ij}|\,, A_{ii} + \sum_{j \ne i} |A_{ij}|]$. By the definition of DD matrices (Eq. (5)), if X is DD, then the lower interval bound $X_{ii} - \sum_{j \ne i} |X_{ij}|$ is non-negative, which implies that every eigenvalue of X is non-negative, which means that X is PSD. We note that the converse need not hold: many PSD matrices are not DD.

The DD cone is polyhedral: its extreme rays are given by the rank-one matrices $e_i e_i^\top$ and $(e_i \pm e_j)(e_i \pm e_j)^\top$ [2]. We therefore proceed to linearize the absolute value term in Eq. (5) by means of an additional $n \times n$ matrix variable T, obtaining the linear reformulation [1, Thm. 3.9]:

$$\forall i \le n \quad X_{ii} \ge \sum_{j \ne i} T_{ij} \qquad \wedge \qquad -T \le X \le T.$$

We remark that linear programming over the DD cone is also known as DD Programming (DDP).

We now put everything back together to yield the following MIDDP formulation:

$$\left.\begin{array}{rrcl}
\min\limits_{X,T,y,z} & \sum\limits_{i<j\le n} \sum\limits_{\ell \le m} (z^+_{ij\ell} & + & z^-_{ij\ell}) \\
\forall i < j \le n, \ell \le m & -z^-_{ij\ell} - M(1 - y_{ij\ell}) & \le & X_{ii} + X_{jj} - 2X_{ij} - d^2_\ell \le z^+_{ij\ell} + M(1 - y_{ij\ell}) \\
\forall \ell \le m & \sum\limits_{i<j\le n} y_{ij\ell} & = & 1 \\
\forall i < j \le n & \sum\limits_{\ell \le m} y_{ij\ell} & \le & 1 \\
\forall i \le n & X_{ii} & \ge & \sum\limits_{j \le n : j \ne i} T_{ij} \\
 & -T & \le & X \le T \\
 & X, T \in \mathbb{R}^{n \times n} & \wedge & y \in \{0,1\}^{Nm} \ \wedge\ z \ge 0.
\end{array}\right\} \tag{6}$$

We note that Eq. (6) is a MILP, for which there exist technically viable solvers even for relatively large sizes [14]. We also remark that, although this reformulation may not immediately appear simple, it is based on a set of well-known building blocks.

Since not every PSD matrix is DD, a DDP is an inner approximation of the corresponding SDP, which means that, potentially, the DDP might be infeasible even if the SDP is not. This is not an issue for the MIDDP formulation in Eq. (6) since it is always feasible: it suffices to satisfy the DD constraints first, and then choose large enough slack variables z^+, z^- to accommodate the error.

In practice, solving Eq. (6) yields an $n \times n$ matrix X. There are two ways to obtain an $n \times K$ (approximate) realization of δ.

1. *Principal Component Analysis* (PCA). Let $X = P \Lambda P^{\top}$ be a spectral decomposition of X. The diagonal matrix Λ is non-negative because X is DD and therefore PSD, so that $\sqrt{\Lambda}$ is a real matrix. Let Λ^K be the matrix obtained by Λ by zeroing all of the diagonal entries aside from the K largest. Then $\hat{x} = P\sqrt{\Lambda^K}$ has the property that $\bar{X} = \hat{x}\hat{x}^{\top} = P\Lambda^K P^{\top}$ is a best rank-K approximation of X.
2. *Barvinok's "naive algorithm"* generalized to K dimensions [21]. Proceeding as for PCA, consider the factor $F = P\sqrt{\Lambda}$ of X, then sample (componentwise) an $n \times K$ matrix $Z \sim \mathsf{Normal}(0, 1/\sqrt{K})$. The $n \times K$ matrix $\check{x} = FZ$ has good proximity properties with the constraints $\|x_i - x_j\|_2 = \bar{y}_{ij\ell}\delta_\ell$ with arbitrarily high probability (here $\bar{y}$ are the assignment variables for α having value 1 after solving the MIDDP: they simply select the correct ℓ to match to the $\{i, j\}$).

Either of these two ways would allow us to retrieve an approximate realization x from the DD matrix X, but experiments in this yielded very wrong realizations.

3 A matheuristic for the UDGP

Matheuristics are heuristic algorithms based on MP formulations [23]. Our matheuristic decomposes the decision on the assignment α and the realization x.

3.1 The way of proteins and their scale

This decomposition idea is not new: the DGP application to protein conformation derives the distance values from NMR experiments, which in fact yield an ambiguous assignment. This assignment is disambiguated in two phases. The first rests on chemical considerations. The second phase is based on a metaheuristic (usually simulated annealing) which returns an assignment together with a poor quality realization, which is discarded. This assignment, however, is used to define a weighted graph that becomes the input of a DGP. The solution of this DGP provides the realization of interest, and the shape of the protein.

There are two technical differences with the Buckminsterfullerene: powder diffraction data does not yield any assignment (not even an ambiguous one), and the first disambiguation phase does not apply because of the atomic uniformity of the Buckminsterfullerene: it consists entirely of carbon atoms.

The main difference w.r.t. methodologies for computing the shape of proteins and of inorganic molecules such as the Buckminsterfullerene, however, is one of scale. Proteins are much larger than the range of distances that can be measured using NMR, which is around 5.5Å. As a consequence, only very few distances can be obtained. The protein graphs that provide the DGP input by means of which we compute realizations are very sparse.

The Buckminsterfullerene — and similar molecules — are much smaller than proteins. As a consequence, we can often obtain lists δ of distance values that have length comparable with the total number N of index pairs. Measurements are never free from errors, so having "complete" lists of distances does not mean that the list is correct. The authors of [15] imply that a source of error is that δ may contain *more* distances than needed: for each given distance value, its number of occurrences in δ may be overestimated. We do not simulate this type of error here, but we do look at how the structure reconstruction changes as $|\delta|$ grows to reach N.

3.2 The algorithm

For the above reasons, we propose a different methodology, based on MP, to implement the decomposition between α and x.

The matheuristic we propose is as follows:

1. compute an assignment α and an approximate realization x' using the MIDDP (MILP) formulation Eq. (6);
2. discard the realization x'
3. construct the weighted graph G from α
4. define a DGP instance on G and solve it, to obtain a realization x^*
5. return x^*.

3.2.1 Solving the MIDDP

Even if MILP solvers are technically very advanced, the MIDDP formulation in Eq. (6) is very challenging to solve at the instance scale yielded by the Buckminsterfullerene. We therefore solve it "locally", which, for a MILP solver, means setting a resource limit (either number of Branch-and-Bound nodes, or time). The solution returned by the solver is an approximate assignment.

3.2.2 Solving the DGP

The DGP consists in finding a realization x satisfying Eq. (1). Many different formulations of the DGP have been discussed in the literature [20, 18]. In this case we use the quartic formulation:

$$\min_{x\in\mathbb{R}^{nK}} \sum_{\{i,j\}\in E} (\|x_i - x_j\|_2^2 - d_{ij}^2)^2. \tag{7}$$

It is unconstrained and nonconvex in the x variables. As for the MIDDP, solving Eq. (7) exactly is out of the technological reach of global optimization solvers, at least at the scale of the Buckminsterfullerene. We therefore use a simple MS heuristic: at each iteration we sample a random starting point, then call the local optimization solver IPOPT [7] to identify a close local minimum, choosing the best over the allowed iterations.

4 Finding the shape of the Buckminsterfullerene

We implemented our formulation using AMPL [12]. We used a smaller M value in solving the MIDDP than the one mentioned in [5, Prop. 2.2], since it is far from being tight. We set $M = \sum_\ell \delta_\ell^2$ in our computational experiments.

We solved the MIDDP formulation using CPLEX 22.1 [14]. The local solver deployed on Eq. (7) is IPOPT 3.4.12. All experiments were carried out on an Apple M1 Max with 64GB RAM running MacOS Ventura 13.6.4. All software was natively compiled for the ARM64 architecture.

CPLEX was given a time limit of 360s (six minutes — extending to one hour yielded no appreciable difference). The MS algorithm used to solve the DGP in Eq. (7) was allowed to run for 10 iterations. IPOPT was given no time limit.

The Buckminsterfullerene has $n = 60$ vertices and $N = 1770$ distance values, many of them with multiplicities, given its molecular symmetry. We computed the exact realization according to [25] and extracted a complete list δ of its interatomic distances. From this instance we also generated instances with smaller $|\delta|$ through a probabilistic Erdős-Rényi edge removal choice. We examined instances with $|\delta| \in [0.1N, 1.0N]$, with $1.0N$ corresponding to the complete list. The original realization of the Buckminsterfullerene is given in Fig. 1 (left). We also show a realization for $|\delta| = 0.5N$ (Fig. 1 center), which is rather typical of the whole range of probabilistically generated instances. Lastly, the perfectly reconstructed molecule corresponding to $|\delta| = 1770$ is shown in Fig. 1 (right). It may not appear identical because it is a rotation of the original: but the error returned by the solver was zero.

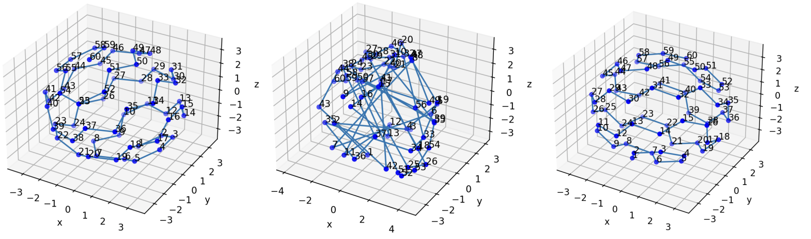

Fig. 1 Left: the Buckminsterfullerene molecule with a spiral order on the vertices. Center: wrong realizations obtained for $|\delta| = 0.5N$. Right: the realization for $|\delta| = N$ yields a perfect reconstruction (rotated).

By trial-and-error we narrowed down an interval of interest for $|\delta|$, i.e. when the resulting shape is not completely wrong but not yet correct, to $|\delta| \in [0.995N, 0.999N]$. For this range we present numerical experiments based on the Mean Distance Error (MDE), i.e. $(1/|E|)\sum_{\{i,j\}\in E} |\, \|x_i - x_j\|_2 - d_{ij} \,|$ computed on the reconstructed weighted graph.

density	0.995	0.996	0.997	0.998	0.999
MDE	0.722	1.057	0.778	0.552	0.603

The realizations corresponding to above densities are shown in Fig. 2. As one can see, the realizations have the correct overall shape with some local errors.

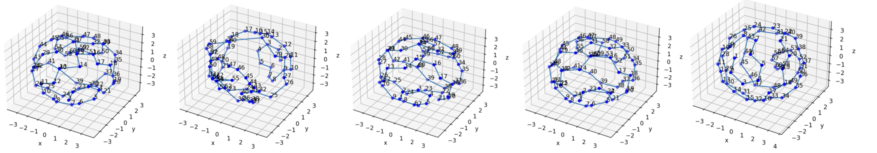

Fig. 2 Slightly wrong realizations for $|\delta| \in [0.995N, 0.999N]$.

We report the mean errors $(1/N)\|\mathsf{Adj}(G') - \mathsf{Adj}(G^*)\|_1$ of the reconstructed graphs G' w.r.t. the correct Buckminsterfullerene graph G^*, where $\mathsf{Adj}(G)$ is the upper triangular part of weighted adjacency matrix of G.

density	0.995	0.996	0.997	0.998	0.999
mean error	1.043	1.291	1.129	0.851	0.720

Finally, the CPU time taken for each instance to be solved was between 1000s and 1200s.

5 Conclusion

The proposed methodology, simpler than that of [15], allowed us to reconstruct the Buckminsterfullerene perfectly from the list of all its distances. It also allowed us to reconstruct locally wrong realizations with very few missing distances. This is in constrast to [15], which required more than N distances (same distance values but with higher multiplicities) in order to reconstruct the molecule perfectly.

References

1. A. Ahmadi and A. Majumdar. DSOS and SDSOS optimization: More tractable alternatives to sum of squares and semidefinite optimization. *SIAM Journal on Applied Algebra and Geometry*, 3(2):193–230, 2019.

2. G. Barker and D. Carlson. Cones of diagonally dominant matrices. *Pacific Journal of Mathematics*, 57(1):15–32, 1975.
3. N. Beeker, S. Gaubert, C. Glusa, and L. Liberti. Is the distance geometry problem in NP? In A. Mucherino, C. Lavor, L. Liberti, and N. Maculan, editors, *Distance Geometry: Theory, Methods, and Applications*, pages 85–94. Springer, New York, 2013.
4. S. Billinge, P. Duxbury, D. Gonçalves, C. Lavor, and A. Mucherino. Assigned and unassigned distance geometry: Applications to biological molecules and nanostructures. *4OR*, 14:337–376, 2016.
5. M. Bruglieri, R. Cordone, and L. Liberti. Maximum feasible subsystems of distance geometry constraints. *Journal of Global Optimization*, 83:29–47, 2022.
6. Y.-H. Cheng, J.-H. Liao, Y.-J. Zhao, J. Ni, and X.-B. Yang. Theoretical investigations on stable structures of $C_{60-n}N_n$ ($n = 2$–12): Symmetry, model interaction, and global optimization. *Carbon*, 154:140–149, 2019.
7. COIN-OR. *Introduction to IPOPT: A tutorial for downloading, installing, and using IPOPT*, 2006.
8. T. Dakić. *On the turnpike problem*. PhD thesis, Simon Fraser University, 2000.
9. I. Dokmanić, R. Parhizkar, J. Ranieri, and M. Vetterli. Euclidean distance matrices: Essential theory, algorithms and applications. *IEEE Signal Processing Magazine*, 1053-5888:12–30, Nov. 2015.
10. J. Doye, R. Leary, M. Locatelli, and F. Schoen. Global optimization of Morse clusters by potential energy transformations. *INFORMS Journal on Computing*, 16(4):371–379, 2004.
11. P. Duxbury, C. Lavor, L. Liberti, and L. de Salles-Neto. Unassigned distance geometry and molecular conformation problems. *Journal of Global Optimization*, 83:73–82, 2022.
12. R. Fourer and D. Gay. *The AMPL Book*. Duxbury Press, Pacific Grove, 2002.
13. S. Gershgorin. über die Abgrenzung der Eigenwerte einer Matrix. *Zvesti Akademii Nauk SSSR. Otdelenie Fizicheskikh i Matematicheskikh Nauk*, 6:749–754, 1931.
14. IBM. *ILOG CPLEX 22.1 User's Manual*. IBM, 2022.
15. P. Juhás, D. Cherba, P. Duxbury, W. Punch, and S. Billinge. *Ab initio* determination of solid-state nanostructure. *Nature*, 440(30):655–658, 2006.
16. H. Kroto, J. Heath, S. O'Brien, R. Curl, and R. Smalley. C_60: Buckminsterfullerene. *Nature*, 318:162–163, 1985.
17. P. Lemke, S. Skiena, and W. Smith. Reconstructing sets from interpoint distances. In B. Aronov and *et al.*, editors, *Discrete and Computational Geometry*, volume 25 of *Algorithms and Combinatorics*, pages 597–631, Berlin, 2003. Springer.
18. L. Liberti. Distance geometry and data science. *TOP*, 28:271–339, 220.
19. L. Liberti and C. Lavor. *Euclidean Distance Geometry: An Introduction*. Springer, New York, 2017.
20. L. Liberti, C. Lavor, N. Maculan, and A. Mucherino. Euclidean distance geometry and applications. *SIAM Review*, 56(1):3–69, 2014.
21. L. Liberti and K. Vu. Barvinok's naive algorithm in distance geometry. *Operations Research Letters*, 46:476–481, 2018.
22. M. Locatelli and F. Schoen. Efficient algorithms for large scale global optimization: Lennard-Jones clusters. *Computational Optimization and Applications*, 26:173–190, 2003.
23. V. Maniezzo, T. Stützle, and S. Voß, editors. *Matheuristics: Hybridizing metaheuristics and mathematical programming*, volume 10 of *Annals of Information Systems*, New York, 2009. Springer.
24. J. 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.
25. P. Senn. Computation of the cartesian coordinates of Buckminsterfullerene. *Journal of Chemical Education*, 72(4):302–303, 1995.
26. K. Wüthrich. NMR studies of structure and function of biological macromolecules (Nobel lecture). *Angewandte Chemie*, 42:3340–3363, 2003.