SIAM REVIEW
Vol. 56, No. 1, pp. 3–69

Euclidean Distance Geometry and Applications*

Leo Liberti[†]
Carlile Lavor[‡]
Nelson Maculan[§]
Antonio Mucherino[¶]

Abstract. Euclidean distance geometry is the study of Euclidean geometry based on the concept of distance. This is useful in several applications where the input data consist of an incomplete set of distances and the output is a set of points in Euclidean space realizing those given distances. We survey the theory of Euclidean distance geometry and its most important applications, with special emphasis on molecular conformation problems.

Key words. matrix completion, bar-and-joint framework, graph rigidity, inverse problem, protein conformation, sensor network

AMS subject classifications. 51K05, 51F15, 92E10, 68R10, 68M10, 90B18, 90C26, 52C25, 70B15, 91C15

DOI. 10.1137/120875909

*Received by the editors May 4, 2012; accepted for publication (in revised form) March 5, 2013; published electronically February 6, 2014. This work was partially supported by the Brazilian research agencies FAPESP, FAPERJ, CNPq, and CAPES and by the French research agency ANR (project ANR-10-BINF-03-08 "Bip:Bip").

http://www.siam.org/journals/sirev/56-1/87590.html

[†]LIX, École Polytechnique, 91128 Palaiseau, France, and IBM T.J. Watson Research Center, Yorktown Heights, NY (liberti@lix.polytechnique.fr, leoliberti@us.ibm.com).

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

[§]Federal University of Rio de Janeiro (COPPE–UFRJ), C.P. 68511, 21945-970, Rio de Janeiro - RJ, Brazil (maculan@cos.ufrj.br).

[¶]IRISA, Université de Rennes I, Rennes, France (antonio.mucherino@irisa.fr).

1. Introduction.

In 1928, Menger gave a characterization of several geometric concepts (e.g., congruence and set convexity) in terms of distances [161]. The results found by Menger, and eventually completed by Blumenthal [28], originated a body of knowledge that goes by the name of *distance geometry* (DG). This survey paper is concerned with what we believe to be the fundamental problem in DG:

> Distance Geometry Problem (DGP). Given an integer $K > 0$ and a simple undirected graph $G = (V, E)$ whose edges are weighted by a nonnegative function $d : E \to \mathbb{R}_+$, determine whether there is a function $x : V \to \mathbb{R}^K$ such that
>
> $$\forall \{u, v\} \in E, \quad \|x(u) - x(v)\| = d(\{u, v\}). \tag{1.1}$$

Throughout this survey, we shall write $x(v)$ as x_v and $d(\{u, v\})$ as d_{uv} or $d(u, v)$; moreover, norms $\| \cdot \|$ will be Euclidean unless marked otherwise (see [60] for an account of existing distances).

Given the vast extent of this field, we make no claim to nor attempt at exhaustiveness. This survey is intended to give the reader an idea of what we believe to be the most important concepts of DG, keeping in mind our own particular application-oriented slant (i.e., molecular conformation).

The function x satisfying (1.1) is also called a *realization* of G in $\mathbb{R}^K$. If H is a subgraph of G and $\bar{x}$ is a realization of H, then $\bar{x}$ is a *partial* realization of G. If G is a given graph, then we sometimes indicate its vertex set by $V(G)$ and its edge set by $E(G)$.

We remark that, for Blumenthal, the fundamental problem of DG was what he called the "subset problem" [28, Chap. IV, sect. 36, p. 91], i.e., finding necessary and sufficient conditions to decide whether a given matrix is a distance matrix (see section 1.1.3). Specifically, for Euclidean distances, necessary conditions were (implicitly) found by Cayley [38], who proved that five points in $\mathbb{R}^3$, four points on a plane, and three points on a line will have zero Cayley–Menger determinant (see section 2). Some sufficient conditions were found by Menger [162], who proved that it suffices to verify that all $(K + 3) \times (K + 3)$ square submatrices of the given matrix are distance matrices (see [28, Thm. 38.1]; other necessary and sufficient conditions are given in Theorem 2.1). The most prominent difference with the DGP is that a distance matrix essentially represents a *complete weighted graph*, whereas the DGP does not impose any structure on G. The first explicit mention we found of the DGP, as defined above, dates from 1978:

> The positioning problem arises when it is necessary to locate a set of geographically distributed objects using measurements of the distances between some object pairs. (Yemini [242])

The explicit note that only some object pairs have known distance makes the crucial transition from classical DG lore to the DGP. In the year following his 1978 paper, Yemini wrote another paper on the computational complexity of some problems in graph rigidity [243], which introduced the *position-location problem* as the problem of determining the coordinates of a set of objects in space from a sparse set of distances. This was in contrast to typical structural rigidity results of the time, whose main focus was the determination of the rigidity of given frameworks (see [233] and references therein). Meanwhile, Saxe published a paper in the same year [198] in which the DGP was introduced as the K*-embeddability problem* and shown to be strongly **NP**-complete when $K = 1$ and strongly **NP**-hard for general $K > 1$.

The interest in the DGP resides in its wealth of applications (molecular conformation, wireless sensor networks, statics, dimensionality reduction, and robotics, among others), as well as in the beauty of the related mathematical theory. Our exposition takes the viewpoint of a specific application that we have studied for a number of years, namely, the determination of protein structure using nuclear magnetic resonance (NMR) data [52]. A discussion of the relationship between DG and real-world problems in computational chemistry is presented in [51].

NMR data is usually presented in the current DG literature as a graph whose edges are weighted with intervals, which represent distance measurements with errors. This, however, is the result of data manipulation carried out by NMR specialists. The actual situation is more complex: the NMR machinery outputs frequency readings for distance values related to pairs of atom types. Formally, one could imagine the NMR machinery to be a black box whose input is a set of distinct atom-type pairs $\{a, b\}$ (e.g., $\{\mathrm{H}, \mathrm{H}\}$, $\{\mathrm{C}, \mathrm{H}\}$, and so on) and whose output is a set of triplets $(\{a, b\}, d, q)$. Their meaning is that q pairs of atoms of type a, b were observed to have (interval) distance d within the molecule being analyzed. The chemical knowledge of a protein also includes other information such as covalent bond and angles, certain torsion angles, and so on (see [199] for definitions of these chemical terms). Armed with this knowledge, NMR specialists are able to output an interval weighted graph that represents the molecule with a subset of its uncertain distances (this process, however, often yields errors, so that a certain percentage of interval distances might be outright wrong [17]). The problem of finding a protein structure given all practically available information about the protein is not formally defined, but we name it the protein structure from raw data (PSRD) for future reference. Several DGP variants discussed in this survey are abstract models for the PSRD.

The rest of this survey paper is organized as follows. Section 1.1 introduces the mathematical notation and basic definitions. Sections 1.2–1.3 present a taxonomy of problems in DG, which we hope will help the reader not get lost in the scores of acronyms we use. Section 2 presents the main fundamental mathematical results in DG. Section 3 discusses applications to molecular conformation, with a special focus on proteins. Section 4 surveys engineering applications of DG, mainly wireless sensor networks and statics, with some notes on dimensionality reduction and robotics.

1.1. Notation and Definitions. In this section, we give a list of the basic mathematical definitions employed in this paper. We focus on graphs, orders, matrices, realizations, and rigidity. This section may be skipped on a first reading and referred to later on if needed.

1.1.1. Graphs. The main objects studied in this survey are weighted graphs. Most of the definitions below can be found in any standard textbook on graph theory [61]. Our definition of paths rests on graph theoretical notions only (most definitions also involve an order on the vertices).

1. A *simple undirected graph* G is a pair (V, E), where V is the set of *vertices* and E is a set of unordered pairs $\{u, v\}$ of vertices, called *edges*. For $U \subseteq V$, we let $E[U] = \{\{u, v\} \in E \mid u, v \in U\}$ be the set of edges *induced* by U.
2. $H = (U, F)$ is a *subgraph* of G if $U \subseteq V$ and $F \subseteq E[U]$. The subgraph H of G is *induced* by U (denoted $H = G[U]$) if $F = E[U]$.
3. A graph $G = (V, E)$ is *complete* (or a *clique* on V) if $E = \{\{u, v\} \mid u, v \in V \wedge u \neq v\}$.
4. Given a graph $G = (V, E)$ and a vertex $v \in V$, we let $N_G(v) = \{u \in V \mid \{u, v\} \in E\}$ be the *neighborhood* of v and $\delta_G(v) = \{\{u, w\} \in E \mid u = v\}$

be the *star* of v in G. If no ambiguity arises, we simply write $N(v)$ and $\delta(v)$.

5. We extend N_G and δ_G to subsets of vertices: given a graph $G = (V, E)$ and $U \subseteq V$, we let $N_G(U) = \bigcup_{v \in U} N_G(v)$ be the neighborhood of U and $\delta_G(U) = \bigcup_{v \in U} \delta_G(v)$ be the *cutset* induced by U in G. A cutset $\delta(U)$ is *proper* if $U \neq \varnothing$ and $U \neq V$. If no ambiguity arises, we write $N(U)$ and $\delta(U)$.
6. A graph $G = (V, E)$ is *connected* if no proper cutset is empty.
7. Given a graph $G = (V, E)$ and $s, t \in V$, a *simple path* H with *endpoints* s, t is a connected subgraph $H = (V', E')$ of G such that $s, t \in V'$, $|N_H(s)| = |N_H(t)| = 1$, and $|N_H(v)| = 2$ for all $v \in V' \smallsetminus \{s, t\}$.
8. A graph $G = (V, E)$ is a *simple cycle* if it is connected and for all $v \in V$ we have $|N(v)| = 2$.
9. Given a simple cycle $C = (V', E')$ in a graph $G = (V, E)$, a *chord* of C in G is a pair $\{u, v\}$ such that $u, v \in V'$ and $\{u, v\} \in E \smallsetminus E'$.
10. A graph $G = (V, E)$ is *chordal* if every simple cycle $C = (V', E')$ with $|E'| > 3$ has a chord.
11. Given a graph $G = (V, E)$, $\{u, v\} \in E$, and $z \notin V$, the graph $G' = (V', E')$ such that $V' = (V \cup \{z\}) \smallsetminus \{u, v\}$ and $E' = (E \cup \{\{w, z\} \mid w \in N_G(u) \cup N_G(v)\}) \smallsetminus \{\{u, v\}\}$ is the *edge contraction* of G w.r.t. $\{u, v\}$.
12. Given a graph $G = (V, E)$, a *minor* of G is any graph obtained from G by repeated edge contraction, edge deletion, and vertex deletion operations.
13. Unless otherwise specified, we let $n = |V|$ and $m = |E|$.

1.1.2. Orders. At first sight, realizing weighted graphs in Euclidean spaces involves a continuous search. If the graph has certain properties, such as rigidity, then the number of embeddings is finite (see section 3.3) and the search becomes combinatorial. This offers numerical advantages in terms of efficiency of reliability. Since rigidity is hard to determine a priori, one often requires stricter conditions that are easier to verify. Most such conditions are concerned with the existence of a vertex order with special topological properties. If such orders can be defined in the input graph, the corresponding realization algorithms usually embed each vertex in turn, following the order. These orders are sometimes inherent to the application (e.g., in molecular conformation we might choose to look at the backbone order), but are more often determined either theoretically for an infinite class of problem instances (see section 3.5) or else algorithmically for a given instance (see section 3.3.3).

The names of the orders listed below refer to acronyms that indicate the problems they originate from; the acronyms themselves will be explained in section 1.2. Orders are defined with respect to a graph and sometimes an integer (which will turn out to be the dimension of the embedding space).

1. For any positive integer $p \in \mathbb{N}$, we let $[p] = \{1, \ldots, p\}$.
2. For a set V, a total order $<$ on V, and $v \in V$, we let $\gamma(v) = \{u \in V \mid u < v\}$ be the set of *predecessors* of v w.r.t. $<$ and let $\rho(v) = |\gamma(v)| + 1$ be the *rank* of v in $<$. We also define $\eta(v) = \{u \in V \mid v < u\}$ to be the set of *successors* of v w.r.t. $<$.
3. The notation $N(v) \cap \gamma(v)$ indicates the set of adjacent predecessors of a vertex v; $N(v) \cap \eta(v)$ indicates the set of adjacent successors of v.
4. It is easy to show that if $G = (V, E)$ is a simple path, then there is an order $<$ on V such that for all $\{u, v\} \in E$ we have $\rho(u) = \rho(v) - 1$ and the vertices of minimum and maximum rank in $<$ are the endpoints of the path.

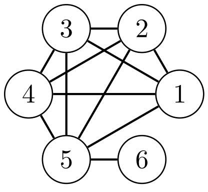


Fig. 1.1 *A graph with a PEO order on V: $N(1) \cap \eta(1) = \{2,3,4,5\}$, $N(2) \cap \eta(2) = \{3,4,5\}$, $N(3) \cap \eta(3) = \{4,5\}$, $N(4) \cap \eta(4) = \{5\}$, $N(5) \cap \eta(5) = \{6\}$, $N(6) \cap \eta(6) = \varnothing$ are all cliques.*

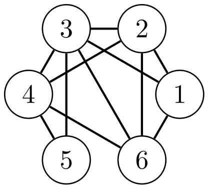


Fig. 1.2 *A graph with a DVOP order on V (for $K = 2$): $\{1,2\}$ induces a clique, $N(v) \cap \gamma(v) = \{v-1, v-2\}$ for all $v \in \{3,4,5\}$, and $N(6) \cap \gamma(6) = \{1,2,3,4\}$.*

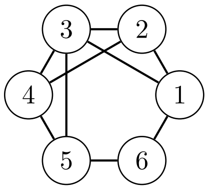


Fig. 1.3 *A graph with a Henneberg type I order on V (for $K = 2$): $\{1,2\}$ induces a clique, $N(v) \cap \gamma(v) = \{v-1, v-2\}$ for all $v \in \{3,4,5\}$, and $N(6) \cap \gamma(6) = \{1,5\}$.*

5. A *perfect elimination order* (PEO) on $G = (V, E)$ is an order on V such that, for each $v \in V$, $G[N(v) \cap \eta(v)]$ is a clique in G (see Figure 1.1).
6. A *DVOP order* on $G = (V, E)$ w.r.t. an integer $K \in [n]$ is an order on V where (a) the first K vertices induce a clique in G and (b) each $v \in V$ of rank $\rho(v) > K$ has $|N(v) \cap \gamma(v)| \geq K$ (see Figure 1.2).
7. A *Henneberg type I order* is a DVOP order where each v with $\rho(v) > K$ has $|N(v) \cap \gamma(v)| = K$ (see Figure 1.3).
8. A *K-trilateration* (or *K-trilaterative*) order is a DVOP order where (a) the first $K+1$ vertices induce a clique in G and (b) each v with $\rho(v) > K+1$ has $|N(v) \cap \gamma(v)| \geq K+1$ (see Figure 1.4).
9. A *DDGP order* is a DVOP order where, for each v with $\rho(v) > K$, there exists $U_v \subseteq N(v) \cap \gamma(v)$ with $|U_v| = K$ and $G[U_v]$ a clique in G (see Figure 1.5).
10. A $^{\mathsf{K}}$*DMDGP order* is a DVOP order where, for each v with $\rho(v) > K$, there exists $U_v \subseteq N(v) \cap \gamma(v)$ with (a) $|U_v| = K$, (b) $G[U_v]$ a clique in G, and (c) for all $u \in U_v$ $(\rho(v) - K - 1 \leq \rho(u) \leq \rho(v) - 1)$ (see Figure 1.6).

From these definitions it is clear that:

- $^{\mathsf{K}}$DMDGP orders are also DDGP orders;
- DDGP, K-trilateration, and Henneberg type I orders are also DVOP orders;

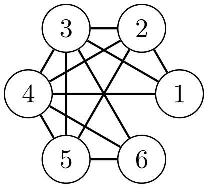


Fig. 1.4 *A graph with a* 2*-trilaterative order on* V*:* $\{1,2,3\}$ *induces a clique* $N(v) \cap \gamma(v) = \{v-1, v-2, v-3\}$ *for all* $v \in \{4,5,6\}$.

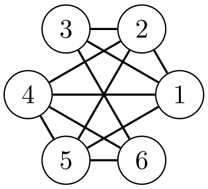


Fig. 1.5 *A graph with a DDGP order on* V *(for* $K=2$*):* $U_3 = U_4 = U_5 = \{1,2\}$, $U_6 = \{3,4\}$.

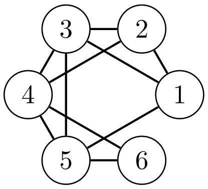


Fig. 1.6 *A graph with a* ${}^{\mathsf{K}}$*DMDGP order on* V *(for* $K=2$*):* $U_3 = \{1,2\}$, $U_4 = \{2,3\}$, $U_5 = \{3,4\}$, $U_6 = \{4,5\}$.

- ${}^{\mathsf{K}}$DMDGP orders on graphs with a minimal number of edges are inverse PEOs;
- K-trilateration orders on graphs with a minimal number of edges are inverse PEOs.

Furthermore, it is easy to show that DDGP, K-trilateration, and Henneberg type I orders have a nonempty symmetric difference and that there are PEO instances not corresponding to any inverse ${}^{\mathsf{K}}$DMDGP or K-trilateration orders.

1.1.3. Matrices. The incidence and adjacency structures of graphs can be well represented using matrices. For this reason, DGPs on graphs can also be seen as problems on matrices.

1. A *distance space* is a pair (X, d), where $X \subseteq \mathbb{R}^K$ and $d : X \times X \to \mathbb{R}_+$ is a distance function (i.e., a metric on X, which by definition must be a nonnegative, symmetric function $X \times X \to \mathbb{R}_+$ satisfying the triangular inequality $d(x,z) \le d(x,y) + d(y,z)$ for any $x, y, z \in X$ and such that $d(x,x) = 0$ for all $x \in X$).
2. A *distance matrix* for a finite distance space $(X = \{x_1, \ldots, x_n\}, d)$ is the $n \times n$ square matrix $D = (d_{uv})$ where, for all $u, v \le |X|$, we have $d_{uv} = d(x_u, x_v)$.
3. A *partial matrix* on a field $\mathbb{F}$ is a pair (A, S), where $A = (a_{ij})$ is an $m \times n$ matrix on $\mathbb{F}$ and S is a set of pairs (i, j) with $i \le m$ and $j \le n$; the *completion*

of a partial matrix is a pair (α, B), where $\alpha : S \to \mathbb{F}$ and $B = (b_{ij})$ is an $m \times n$ matrix on $\mathbb{F}$, such that for all $(i,j) \in S$ $(b_{ij} = \alpha_{ij})$ and for all $(i,j) \notin S$ $(b_{ij} = a_{ij})$.

4. An $n \times n$ matrix $D = (d_{ij})$ is a *Euclidean distance matrix* if there exists an integer $K > 0$ and a set $X = \{x_1, \dots, x_n\} \subseteq \mathbb{R}^K$ such that, for all $i, j \le n$, we have $d_{ij} = \|x_i - x_j\|$.
5. An $n \times n$ symmetric matrix $A = (a_{ij})$ is *positive semidefinite* if all its eigenvalues are nonnegative.
6. Given two $n \times n$ matrices $A = (a_{ij})$, $B = (b_{ij})$, the *Hadamard product* $C = A \circ B$ is the $n \times n$ matrix $C = (c_{ij})$, where $c_{ij} = a_{ij} b_{ij}$ for all $i, j \le n$.
7. Given two $n \times n$ matrices $A = (a_{ij})$, $B = (b_{ij})$, the *Frobenius (inner) product* $C = A \bullet B$ is defined as $\mathrm{trace}(A^\top B) = \sum_{i,j \le n} a_{ij} b_{ij}$.

1.1.4. Realizations and Rigidity. The definitions below give enough information to define the concept of rigid graph, but there are several more definitions concerning rigidity concepts. For a more extensive discussion, see section 4.2.

1. Given a graph $G = (V, E)$ and a manifold $M \subseteq \mathbb{R}^K$, a function $x : G \to M$ is an *embedding* of G in M if (i) x maps V to a set of n points in M; (ii) x maps E to a set of m simple arcs (i.e., homeomorphic images of $[0,1]$) in M; (iii) for each $\{u, v\} \in E$, the endpoints of the simple arc x_{uv} are x_u and x_v. We remark that the restriction of x to V can also be seen as a vector in $\mathbb{R}^{nK}$ or as an $K \times n$ real matrix.
2. An embedding such that $M = \mathbb{R}^K$ and the simple arcs are line segments is called a *realization* of the graph in $\mathbb{R}^K$. A realization is *valid* if it satisfies (1.1). In practice we neglect the action of x on E (because it is naturally induced by the action of x on V, since the arcs are line segments in $\mathbb{R}^K$) and only denote realizations as functions $x : V \to \mathbb{R}^K$.
3. Two realizations x, y of a graph $G = (V, E)$ are *congruent* if, for every $u, v \in V$, we have $\|x_u - x_v\| = \|y_u - y_v\|$. If x, y are not congruent, then they are *incongruent*. If R is a rotation, translation, or reflection and $Rx = (Rx_1, \dots, Rx_n)$, then Rx is congruent to x [28].
4. A *framework* in $\mathbb{R}^K$ is a pair (G, x) where x is a realization of G in $\mathbb{R}^K$.
5. A *displacement* of a framework (G, x) is a continuous function $y : [0,1] \to \mathbb{R}^{nK}$ such that (i) $y(0) = x$; (ii) $y(t)$ is a valid realization of G for all $t \in [0,1]$.
6. A *flexing* of a framework (G, x) is a displacement y of x such that $y(t)$ is incongruent to x for any $t \in (0,1]$.
7. A framework is *flexible* if it has a flexing; otherwise it is *rigid*.
8. Let (G, x) be a framework. Consider the linear system $R\alpha = 0$, where R is the $m \times nK$ matrix each $\{u, v\}$th row of which has exactly $2K$ nonzero entries, $x_{ui} - x_{vi}$ and $x_{vi} - x_{ui}$ (for $\{u, v\} \in E$ and $i \le K$), and $\alpha \in \mathbb{R}^{nK}$ is a vector of indeterminates. The framework is *infinitesimally rigid* if the only solutions of $R\alpha = 0$ are translations or rotations [218], and *infinitesimally flexible* otherwise. By [81, Thm. 4.1], infinitesimal rigidity implies rigidity.
9. By [94, Thm. 2.1], if a graph has a unique infinitesimally rigid framework, then almost all its frameworks are rigid. Thus, it makes sense to define a *rigid graph* as a graph having an infinitesimally rigid framework. The notion of a graph being rigid independent of the framework assigned to it is also known as *generic rigidity* [45].

A few remarks on the concepts of embedding and congruence, which are of paramount importance throughout this survey, are in order. The definition of an embedding (item 1 above) is similar to that of a *topological embedding*. The latter, however, also satisfies other properties: no graph vertex is embedded in the interior of any simple arc (for all $v \in V, \{u,w\} \in E$ $(x_v \notin x^\circ_{uw})$, where S° is the interior of the set S), and no two simple arcs intersect (for all $\{u,v\} \neq \{v,z\} \in E$ $(x^\circ_{uv} \cap x^\circ_{vz} = \varnothing)$). The graph embedding problem on a given manifold, in the topological sense, is the problem of finding a topological embedding for a graph in the manifold: the constraints are given not by the distances, but rather by the requirement that no two edges must be mapped to intersecting simple arcs. Garey and Johnson list a variant of this problem as the open problem Graph Genus [79, OPEN3]. The problem was subsequently shown to be NP-complete by Thomassen in 1989 [221].

The definition of congruence concerns pairs of points: two distinct pairs of points $\{x_1, x_2\}$ and $\{y_1, y_2\}$ are congruent if the distance between x_1 and x_2 is equal to the distance between y_1 and y_2. This definition is extended to sets of points X, Y in a natural way: X and Y are congruent if there is a surjective function $f : X \to Y$ such that each pair $\{x_1, x_2\} \subseteq X$ is congruent to $\{f(x_1), f(x_2)\}$. Set congruence implies that f is actually a bijection; moreover, it is an equivalence relation [28, Chap. II, sect. 12].

1.2. A Taxonomy of Problems in Distance Geometry.

Given the broad scope of the presented material (and the considerable number of acronyms attached to problem variants), we believe that the reader will appreciate this introductory taxonomy, which defines the problems we shall discuss in the rest of this paper. Figure 1.7 and Table 1.1 provide a graphical description of the existing logical/topical relations between problems. Some of our terminology has changed from past papers, as we are now attempting to standardize the problem names in a consistent manner.

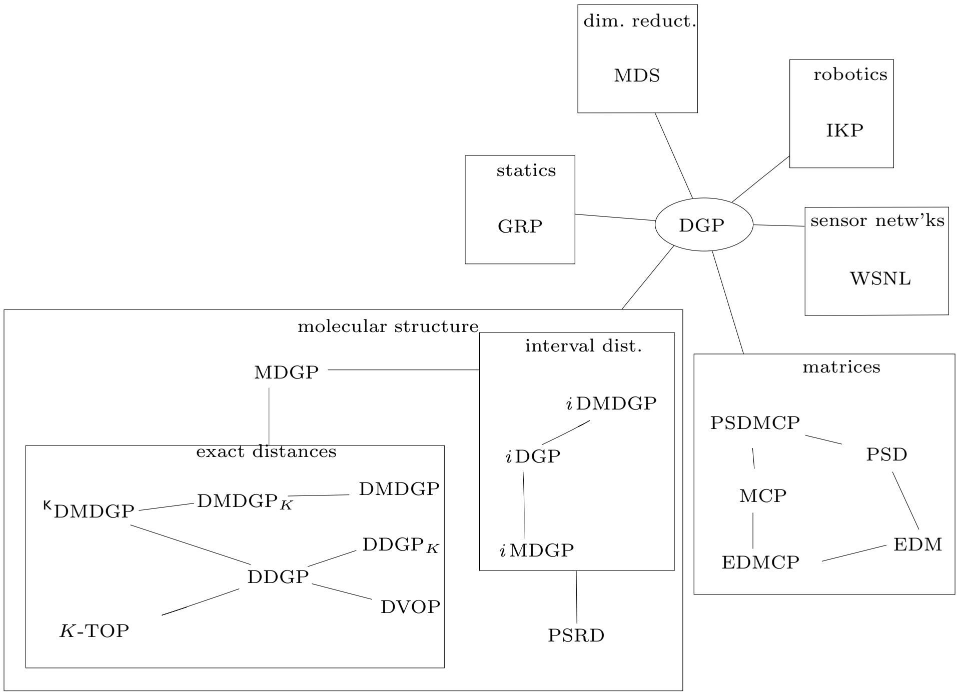


Fig. 1.7 *Classification of DGPs.*

Table 1.1 *Distance geometry problems and their acronyms.*

Acronym	Full Name
Distance geometry	
DGP	Distance Geometry Problem [28]
MDGP	Molecular DGP (in 3 dimensions) [52]
DDGP	Discretizable DGP [121]
DDGP_K	DDGP in fixed dimension [168]
$^{\mathsf{K}}$DMDGP	Discretizable MDGP (a.k.a. GDMDGP [151])
DMDGP_K	DMDGP in fixed dimension [147]
DMDGP	DMDGP_K with $K = 3$ [127]
iDGP	interval DGP [52]
iMDGP	interval MDGP [165]
iDMDGP	interval DMDGP [129]
Vertex orders	
DVOP	Discretizable Vertex Order Problem [121]
K-TOP	K-Trilateration order problem [73]
Applications	
PSRD	Protein Structure from Raw Data
MDS	Multidimensional Scaling [58]
WSNL	Wireless Sensor Network Localization [242]
IKP	Inverse Kinematic Problem [222]
Mathematics	
GRP	Graph Rigidity Problem [243]
MCP	Matrix Completion Problem [119]
EDM	Euclidean Distance Matrix Problem [28]
EDMCP	Euclidean Distance MCP [117]
PSD	Positive Semidefinite determination [118]
PSDMCP	Positive Semidefinite MCP [117]

We sometimes emphasize problem variants where the dimension K is "fixed." This is common in theoretical computer science: it simply means that K is a given constant that is not part of the problem input. This is important because the worst-case complexity expression for the corresponding solution algorithms decreases. For example, in section 3.3.3 we give an $O(n^{K+3})$ algorithm for a problem parametrized on K. This has exponential time whenever K is part of the input, but it becomes polynomial when K is a fixed constant.

1. *Distance Geometry Problem* (DGP) [28, Chap. IV, sects. 36–42], [128]: given an integer $K > 0$ and a nonnegatively weighted simple undirected graph, find a realization in $\mathbb{R}^K$ such that Euclidean distances between pairs of points are equal to the edge weights (formal definition in section 1). We denote by DGP_K the subclass of DGP instances for a fixed K.
2. *Protein Structure from Raw Data* (PSRD): we do not mean this as a formal decision problem, but rather as a practical problem; i.e., given all possible raw data concerning a protein, find the protein structure in space. Notice that the "raw data" might contain raw output from the NMR machinery, covalent bonds and angles, a subset of torsion angles, information about the secondary structure of the protein, information about the potential energy function, and so on (discussed above) [199].
3. *Molecular Distance Geometry Problem* (MDGP) [52, sect. 1.3], [148]: same as DGP_3 (discussed in section 3.2).
4. *Discretizable Distance Geometry Problem* (DDGP) [121]: subset of DGP instances for which a vertex order is given such that (a) a realization for the

first K vertices is also given; (b) each vertex v of rank $>K$ has $\geq K$ adjacent predecessors (discussed in section 3.3.4).

5. *Discretizable Distance Geometry Problem* with a fixed number of dimensions (DDGP$_K$) [168]: subset of DDGP for which the dimension of the embedding space is fixed to a constant value K (discussed in section 3.3.4). The case $K = 3$ was specifically discussed in [168].
6. *Discretizable Vertex Order Problem* (DVOP) [121]: given an integer $K > 0$ and a simple undirected graph, find a vertex order such that the first K vertices induce a clique and each vertex of rank $> K$ has $\geq K$ adjacent predecessors (discussed in section 3.3.3).
7. *K-Trilateration* order problem (K-TOP) [73]: like the DVOP, with "K" replaced by "$K + 1$" (discussed in section 3.3).
8. *Discretizable Molecular Distance Geometry Problem* ($^{\mathsf{K}}$DMDGP) [151]: subset of DDGP instances for which the K immediate predecessors of v are adjacent to v (discussed in section 3.3).
9. *Discretizable Molecular Distance Geometry Problem* in fixed dimension (DMDGP$_K$) [150]: subset of $^{\mathsf{K}}$DMDGP for which the dimension of the embedding space is fixed to a constant value K (discussed in section 3.3).
10. *Discretizable Molecular Distance Geometry Problem* (DMDGP) [127]: the DMDGP$_K$ with $K = 3$ (discussed in section 3.3).
11. *Interval Distance Geometry Problem* (iDGP) [52, 128]: given an integer $K > 0$ and a simple undirected graph whose edges are weighted with intervals, find a realization in $\mathbb{R}^K$ such that Euclidean distances between pairs of points belong to the edge intervals (discussed in section 3.4).
12. *Interval Molecular Distance Geometry Problem* (iMDGP) [165, 128]: the iDGP with $K = 3$ (discussed in section 3.4).
13. *Interval Discretizable Molecular Distance Geometry Problem* (iDMDGP) [176]: given (i) an integer $K > 0$; (ii) a simple undirected graph whose edges can be partitioned in three sets E_N, E_S, E_I such that edges in E_N are weighted with nonnegative scalars, edges in E_S are weighted with finite sets of nonnegative scalars, and edges in E_I are weighted with intervals; (iii) a vertex order such that each vertex v of rank $>K$ has at least K immediate predecessors that are adjacent to v using only edges in $E_N \cup E_S$, find a realization in $\mathbb{R}^3$ such that Euclidean distances between pairs of points are equal to the edge weights (for edges in E_N), or belong to the edge set (for edges in E_S), or belong to the edge interval (for edges in E_I) (discussed in section 3.4).
14. *Wireless Sensor Network Localization* problem (WSNL) [242, 197, 73]: like the DGP, but with a subset A of vertices (called *anchors*) whose position in $\mathbb{R}^K$ is known a priori (discussed in section 4.1). The variants of practical interest have K fixed to 2 or 3.
15. *Inverse Kinematic Problem* (IKP) [222]: subset of WSNL instances such that the graph is a simple path whose endpoints are anchors (discussed in section 4.3.2).
16. *Multidimensional Scaling* problem (MDS) [58]: given a set X of vectors, find a set Y of smaller dimensional vectors (with $|X| = |Y|$) such that the distance between the ith and jth vectors of Y approximates the distance of the corresponding pair of vectors of X (discussed in section 4.3.1).
17. *Graph Rigidity Problem* (GRP) [243, 117]: given a simple undirected graph, find an integer $K' > 0$ such that the graph is (generically) rigid in $\mathbb{R}^K$ for all $K \geq K'$ (discussed in section 4.2).

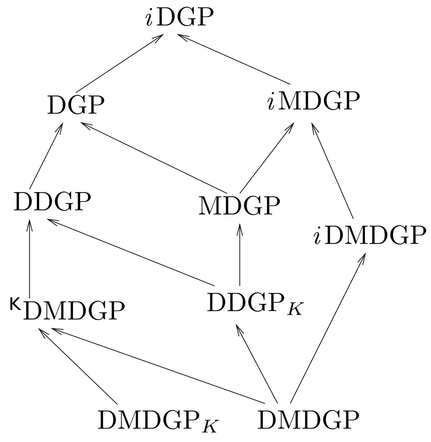


Fig. 1.8 *Inclusionwise lattice of DGP variants (arrows mean $\subset$).*

18. *Matrix Completion Problem* (MCP) [119]: given a square "partial matrix" (i.e., a matrix with some missing entries) and a matrix property P, determine whether there exists a completion of the partial matrix that satisfies P (discussed in section 2).
19. *Euclidean Distance Matrix* (EDM) problem [28]: determine whether a given matrix is an EDM (discussed in section 2).
20. *Euclidean Distance Matrix Completion Problem* (EDMCP) [117, 118, 100]: subset of MCP instances with P corresponding to "Euclidean distance matrix for a set of points in $\mathbb{R}^K$ for some K" (discussed in section 2).
21. *Positive Semidefinite* (PSD) determination [118]: determine whether a given matrix is positive semidefinite (discussed in section 2).
22. *Positive Semidefinite Matrix Completion Problem* (PSDMCP) [117, 118, 100]: subset of MCP instances with P corresponding to "positive semidefinite matrix" (discussed in section 2).

1.3. DGP Variants by Inclusion.

The research carried out by the authors of this survey focuses mostly on the subset of problems in the *DG* category mentioned in Figure 1.7. These problems, seen as sets of instances, are related by the inclusionwise lattice shown in Figure 1.8.

2. The Mathematics of Distance Geometry.

This section will briefly discuss some fundamental mathematical notions related to DG. As is well known, DG has strong connections to matrix analysis, semidefinite programming (SDP), convex geometry, and graph rigidity [56]. On the other hand, Gödel's extensions to differentiable manifolds (section 2.2) and the exterior algebra formalization (section 2.3) are perhaps less well known.

Given a set $\mathcal{U} = \{p_0, \ldots, p_K\}$ of $K+1$ points in $\subseteq \mathbb{R}^K$, the volume of the K-simplex defined by the points in $\mathcal{U}$ is given by the so-called Cayley–Menger formula [161, 162, 28]

$$\Delta_K(\mathcal{U}) = \sqrt{\frac{(-1)^{K+1}}{2^K (K!)^2} \mathsf{CM}(\mathcal{U})}, \tag{2.1}$$

where $\mathsf{CM}(\mathcal{U})$ is the Cayley–Menger determinant [161, 162, 28]

$$\mathsf{CM}(\mathcal{U}) = \begin{vmatrix} 0 & 1 & 1 & \dots & 1 \\ 1 & 0 & d_{01}^2 & \dots & d_{0K}^2 \\ 1 & d_{01}^2 & 0 & \dots & d_{1K}^2 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & d_{0K}^2 & d_{1K}^2 & \dots & 0 \end{vmatrix}, \tag{2.2}$$

with $d_{uv} = \|p_u - p_v\|$ for all $u, v \in \{0, \dots, K\}$. The Cayley–Menger determinant is proportional to the quantity known as the *oriented volume* [52] (sometimes also called the *signed volume*), which plays an important role in the theory of oriented matroids [27]. Opposite signed volume values correspond to the two possible orientations of a simplex keeping one of its facets fixed (see, e.g., the two positions for vertex 4 in Figure 3.6, center). In [241], a generalization of DG is proposed to solve spatial constraints, using an extension of the Cayley–Menger determinant.

2.1. The Euclidean Distance Matrix Problem.

Cayley–Menger determinants were used in [28] to give necessary and sufficient conditions for the EDM problem, i.e., determining whether for a given $n \times n$ matrix $D = (d_{ij})$ there exist an integer K and a set $\{p_1, \dots, p_n\}$ of points of $\mathbb{R}^K$ such that $d_{ij} = \|p_i - p_j\|$ for all $i, j \le n$. Necessary and sufficient conditions for a matrix to be an EDM are given in [209]. For a square $h \times h$ matrix R and any $i \le h$, let $R^{(i)}$ the submatrix of R consisting of the first i rows and columns.

Theorem 2.1 (Theorem 4 in [209]). *An $n \times n$ distance matrix D is embeddable in $\mathbb{R}^K$ but not in $\mathbb{R}^{K-1}$ if and only if* (a) *there is a principal $(K+1) \times (K+1)$ submatrix R of D (whose rows and columns are not necessarily of the same order as in D) such that, for all $i \in \{2, \dots, K+1\}$, the sign of the Cayley–Menger determinant of $R^{(i)}$ is $(-1)^i$;* (b) *for $\mu \in \{2, 3\}$, every principal $(K+\mu) \times (K+\mu)$ submatrix of D containing R has zero Cayley–Menger determinant.*

In other words, the two conditions of this theorem state that there must be a K-simplex S of reference with nonzero volume in $\mathbb{R}^K$, and all $(K+1)$- and $(K+2)$-simplices containing S as a face must be contained in $\mathbb{R}^K$.

2.2. Differentiable Manifolds.

Condition (ii) in Theorem 2.1 fails to hold in the cases of (curved) manifolds. Gödel showed that for $K = 3$, the condition can be updated as follows (see paper 1933h in [75]): for any quadruplet $\mathcal{U}_n$ of point sequences p_u^n (for $u \in \{0, \dots, 3\}$) converging to a single nondegenerate point p_0, the following holds:

$$\lim_{n \to \infty} \frac{\mathsf{CM}(\mathcal{U}_n)}{\sum\limits_{u<v} \|p_u^n - p_v^n\|^6} = 0.$$

In a related note, Gödel also showed that if $\mathcal{U} = \{p_0, \dots, p_3\}$ with $\mathsf{CM}(\mathcal{U}) \neq 0$, then the distance matrix over $\mathcal{U}$ can be realized on the surface of a 2-sphere where the distances between the points are the lengths of the arcs on the spherical surface (see paper 1933b in [75]). This observation establishes a relationship between DG and the kissing number problem [114] and, more generally, to coding theory [46]. The specializations of the "subset problem" (see the introduction) and the DGP to kissing arrangements of spheres in space is studied from a theoretical point of view in [39].

2.3. Exterior Algebras.

Cayley–Menger determinants are exterior products [10]. The set of all possible exterior products of a vector space forms an *exterior algebra*, which is a special type of Clifford algebra [40]; specifically, exterior algebras are tensor algebras modulo the ideal generated by x^2. The fact that any square element of the algebra is zero implies $0 = (x+y)^2 = x^2+xy+yx+y^2 = xy+yx$, and hence $xy = -yx$. Accordingly, exterior algebras are used in the study of alternating multilinear forms. The paper [67] gives an in-depth view of the connection between DG and Clifford algebras.

In the setting of DG, we define the product of vectors $x_1, \ldots, x_n \in \mathbb{R}^K$ (for $n \geq K$) using the corresponding Cayley–Menger determinant on $\mathcal{U} = \{x_0, \ldots, x_n\}$, where x_0 is the origin. It is clear that if $x_i = x_j$ for some $i \neq j$, then the corresponding n-simplex is degenerate and certainly has volume 0 in $\mathbb{R}^K$ (even if $n = K$), hence $\mathsf{CM}(\mathcal{U}) = 0$. Equivalently, if a product $\prod_i x_i$ can be written as $x_j^2 \prod_{i \neq j} x_i$, then it belongs to the ideal $\langle x^2 \rangle$ and is replaced by 0 in the exterior algebra.

Abstract relationships between an exterior algebra and its corresponding vector space are specialized to relationships between Cayley–Menger determinants and vectors in $\mathbb{R}^K$. Thus, for example, one can derive the following well-known result in linear algebra: $x_1, \ldots, x_K$ are linearly independent if and only if $\mathsf{CM}(\mathcal{U}) \neq 0$, where $\mathcal{U} = \{x_0, \ldots, x_K\}$ with x_0 being the origin [10, 40]. A more interesting example consists in deriving certain invariants expressed in *Plücker coordinates* [40]: given a basis $x_1, \ldots, x_K$ of $\mathbb{R}^K$ and a basis $y_1, \ldots, y_h$ of $\mathbb{R}^h$, where $h \leq K$, it can be shown that for any subset S of $\{1, \ldots, K\}$ of size h there exist constants α_S such that $\sum_S \alpha_S \prod_{i \in S} x_i = \prod_{i \leq h} y_i$. In our setting, product vectors correspond to Cayley–Menger determinants derived from the given points $x_1, \ldots, x_K$ and an origin x_0. It turns out that the ratios of various α_S's are invariant over different bases $y'_1, \ldots, y'_h$ of $\mathbb{R}^h$, which allows their employment as a convenient coordinate system for $\mathbb{R}^h$. Invariants related to the Plücker coordinates are exploited in [52] to find realizations of chirotopes (orientations of vector configurations [27]).

2.4. Bideterminants.

For sets of more than $K + 1$ points, the determination of the relative orientation of each K-simplex in terms of a K-simplex of reference (see, e.g., Figure 3.10, center and right) is important. Such relative orientations are given by the *Cayley–Menger bideterminant* of two K-simplices $\mathcal{U} = \{p_0, \ldots, p_K\}$ and $\mathcal{V} = \{q_0, \ldots, q_K\}$, with $d_{ij} = \|p_i - q_j\|$:

$$\mathsf{CM}(\mathcal{U}, \mathcal{V}) = \begin{vmatrix} 0 & 1 & \ldots & 1 \\ 1 & d_{00}^2 & \ldots & d_{0K}^2 \\ 1 & d_{10}^2 & \ldots & d_{1K}^2 \\ \vdots & \vdots & \ddots & \vdots \\ 1 & d_{K0}^2 & \ldots & d_{KK}^2 \end{vmatrix}. \tag{2.3}$$

These bideterminants allow, for example, the determination of stereoisometries in chemistry [27].

2.5. Positive Semidefinite and Euclidean Distance Matrices.

In [200] Schoenberg proved that there is a one-to-one relationship between EDMs and PSD matrices. Let $D = (d_{ij})$ be an $(n+1) \times (n+1)$ matrix and $A = (a_{ij})$ the $(n+1) \times (n+1)$ matrix given by $a_{ij} = \frac{1}{2}(d_{0i}^2 + d_{0j}^2 - d_{ij}^2)$.

The bijection given by Theorem 2.2 below can be exploited to show that determining whether a matrix is PSD or EDM is essentially the same thing [208].

Theorem 2.2 (Theorem 1 in [208]). *A necessary and sufficient condition for the matrix D to be an EDM with respect to a set $\mathcal{U} = \{p_0, \ldots, p_n\}$ of points in $\mathbb{R}^K$*

but not in $\mathbb{R}^{K-1}$ *is that the quadratic form* $x^\top Ax$ *(where* A *is given above) is PSD of rank* K*.*

Schoenberg's theorem was cast in a very compact and elegant form in [57]:

$$\mathbb{EDM} = \mathbb{S}_h \cap (\mathbb{S}_c^\perp - \mathbb{S}_+), \tag{2.4}$$

where $\mathbb{EDM}$ is the set of $n \times n$ EDMs, $\mathbb{S}$ is the set of $n \times n$ symmetric matrices, $\mathbb{S}_h$ is the projection of $\mathbb{S}$ on the subspace of matrices having zero diagonal, $\mathbb{S}_c$ is the kernel of the matrix map $Y \to Y\mathbf{1}$ (with $\mathbf{1}$ the all-one n-vector), $\mathbb{S}_c^\perp$ is the orthogonal complement of $\mathbb{S}_c$, and $\mathbb{S}_+$ is the set of symmetric PSD $n \times n$ matrices. The matrix representation in (2.4) was exploited in the alternating projection algorithm (APA) discussed in section 3.4.4.

2.6. Matrix Completion Problem.

Given an appropriate property P applicable to square matrices, the MCP schema asks whether a given $n \times n$ partial matrix A' can be completed to a matrix A such that $P(A)$ holds. MCPs are naturally formulated in terms of graphs: given a weighted graph $G = (V, E, a')$, with $a' : E \to \mathbb{R}$, is there a complete graph K on V (possibly with loops) with an edge weight function a such that $a_{uv} = a'_{uv}$ for all $\{u, v\} \in E$? This problem schema is parametrized over $P(\cdot)$. In two specializations mentioned below, a is completed so that the whole matrix is an EDM or a PSD matrix.

MCPs are an interesting class of inverse problems that find applications in the analysis of data, such as, for example, the reconstruction of 3D images from several 2D projections on random planes in cryoelectron microscopy [207]. When $P(A)$ is the (informal) statement "A has low rank," there is an interesting application to recommender systems: voters submit rankings for a few items, and consistent rankings for all items are required. Since few factors are believed to impact a user's preferences, the data matrix is expected to have low rank [206].

Two celebrated specializations of this problem schema are the EDMCP and the PSDMCP. These two problems have a strong link by virtue of Theorem 2.2 and, in fact, there is a bijection between EDMCP and PSDMCP instances [117]. MCP variants where a'_{ij} is an interval and the condition (i) is replaced by $a_{ij} \in a'_{ij}$ also exist (see, e.g., [100], where a modification of the EDMCP in this sense is given).

2.6.1. Positive Semidefinite Completion.

Laurent [118] remarks that the PSDMCP can be reduced to the SDP feasibility problem: given integral $n \times n$ symmetric matrices $Q_0, \dots, Q_m$, determine whether there exist scalars $z_1, \dots, z_m$ satisfying $Q_0 + \sum_{i \le m} z_i Q_i \succeq 0$. Thus, by Theorem 2.2, the EDMCP has the same property. Since the complexity status of the SDP problem is currently unknown (and, in particular, it is not even known whether this problem is in NP), the same holds for the PSDMCP, and hence also for the EDMCP. If one allows ε-approximate solutions, however, the situation changes. The following SDP formulation correctly models the PSDMCP:

$$\begin{array}{rl} \max & \displaystyle\sum_{(i,j) \notin E} a_{ij}, \\ & A = (a_{ij}) \succeq 0, \\ \forall i \in V & \quad a_{ii} = a'_{ii}, \\ \forall \{i, j\} \in E & \quad a_{ij} = a'_{ij}. \end{array}$$

Accordingly, SDP-based formulations and techniques are common in DG (see, e.g., section 4.1.2).

Polynomial cases of the PSDMCP are discussed in [117, 118] (and citations therein). These include chordal graphs, graphs without K_4 minors, and graphs without certain induced subgraphs (e.g., wheels W_n with $n \geq 5$). Specifically, in [118] it is shown that if a graph G is such that adding m edges makes it chordal, then the PSDMCP is polynomial on G for fixed m. All these results naturally extend to the EDMCP.

Another interesting challenge, aside from actually *solving* the problem, is to determine conditions on the given partial matrix to bound the cardinality of the solution set (specifically, the cases of one or finitely many solutions are addressed). This question is addressed in [100], where explicit bounds on the number of nondiagonal entries of A' are found in order to ensure uniqueness or finiteness of the solution set.

2.6.2. Euclidean Distance Completion.

The EDMCP differs from the DGP in that the dimension K of the embedding space is not provided as part of the input. An upper bound to the minimum possible K that is better than the trivial one ($K \leq n$) was given in [12] as

$$K \leq \frac{\sqrt{8|E|+1}-1}{2}. \tag{2.5}$$

Because of Theorem 2.2, the EDMCP inherits many of the properties of the PSDMCP. We believe that Menger was the first to explicitly state a case of EDMCP in the literature: in [161, p. 121] (see also [162, p. 738]) he refers to the matrices appearing in Cayley–Menger determinants with one missing entry. These, incidentally, are also used in the dual branch-and-prune (BP) algorithm (see section 3.3.6.1).

As mentioned in section 2.6.1, the EDMCP can be solved in polynomial time on chordal graphs $G = (V, E)$ [90, 117]. This is because a graph is chordal if and only if it has a *perfect elimination order* (PEO) [62], i.e., a vertex order on V such that, for all $v \in V$, the set of adjacent successors $N(v) \cap \eta(v)$ is a clique in G. PEOs can be found in $O(|V| + |E|)$ [190] and can be used to construct a sequence of graphs $G = (V, E) = G_0, G_1, \ldots, G_s$, where G_s is a clique on V and $E(G_i) = E(G_{i-1}) \cup \{\{u, v\}\}$, where u is the maximum ranking vertex in the PEO of G_{i-1} such that there exists $v \in \eta(u)$ with $\{u, v\} \notin E(G_{i-1})$. Assigning to $\{u, v\}$ the weight $d_{uv} = \sqrt{d_{1u}^2 + d_{1v}^2}$ guarantees that the weighted (complete) adjacency matrix of G_s is a distance matrix completion of the weighted adjacency matrix of G, as required [90]. This result is introduced in [90] (for the PSDMCP rather than the EDMCP) and summarized in [117].

3. Molecular Conformation.

Following the authors' personal interests, this is the largest section in the present survey. DG is mainly (but not exclusively [29]) used in molecular conformation as a model of an inverse problem connected to the interpretation of NMR data. We survey continuous search methods, then focus on discrete search methods, discuss the extension to interval distances, and finally present recent results specific to the NMR application.

3.1. Test Instances.

The methods described in this section have been empirically tested according to different instance sets and on different computational testbeds, so a comparison is difficult. In general, researchers in this area try to provide a "realistic" setting; the most common choices are as follows.

- **Geometrical instances**: instances are generated randomly from a geometrical model that is also found in nature, such as grids [164]; see Figure 3.1.

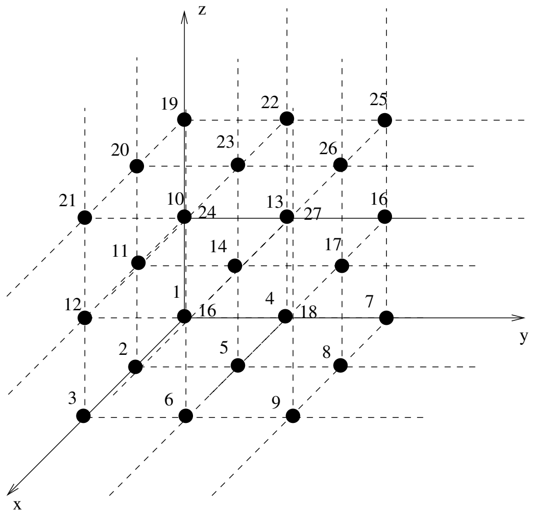


Fig. 3.1 *A Moré–Wu* $3 \times 3 \times 3$ *cubic instance, with its* 3D *realization (similar to a crystal).*

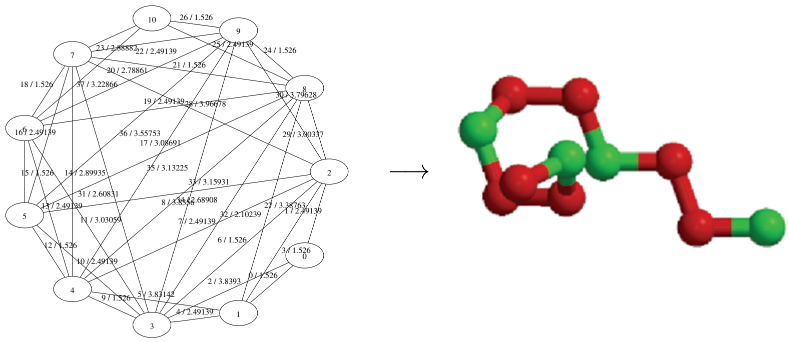


Fig. 3.2 *A Lavor instance with* 7 *vertices and* 11 *edges: graph and* 3D *realization (similar to a protein backbone).*

- **Random instances**: instances are generated randomly from a physical model that is close to reality, such as [120, 146]; see Figure 3.2.
- **Dense PDB instances**: real protein conformations (or backbones) are downloaded from the Protein Data Bank (PDB) [18] and then, for each residue, all within-residue distances as well as all distances between each residue and its two neighbors are generated [165, 98, 99]; see Figure 3.3.
- **Sparse PDB instances**: real protein conformations (or backbones) are downloaded from the PDB [18] and then all distances within a given threshold are generated [137, 127]; see Figure 3.4.

When the target application is the analysis of NMR data, as in the present case, the best test setting is provided by sparse PDB instances, as NMR can only measure distances up to a given threshold. Random instances are only useful when the underlying physical model is meaningful (as is the case in [120]). Geometrical instances could be useful in specific cases, e.g., the analysis of crystals. The problem with dense PDB instances is that, using the notions given in section 3.3 and the fact that a

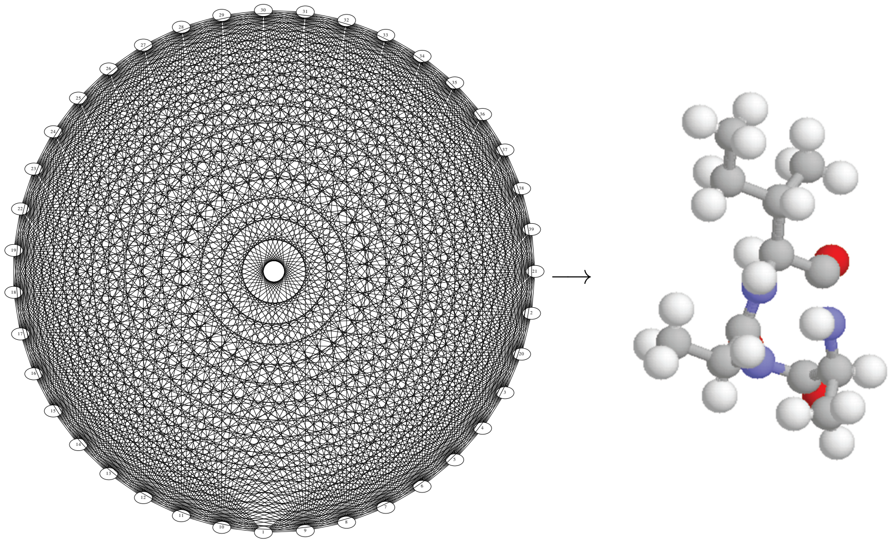


Fig. 3.3 *A fragment of* `2erl` *with all within-residue and contiguous residue distances, and one of two possible solutions.*

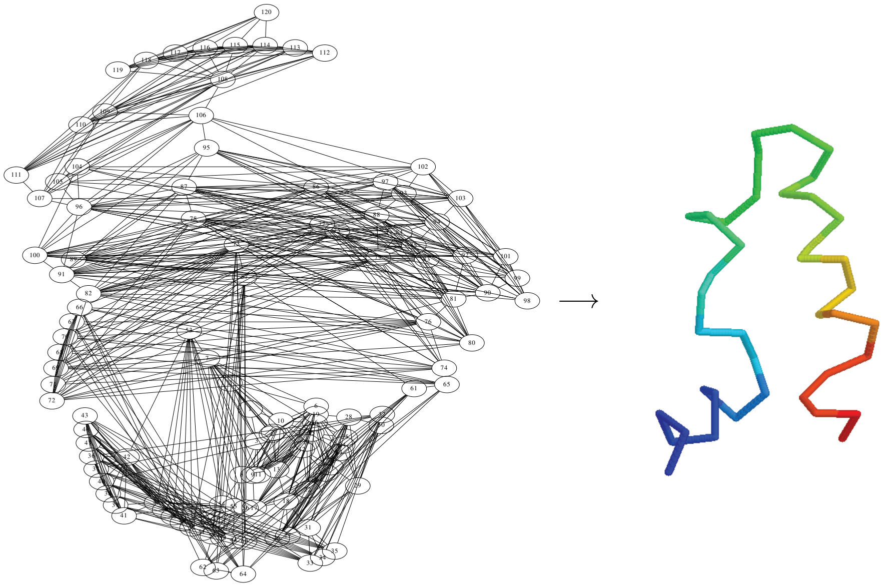


Fig. 3.4 *The backbone of the* `2erl` *instance from the PDB: graph and* 3D *realization.*

residue contains more than three atoms, it is easy to show that the backbone order on these protein instances induces a 3-trilateration order in $\mathbb{R}^3$ (see section 4.1.1). Since graphs with such orders can be realized in polynomial time [73], they do not provide a particularly hard class of test instances. Moreover, since there are actually nine backbone atoms in each set of three consecutive residues, the backbone order is actually a 7-trilateration order. In other words, there is a surplus of distances, and the problem is overdetermined.

Aside from a few early papers (e.g., [123, 145, 146]), we (the authors of this survey) have always used test sets consisting mostly of sparse PDB instances. We have also occasionally used geometric and (hard) random instances, but have never employed "easy" dense PDB instances.

3.1.1. Test Result Evaluation. The test results always yield a realization x for the given instance; accuracy measures for x, which quantify either how far x is from being valid, or how far it is from a known optimal solution; and a CPU time taken by the method to output x. Optionally, certain methods (such as the BP algorithm; see section 3.3.5) might also yield a whole set of valid realizations. Different methods are usually compared according to their accuracy and speed.

There are three popular accuracy measures. The *penalty* is the evaluation of the function defined in (3.3) for a given realization x. The *largest distance error* (LDE) is a scaled, averaged, and square-rooted version of the penalty, given by $\frac{1}{|E|}\sum_{\{u,v\}\in E}\frac{|\|x_u-x_v\|-d_{uv}|}{d_{uv}}$. The *root mean square deviation* (RMSD) is a difference measure for sets of points in Euclidean space having the same center of mass. Specifically, if x, y are embeddings of $G=(V,E)$, then $\mathrm{RMSD}(x,y)=\min_T\|y-Tx\|$, where T varies over all rotations and translations in $\mathbb{R}^K$. Accordingly, if y is the known optimal configuration of a given protein, different realizations of the same protein yield different RMSD values. Evidently, RMSD is a meaningful accuracy measure only for test sets where the optimal conformations are already known (such as PDB instances).

3.2. The Molecular Distance Geometry Problem. The MDGP is the same as DGP_3. The name "molecular" indicates that the problem originates from the study of molecular structures.

The relationship between molecules and graphs is probably the deepest existing between chemistry and discrete mathematics: a wonderful account is given in [20, Chap. 4]. Molecules were initially identified by *atomic formulae* (such as $\mathrm{H_2O}$), which indicate the relative numbers of atoms in each molecule. When chemists started to realize that some compounds with the same atomic formula have different physical properties, they sought the answer in the way the same numbers of atoms were linked to each other through chemical bonds. Displaying this type of information required more than an atomic formula, and, accordingly, several ways to represent molecules using diagrams were independently invented. The one which is still essentially in use today, consisting in a set of atom symbols linked by segments, was originally described in [53]. The very origin of the word "graph" is due to the representation of molecules [215].

The function of molecules rests on their chemical composition and three-dimensional (3D) shape in space (also called *structure* or *conformation*). As mentioned in section 1, NMR experiments can be used to determine a subset of short Euclidean distances between atoms in a molecule. These, in turn, can be used to determine its structure, i.e., the relative positions of atoms in $\mathbb{R}^3$. The MDGP provides the simplest model for this inverse problem: V models the set of atoms, E the set of atom pairs for which a distance is available, and the function $d: E \to \mathbb{R}_+$ assigns distance values to each pair, so that $G=(V,E)$ is the graph of the molecule. Assuming the input data are correct, the set X of solutions of the MDGP on G will yield all the structures of the molecule that are compatible with the observed distances.

In this section we review the existing methods for solving the MDGP with exact distances on general molecule graphs.

3.2.1. General-Purpose Approaches. Finding a solution of the set of nonlinear equations (1.1) poses several numerical difficulties. Recent (unpublished) tests performed by the authors of this survey determined that tiny, randomly generated weighted graph instances with fewer than ten vertices could not be solved using Octave's nonlinear equation solver `fsolve` [70]. The spatial branch-and-bound (sBB) code COUENNE [14] could solve instances with $|V| \in \{2,3,4\}$ but no larger in reasonable CPU times: attaining feasibility of local iterates with respect to the nonlinear manifold defined by (1.1) is a serious computational challenge. This motivates the following formulation using mathematical programming (MP):

$$\min_{x \in \mathbb{R}^K} \sum_{\{u,v\} \in E} (\|x_u - x_v\|^2 - d_{uv}^2)^2. \tag{3.1}$$

The global optimization (GO) problem (3.1) aims to minimize the squared infeasibility of points in $\mathbb{R}^K$ with respect to the manifold (1.1). Both terms in the squared difference are themselves squared in order to decrease floating point errors (`NaN` occurrences due to the square root) while evaluating the objective function of (3.1) when $\|x_u - x_v\|$ is very close to 0. We remark that (3.1) is an unconstrained nonconvex nonlinear program (NLP) whose objective function is a nonnegative polynomial of fourth degree, with the property that $x \in X$ if and only if the evaluation of the objective function at x yields 0.

In [123] we tested formulation (3.1) and some variants thereof with three GO solvers: a multilevel single linkage (MLSL) multistart method [115], a variable neighborhood search (VNS) metaheuristic for nonconvex NLPs [142], and an early implementation of sBB [153, 140, 143] (the only solver in the set that guarantees global optimality of the solution to within a given $\varepsilon > 0$ tolerance). We found that it was possible to solve geometrical and random instances [120] with up to 30 atoms using the sBB solver, whereas the two stochastic heuristics could scale up to 50 atoms, with VNS yielding the best performance.

3.2.2. Smoothing-Based Methods. A *smoothing* of a multivariate multimodal function $f(x)$ is a family of functions $F_\lambda(x)$ such that $F_0(x) = f(x)$ for all $x \in \mathbb{R}^K$ and $F_\lambda(x)$ has a decreasing number of local optima as λ increases. Eventually, F_λ becomes convex, or at least invex [15], and its optimum x^λ can be found using a single run of a local NLP solver. A homotopy continuation algorithm then traces the sequence x^λ in reverse as $\lambda \to 0$ by locally optimizing $F_{\lambda - \Delta\lambda}(x)$ for a given step $\Delta\lambda$ with x^λ as a starting point, hoping to identify the global optimum x^* of the original function $f(x)$ [108]. Since the reverse tracing is based on a local optimization step, rather than a global one, global optima in the smoothing sometimes fail to be traced to global optima in the original function.

Of course, the intuitive geometrical meaning of F_λ with respect to f really depends on the kind of smoothing operator we employ. It was shown in [146, Thm. 2.1] that the smoothing $\langle f \rangle_\lambda$ of (3.4) decreases the squares of the distance values, so that eventually they become negative;[1] this implies that the problematic nonconvex terms $(\|x_u - x_v\|^2 - d_{uv}^2)^2$ become convex. The higher the value of λ, the more nonconvex terms become convex. Those terms (indexed on u, v) that remain nonconvex have a smaller value for d_{uv}^2. Thus λ can be seen as a sliding rule controlling the convexity/nonconvexity of any number of terms via the size and sign of the d_{uv}^2 values. The upshot of this is that $\langle f \rangle_\lambda$ clusters closer vertices and shortens the distance to farther vertices; in other words, this smoothing provides a "zoomed-out view" of the realization.

[1] By mentioning negative squares we do not invoke complex numbers here; we merely mean that the values assigned to the symbols denoted by d_{uv}^2 eventually become negative.

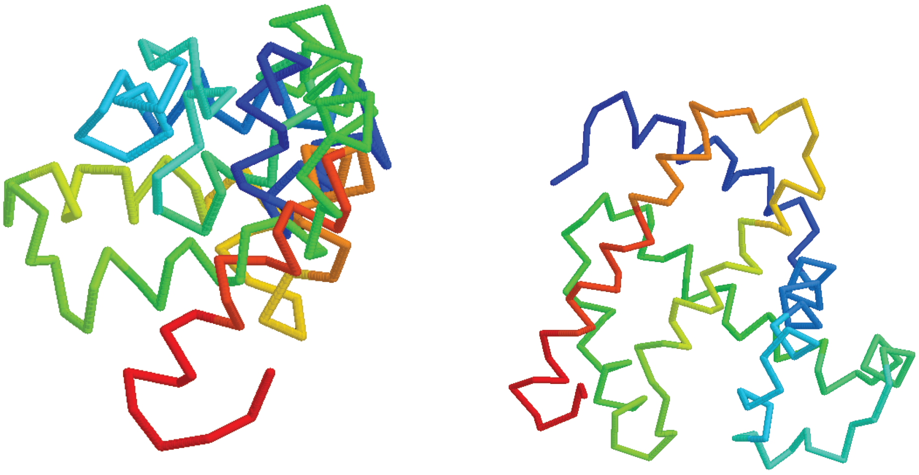


Fig. 3.5 *Comparison of a wrong molecular conformation for* `1mbn` *found by DGSOL (left) with the correct one found by the BP Algorithm* 1 *(right). Because of the local optimization step, DGSOL traced a smoothed global optimum to a strictly local optimum of the original function.*

A smoothing operator based on the many-dimensional diffusion equation $\Delta F = \frac{\partial F}{\partial \lambda}$, where Δ is the Laplacian $\sum_{i \le n} \partial^2 / \partial x_i^2$, is derived in [108] as the Fourier–Poisson formula

$$F_\lambda(x) = \frac{1}{\pi^{n/2}\lambda^n} \int_{\mathbb{R}^n} f(y) e^{-\frac{||y-x||^2}{\lambda^2}} dy, \tag{3.2}$$

also called the *Gaussian transform* in [164]. The Gaussian transform with the homotopy method provides a successful methodology for optimizing the objective function

$$f(x) = \sum_{\{u,v\} \in E} (\|x_u - x_v\|^2 - d_{uv}^2)^2, \tag{3.3}$$

where $x \in \mathbb{R}^3$. More information on continuation and smoothing-based methods applied to the iMDGP can be found in section 3.4.

In [164], it is shown that the closed form of the Gaussian transform applied to (3.3) is

$$\langle f \rangle_\lambda = f(x) + 10\lambda^2 \sum_{\{u,v\} \in E} (\|x_u - x_v\|^2 - 6 d_{uv}^2 \lambda^2) + 15\lambda^4 |E|. \tag{3.4}$$

Based on this, a continuation method is proposed and successfully tested on a set of geometrical instances (cubical grids). The implementation of this method, DGSOL, is one of the few MDGP solution codes that are freely available (source included); see http://www.mcs.anl.gov/˜more/dgsol/. DGSOL has several advantages: it is efficient, effective for small-to-medium-sized instances, and, more importantly, can be naturally extended to solve iMDGP instances (which replace the real edge weights with intervals). The one disadvantage we found with DGSOL is that it does not scale well to large-sized instances: although the method is reasonably fast even on large instances, the solution quality decreases. On large instances, DGSOL often finds infeasibilities that denote not just an offset from an optimal solution, but a completely wrong conformation (see Figure 3.5).

In [98, 99] an exact reformulation of a Gaussian transform of (3.1) as a *difference of convex* (d.c.) function is proposed and then solved using a method similar to DGSOL, but where the local NLP solution is carried out using a different algorithm, called the DCA. Although the method does not guarantee global optimality, there are empirical indications that the DCA works well in that sense. This method has been tested on three sets of data: the artificial data from Moré and Wu [164] (with up to 4096 atoms), 16 proteins in the PDB [18] (from 146 up to 4189 atoms), and the data from Hendrickson [95] (from 63 up to 777 atoms).

In [146], VNS and DGSOL were combined into a heuristic method called *double VNS with smoothing* (DVS). DVS consists in running VNS twice: first on a smoothed version $\langle f\rangle_\lambda$ of the objective function $f(x)$ of (3.1), and then on the original function $f(x)$ with tightened ranges. The rationale behind DVS is that $\langle f\rangle_\lambda$ is easier to solve, and the homotopy defined by λ should increase the probability that the global optimum x^λ of $\langle f\rangle_\lambda$ is close to the global optimum x^* of $f(x)$. The range tightening that allows VNS to be more efficient in locating x^* is based on a "Gaussian transform calculus" that gives explicit formulae that relate $\langle f\rangle_\lambda$ to $f(x)$ whenever λ and d change. These formulae are then used to identify smaller ranges for x^*. DVS is more accurate but slower than DGSOL.

It is worth remarking that both DGSOL and the DCA methods were tested using geometrical and (easy) dense PDB instances, whereas the DVS was tested using geometric and random instances (see section 3.1).

3.2.3. Geometric Build-Up Methods. In [66], a combinatorial method called the *geometric build-up* (GBU) algorithm is proposed to solve the MDGP on sufficiently dense graphs. A subgraph H of G, initially chosen to consist of only four vertices, is given together with a valid realization $\bar{x}$. The algorithm proceeds iteratively by finding x_v for each vertex $v \in V(G) \smallsetminus V(H)$. When x_v is determined, v and $\delta_H(v)$ are removed from G and added to H. For this to work, at every iteration three conditions must hold:

1. $|\delta_H(v)| \geq 4$.
2. At least one subgraph H' of H, with $V(H') = \{u_1, u_2, u_3, u_4\}$ and $|\delta_{H'}(v)| = 4$, must be such that the realization $\bar{x}$ restricted to H' is noncoplanar.
3. The input instance must be a YES one.

These conditions ensure that the position x_v can be determined using triangulation. More specifically, let $\bar{x}|_{H'} = \{x_{u_i} \mid i \leq 4\} \subseteq \mathbb{R}^3$. Then x_v is a solution of the following system:

$$\begin{aligned}||x_v - x_{u_1}|| &= d_{vu_1},\\ ||x_v - x_{u_2}|| &= d_{vu_2},\\ ||x_v - x_{u_3}|| &= d_{vu_3},\\ ||x_v - x_{u_4}|| &= d_{vu_4}.\end{aligned}$$

Squaring both sides of these equations, we have

$$\begin{aligned}||x_v||^2 - 2{x_v}^\top x_{u_1} + ||x_{u_1}||^2 &= d^2_{vu_1},\\ ||x_v||^2 - 2{x_v}^\top x_{u_2} + ||x_{u_2}||^2 &= d^2_{vu_2},\\ ||x_v||^2 - 2{x_v}^\top x_{u_3} + ||x_{u_3}||^2 &= d^2_{vu_3},\\ ||x_v||^2 - 2{x_v}^\top x_{u_4} + ||x_{u_4}||^2 &= d^2_{vu_4}.\end{aligned}$$

By subtracting one of the above equations from the others, one obtains a linear system that can be used to determine x_v. For example, subtracting the first equation from the others, we obtain

$$Ax = b, \tag{3.5}$$

where

$$A = -2 \begin{pmatrix} (x_{u_1} - x_{u_2})^\top \\ (x_{u_1} - x_{u_3})^\top \\ (x_{u_1} - x_{u_4})^\top \end{pmatrix}$$

and

$$b = \begin{pmatrix} \left(d^2_{vu_1} - d^2_{vu_2}\right) - \left(||x_{u_1}||^2 - ||x_{u_2}||^2\right) \\ \left(d^2_{vu_1} - d^2_{vu_3}\right) - \left(||x_{u_1}||^2 - ||x_{u_3}||^2\right) \\ \left(d^2_{vu_1} - d^2_{vu_4}\right) - \left(||x_{u_1}||^2 - ||x_{u_4}||^2\right) \end{pmatrix}.$$

Since $x_{u_1}, x_{u_2}, x_{u_3}, x_{u_4}$ are noncoplanar, (3.5) has a unique solution.

The GBU is very sensitive to numerical errors [66]. In [236], Wu and Wu proposed an updated GBU algorithm in which the accumulated errors can be controlled. Their algorithm was tested on a set of sparse PDB instances consisting of 10 proteins with 404 up to 4201 atoms. The results yielded RMSD measures ranging from $O(10^{-8})$ to $O(10^{-13})$. It is interesting to remark that if G is a complete graph and $d_{uv} \in \mathbb{Q}_+$ for all $\{u, v\} \in E$, this approach solves the MDGP in linear time $O(n)$ [65]. A more complete treatment of MDGP instances satisfying the K-dimensional generalization of conditions 1–2 above is given in [73, 8] in the framework of the WSNL and K-TOP problems.

An extension of the GBU that is able to deal with sparser graphs (more precisely, $\delta_H(v) \geq 3$) is given in [36]; another extension along the same lines is given in [237]. We remark that the set of graphs such that $\delta_H(v) \geq 3$ and the condition 2 above hold are precisely the instances of the DDGP such that $K = 3$ (see section 3.3.4); this problem is discussed extensively in [168]. The main conceptual difference between these GBU extensions and the BP algorithm for the DDGP [168] (see section 3.3 below) is that BP exploits a given order on V (see section 1.1.2). Since the GBU extensions do not make use of this order, they are heuristic algorithms: if $\delta_H(v) < 3$ at iteration v, then the GBU stops, but there is no guarantee that a different choice of "next vertex" might not have carried the GBU to termination. A very recent review on methods based on the GBU approach and on the formulation of other DGPs with inexact distances is given in [228]. The BP algorithm (Algorithm 1) marks a striking difference insofar as the knowledge of the order guarantees the exactness of the algorithm.

3.2.4. Graph Decomposition Methods. Graph decomposition methods are in essence mixed-combinatorial algorithms based on graph decomposition: the input graph $G = (V, E)$ is partitioned or covered by subgraphs H, each of which is realized independently (the local phase), and then the realizations of the subgraphs are "stitched together" using MP techniques (the global phase). The global phase is equivalent to applying MDGP techniques to the minor G' of G obtained by contracting each subgraph H to a single vertex. The nice feature of these methods is that the local phase is amenable to efficient yet exact solutions. For example, if H is uniquely realizable, then it is likely to be realizable in polynomial time. More precisely, a graph H is *uniquely realizable* if it has exactly one valid realization in $\mathbb{R}^K$ modulo rotations and translations; see section 4.1.1. A graph H is *uniquely localizable* if it is uniquely realizable and there is no $K' > K$ such that H also has a valid realization affinely spanning $\mathbb{R}^{K'}$. It was shown in [157] that uniquely localizable graphs are realizable in polynomial time up to a given $\varepsilon > 0$ tolerance (see section 4.1.2). On the other hand, no graph decomposition algorithm currently makes a claim to overall exactness: in order to make them practically useful, several heuristic steps must also be employed.

In ABBIE [95], both local and global phases are solved using local NLP solution techniques. Once a realization for all subgraphs H is known, the coordinates of the vertex set V_H of H can be expressed relative to the coordinates of a single vertex in V_H; this corresponds to a starting point for the realization of the minor G'. ABBIE was the first graph decomposition algorithm for the DGP, and was able to realize sparse PDB instances with up to 124 amino acids, a considerable feat in 1995.

In DISCO [139], V is covered by appropriately sized subgraphs sharing at least K vertices. The local phase is solved using an SDP formulation similar to the one given in [25]. The local phase is solved using the positions of common vertices: these are aligned, and the corresponding subgraph is then rotated, reflected, and translated accordingly.

In [24], G is covered by appropriate subgraphs H which are determined using a swap-based heuristic from an initial covering. Both local and global phases are solved using the SDP formulation in [25]. A version of this algorithm targeting the WSNL (see section 4.1) was proposed in [26]; the difference is that, since the positions of some vertices are known a priori, the subgraphs H are clusters formed around these vertices (see section 4.1.2).

In [111], the subgraphs include one or more $(K+1)$-cliques. The local phase is very efficient, as cliques can be realized in linear time [209, 65]. The global phase is solved using an SDP formulation proposed in [2] (also see section 4.1.2).

A very recent method called 3D-ASAP [55], designed to be scalable, distributable, and robust with respect to data noise, employs either a weak form of unique localizability (for exact distances) or spectral graph partitioning (for noisy distance data) to identify clusters. The local phase is solved using either local NLP- or SDP-based techniques (whose solutions are refined using appropriate heuristics), while the global phase reduces to a 3D synchronization problem, i.e., finding rotations in the special orthogonal group $SO(3,\mathbb{R})$, reflections in $\mathbb{Z}_2$, and translations in $\mathbb{R}^3$ such that two similar distance spaces have the best possible alignment in $\mathbb{R}^3$. This is addressed using a 3D extension of a spectral technique introduced in [205]. A somewhat simpler version of the same algorithm tailored to the case $K = 2$ (with the WSNL as motivating application; see section 4.1) is discussed in [54].

3.3. Discretizability.

Some DGP instances can be solved using mixed-combinatorial algorithms such as GBU-based (section 3.2.3) and graph-decomposition-based (section 3.2.4) methods. Combinatorial methods offer several advantages with respect to continuous ones, for example, accuracy and efficiency. In this section, we shall give an in-depth view of discretizability of the DGP and discuss at length an exact combinatorial algorithm for finding *all solutions* to those DGP instances which can be discretized.

We let X be the set of all valid realizations in $\mathbb{R}^K$ of a given weighted graph $G = (V, E, d)$ modulo rotations and translations (i.e., if $x \in X$, then no other valid realization y for which there exists a rotation and/or translation operator T with $y = Tx$ is in X). We remark that we allow reflections for technical reasons: much of the theory of discretizability is based on partial reflections, and since any reflection is also a partial (improper) reflection, disallowing reflections would complicate notation later on. In practice, the DGP system (1.1) can be reduced modulo translations by

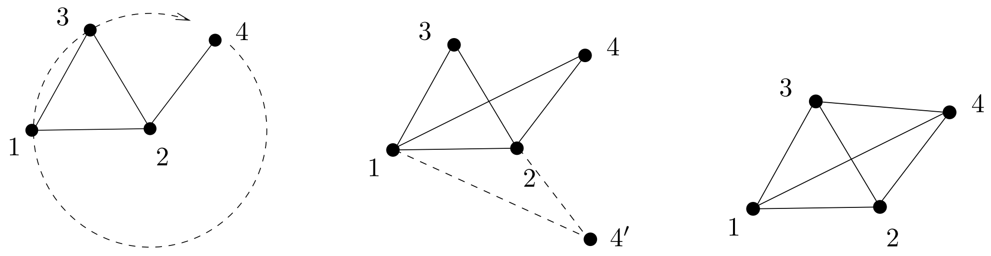


Fig. 3.6 *A flexible framework (left), a rigid graph (center), and a uniquely localizable (rigid) graph (right).*

fixing a vertex v_1 to $x_{v_1} = (0, \dots, 0)$ and modulo rotations by fixing an appropriate set of components out of the realizations of the other $K - 1$ vertices $\{v_2, \dots, v_K\}$ to values that are consistent with the distances in the subgraph of G induced by $\{v_i \mid 1 \le i \le K\}$.

Assuming $X \neq \varnothing$, every $x \in X$ is a solution of the polynomial system

$$\forall \{u, v\} \in E \quad \|x_u - x_v\|^2 = d_{uv}^2, \tag{3.6}$$

and as such it has either finite or uncountable cardinality (this follows from a fundamental result on the structure of semialgebraic sets [16, Thm. 2.2.1]; also see [163]). This feature is strongly related to graph rigidity (see sections 1.1.4 and 4.2.2); specifically, $|X|$ is finite for a rigid graph and almost all nonrigid graphs yield uncountable cardinalities for X whenever X is nonempty. If we know that G is rigid, then $|X|$ is finite and, a posteriori, we only need to look for a finite number of realizations in $\mathbb{R}^K$: a combinatorial search is better suited than a continuous one.

When $K = 2$, it is instructive to inspect a graphical representation of the situation (Figure 3.6). The framework for the graph $(\{1, 2, 3, 4\}, \{\{1, 2\}, \{1, 3\}, \{2, 3\}, \{2, 4\}\})$ shown in Figure 3.6 (left) is flexible: any of the uncountably many positions for vertex 4 (shown by the dashed arrow) yield a valid realization of the graph. If we add the edge $\{1, 4\}$, there are exactly two positions for vertex 4 (Figure 3.6, center), and if we also add $\{3, 4\}$, there is only one possible position (Figure 3.6, right). Accordingly, if we can only use one distance d_{24} to realize x_4 in Figure 3.6 (left), X is uncountable, but if we can use $K = 2$ distances (Figure 3.6, center) or $K + 1 = 3$ distances (Figure 3.6, right), then $|X|$ becomes finite. The GBU algorithm [66] and the triangulation method in [73] exploit the situation shown in Figure 3.6 (right); the difference between these two methods is that the latter exploits a vertex order given a priori that ensures that a solution can be found for every realizable graph.

The core of the work that the authors of this survey have been carrying out (with the help of several colleagues) since 2005 is focused on the situation shown in Figure 3.6 (center): we do not have *one* position to realize the next vertex v in the given order, but (in almost all cases) *two*, x_v^0, x_v^1, so that the graph is rigid but not uniquely so. In order to disregard translations and rotations, we assume a realization $\bar{x}$ of the first K vertices is given as part of the input. This means that there will be two possible positions for x_{K+1}, four for x_{K+2}, and so on. All in all, $|X| = 2^{n-K}$. The situation becomes more interesting if we consider additional edges in the graph, which sometimes make one or both of x_v^0, x_v^1 infeasible with respect to (1.1). A natural methodology to exploit this situation is to follow the binary branching process whenever possible, pruning a branch x_v^ℓ ($\ell \in \{0, 1\}$) only when there is an additional edge $\{u, v\}$ whose associated distance d_{uv} is incompatible with the position x_v^ℓ. We call this methodology *branch-and-prune* (BP).

Our motivation for studying nonuniquely rigid graphs arises from protein conformation; realizing the protein backbone in $\mathbb{R}^3$ is possibly the most difficult step to realizing the whole protein (arranging the side chains can be seen as a subproblem [194, 193]). As discussed in the rest of this section, protein backbones also conveniently supply a natural atomic ordering, which can be exploited in various ways to produce a vertex order that will guarantee exactness of the BP method. The edges necessary to pruning are supplied by NMR experiments. A definite advantage of the BP method is that it offers a theoretical guarantee of finding *all* realizations in X instead of just *one*, as most other methods do.

3.3.1. Rigid Geometry Hypothesis and Molecular Graphs. Discretizability of the search space turns out to be possible only if the molecule is rigid in physical space, which fails to be the case in practice. In order to realistically model the flexing of a molecule in space, it is necessary to consider the bond-stretching and bond-bending effects, which increase the number of variables of the problem and also the computational effort to solve it. However, it is common in molecular conformational calculations to assume that all bond lengths and bond angles are fixed at their equilibrium values, which is known as the *rigid geometry hypothesis* [80].

It follows that for each pair of atomic bonds, say, $\{u, v\}, \{v, w\}$, the covalent bond lengths d_{uv}, d_{vw} are known, as well as the angle between them. With this information, it is possible to compute the remaining distance d_{uw}. Every weighted graph G representing bonds (and their lengths) in a molecule can therefore be trivially completed with weighted edges $\{u, w\}$ whenever there is a path with two edges connecting u and w. Such a completion, denoted G^2, is called a *molecular graph* [104]. We remark that all graphs that the BP method can realize are molecular, but not vice versa.

3.3.2. Sphere Intersections and Probability. For a center $c \in \mathbb{R}^K$ and a radius $r \in \mathbb{R}_+$, we denote by $S^{K-1}(c, r)$ the sphere centered at c with radius r in $\mathbb{R}^K$. The intersection of K spheres in $\mathbb{R}^K$ might contain zero, one, two, or uncountably many points depending on the position of the centers $x_1, \ldots, x_K$ and the lengths $d_{1,K+1}, \ldots, d_{K,K+1}$ of the radii [47]. Let $P = \bigcap_{i \le K} S^{K-1}(x_i, d_{i,K+1})$ be the intersection of these K spheres and $\mathcal{U}^- = \{x_i \mid i \le K\}$. If dim aff$(\mathcal{U}^-) < K - 1$, then $|P|$ is uncountable [121, Lemma 3] (see Figure 3.7). Otherwise, if dim aff$(\mathcal{U}^-) = K-1$, then

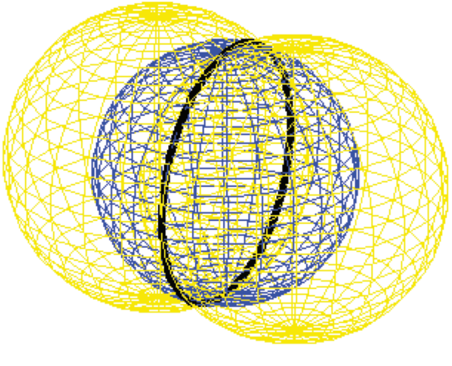


Fig. 3.7 *When three sphere centers are collinear in three dimensions, a nonempty sphere intersection (the thick circle) has uncountable cardinality.*

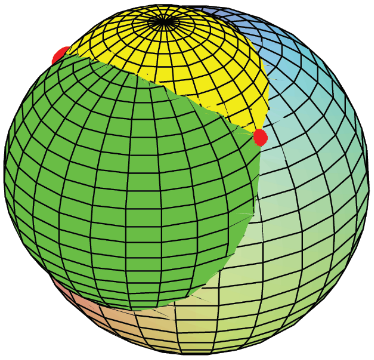


Fig. 3.8 *General case for the intersection P of three spheres in $\mathbb{R}^3$.*

$|P| \in \{0,1,2\}$ [121, Lemmata 1–2]. We also remark that the condition dim aff$(\mathcal{U}^-) < K-1$ corresponds to requiring that $\mathsf{CM}(\mathcal{U}^-) = 0$. See [182] for a detailed treatment of sphere intersections in molecular modeling.

Now assume dim aff$(\mathcal{U}^-) = K-1$, let x_{K+1} be a given point in P, and let $\mathcal{U} = \mathcal{U}^- \cup \{x_{K+1}\}$. The inequalities $\Delta_K(\mathcal{U}) \geq 0$ (see (2.1)) are called *simplex inequalities* (or *strict* simplex inequalities if $\Delta_K(\mathcal{U}) > 0$). We remark that, by definition of the Cayley–Menger determinant, the simplex inequalities are expressed in terms of the squared values d_{uv} of the distance function, rather than the points in $\mathcal{U}$. Accordingly, given a weighted clique $\mathbf{K} = (U,E,d)$ where $|U| = K+1$, we can also denote the simplex inequalities as $\Delta_K(U,d) \geq 0$. If the simplex inequalities fail to hold, then the clique cannot be realized in $\mathbb{R}^K$, and $P = \varnothing$. If $\Delta_K(U,d) = 0$, the simplex has zero volume, which implies that $|P| = 1$ by [121, Lemma 1]. If the strict simplex inequalities hold, then $|P| = 2$ by [121, Lemma 2] (see Figure 3.8). In summary, if $\mathsf{CM}(\mathcal{U}^-) = 0$, then P is uncountable; if $\Delta_K(U,d) = 0$, then $|P| = 1$; and all other cases lead to $|P| \in \{0,2\}$.

Next, we are going to consider a probability distribution in $\mathbb{R}^K$. Notice that any realization $x = (x_1,\ldots,x_n)$ of any weighted graph G in $\mathbb{R}^K$ is contained in a ball $\mathbb{B}$ centered at, say, x_1, with radius $\sum_{\{u,v\}\in T} d_{uv}$, where T is a minimum spanning tree of G (this is a "worst case" corresponding to every point in the realization being collinear). Thus the set X of all realizations of a given graph is bounded, say, $X \subseteq \mathbb{B}'$, where the ball $\mathbb{B}' \subseteq \mathbb{R}^{Kn}$ is induced by $\mathbb{B} \subseteq \mathbb{R}^K$.

We now focus on the case where $G = \mathbf{K}$ only has K vertices. Consider the uniform probability distribution on $\mathbb{B}'$. The probability that any randomly sampled realization in $\mathbb{B}'$ belongs to any given subset with Lebesgue measure zero is equal to zero. Since both $\{x \in \mathbb{R}^{K^2} \mid \mathsf{CM}(\mathcal{U}^-) = 0\}$ and $\{x \in \mathbb{R}^{K^2} \mid \Delta_K(U,d) = 0\}$ are (strictly) lower-dimensional manifolds in $\mathbb{R}^{K^2}$, they have Lebesgue measure zero, and so do their restrictions to $\mathbb{B}'$. Thus the probability that $|P| = 1$ or P is uncountable for any given $x \in \mathbb{B}'$ is zero. Furthermore, if we assume $P \neq \varnothing$, then $|P| = 2$ with probability 1.

We extend this notion to hold for any given sentence $\mathsf{p}(x)$: the statement "for all $x \in Y$ ($\mathsf{p}(x)$ with probability 1)" (where Y is a bounded set in a Euclidean space) means that the statement $\mathsf{p}(x)$ holds over a subset of Y having the same Lebesgue measure as Y. Typically, this occurs whenever p is a geometrical statement that fails to hold for strictly lower-dimensional manifolds. These situations, such as collinearity causing an uncountable P in Figure 3.7, are generally described by equations. Notice that an event can occur with probability 1 conditional to another event happening with probability 0. For example, we shall show in section 3.3.8 that the cardinality of the solution set of YES instances of the ^{K}DMDGP is a power of two with probability 1, even though a ^{K}DMDGP instance has probability 0 of being a YES instance, when sampled uniformly in the set of all ^{K}DMDGP instances.

We remark that our notion of "statement holding with probability 1" is different from the genericity assumption that is used in early works in graph rigidity (see section 4.2 and [45]): a finite set S of real values is *generic* if the elements of S are algebraically independent over $\mathbb{Q}$, i.e., there exists no rational polynomial whose set of roots is S. This requirement is too stringent for our aims. The notion we propose is similar to Graver's definition of genericity: all minors of the complete rigidity matrix must be nontrivial (see section 4.2.2 and [87]).

Lastly, most computer implementations will only employ (a subset of) rational numbers. This means that the genericity assumption based on algebraic independence can only ever work for sets of at most one floating point number (any other being trivially linearly dependent on it), which makes the whole exercise futile (as remarked in [95]). The fact that $\mathbb{Q}$ has Lebesgue measure zero in $\mathbb{R}$ also makes our notion theoretically void, since it destroys the possibility of sampling in a set of positive Lebesgue measure. But the practical implications of the two notions are different: whereas no two floating points will ever be algebraically independent, it is empirically extremely unlikely that any sampled vector of floating point numbers should belong to the manifold defined by a given set of rational equations. This is one more reason why we prefer our "probability 1" notion to genericity.

3.3.3. The Discretizable Vertex Ordering Problem. The theory of sphere intersections, as described in section 3.3.2, implies that if there exists a vertex order on V such that each vertex v such that $\rho(v) > K$ has exactly K adjacent predecessors, then with probability 1 we have $|X| = 2^{n-K}$. If there are at least K adjacent predecessors, $|X| \leq 2^{n-K}$ as either or both positions x_v^0, x_v^1 for v might be infeasible with respect to some distances. In the rest of this paper, to simplify notation we identify each vertex $v \in V$ with its (unique) rank $\rho(v)$, let $V = \{1, \ldots, n\}$, and write, e.g., $u - v$ to mean $\rho(u) - \rho(v)$ or $v > K$ to mean $\rho(v) > K$.

In this section we discuss the problem of identifying an order with the above properties. Formally, the DVOP asks to find a vertex order on V such that $G[\{1, \ldots, K\}]$ is a K-clique and such that for all $v > K$ $(|N(v) \cap \gamma(v)| \geq K)$. We specify that the first K vertices induce a clique in G, because this allows us to realize the first K vertices uniquely—it is a requirement of DDGPs that a realization should be known for the first K vertices.

The DVOP is NP-complete by reduction from the K-clique. An exponential time solution algorithm consists in testing each subset of K vertices; if one is a clique, then try to build an order by greedily choosing a next vertex with the largest number of adjacent predecessors, stopping whenever this is smaller than K. This yields an $O(n^{K+3})$ algorithm. If K is a fixed constant, then of course this becomes a polynomial algorithm, showing that the DVOP with fixed K is in **P**. Since DGP applications rarely require a variable K, this is a positive result.

The computational results given in [121] show that solving the DVOP as a preprocessing step sometimes allows the solution of a sparse PDB instance whose backbone order is not a DVOP order. This may happen if the distance threshold used to generate sparse PDB instances is set to values that are lower than usual (e.g., 5.5Å instead of 6Å).

3.3.4. The Discretizable Distance Geometry Problem. The input of the DDGP consists of

- a simple weighted undirected graph $G = (V, E, d)$;
- an integer $K > 0$;
- an order on V such that
  - for each $v > K$, the set $N(v) \cap \gamma(v)$ of adjacent predecessors has *at least* K elements;
  - for each $v > K$, $N(v) \cap \gamma(v)$ contains a subset U_v of *exactly* K elements such that
    - $G[U_v]$ is a K-clique in G;
    - strict triangular inequalities $\Delta_{K-1}(U_v, d) > 0$ hold (see (2.1));
- a valid realization $\bar{x}$ of the first K vertices.

The DDGP asks whether $\bar{x}$ can be extended to a valid realization of G [121]. The DDGP with fixed K is denoted by DDGP$_K$; the DDGP$_3$ is discussed in [168].

We remark that any method that computes x_v in terms of its adjacent predecessors is able to employ a current realization of the vertices in U_v during the computation of x_v. As a consequence, $\Delta_{K-1}(U_v, d)$ is well defined (during the execution of the algorithm), even though $G[U_v]$ might fail to be a clique in G. Thus, potentially more DGP instances besides those in the DDGP can be solved with a DDGP method of this kind. The DDGP is NP-hard because it contains the DMDGP (see section 3.3.7 below), and there is a reduction from Subset-Sum [79] to the DMDGP [127].

3.3.5. The Branch-and-Prune Algorithm. The recursive step of an algorithm for realizing a vertex v given an embedding x' for $G[\gamma_v]$, where γ_v is the set of predecessors of v, is shown in Algorithm 1. We recall that $S^{K-1}(y, r)$ denotes the sphere in $\mathbb{R}^K$ centered at y with radius r. By the discretization due to sphere intersections, we note that $|P| \leq 2$. The BP algorithm consists in calling BP$(K+1, \bar{x}, \varnothing)$. The BP method finds the set X of all valid realizations of a DDGP instance graph $G = (V, E, d)$ in $\mathbb{R}^K$ modulo rotations and translations [145, 127, 168]. The structure of its recursive calls is a binary tree (called the *BP tree*), which contains 2^{n-K} nodes in the worst case; this makes BP a worst-case exponential algorithm. Figure 3.9 gives an example of a BP tree.

Algorithm 1 BP(v, $\bar{x}$, X)

```
Require: A vertex $v \in V \smallsetminus [K]$, an embedding $x'$ for $G[\gamma_v]$, a set $X$.
$P = \bigcap_{u \in N(v), u < v} S^{K-1}(x'_u, d_{uv})$;
for $x_v \in P$ do
    $x = (x', x_v)$
    if $v = n$ then
        $X \leftarrow X \cup \{x\}$
    else
        BP$(v+1, x, X)$
    end if
end for
```

Realizations $x \in X$ can also be represented by sequences $\chi(x) \in \{-1, 1\}^n$ such that (i) $\chi(x)_v = 1$ for all $v \leq K$; (ii) for all $v > K$, $\chi(x)_v = -1$ if $ax_v < a_0$ and $\chi(x)_v = 1$ if $ax_v \geq a_0$, where $ax = a_0$ is an equation of the hyperplane through $x(U_v) = \{x_u \mid u \in U_v\}$, which is unique with probability 1. The vector $\chi(x)$ is also

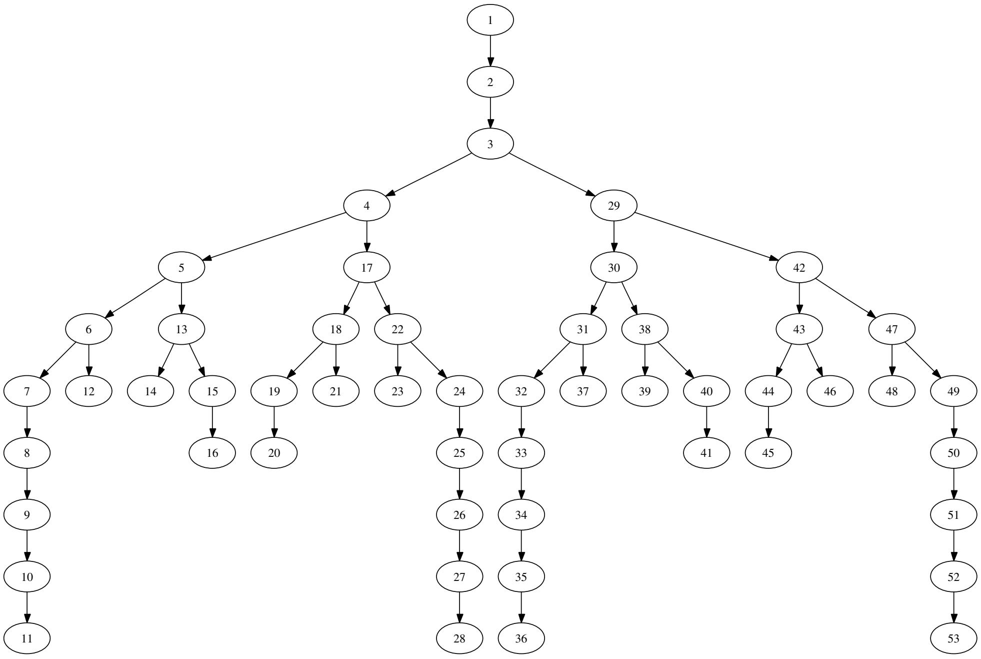


Fig. 3.9 *An example of a BP tree on the random instance* `lavor11_7` [120]*. Pruning edges (see section* 3.3.5.1*) are as follows:* $N(2) = \{9\}$*,* $N(3) = N(4) = \{8, 9, 10\}$*,* $N(5) = \{9, 10\}$*,* $N(6) = \{10\}$*,* $N(7) = \{11\}$*.*

known as the *chirality* [52] of x (formally, the chirality is defined to be $\chi(x)_v = 0$ if $ax = a_0$, but since this case holds with probability 0, we disregard it).

The BP algorithm (Algorithm 1) can be run to termination to find all possible valid realizations of G, or stopped after the first leaf node at level n is reached, in order to find just one valid realization of G. Compared to most continuous search algorithms we tested for DGP variants, the performance of the BP algorithm is impressive from the points of view of both efficiency and reliability, and, to the best of our knowledge, it is currently the only method that is able to find all valid realizations of DDGP graphs. The computational results in [127], obtained using sparse PDB instances as well as hard random instances [120], show that graphs with thousands of vertices and edges can be realized on standard PC hardware from 2007 in fewer than 5 seconds, to an LDE accuracy of at worst $O(10^{-8})$. Complete sets X of incongruent realizations were obtained for 25 sparse PDB instances (generation threshold fixed at 6Å) with sizes ranging from $n = 57, m = 476$ to $n = 3861, m = 35028$. All such sets contain exactly one realization with RMSD value of at worst $O(10^{-6})$, together with one or more isomers, all of which have LDE values of at worst $O(10^{-7})$ (and most often $O(10^{-12})$ or less). The cumulative CPU time taken to obtain all these solution sets is 5.87s of user CPU time, with one outlier taking 90% of the total.

3.3.5.1. Pruning Devices. We partition E into the sets $E_D = \{\{u, v\} \in E \mid u \in U_v\}$ and $E_P = E \setminus E_D$. We call E_D the *discretization edges* and E_P the *pruning edges*. Discretization edges guarantee that a DGP instance is in the DDGP. Pruning edges are used to reduce the BP search space by pruning its tree. In practice, pruning edges might cause the set T in Algorithm 1 to have cardinality 0 or 1 instead of 2, if the distance associated with them is incompatible with the distances of the discretization edges.

The pruning carried out using pruning edges is called *direct distance feasibility* (DDF) and is by far the easiest, most efficient, and most generally useful. Other pruning tests have been defined. A pruning technique called *Dijkstra shortest path* (DSP) was considered in [127, sect. 4.2], based on the fact that G is a Euclidean network. Specifically, the total weight of a shortest path from u to v provides an upper bound to the Euclidean distance between x_u and x_v, and can therefore be employed to prune positions x_v that are too far from x_u. The DSP was found to be effective in some instances, but too often very costly. Other more effective pruning tests based on chemical observations, including secondary structures provided by NMR data, have been considered in [176].

3.3.6. Dual Branch-and-Prune. There is a close relationship between the DGP_K and the EDMCP (see section 2.6.2) with K fixed: each DGP_K instance G can be transformed in linear time to an EDMCP instance (and vice versa) by just considering the weighted adjacency matrix of G, where vertex pairs $\{u,v\} \notin E$ correspond to entries missing from the matrix. We shall call $\mathscr{M}(G)$ the EDMCP instance corresponding to G and $\mathscr{G}(A)$ the DGP_K instance corresponding to an EDMCP instance A.

As remarked in [184], the completion in $\mathbb{R}^3$ of a distance (sub)matrix D with the structure

$$\begin{pmatrix} 0 & d_{12} & d_{13} & d_{14} & \boxed{\delta} \\ d_{21} & 0 & d_{23} & d_{24} & d_{25} \\ d_{31} & d_{32} & 0 & d_{34} & d_{35} \\ d_{41} & d_{42} & d_{43} & 0 & d_{45} \\ \boxed{\delta} & d_{52} & d_{53} & d_{54} & 0 \end{pmatrix} \tag{3.7}$$

can be carried out in constant time by solving a quadratic system in the unknown δ derived from setting the Cayley–Menger determinant (see section 2) of the distance space (X,d) to zero, where $X = \{x_1, \ldots, x_5\}$ and d is given by (3.7). This is because the Cayley–Menger determinant is proportional to the volume of a 4-simplex, which is the (unique, up to congruences) realization of the weighted 5-clique defined by a full distance matrix. Since a simplex on five points embedded in $\mathbb{R}^3$ necessarily has 4-volume equal to zero, it suffices to set the Cayley–Menger determinant of (3.7) to zero to obtain a quadratic equation in δ.

We denote the pair $\{u,v\}$ indexing the unknown distance δ by $\mathsf{e}(D)$, the Cayley–Menger determinant of D by $\mathsf{CM}(D)$, and the corresponding quadratic equation in δ by $\mathsf{CM}(D,\delta) = 0$. If D is a distance matrix, then $\mathsf{CM}(D,\delta) = 0$ has real solutions; furthermore, in this case it has two distinct solutions δ^1, δ^2 with probability 1, as remarked in section 3.3. These are two valid values for the missing distance d_{15}. This observation extends to general K, where we consider a $(K+1)$-simplex realization of a weighted near-clique (defined as a clique with a missing edge) on $K+2$ vertices.

3.3.6.1. BP in Distance Space. In this section we discuss a coordinate-free BP variant that takes decisions about distance values on missing edges rather than on realization of vertices in $\mathbb{R}^K$. We are given a DDGP instance with a graph $G = (V,E)$ and a partial embedding $\bar{x}$ for the subgraph $G[[K]]$ of G induced by the set $[K]$ of the first K vertices. The DDGP order on V guarantees that the vertex of rank $K+1$ has K adjacent predecessors, hence it is adjacent to all the vertices of rank $v \in [K]$. Thus, $G[[K+1]]$ is a full $(K+1)$-clique. Consider now the vertex of rank $K+2$: again, the DDGP order guarantees that it has at least K adjacent predecessors. If it has

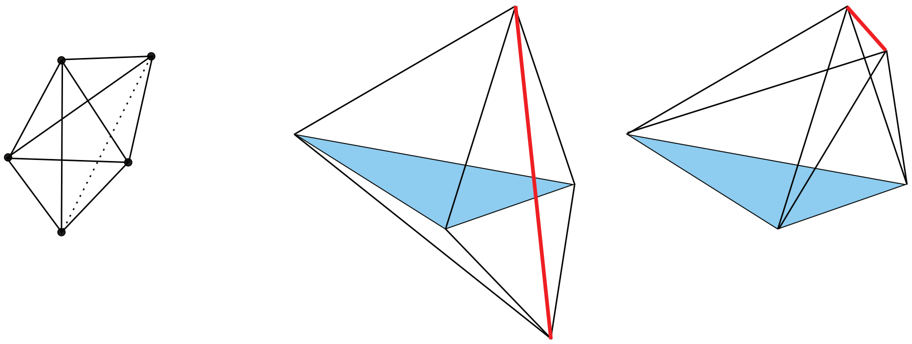


Fig. 3.10 *On the left, a near-clique on five vertices with one missing edge (dotted line). Center and right, its two possible realizations in* $\mathbb{R}^3$ *(missing distance shown in red).*

$K+1$, then $G[[K+2]]$ is the full $(K+2)$-clique. Otherwise $G[[K+2]]$ is a near-clique on $K+2$ vertices with a missing edge $\{u, K+2\}$ for some $u \in [K+1]$. We can therefore use the Cayley–Menger determinant (see (3.7) for the special case $K=3$ and section 2 for the general case) to compute two possible values for $d_{u,K+2}$. Because the vertex order always guarantees at least K adjacent predecessors, this procedure can be generalized to vertices of any rank v in $V \smallsetminus [K]$, and so it defines a recursive algorithm that

- branches whenever a distance can be assigned two different values;
- simply continues to the next rank whenever the subgraph induced by the current $K+2$ vertices is a full clique;
- prunes all branches whenever the partial distance matrix defined on the current $K+2$ vertices has no Euclidean completion.

In general, this procedure holds for DDGP instances G whenever there is a vertex order such that each next vertex v is adjacent to K predecessors. This ensures G has a subgraph (containing v and $K+1$ predecessors) consisting of two $(K+1)$-cliques whose intersection is a K-clique, i.e., a near-clique with one missing edge. There are in general two possible realizations in $\mathbb{R}^K$ for such subgraphs, as shown in Figure 3.10.

Algorithm 2 presents the dual BP method. It takes as input a vertex v of rank greater than $K+1$, a partial matrix A, and a set $\mathscr{A}$ which will eventually contain all the possible completions of the partial matrix given as the problem input. For a given partial matrix A, a vertex v of $\mathscr{G}(A)$, and an integer $\ell \leq K$, let A_v^ℓ be the $\ell \times \ell$ symmetric submatrix of A including row and column v that has fewest missing components. Whenever A_v^{K+2} has no missing elements, the equation $\mathsf{CM}(A_v^{K+2}, \delta) = 0$ is either a tautology if A_v^{K+2} is an EDM, or unsatisfiable in $\mathbb{R}$ otherwise. In the first case, we define it to have $\delta = d_{uv}$ as a solution, where u is the smallest row/column index of A_v^{K+2}. In the second case, it has no solutions.

Theorem 3.1 (see [144]). *At the end of Algorithm 2,* $\mathscr{A}$ *contains all possible completions of the input partial matrix.*

The similarity of Algorithms 1 and 2 is such that it is very easy to assign dual meanings to the original (otherwise known as *primal*) BP algorithms. This duality stems from the fact that weighted graphs and partial symmetric matrices are "dual" to each other through the inverse mappings $\mathscr{M}$ and $\mathscr{G}$. Whereas in the primal BP algorithm we decide realizations of the graph, in the dual BP algorithm we decide the completions of partial matrices, so realizations and distance matrix completions are

Algorithm 2 dBP(v, A, $\mathscr{A}$)

```
Require: A vertex $v \in V \smallsetminus [K+1]$, a partial matrix $A$, a set $\mathscr{A}$.
$P = \{\delta \mid \mathsf{CM}(A_v^{K+2}, \delta) = 0\}$
for $\delta \in P$ do
    $\{u, v\} \leftarrow \mathsf{e}(A_v^{K+2})$
    $d_{uv} \leftarrow \delta$
    if $A$ is complete then
        $\mathscr{A} \leftarrow \mathscr{A} \cup \{A\}$
    else
        dBP($v+1$, $A$, $\mathscr{A}$)
    end if
end for
```

dual to each other. The primal BP method decides on points $x_v \in \mathbb{R}^K$ to assign to the next vertex v, whereas the dual BP method decides on distances δ to assign to the next missing distance incident to v and to a predecessor of v; there are at most two choices of x_v as there are at most two choices for δ; only one choice of x_v is available whenever v is adjacent to strictly more than K predecessors, and the same happens for δ; finally, no choices for x_v are available in the case that the current partial realization cannot be extended to a full realization of the graph, and no choices for δ are available in the case that the current partial matrix cannot be completed to an EDM. Thus, point vectors and distance values are dual to each other. The same vertex order can be used by both the primal and the dual BP algorithms (so the order is self-dual).

There is one clear difference between primal and dual BP methods: namely, that the dual BP method needs an initial $(K+1)$-clique, whereas the primal BP method only needs an initial K-clique. This difference also has a dual interpretation: a complete EDM corresponds to two (rather than one) realizations, one being the reflection of the other through the hyperplane defined by the first K points (this is the "fourth level symmetry" referred to in [127, sect. 2.1] for the case $K = 3$). We remark that this difference is related to the reason why the exact SDP-based polynomial method for realizing uniquely localizable (see section 3.2.4) networks proposed in [157] needs the presence of at least $K+1$ anchors.

3.3.7. The Discretizable Molecular Distance Geometry Problem. The DMDGP is a subset of instances of the DDGP$_3$; its generalization to arbitrary K is denoted ^{K}DMDGP. The difference between the DMDGP and the DDGP is that U_v is required to be the set of K *immediate* (rather than arbitrary) predecessors of v. So, for example, the discretization edges can also be expressed as $E_D = \{\{u, v\} \in E \mid |u - v| \leq K\}$ (see section 3.3.5.1), and $x(U_v) = \{x_{v-K}, \ldots, x_{v-1}\}$. This restriction originates from the practically interesting case of realizing protein backbones with NMR data.

Since such graphs are molecular (see section 3.3.1), they have vertex orders guaranteeing that each vertex $v > 3$ is adjacent to *two* immediate predecessors, as shown in Figure 3.11. The distance $d_{v,v-2}$ is computed using the covalent bond lengths and the angle $(v-2, v-1, v)$, which are known because of the rigid geometry hypothesis [80]. In general, this is only enough to guarantee discretizability for $K = 2$. By exploiting further protein properties, however, we were able to find a vertex order (different from the natural backbone order) that satisfies the DMDGP definition (see section 3.5.2).

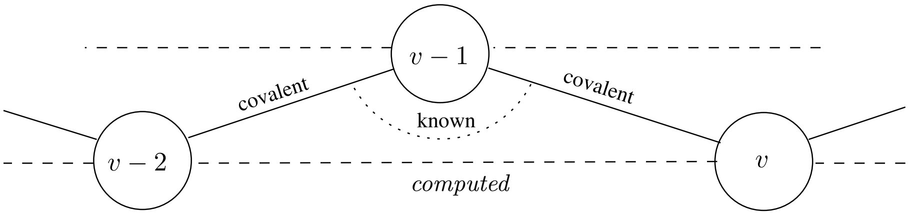


Fig. 3.11 *Vertex v is adjacent to its two immediate predecessors.*

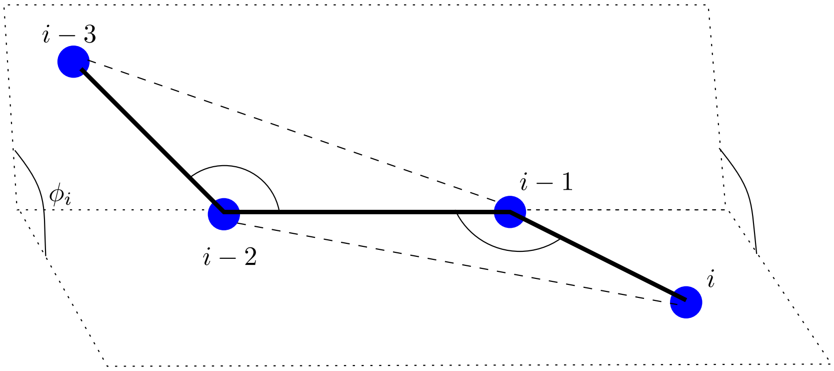


Fig. 3.12 *The torsion angle ϕ_i.*

Requiring that all adjacent predecessors of v must be immediate provides sufficient structure to prove several results about the symmetry of the solution set X (section 3.3.8) and about the fixed-parameter tractability of the BP algorithm (Algorithm 1) when solving ^{K}DMDGPs on protein backbones with NMR data (section 3.3.9). The DMDGP is NP-hard by reduction from SUBSET-SUM [127]. The result can be generalized to the ^{K}DMDGP [147].

3.3.7.1. Mathematical Programming Formulation. For completeness, and for the convenience of MP-versed readers, we provide here an MP formulation of the DMDGP. We model the choice between x_v^0, x_v^1 by using *torsion angles* [126]: these are the angles ϕ_v defined for each $v > 3$ by the planes passing through $x_{v-3}, x_{v-2}, x_{v-1}$ and x_{v-2}, x_{v-1}, x_v (Figure 3.12). More precisely, we suppose that the cosines $c_v = \cos(\phi_v)$ of such angles are also part of the input. In fact, the values for $c : V \smallsetminus \{1,2,3\} \to \mathbb{R}$ can be computed using the DMDGP structure of the weighted graph in constant time using [93, eq. (2.15)]. Conversely, if one is given precise values for the torsion angle cosines, then every quadruplet $(x_{v-3}, x_{v-2}, x_{v-1}, x_v)$ must be a rigid framework (for $v > 3$). We let $\alpha : V \smallsetminus \{1,2\} \to \mathbb{R}^3$ be the normal vector to the plane defined by three consecutive vertices,

$$\begin{aligned}\forall v \geq 3 \quad \alpha_v &= \begin{vmatrix} \mathbf{i} & \mathbf{j} & \mathbf{k} \\ x_{v-2,1} - x_{v-1,1} & x_{v-2,2} - x_{v-1,2} & x_{v-2,3} - x_{v-1,3} \\ x_{v,1} - x_{v-1,1} & x_{v,2} - x_{v-1,2} & x_{v,3} - x_{v-1,3} \end{vmatrix} \\ &= \begin{pmatrix} (x_{v-2,2} - x_{v-1,2})(x_{v,3} - x_{v-1,3}) - (x_{v-2,3} - x_{v-1,3})(x_{v,2} - x_{v-1,2}) \\ (x_{v-2,1} - x_{v-1,1})(x_{v,3} - x_{v-1,3}) - (x_{v-2,3} - x_{v-1,3})(x_{v,1} - x_{v-1,1}) \\ (x_{v-2,1} - x_{v-1,1})(x_{v,2} - x_{v-1,2}) - (x_{v-2,2} - x_{v-1,2})(x_{v,1} - x_{v-1,1}) \end{pmatrix},\end{aligned}$$

so that α_v is expressed a function $\alpha_v(x)$ of x and represented as a matrix with entries x_{vk}. Now, for every $v > 3$, the cosine of the torsion angle ϕ_v is proportional to the scalar product of the normal vectors α_{v-1} and α_v:

$$\forall v > 3 \quad \alpha_{v-1}(x) \cdot \alpha_v(x) = \|\alpha_{v-1}(x)\| \|\alpha_v(x)\| \cos \phi_v.$$

Thus, the following provides an MP formulation for the DMDGP:

$$\begin{array}{ll} \min_x & \sum_{\{u,v\} \in E} (\|x_u - x_v\|^2 - d_{uv}^2)^2 \\ \text{s.t.} & \forall v > 3 \quad \alpha_{v-1}(x) \cdot \alpha_v(x) \quad = \quad \|\alpha_{v-1}(x)\| \|\alpha_v(x)\| c_v. \end{array} \tag{3.8}$$

We remark that generalizations of (3.8) to arbitrary (fixed) K are made possible by using Graßmann–Plücker relations [30] (also see [52, Chap. 2]).

3.3.8. Symmetry of the Solution Set. When we first experimented with the BP method on the DMDGP, we observed that $|X|$ was always a power of two. An initial conjecture in this direction was quickly disproved by hand-crafting an instance with 54 solutions derived by the polynomial reduction of the Subset-Sum to the DMDGP used in the NP-hardness proof of the DMDGP [127]. Notwithstanding, all protein and protein-like instances we tested yielded $|X| = 2^\ell$ for some integer ℓ. Years later, we were able to prove that the conjecture holds on ^{K}DMDGP instances with probability 1, and also derived an infinite class of counterexamples [151]. Aside from explaining our conjecture arising from empirical evidence, our result is also important insofar as it provides the core of a theory of partial reflections for the ^{K}DMDGP. References to partial reflections are occasionally found in the DGP literature [94, 157], but our group-theoretical treatment is an extensive addition to the current body of knowledge.

In this section we give an exposition that is more compact and hopefully clearer than the one in [151]. We focus on ^{K}DMDGP and therefore assume that U_v contains the K immediate predecessors of v for each $v > K$. We also assume G is a YES instance of the ^{K}DMDGP, so that $|P| = 2$ with probability 1.

3.3.8.1. The Discretization Group. Let $G_D = (V, E_D, d)$ be the subgraph of G consisting of the discretization edges, and let X_D be the set of realizations of G_D; since G_D has no pruning edges by definition, the BP search tree for G_D is a full binary tree and $|X_D| = 2^{n-K}$. The discretization edges arrange the realizations so that, at level $\ell > K$, there are $2^{\ell-K}$ possible positions for the vertex v with rank ℓ. We assume that $|P| = 2$ (see Algorithm 1) at each level v of the BP tree, an event which, for YES instances, in the absence of pruning edges, happens with probability 1. Let $P = \{x_v^0, x_v^1\}$ be the two possible realizations of v at some recursive call of Algorithm 1 at level v of the BP tree; then because P is an intersection of K spheres, x_v^1 is the reflection of x_v^0 through the hyperplane defined by $x(U_v) = \{x_{v-K}, \ldots, x_{v-1}\}$. We denote this reflection operator by R_x^v.

Theorem 3.2 (Cor. 4.6 and Thm. 4.9 of [151]). *With probability* 1, *for all $v > K$ and $u < v - K$, there is a set H^{uv} of 2^{v-u-K} real positive values such that for each $x \in X$ we have $\|x_u - x_v\| \in H^{uv}$. Furthermore, for all $x' \in X$ such that $x' \neq x$ and $x'_t = x_t$ for all $t \leq u + K - 1$, $\|x_u - x_v\| = \|x'_u - x'_v\|$ if and only if $x'_v = R_x^{u+K}(x_v)$.*

We sketch the proof in Figure 3.13 for $K = 2$; the solid arcs at levels $3, 4, 5$ mark the locus of feasible realizations for vertices at rank $3, 4, 5$ in the ^{K}DMDGP order. The dashed arcs represent the spheres S_{uv}^x (see Algorithm 1). Intuitively, two branches from level 1 to level 4 or 5 will have equal segment lengths but different angles between consecutive segments, which will cause the end nodes to be at different distances from

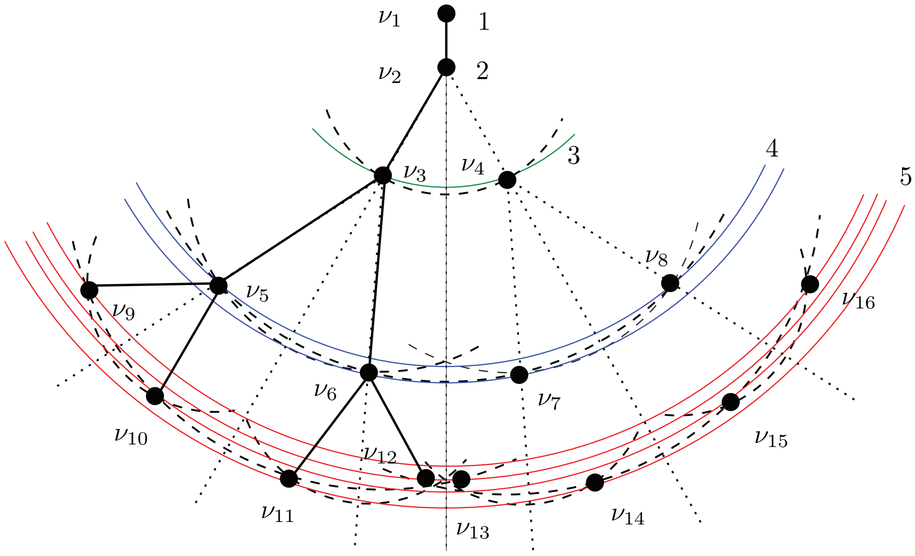


Fig. 3.13 *A pruning edge* $\{1,4\}$ *prunes either* ν_6, ν_7 *or* ν_5, ν_8.

the node at level 1. Observe that the number of solid arcs at each level is a power of two, where the exponent depends on the level index ℓ, and each solid circle contains exactly two realizations (that are reflections of each other) of the same vertex at rank ℓ.

We now give a basic result on reflections in $\mathbb{R}^K$. For any nonzero vector $y \in \mathbb{R}^K$, let $\mathcal{R}(y)$ be the reflection operator through the hyperplane passing through the origin and normal to y. If y is normal to the hyperplane defined by $x_{v-K}, \ldots, x_{v-1}$, then $\mathcal{R}(y) = R_x^v$.

Lemma 3.3 (Lemma 4.2 in [147]). *Let* $x \neq y \in \mathbb{R}^K$ *and* $z \in \mathbb{R}^K$ *such that* z *is not in the hyperplanes through the origin and normal to* x, y. *Then* $\mathcal{R}(x)\mathcal{R}(y)z = \mathcal{R}(\mathcal{R}(x)y)\mathcal{R}(x)z$.

Theorem 3.3 provides a commutativity for reflections acting on points and hyperplanes. Figure 3.14 illustrates the proof for $K = 2$.

For $v > K$ and $x \in X$ we now define partial reflection operators as

$$g_v(x) = (x_1, \ldots, x_{v-1}, R_x^v(x_v), \ldots, R_x^v(x_n)). \tag{3.9}$$

The g_v's map a realization x to its partial reflection with the first branch at v. It is easy to show that the g_v's are injective with probability 1 and idempotent.

Lemma 3.4 (Lemma 4.3 in [147]). *For* $x \in X$ *and* $u, v \in V$ *such that* $u, v > K$, $g_u g_v(x) = g_v g_u(x)$.

We define the *discretization group* to be the symmetry group $\mathcal{G}_D = \langle g_v \mid v > K \rangle$ generated by the partial reflection operators g_v.

Corollary 3.5. *With probability* 1, $\mathcal{G}_D$ *is an Abelian group isomorphic to* C_2^{n-K} *(the Cartesian product consisting of* $n - K$ *copies of the cyclic group of order* 2*).*

For all $v > K$ let $\xi_v = (1, \ldots, 1, -1_v, \ldots, -1)$ be the vector consisting of ones in the first $v - 1$ components and -1 in the last components. Then the g_v actions are naturally mapped onto the chirality functions.

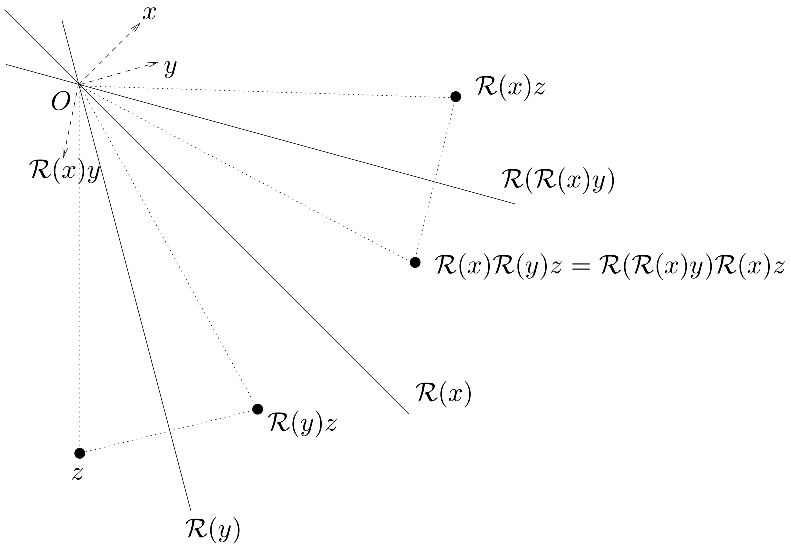

x
y
O
$\mathcal{R}(x)z$
$\mathcal{R}(x)y$
$\mathcal{R}(\mathcal{R}(x)y)$
$\mathcal{R}(x)\mathcal{R}(y)z = \mathcal{R}(\mathcal{R}(x)y)\mathcal{R}(x)z$
$\mathcal{R}(x)$
$\mathcal{R}(y)z$
z
$\mathcal{R}(y)$

Fig. 3.14 *Reflecting through $\mathcal{R}(y)$ first and $\mathcal{R}(x)$ later is equivalent to reflecting through $\mathcal{R}(x)$ first and the reflection of $\mathcal{R}(y)$ through $\mathcal{R}(x)$ later.*

Lemma 3.6 (Lemma 4.5 in [147]). *For all $x \in X$, $\chi(g_v(x)) = \chi(x) \circ \xi_v$, where $\circ$ is the Hadamard product.*

This follows by definitions of g_v and of chirality of a realization. Since, by Algorithm 1, each $x \in X$ has a different chirality, for all $x, x' \in X$ there is $g \in \mathcal{G}_D$ such that $x' = g(x)$, i.e., the action of $\mathcal{G}_D$ on X is transitive. By Theorem 3.2, the distances associated to the discretization edges are invariant with respect to the discretization group.

3.3.8.2. The Pruning Group. Consider a pruning edge $\{u, v\} \in E_P$. By Theorem 3.2, with probability 1 we have $d_{uv} \in H^{uv}$, otherwise G cannot be a YES instance (against the initial assumption). Also, again by Theorem 3.2, $d_{uv} = \|x_u - x_v\| \neq \|g_w(x)_u - g_w(x)_v\|$ for all $w \in \{u + K + 1, \dots, v\}$ (e.g., the distance $\|\nu_1 - \nu_9\|$ in Figure 3.13 is different from all its reflections $\|\nu_1 - \nu_h\|$, with $h \in \{10, 11, 12\}$, w.r.t. g_4, g_5). We therefore define the *pruning group*

$$\mathcal{G}_P = \langle g_w \mid w > K \wedge \forall \{u, v\} \in E_P \ (w \notin \{u + K + 1, \dots, v\})\rangle.$$

By definition, $\mathcal{G}_P \leq \mathcal{G}_D$ and the distances associated with the pruning edges are invariant with respect to $\mathcal{G}_P$.

Theorem 3.7 (Theorem 4.6 in [151]). *The action of $\mathcal{G}_P$ on X is transitive with probability* 1.

Theorem 3.8 (Theorem 4.7 in [147]). *With probability* 1, $\exists \ell \in \mathbb{N}\ |X| = 2^\ell$.

Proof. The argument below holds with probability 1. Since $\mathcal{G}_D \cong C_2^{n-K}$, $|\mathcal{G}_D| = 2^{n-K}$. Since $\mathcal{G}_P \leq \mathcal{G}_D$, $|\mathcal{G}_P|$ divides the order of $|\mathcal{G}_D|$, which implies that there is an integer ℓ with $|\mathcal{G}_P| = 2^\ell$. By Theorem 3.7, the action of $\mathcal{G}_P$ on X has only one orbit, i.e., $\mathcal{G}_P x = X$ for any $x \in X$. By idempotency, for $g, g' \in \mathcal{G}_P$, if $gx = g'x$, then $g = g'$. This implies $|\mathcal{G}_P x| = |\mathcal{G}_P|$. Thus, for any $x \in X$, $|X| = |\mathcal{G}_P x| = |\mathcal{G}_P| = 2^\ell$. ∎

3.3.8.3. Practical Exploitation of Symmetry. These results naturally find a practical application to speed-up the BP algorithm. The BP algorithm proceeds until a first valid realization is identified. It can be shown that, at that point, a set of generators for the group $\mathcal{G}_P$ is known. These are used to generate all other valid realizations of the input graph, up to rotations and translations [167, 169]. Empirically, this cuts the CPU time to roughly $2/|X|$ (the factor 2 is due to the fact that the original BP algorithm already takes one reflection symmetry into account; see [127, Thm. 2]).

3.3.9. Fixed-Parameter Tractability. As the theory of partial reflections, the proof that the BP algorithm is fixed-parameter tractable (FPT) on proteins also stems from empirical evidence. All the CPU time plots versus instance size for the BP algorithm on protein backbones look roughly linear, suggesting that perhaps such instances are a "polynomial case" of the DMDGP. The results that follow provide sufficient conditions for this to be the case, and we were able to verify empirically that PDB proteins conform to these conditions. These results are a consequence of the theory in section 3.3.8 insofar as they rely on an exact count of the BP tree nodes at each level. We formalize this in a directed acyclic graph (DAG) $\mathcal{D}_{uv}$ that represents the number of valid BP search tree nodes in terms of pruning edges between two vertices $u, v \in V$ such that $v > K$ and $u < v - K$ (see Figure 3.15). The first row in Figure 3.15 shows different values for the rank of v w.r.t. u; an arc labeled with an integer i implies the existence of a pruning edge $\{u + i, v\}$ (arcs with $\vee$-expressions replace parallel arcs with different labels). An arc is unlabeled if there is no pruning edge $\{w, v\}$ for any $w \in \{u, \ldots, v - K - 1\}$. The vertices of the DAG are arranged vertically by BP search tree level, and are labeled with the number of BP nodes at a given level, which is always a power of two by Theorem 3.8. A path in this DAG represents the set of pruning edges between u and v, and its incident vertices show the number of valid nodes at the corresponding levels. For example, following unlabeled arcs corresponds to no pruning edge between u and v and leads to a full binary BP search tree with 2^{v-K} nodes at level v.

For a given G_D, each possible pruning edge set E_P corresponds to a path spanning all columns in $\mathcal{D}_{1n}$. Instances with diagonal (Proposition 3.9) or below-diagonal (Proposition 3.10) E_P paths yield BP trees whose width is bounded by $O(2^{v_0})$, where

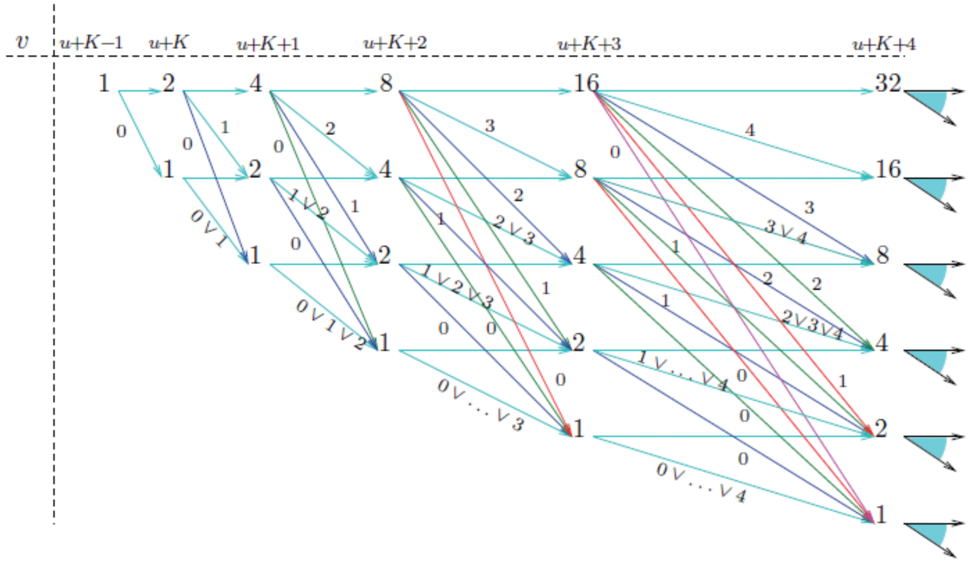


Fig. 3.15 *Number of valid BP nodes (vertex label) at level $u + K + \ell$ (column) in terms of the pruning edges (path spanning all columns).*

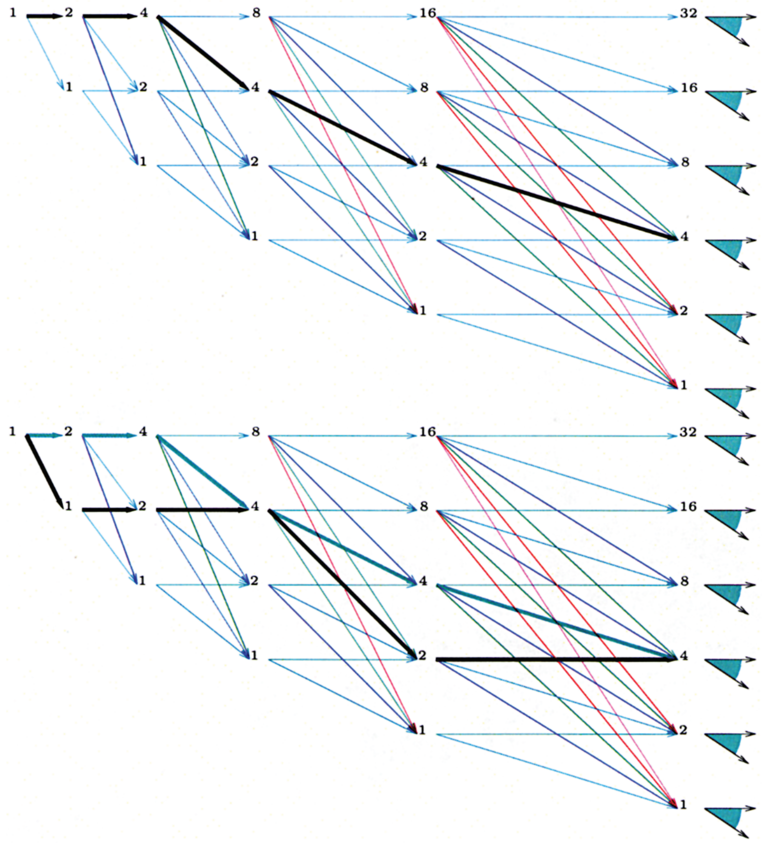


Fig. 3.16 *A path* p_0 *yielding treewidth* 4 *(top) and another path below* p_0 *(bottom).*

v_0 is some fixed vertex in V. Since v_0 is usually small w.r.t. n, the multiplying constant 2^{v_0} is not prohibitively large.

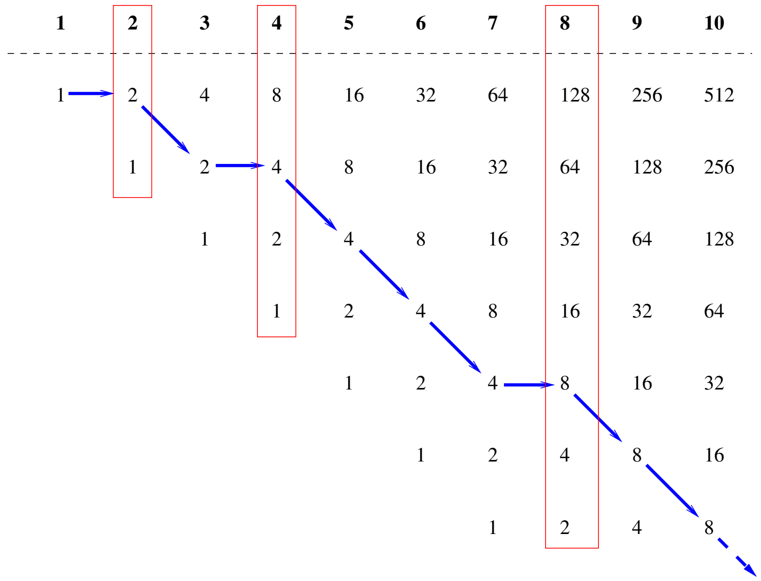
Proposition 3.9 (Proposition 5.1 in [147]). *If* $\exists v_0 > K$ *s.t. for all* $v > v_0$ $\exists u < v - K$ *with* $\{u, v\} \in E_P$*, then the BP search tree width is bounded by* $2^{v_0 - K}$.

This corresponds to a path $\mathsf{p}_0 = (1, 2, \dots, 2^{v_0-K}, \dots, 2^{v_0-K})$ that follows unlabeled arcs up to level v_0 and then arcs labeled $v_0 - K - 1$, $v_0 - K - 1 \vee v_0 - K$, and so on, leading to nodes that are all labeled with 2^{v_0-K} (Figure 3.16, top).

Proposition 3.10 (Proposition 5.2 in [147]). *If* $\exists v_0 > K$ *such that every subsequence* s *of consecutive vertices* $>v_0$ *with no incident pruning edge is preceded by a vertex* v_s *such that* $\exists u_s < v_s$ $(v_s - u_s \geq |s| \wedge \{u_s, v_s\} \in E_P)$*, then the BP search tree width is bounded by* 2^{v_0-K}.

This situation corresponds to a below-diagonal path (Figure 3.16, bottom). In general, for those instances for which the BP search tree width has an $O(2^{v_0} \log n)$ bound, the BP algorithm has a worst-case running time $O(2^{v_0} L 2^{\log n}) = O(Ln)$, where L is the complexity of computing P as defined in Algorithm 1. Since L is typically constant in n [66], for such cases the BP algorithm runs in time $O(2^{v_0} n)$. Let $V' = \{v \in V \mid \exists \ell \in \mathbb{N}\ (v = 2^\ell)\}$.

Proposition 3.11 (Proposition 5.3 in [147]). *If* $\exists v_0 > K$ *s.t. for all* $v \in V \smallsetminus V'$ *with* $v > v_0$ *there is* $u < v - K$ *with* $\{u, v\} \in E_P$*, then the BP search tree width at level* n *is bounded by* $2^{v_0} n$.



1	2	3	4	5	6	7	8	9	10
1	2	4	8	16	32	64	128	256	512
	1	2	4	8	16	32	64	128	256
		1	2	4	8	16	32	64	128
			1	2	4	8	16	32	64
				1	2	4	8	16	32
					1	2	4	8	16
						1	2	4	8

Fig. 3.17 *A path yielding treewidth $O(n)$.*

This corresponds to a path roughly along the diagonal apart from logarithmically many vertices in V (those in V'), at which levels the BP algorithm doubles the number of search nodes (Figure 3.17). For a pruning edge set E_P as in Proposition 3.11, or yielding a path below it, the BP algorithm runs in $O(2^{v_0}n^2)$.

3.3.9.1. Empirical Verification. On a set of 45 protein instances from the PDB, 40 satisfy Proposition 3.9 and 5 satisfy Proposition 3.10, all with $v_0 = 4$ [147]. This is consistent with the computational insight [127] that BP empirically displays a polynomial (specifically, linear) complexity on real proteins.

3.3.10. Development of the Branch-and-Prune Algorithm. To the best of our knowledge, the first discrete search method for the MDGP that exploited the intersection of three spheres in $\mathbb{R}^3$ was proposed by three of the coauthors of this survey (CL, LL, NM) in 2005 [122], in the framework of a quantum computing algorithm. Quite independently, the GBU algorithm was extended in 2008 [237] to deal with intersections of three rather than four spheres. Interestingly, as remarked in section 3.2.3, another extension to the same case was proposed by a different research group in the same year [36]. By contrast, the idea of a vertex order used to find realizations iteratively was already present in early works in statics [195, 96] (see section 4.2) and was first properly formalized in [97] (see section 4.2.3).

The crucial idea of combining the intersection of three spheres with a vertex ordering, which would offer a theoretical guarantee of exactness, occurred in June 2005, when two of the coauthors of this survey (CL, LL) met in Milan, Italy. The first version of the BP algorithm was conceived, implemented, and computationally validated during the summer of 2005; this work, however, only appeared in 2008 [145] due to various editorial mishaps. Between 2005 and 2008 we kept working on the theory of the DMDGP; we were able to publish an arXiv technical report in 2006 [124], which was eventually completed in 2009 and published online in 2011 [127]. Remarkably, our own early work on BP and an early version of [237] were both presented at the International Symposium on Mathematical Programming (ISMP) in Rio de Janeiro in 2006.

Along the years we improved and adapted the original BP algorithm [145] to further settings. We defined precisely the DGP subclasses on which it works, and proved that it finds *all* realizations in X for these subclasses [124, 130, 127, 168]. We discussed how to determine a good vertex order automatically [121]. We tested and fine-tuned the BP algorithm for proteins [175]. We compared it with other methods [178]. We tried to decompose the protein backbone in order to reduce the size of the BP trees [181]. We adapted the BP to work with intervals instead of exact distances [166, 134, 171, 129]. We engineered it to work on distances between atoms of given type (this is an important restriction of NMR experiments) [131, 132, 170, 133, 135]. We generalized it to arbitrary values of K and developed a theory of symmetries in protein backbones [149, 151, 152]. We exploited these symmetries in order to immediately reconstruct all solutions from just one [167, 169]. We showed that the BP algorithm is FPT on protein-like instances and empirically appears to be polynomial on proteins [150, 147]. We derived a dual BP algorithm that works in distance rather than realization space [144]. We put all this together so that it would work on real NMR data [176, 156]. We started working on embedding the side chains [193, 48]. We took some first steps toward applying the BP algorithm to more general molecular conformation problems involving energy minimization [136]. We provided an open-source [177] implementation and tested some parallel ones [174, 86]. We wrote a number of other surveys [125, 148, 128, 172], but none as extensive as the present one. We also edited a book on the subject of DG and applications [173].

3.4. Interval Data. In this section we discuss methods that target an MDGP variant, called iMDGP, which is closer to the real NMR data: edges $\{u, v\} \in E$ are weighted with real intervals $\mathbf{d}_{uv} = [d^L_{uv}, d^U_{uv}]$ instead of real values. These intervals occur in practice because, as in all other physical experiments, NMR outputs data with some uncertainty, which can be modeled using intervals. The iMDGP therefore consists of finding $x \in \mathbb{R}^K$ that satisfies the following set of nonlinear inequalities:

$$\forall \{u, v\} \in E \quad d^L_{uv} \leq \|x_u - x_v\| \leq d^U_{uv}. \tag{3.10}$$

The MP formulation (3.1) can be adapted to deal with this situation in a number of ways, such as

$$\min_x \sum_{\{u,v\}\in E} (\max(d^L_{uv} - ||x_u - x_v||, 0) + \max(||x_u - x_v|| - d^U_{uv}, 0)), \tag{3.11}$$

$$\min_x \sum_{\{u,v\}\in E} (\max((d^L_{uv})^2 - ||x_u - x_v||^2, 0) + \max(||x_u - x_v||^2 - (d^U_{uv})^2, 0), \tag{3.12}$$

$$\min_x \sum_{\{u,v\}\in E} (\max{}^2((d^L_{uv})^2 - ||x_u - x_v||^2, 0) + \max{}^2(||x_u - x_v||^2 - (d^U_{uv})^2, 0)). \tag{3.13}$$

Problem (3.13) is often appropriately modified to avoid bad scaling (which occurs whenever the observed distances differ in order of magnitude):

$$\min_x \sum_{\{u,v\}\in E} \left(\max{}^2 \left(\frac{(d^L_{uv})^2 - ||x_u - x_v||^2}{(d^L_{uv})^2}, 0 \right) + \max{}^2 \left(\frac{||x_u - x_v||^2 - (d^U_{uv})^2}{(d^U_{uv})^2}, 0 \right) \right). \tag{3.14}$$

3.4.1. Smoothing-Based Methods. Several smoothing-based methods (e.g., DG-SOL and DCA; see section 3.2.2) have been trivially adapted to solve (3.13) and/or (3.14).

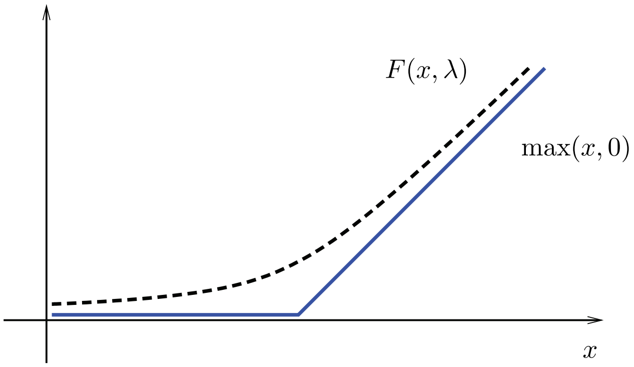


Fig. 3.18 *The function* $\max(x, 0)$ *and its hyperbolic smoothing* $F(x, \lambda)$.

3.4.1.1. Hyperbolic Smoothing. The *hyperbolic smoothing* described in [212, 211] is specifically suited to the shape of each summand in (3.11), as shown in Figure 3.18. The actual solution algorithm is very close to the one employed by DGSOL (see section 3.2.2). Given the fact that the smoothing is not "general purpose" (as the Gaussian transform is), but is specific to the problem at hand, the computational results improve.

3.4.2. The EMBED Algorithm. The EMBED algorithm, proposed by Crippen and Havel [52], first completes the missing bounds and refines the given bounds using triangle and tetrangle inequalities. Then a trial distance matrix D' is randomly generated, and a solution is sought using a matrix decomposition method [28]. Since the distance matrix D' is not necessarily Euclidean [71], the solution may not satisfy (3.10). If this is the case, the final step of the algorithm is to minimize the distance violations using the previous solution as the initial guess. More details can be found in [229, 92].

3.4.3. Monotonic Basin Hopping. A monotonic basin hopping (MBH) algorithm for solving (3.13)–(3.14) is employed in [91]. Let $\mathscr{L}$ be the set of local optima of (3.3) and $\mathscr{N} : \mathbb{R}^3 \to \mathscr{P}(\mathbb{R}^3)$ (where $\mathscr{P}(S)$ denotes the power set of S) be some appropriate neighborhood structure. A partial order $\sqsupset$ on $\mathscr{L}$ is assumed to exist: $x \sqsupset y$ implies $y \in \mathscr{N}(x)$ and $f(x) > f(y)$. A *funnel* is a subset $\mathscr{F} \subseteq \mathscr{L}$ such that for each $x \in \mathscr{F}$ there exists a chain $x = x^0 \sqsupset x^1 \sqsupset \cdots \sqsupset x^t = \min \mathscr{F}$ (this situation is described in Figure 3.19). The MBH algorithm is as follows. Starting with a current solution $x \in \mathscr{F}$, sample a new point $x' \in \mathscr{N}(x)$ and use it as the starting point for a local NLP solver; repeating this sufficiently many times will yield the next optimum x^1 in the funnel. This is repeated until improvements are no longer possible. The MBH is also employed within a population-based metaheuristic called population basin hopping (PBH), which explores several funnels in parallel.

3.4.4. Alternating Projections Algorithm. The alternating projection algorithm (APA) [187] is an application of the more general successive projection methodology (SPM) [89, 225] to the iMDGP. The SPM takes a starting point and projects it alternately on the two convex sets, attempting to reach a point in their intersection (Figure 3.20).

In the APA, the starting point is a given predistance matrix $D = (\delta_{uv})$, i.e., an $n \times n$ symmetric matrix with nonnegative components and zero diagonal. D is

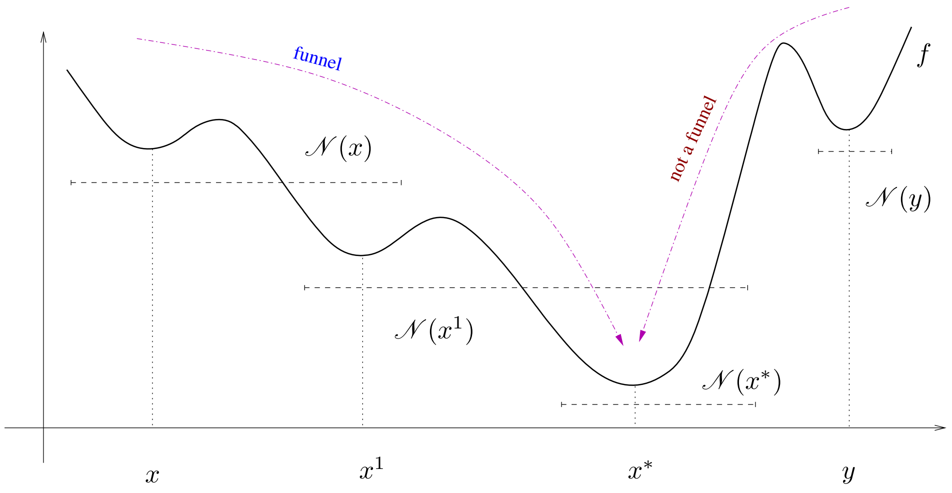


Fig. 3.19 *The dashed horizontal lines indicate the extent of the neighborhoods. The set* $\mathscr{F} = \{x, x^1, x^*\}$ *is a funnel, because* $x \sqsupset x^1 \sqsupset x^* = \min \mathscr{F}$. *The set* $\{x^*, y\}$ *is not a funnel, as* $x^* \notin \mathscr{N}(y)$.

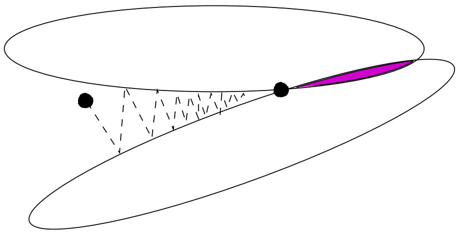


Fig. 3.20 *The SPM attempts to find a point in the intersection of two convex sets.*

generated randomly so that $d^L_{uv} \le \delta_{uv} \le d^U_{uv}$ for all $\{u, v\} \in E$, and $\delta_{uv} = 0$ otherwise. By Schoenberg's Theorem 2.2 and (2.4), if we let $P = I - \frac{1}{n}\mathbf{1}\mathbf{1}^\top$ and $A = -\frac{1}{2}PDP$, where I is the $n \times n$ identity matrix and $\mathbf{1}$ is the all-one n-vector, D is an EDM if and only if A is PSD. Notice that P is the orthogonal projection operator on the subspace $M = \{x \in \mathbb{R}^n \mid x^\top \mathbf{1} = 0\}$ of vectors orthogonal to $\mathbf{1}$, so D is an EDM if and only if D is negative semidefinite on M [82]. On the other hand, a necessary condition for any matrix to be an EDM is that it should have zero diagonal. This identifies the two convex sets on which the SPM is run: the set $\mathcal{P}$ of matrices which are negative semidefinite on M, and the set $\mathcal{Z}$ of zero-diagonal matrices. The projection operator for $\mathcal{P}$ is $Q(D) = PU\Lambda^- UP$, where $U\Lambda U$ is the spectral decomposition of D and Λ^- is the nonpositive part of Λ, and the projection operator for $\mathcal{Z}$ is $Q'(D) = D - \mathrm{diag}(D)$.

Although the convergence proofs for the SPM assume an infinite number of iterations in the worst case, empirical tests suggest that five iterations of the APA are enough to get satisfactory results. The APA was tested on the bovine pancreatic trypsin inhibitor protein (`q1q`), which has 588 atoms including side chains.

3.4.5. The GNOMAD Iterative Method. The GNOMAD algorithm [235] (see Algorithm 3) is a multilevel iterative method, which tries to arrange groups of atoms

Algorithm 3 GNOMAD

```
1: ${C_1, ..., C_ℓ}$ is a vertex cover for $V$
2: for $i \in \{1,\dots,\ell\}$ do
3:   while termination condition not met do
4:     determine an order $<$ on $C_i$
5:     for $v \in (C_i, <)$ do
6:       find search direction $\Delta_v$ for $x_v$ (obtained by solving an NLP locally)
7:       determine step $s_v$ minimizing constraint infeasibility
8:       $x_v \leftarrow x_v + s_v\Delta_v$
9:     end for
10:  end while
11: end for
```

at the highest level, then determines an appropriate order within each group using the contribution of each atom to the total error, then finally, at the lowest level, performs a set of atom moves within each group in the prescribed order. The method exploits several local NLP searches (in low dimension) at each iteration, as detailed in Algorithm 3. The constraints exploited in step 7 are mostly given by van der Waals distances [199], which are physically inviolable separation distances between atoms.

3.4.6. Stochastic Proximity Embedding Heuristic. The basic idea of the stochastic proximity embedding (SPE) [240] heuristic is as follows. All the atoms are initially placed randomly into a cube of a given size. Pairs of atoms in E are repeatedly and randomly selected; for each pair $\{u,v\}$, the algorithm checks satisfaction of the corresponding constraint in (3.10). If the constraint is violated, the positions of the two atoms are changed according to explicit formulae in order to improve the current embedding (two examples are shown in Figure 3.21).

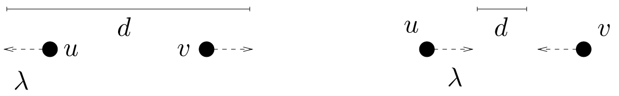


Fig. 3.21 *Local changes to positions according to discrepancy with respect to the corresponding distance.*

The SPE heuristic is shown in Algorithm 4. SPE offers no guarantee to obtain a solution satisfying all constraints in (3.10); however, the "success stories" reported in [102] seem to indicate this as a valid methodology.

Algorithm 4 SPE Heuristic

```
while termination condition not met do
  Pick $\{u,v\} \in E$ ($\|x_u - x_v\| \notin d_{uv}$)
  Update $\lambda$
  Let $x_u \leftarrow x_u + \lambda(x_u - x_v)$
  Let $x_v \leftarrow x_v + \lambda(x_v - x_u)$.
end while
```

3.5. NMR Data. NMR experiments are performed in order to estimate distances between some pairs of atoms forming a given molecule [238]. In solution, the molecule is subjected to a strong external magnetic field, which induces the alignment of the spin magnetic moment of the observed nuclei. The analysis of this process allows the identification of a subset of distances for certain pairs of atoms, mostly those involving hydrogens, as explained in the introduction. In proteins, nuclei of carbons and nitrogens are also sometimes considered.

It is important to remark that some NMR signals may be imprecise, because it is not always possible to distinguish between the atoms of the molecule. We can have this situation, for example, in proteins containing amino acids such as valines and leucines. In such a case, the distance *restraints* (a term used in proteomics meaning "constraints") involve a "pseudoatom" that is placed halfway between the two undistinguished atoms [239]. Once the upper bound for the distance has been chosen when considering the pseudoatom, its value is successively increased in order to obtain an upper bound for the real atoms.

There are also other potential sources of errors that can affect NMR data. If the molecule is not stable in solution, its conformation may change during the NMR experiments, and therefore the information obtained could be inconsistent. Depending on the machine and on the magnetic field, some noise may spoil the quality of the NMR signals from which the intervals are derived. Moreover, due to a phenomenon called "spin diffusion," the NMR signals related to two atoms could also be influenced by neighboring atoms [42]. Thus, the distances provided by NMR are imprecise, not only due to noise, but also due to dynamics of the molecule in solution.

Fortunately, for molecules having a known chemical composition such as proteins, there are a priori known distances that can be considered together with those obtained through NMR experiments. If two atoms are chemically bonded, their relative distance is known; this distance is subject to small variations, but it can still be considered as fixed in several applications (see the rigid geometry hypothesis; section 3.3.1). Moreover, the distance between two atoms bonded to a common atom can also be estimated, because they generally form a specific angle that depends upon the kind of atoms involved. Such distances can therefore be considered precise and provide valuable information for the solution of DGPs (this follows because protein graphs are molecular; see section 3.3.1).

As explained in the introduction, the output of an NMR experiment on a given molecule can be taken to consist of a set of triplets $(\{a, b\}, d, q)$, meaning that q pairs of atoms of type a, b were observed to have distance d [17]. It turns out that NMR data can be further manipulated so that it yields a list of pairs $\{u, v\}$ of atoms with a corresponding nonnegative distance d_{uv}. Unfortunately, this manipulation is rather error-prone, resulting in interval-type errors, so that the exact interatomic distances d_{uv} are in fact contained in given intervals $[d^L_{uv}, d^U_{uv}]$ [17]. For practical reasons, NMR experiments are most often performed on hydrogen atoms [17] (although sometimes carbons and nitrogens are also considered). Other known molecular information includes [199, 64] the number and type of atoms in the molecules, all the covalent bonds with corresponding Euclidean distances, and all distances between atoms separated by exactly two covalent bonds.

3.5.1. Virtual Backbones of Hydrogens. In order to address the NMR limitation concerning the lack of data reliability for interatomic distances of non-hydrogen atoms, we define atomic orders limited to hydrogens and disregard the natural backbone order during discretization. Even though we showed that this approach works on a set of artificially generated instances [135], we remarked on its limitations when we tried to apply it to real NMR data. These limitations have been addressed by using reorders (see section 3.5.2).

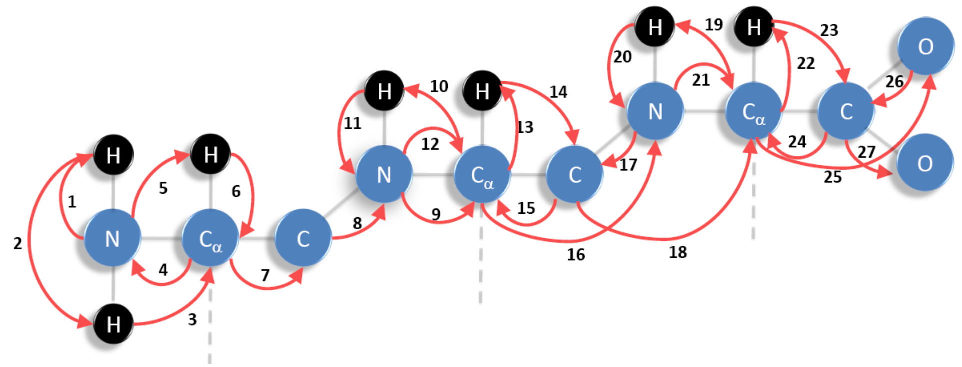


Fig. 3.22 *The order used for discretizing MDGPs with interval data.*

3.5.2. Re-orders and Interval Discretization. In [129] we defined an atomic ordering that ensures that every atom of rank > 3 is adjacent to its three immediate predecessors by means of either real-valued distances d or interval distances $\bar{d}$ that arise from geometrical considerations rather than NMR experiments. Specifically, with reference to Figure 3.12, the distance $d_{i-3,i}$ belongs to a range determined by the uncertainty associated with the torsion angle ϕ_i.

We exploited three protein features to this aim: (i) using hydrogen atoms off the main backbone whenever appropriate; (ii) using the same atom more than once; (iii) remarking that some interval distances $\bar{d}$ can be replaced with finite (small) sets D of real-valued distances. Considering these properties, we were able to define a new atomic ordering for which v can be placed in a finite number of positions in the set $\{0, 1, 2, 2|D|\}$, consistent with the known positions of the three immediate predecessors of v. Feature (i) allows us to exploit atoms for which NMR data are available. Feature (ii) allows us to exploit more than just two bond lengths on atoms with valence > 2, such as carbons and nitrogens, by defining an order that includes the atom more than once; these orders are called *re-orders*, which is short for "repetition orders" [129]. Feature (iii) is related to some interval distances whose interval can be exactly computed (rather than estimated via NMR experiments) by relating it to the torsion angles: a torsion angle of 0 gives the lower bound, and a torsion angle of π radians gives the upper bound. Figure 3.22 shows a re-order for a small protein backbone containing three amino acids.

Re-orders $(v_1, \ldots, v_p)$ deserve a further remark. We stressed the importance of strict simplex inequalities in section 3.3.2, but requiring that $v_i = v_j$ for some $i \neq j$ introduces a zero distance $d(v_i, v_j) = 0$. If this distance is ever used inappropriately, we might end up with a triangle with a side of zero length, which might in turn imply an infinity of possible positions for the next atom. We recall that, for any $v > K$, strict simplex inequalities $\Delta_{K-1}(U_v) > 0$ in dimension $K-1$ are *necessary* for discretization, as they avoid unwanted affine dependencies (see, e.g., Figure 3.7). By contrast, if $\Delta_K(U_v \cup \{v\}) > 0$ hold, then we have a K-simplex with nonzero volume, which has two possible orientations in $\mathbb{R}^K$; in other words, the two possible positions for x_v are distinct. If $\Delta_K(U_v \cup \{v\}) = 0$, however, then there is just one possible position for x_v. Thus, to preserve discretization, zero distances can never occur between pairs v_i, v_j fewer than K atoms apart, but they may occur for $|i - j| = K$; in this case there is no branching at level $\max(i, j)$.

Re-orders make it possible to consider only non-NMR distances for discretization. More precisely, over each set of three adjacent predecessors, only one is related by an interval distance; this interval, however, is not due to experimental imprecision in NMR, but rather to a molecular property of torsion angles; in particular, we can compute lower and upper bounds to these intervals, as mentioned above [129]. The number of discretization steps of these intervals is usually heuristically set so that it becomes smaller than the resolution scope of NMR experimental techniques [180]. We refer to such intervals as *discretizable*. A new, very recent development in this direction replaces the discretization of interval distances with Clifford algebra computations, which reduce the loss of the precision resulting from discretizing intervals [3].

3.5.3. Discrete Search with Interval Distances. The *interval* BP (iBP) [129] is an extension of the BP algorithm that is able to manage interval data. The main idea is to replace, in the sphere intersections necessary for computing candidate atomic positions, a sphere by a *spherical shell*. Given a center $c \in \mathbb{R}^K$ and an interval $d = [d^L, d^U]$, the spherical shell centered at c w.r.t. d is $S^{K-1}(c, d^U) \smallsetminus S^{K-1}(c, d^L)$. With $K = 3$, the intersection of two spheres and a spherical shell gives, with probability 1, two disjoint curves in 3D space (see Figure 3.23). The discretization is still possible if some sample distances are chosen from the interval associated with the curves [180].

Similar to the basic BP algorithm, the two main components of iBP are the branching and the pruning phases. In the branching phase, we can have three different situations, depending on the distance $d(i-3, i)$ (see Figure 3.22). If $d(i-3, i) = 0$, the current atom i has already appeared previously in the order, which means that the only feasible position for i is the same as $i - 3$. If $d(i-3, i)$ is a precise distance, then three spheres are intersected, and only two positions are found with probability 1. Finally, if $d(i-3, i)$ is a discretizable interval $[d^L_{i-3,i}, d^U_{i-3,i}]$, as specified in section 3.5.2, we choose D values from the interval. This yields a choice of $2D$ candidate atomic solutions for i.

If the discretization order in Figure 3.22 is employed for solving NMR instances, (precise) distances derived from the chemical composition of proteins are used for performing the discretization, whereas interval distances from NMR experiments are used for pruning purposes only. The consequent search tree is no longer binary: every time a discretizable interval is used for branching, the current node has at most $2D$

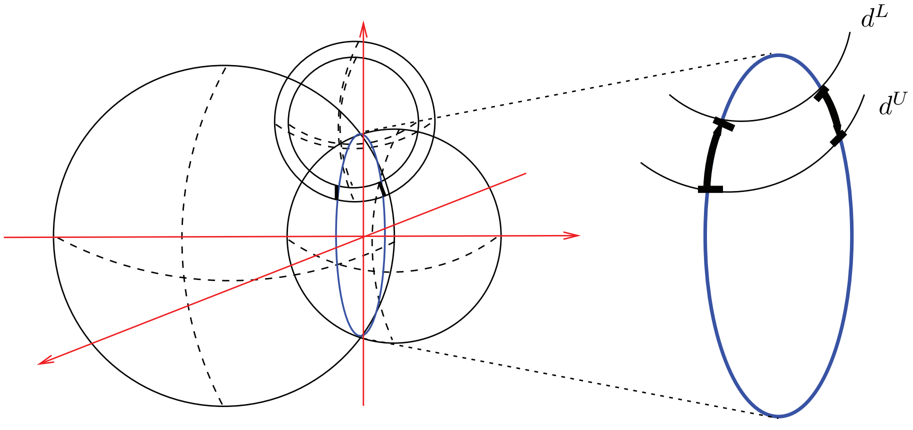


Fig. 3.23 *The intersection of two spheres with a spherical shell.*

subnodes. The advantage is that the generation of the search tree is not affected by experimental errors caused by the NMR machinery.

In order to discretize instances related to entire protein conformations, it is necessary to identify a discretization order for all side chains for the 20 amino acids that can be involved in the protein synthesis. This is a nontrivial task, because side chains have more complex structures with respect to the part that is common to each amino acid, and they may contain many atoms. However, side chains can be of fundamental importance in the identification of protein conformations, because many distances obtained by NMR experiments may regard hydrogen atoms contained in side chains. First efforts toward extending the BP algorithm so that it can calculate the whole 3D structure of a protein, including its side chains, can be found in [193].

4. Engineering Applications.

In this section, we discuss other well-known applications of DG: wireless networks, statics, dimensionality reduction, and robotics. In wireless networks, mobile sensors can usually estimate their pairwise distance by measuring how much battery they use in order to communicate. These distances are then used to find the positions of each sensor (see section 4.1). Statics is the field of study of the equilibrium of rigid structures (mostly man-made, such as buildings or bridges) under the action of external forces. A well-known model for such structures is the *bar-and-joint framework*, which is essentially a weighted graph. The main problem is that of deciding whether a given graph, with a given distance function on the edges, is rigid or not. An associated problem is that of deciding whether a given graph models a rigid structure independently of the distance function (see section 4.2).

4.1. Wireless Sensor Networks.

The position of wireless mobile sensors (e.g., smartphones, identification badges, and so on) is, by its very definition, local to the sensor carrier at any given time. Notwithstanding, in order to be able to properly route communication signals, the network routers must be aware of the sensor positions, as they adapt routes, frequencies, and network ID data accordingly. The information available to solve this problem is given by the fact that mobile sensors are always aware of their neighboring peers (to within a certain radius r of their positions, which we shall assume constant) as well as the amount of battery charge they use in order to communicate with other sensors in their neighborhood. It turns out that this quantity is strongly correlated with the Euclidean distance between the communicating sensors [197]. Moreover, certain network elements, such as routers and wireless repeaters, are fixed, hence their positions are known (such elements are called *anchors* or *beacons*). The problem of determining the sensor positions using these data was deemed to be important at the very inception of wireless networks [232, 77]. There are several good reasons why global positioning system (GPS) enabled devices may not be a valid alternative: they are usually too large, they consume too much power, and they need a line of sight with the satellites, which may not always be the case in practice (think, for example, of localizing sensors within a building) [197]. This problem is formalized as the WSNL (see item 14 in the list of section 1.2).

In wireless sensor networks, $K \in \{2, 3\}$. The 3D case might occur when a single network is spread over several floors of a building, or whenever a mobile battlefield network is parachuted over a mountainous region. Moreover, because the realization represents a practically existing network, an important question is to determine the amount of data that suffices for the graph to have a *unique* realization in $\mathbb{R}^K$. This marks a striking difference with the application of DG techniques to molecular conformation, where molecules can exist in different isomers.

The earliest connections of WSNL with DG are an SDP formulation [63] for a relaxation of the problem where the Euclidean distance between two sensors is *at most* the corresponding edge weight, and an in-depth theoretical study of the WSNL from the point of view of graph rigidity [73] (see section 4.2).

4.1.1. Unique Realizability.

In [73, 8], the WSNL is defined to be *solvable* if the given graph has a unique valid realization, a notion also known as global rigidity. A graph is *globally rigid* if it has a generic realization x and for all other realizations x', x is congruent to x'. For example, if a graph has a K-trilateration order, then it is globally rigid: comparing with DVOP orders, where each vertex is adjacent to K predecessors, the additional adjacency makes it possible to identify *at most one* position in $\mathbb{R}^K$ where the next vertex in the order will be placed, if a position for all predecessors is already known. Any graph possessing a K-trilateration order is called a *K-trilateration graph*. Such graphs are globally rigid and can be realized in polynomial time by simply remarking that the BP algorithm would never branch on such instances.

A graph $G = (V, E)$ is *redundantly rigid* if $(V, E \smallsetminus \{e\})$ is rigid for all $e \in E$. It was shown in [103, 45] that G is globally rigid for $K = 2$ if and only if either G is the 2-clique or the 3-clique, or G is 3-connected and redundantly rigid. Hendrickson conjectured in [94] that these conditions would be sufficient for any value of K, but this was disproved by Connelly [44]. He also proved, in [45], that if a generic framework (G, x) has a self-stress (see section 4.2.1) $\omega : E \to \mathbb{R}$ such that the $n \times n$ *stress matrix*, with (u, v)th entry $(-\omega_{uv})$ if $\{u, v\} \in E$, $\sum_{t\in\delta(v)} \omega_{ut}$ if $u = v$, and 0 otherwise, has rank $n - K - 1$, then (G, x) is globally rigid in any dimension K [45]. This condition was also proved to be necessary in [83]. Some graph properties ensuring global rigidity for $K \in \{2, 3\}$ are given in [4]. A related problem, that of choosing a given subset of vertices to take the role of anchors such that the resulting sensor network is uniquely localizable (see section 3.2.4), is discussed in [76]. Several results on global rigidity (with particular attention to the case $K = 2$) are surveyed in [105]. In particular, it is shown in [105, Thm. 11.3] that Henneberg type II steps (replace an edge $\{u, w\}$ by two edges $\{u, v\}$ and $\{v, w\}$, where v is a new vertex, then add new edges from v to $K - 1$ other vertices different from u, w) are related to global rigidity in a similar way that Henneberg type I steps (see section 4.2.3) are related to rigidity: if a globally rigid graph H is derived from a graph G with at least $K+2$ vertices using a Henneberg type II step in $\mathbb{R}^K$, then G is also globally rigid.

There is an interesting variant of unique localizability which yields a subclass of DGP instances that can be realized in polynomial time, up to a given $\varepsilon > 0$ tolerance. Recall that the DGP is strongly NP-hard [198] in general. Moreover, it remains NP-hard even when the input is a unit disk graph (see section 4.1.4) [8] and there exists no randomized efficient algorithm even when it is known that the input graph is globally rigid [9]. The problem becomes tractable under the equivalent assumptions of K-unique localizability (a sort of unique localizability for fixed K) [157] and universal rigidity [245] (see section 3.2.4). Specifically, a graph is *K-uniquely localizable* if (i) it has a unique realization $x : V \to \mathbb{R}^K$; (ii) it has a unique realization $y^\ell : V \to \mathbb{R}^\ell$ for all $\ell > K$; and (iii) for all $v \in V, \ell > K$, we have $y^\ell_v = (x_v, \mathbf{0})$, where $\mathbf{0}$ is the zero vector in $\mathbb{R}^{\ell-K}$. Anchors play a crucial role in ensuring that the graph should be globally rigid in $\mathbb{R}^K$: the subgraph induced by the anchors should yield a generic globally rigid framework in $\mathbb{R}^K$, and thus the set of anchors must have at least $K+1$ elements. Under these assumptions, a polynomial algorithm (exploiting the SDP formulation and its dual) for realizing K-uniquely localizable graphs, up to a given tolerance, was described in [157].

4.1.2. Semidefinite Programming. Most of the recent methods addressing the WNSL make use of SDP techniques. This is understandable in view of the relationship between DG and SDP via Theorem 2.2, and because the PSDMCP is actually a special case of the general SDP feasibility problem (see section 2.6.1). We also mention that most SDP methods can target DGP variants where the edge weight d maps into bounded intervals, not only reals, and are therefore suitable for applications where distance measurements are not precise.

We believe [106] is the first reference in the literature that proposes an SDP-based method for solving MCPs (specifically, the PSDMCP). In [2], the same approach is adapted to a slightly different EDMCP formulation. Instead of a partial matrix, an $n \times n$ *predistance matrix* A is given, i.e., a matrix with zero diagonal and nonnegative off-diagonal elements. We look for an $n \times n$ EDM D that minimizes $\|H \circ (A - D)\|_F$, where H is a given matrix of weights, $\circ$ is the Hadamard product, and $\|\cdot\|_F$ is the Frobenius norm ($\|Q\|_F = \sqrt{\sum_{i,j \le n} q_{ij}^2}$). An optional linear constraint can be used to fix some of the values of D. A reformulation of the constraint "D is an EDM" to $X \succeq 0$ is derived by means of the statement that D is an EDM if and only if D is negative semidefinite on the orthogonal complement of the all-one vector [85, 187] (see section 3.4.4). In turn, this is related to Theorem 2.2.

In [32, 63], interestingly, the connection with SDP is *not* given by Theorem 2.2, but rather because the WSNL variants mentioned in these papers make use of convex norm constraints which are reformulated using linear matrix inequalities (LMIs). For example, if there is a direct communication link between two nodes $u, v \in V$, then $\|x_u - x_v\| \le r$, where r is a scalar threshold given by the maximum communication range, can be reformulated to the LMI

$$\begin{pmatrix} rI_2 & x_u - x_v \\ (x_u - x_v)^\top & r \end{pmatrix} \succeq 0,$$

where I_2 is the 2×2 identity matrix.

Biswas and Ye proposed in [25] an SDP formulation of the WSNL problem which then gave rise to a series of papers [22, 26, 23, 21, 24] focusing on algorithmic exploitations of their formulation. In the spirit of [141], this can be derived from the "classic" WSNL feasibility formulation below by means of a sequence of basic reformulations

$$\begin{aligned} \forall \{u, v\} \in E \quad & (\|x_u - x_v\|_2 = d_{uv}), \\ \forall u \in A, v \notin A \quad & (\{u, v\} \in E \to \|a_u - x_v\| = d_{uv}), \end{aligned}$$

where $A \subseteq V$ is the set of anchors whose positions $\{a_u \mid u \in A\} \subseteq \mathbb{R}^K$ are known a priori. Let X be the $K \times n$ decision variable matrix whose vth column is x_v. The authors remark that

- for all $u < v \in V$, $\|x_u - x_v\|^2 = {e_{uv}}^\top X^\top X e_{uv}$, where $e_{uv} = 1$ at component u, -1 at component v, and 0 elsewhere;
- for all $u \in A, v \in V$, $\|a_u - x_v\|^2 = (a_u; e_v)^\top [I_K; X]^\top [I_K; X](a_u; e_v)$, where $(a_u; e_v)$ is the column $(K + n)$ vector consisting of a_u on top of e_v, with $e_V = 1$ at component v and 0 elsewhere, and $[I_K; X]$ is the $K \times (K + n)$ matrix consisting of I_K followed by X;
- $[I_K; X]^\top [I_K; X] = \left(\begin{smallmatrix} I_K & X \\ X^\top & X^\top X \end{smallmatrix}\right)$, a $(K + n) \times (K + n)$ matrix denoted by Z;
- the scalar products of decision variable vectors in $X^\top X$ (rows of $X^\top$ by columns of X) can be linearized, replacing each $x_u x_v$ by y_{uv}, which results in substituting $X^\top X$ by an $n \times n$ matrix $Y = (y_{uv})$ such that $Y = X^\top X$.

This yields the following formulation of the WSNL:

$$\begin{aligned} \forall \{u,v\} \in E \quad & {e_{uv}}^\top Y e_{uv} = d_{uv}^2, \\ \forall u \in A, v \notin A \quad & (\{u,v\} \in E \to (a_u; e_v)^\top Z (a_u; e_v) = d_{uv}^2), \\ & Y = X^\top X. \end{aligned}$$

The SDP relaxation of the constraint $Y = X^\top X$, which is equivalent to requiring that Y has rank K, consists in replacing it with $Y - X^\top X \succeq 0$, which is equivalent to $Z \succeq 0$. The whole SDP can be written in terms of the indeterminate matrix Z as follows, using MATLAB-like notation to indicate submatrices:

$$Z_{1:K,1:K} = I_K, \tag{4.1}$$

$$\forall u, v \in V \smallsetminus A \quad (\{u,v\} \in E \to (\mathbf{0}; e_{uv})(\mathbf{0}; e_{uv})^\top \bullet Z = d_{uv}^2), \tag{4.2}$$

$$\forall u \in A, v \in V \smallsetminus A \quad (\{u,v\} \in E \to (a_u; e_v)(a_u; e_v)^\top \bullet Z = d_{uv}^2), \tag{4.3}$$

$$Z \succeq 0, \tag{4.4}$$

where $\bullet$ is the Frobenius product. This formulation was exploited algorithmically in a number of ways. As mentioned in sections 4.1.1 and 3.2.4, solving the SDP formulation (4.1)–(4.4) yields a polynomial-time algorithm for the DGP on uniquely localizable graphs (see section 3.2.4). The proof uses the dual SDP formulation to (4.1)–(4.4) in order to show that the interior point method for SDP yields an exact solution [157, Cor. 1] and the fact that the SDP solution on uniquely localizable graphs has rank K [157, Thm. 2]. Another interesting research direction employing (4.1)–(4.4) is the edge-based SDP (ESDP) relaxation [231]: this consists in relaxing (4.4) to only hold on principal submatrices of Z indexed by A. To address the fact that SDP and ESDP formulations are very sensitive to noisy data, a robust version of the ESDP relaxation was discussed in [183] (see section 3.2.4).

Among the methods based on formulation (4.1)–(4.4), those in [26, 24] are particularly interesting. They address the limited scaling capabilities of SDP solution techniques by identifying vertex clusters where embedding is easier, and then match those embeddings in space using a modified SDP formulation. The vertex clusters cover V in such a way that neighboring clusters share some vertices (these are used to "stitch together" the embeddings restricted to each cluster). The clustering technique is based on permuting columns of the distance matrix (d_{ij}) to try to pool the nonzeros along the main diagonal. The partial embeddings for each cluster are computed by first solving an SDP relaxation of the quadratic system in equation (3.10) restricted to edges in the cluster, and then applying a local NLP optimization algorithm that uses the optimal SDP solution as a starting point. When the distances have errors, there may not exist any valid embedding satisfying all the distance constraints. In this case, it is likely that the SDP approach (which relaxes these constraints anyway) will end up yielding an embedding x' that is valid in a higher-dimensional space $\mathbb{R}^{K'}$, where $K' > K$. In such cases, x' is projected onto a realization x in $\mathbb{R}^K$. Such projected embeddings usually exhibit clusters of close vertices (none of which satisfies the corresponding distance constraints), due to correct distances in the higher-dimensional space being "squeezed" to their orthogonal projection into the lower-dimensional space. In order to counter this type of behavior, a regularization objective function $\max \sum_{i,j \in V} ||x_i - x_j||^2$ is added to the feasibility SDP.

In [111, 110], Krislock and Wolkowicz also exploit the SDP formulations of [2] together with vertex clustering techniques in order to improve the scaling abilities of

SDP solution methods (see also section 3.2.4). Their facial reduction algorithm identifies cliques in the input graph G and iteratively expands them using a K-trilateration order (see section 3.3). Rather than "stitching together" pieces, as in [24], the theory of facial reduction methods works by considering the SDP relaxation of the whole problem and showing how it can be simplified in the presence of one or more cliques (be they intersecting or disjoint). The computational results of [111] show that the facial reduction algorithm scales extremely well (graphs up to 100,000 vertices were embedded in $\mathbb{R}^2$). A comparison with the BP algorithm (see section 3.3.5) appears in [127, Table 6]. The BP algorithm is slightly less accurate (the most common LDE values are $O(10^{-12})$ for BP and $O(10^{-13})$ for facial reduction) but much faster (BP takes between 1% and 10% of the time taken by facial reduction).

4.1.3. Second-Order Cone Programming. A second-order cone programming (SOCP) relaxation of the WSNL was discussed in [226]. The NLP formulation (3.1) is first reformulated as follows:

$$
\begin{array}{rrcl}
 & \min \displaystyle\sum_{\{u,v\}\in E} z_{uv}, & & \\
\forall\{u,v\}\in E & x_u - x_v & = & w_{uv}, \\
\forall\{u,v\}\in E & y_{uv} - z_{uv} & = & d_{uv}^2, \\
\forall\{u,v\}\in E & \|w_{uv}\|^2 & = & y_{uv}, \\
 & u & \geq & 0.
\end{array}
\tag{4.5}
$$

Next, the constraint $\|w_{uv}\|^2 = y_{uv}$ is relaxed to $\|w_{uv}\|^2 \leq y_{uv}$. The SOCP relaxation is weaker than the SDP one (4.1)–(4.4), but scales much better (4000 vs. 500 vertices). It was abandoned by Tseng in favor of the ESDP [183], which is stronger than the SOCP relaxation but scales similarly.

4.1.4. Unit Disk Graphs. Unit disk graphs are intersection graphs of equal circles in the plane, i.e., vertices are the circle centers, and there is an edge between two vertices u, v if their Euclidean distance is at most twice the radius. Unit disk graphs provide a good model for broadcast networks, with each center representing a mobile transmitter/receiver and the radius representing the range. In [41], it is shown that several standard NP-complete graph problems are just as difficult on unit disk graphs as on general graphs, but that the maximum clique problem is polynomial on unit disk graphs (the problem is reduced to finding a maximum independent set in a bipartite graph). In [33], it is shown that even recognizing whether a graph is a unit disk graph is NP-hard. A slightly different version of the problem, consisting in determining whether a given weighted graph can be realized in $\mathbb{R}^2$ as a unit disk graph of given radius, is also NP-hard [8]. From the point of view of DG, it is interesting to remark that the DGP, restricted to sufficiently dense unit disk graphs and provided a partial realization is known for a subset of at least $K+1$ vertices, can be solved in polynomial time [157]. If the graph is sparse, however, the DGP is still NP-hard [9].

The study of unit disk graphs also arises when packing equal spheres in a subset of Euclidean space [46]; the contact graphs of the sphere configuration are unit disk graphs.

4.2. Statics. Statics is the study of forces acting on physical systems in static equilibrium. This means that the barycenter of the system undergoes no linear acceleration (we actually assume the barycenter to have zero velocity), and that the system does not rotate. Geometrically, with respect to a frame of reference, the system undergoes no translations and no rotations. The physical systems we are concerned with are bar-and-joint structures, i.e., 3D embodiments of graph frameworks (G, x), where G is a simple weighted undirected graph and x is a valid realization thereof: joints are vertices, bars are edges, and bar lengths are edge weights. The placement of the structure in physical space provides a valid realization of the underlying graph. Because we suppose the structures to be stiff, they cannot undergo reflections either. In short, the equivalence class of a rigid graph framework modulo congruences is a good representation of a structure in static equilibrium. Naturally, the supporting bar-and-joint structures of man-made constructions such as houses, buildings, skyscrapers, bridges, and so on must always be in static equilibrium, for otherwise the construction would collapse.

Statics has been a field of study ever since humans have had roofs over their heads. The main challenge is the estimation of reaction forces that man-made structures have to provide in order to remain in static equilibrium under the action of external forces. In 1725, Varignon published a textbook [227] that implemented ideas he had sketched in 1687 about the application of systems of forces to different points of static structures. By the mid-1800s there was both an algebraic and a graphical method for testing rigidity of structures. Because of the absence of computing machinery, the latter (called *graphical statics*) was preferred to the former [49, 196, 97]. Cremona proposed a graphical axiomatization of arithmetic operations in [50], whose purpose was probably to give an implied equivalence between the two methods. J. C. Maxwell worked on both methods, publishing his results in 1864: the graphical result in [159] and the algebraic result in [160].

The link between statics and DG is rigidity, which we have seen to be a fundamental idea in the conception of efficient and reliable mixed-combinatorial algorithms for the DGP and its variants. Furthermore, since statics is the most ancient application field related to DG, it contains many of its historical roots and seminal ideas (this is clear when looking at the drawings contained in the tables in the early books mentioned above). Accordingly, in this section we present a summary of rigidity in statics.

4.2.1. Infinitesimal Rigidity. Since statics is mainly concerned with the physical 3D world, we fix $K = 3$ for the rest of this section. Consider a function $F : V \to \mathbb{R}^3$ that assigns a force vector $F_v \in \mathbb{R}^3$ to each point $x_v \in \mathbb{R}^3$ of a framework (G, x). If the framework is to be stationary, the total force and torque acting on it must be null to prevent translations (assuming a zero initial velocity of the barycenter) and rotations. This can be written algebraically [191, 218] as

$$\sum_{v\in V} F_v = 0, \tag{4.6}$$

$$\forall i < j \le K \quad \sum_{v\in V} (F_{vi}x_{vj} - F_{vj}x_{vi}) = 0. \tag{4.7}$$

A force F satisfying (4.6)–(4.7) is called an *equilibrium force* (or, *equilibrium load*). Applied to bar-and-joint structures, equilibrium forces tend to compress or extend the bars without moving the joints in space. Since bars are assumed to be stiff (or, equivalently, the graph edge weights are given constants), the corresponding reaction forces at the endpoint of each bar should be equal in magnitude and opposite in sign. We can define these reaction forces by means of an edge weighting $\omega : E \to \mathbb{R}$ representing the amount of force in each bar per unit length (ω is negative for bar tensions and positive for bar compressions). Stiffness of the structure translates algebraically to a balance of equilibrium force and reaction:

$$\forall u \in V \quad F_u + \sum_{v \in N(u)} \omega_{uv}(x_u - x_v) = 0. \tag{4.8}$$

A vector $\omega \in \mathbb{R}^m$ satisfying (4.8) is called a *resolution*, or *resolving stress*, of the equilibrium force F [191]. If $F = 0$, then ω is a *self-stress*.

For the following, we introduce (squared) edge functions and displacements. The *edge function* of a framework (G, x) is a function $\phi : \mathbb{R}^{nK} \to \mathbb{R}^m$ given by $\phi(x) = (\|x_u - x_v\| \mid \{u, v\} \in E)$. We denote the *squared* edge function $(\|x_u - x_v\|^2 \mid \{u, v\} \in E)$ by ϕ^2. The *edge displacement* of a framework (G, x), with respect to a displacement $y : [0, 1] \to \mathbb{R}^{nk}$, is a continuous function $\mu : [0, 1] \to \mathbb{R}^m$ given by $\mu(t) = (\|y_u(t) - y_v(t)\| \mid \{u, v\} \in E)$. We denote the squared edge displacement $(\|y_u(t) - y_v(t)\|^2 \mid \{u, v\} \in E)$ by μ^2.

Equation (4.8) can also be written as

$$\frac{1}{2}(\mathsf{d}\phi^2)^\top \omega = -F, \tag{4.9}$$

where $\mathsf{d}\phi^2$ is the matrix whose $\{u, v\}$th row encodes the derivatives of the $\{u, v\}$th component of the squared edge function $\phi^2(x)$ with respect to each component x_{vi} of x. Observe that the $\{u, v\}$th row of this matrix contains only the six nonzero components $2(x_{ui} - x_{vi})$ and $2(x_{vi} - x_{ui})$ for $i \in \{1, 2, 3\}$ (see [191, p. 13]). If we now consider (4.9) applied to a displacement y of x, differentiate it with respect to t and evaluate it at $t = 0$, we obtain the linear system $\omega A = 0$, where $A = \frac{1}{2}\mathsf{d}\phi^2$, i.e., the homogeneous version of (4.9).

Consider now a squared edge displacement $\mu^2(t)$ with respect to a flexing y of the framework (G, x). By definition of flexing, we have $\mu^2(t) = (d^2_{uv} \mid \{u, v\} \in E)$ for all $t \in [0, 1]$. Differentiating with respect to t, we obtain the scalar product relation $2(y_u(t) - y_v(t)) \cdot (\frac{\mathsf{d}y_u(t)}{\mathsf{d}t} - \frac{\mathsf{d}y_v(t)}{\mathsf{d}t}) = 0$ (because the edge weights d_{uv} are constant w.r.t. t) for all $\{u, v\} \in E$. Evaluating the derivative at $t = 0$ yields

$$\forall \{u, v\} \in E \quad (x_u - x_v) \cdot (\alpha_u - \alpha_v) = 0, \tag{4.10}$$

where $\alpha : V \to \mathbb{R}^3$ is a map that assigns initial velocities $\alpha_v = \frac{\mathsf{d}x_u}{\mathsf{d}t}|_0$ to each $v \in V$. We note that the system (4.10) can be written as $A\alpha = 0$ [81, Thm. 3.9]. We therefore have the dual relationship $\omega A = 0 = A\alpha$ between α and ω.

By definition, (G, x) is infinitesimally rigid if α only encodes rotations and translations. The above discussion should give an intuition as to why this is equivalent to stating that every equilibrium force has a resolution (see [81, 191, 218] for a full description). Indeed, infinitesimal rigidity was defined in this dual way by Whiteley [233] (who called it *static rigidity*). The matrix A above is called the *rigidity matrix* of the framework (G, x). Notice that when a valid realization x is known for G, then even those distances for $\{u, v\} \notin E$ can be computed for G: when the rows of A are indexed by all unordered pairs $\{u, v\}$, we call A the *complete rigidity matrix* of (G, x).

Infinitesimal rigidity is a stricter notion than rigidity: all infinitesimally rigid frameworks are also rigid [81, Thm. 4.1]. Counterexamples to the converse of this statement, i.e., rigid frameworks which are infinitesimally flexible, usually turn out to have some kind of degeneracy: a flat triangle, for example, is rigid but infinitesimally flexible [191, Ex. 4.2]. In general, infinitesimally rigid frameworks in $\mathbb{R}^K$ (for some integer $K > 0$) might fail to be infinitesimally rigid in higher-dimensional spaces [201].

4.2.2. Graph Rigidity. An important practical question to be asked about rigidity is whether certain graphs give rise to infinitesimally rigid frameworks just because of their graph topology, independent of their edge weights. Bar-and-joint frameworks derived from such graphs are extremely useful in architecture and construction engineering. An important concept in answering this question is that of genericity: a realization is *generic* if all its vertex coordinates are algebraically independent over $\mathbb{Q}$. Because the algebraic numbers have Lebesgue measure zero in the real numbers, this means that the set of nongeneric realizations has Lebesgue measure zero in the set of all realizations.

Rigidity and infinitesimal rigidity are defined as properties of frameworks, rather than of graphs. It turns out, however, that if a graph possesses a single generic rigid framework, then all its generic frameworks are rigid [6, Cor. 2]. This also holds for infinitesimal rigidity [7]. Moreover, rigidity and infinitesimal rigidity are the same notion over the set of all generic frameworks [7, sect. 3]. By genericity, this implies that in almost all cases it makes sense to speak of a "rigid graph" (rather than a rigid framework). The Graph Rigidity Problem asks, given a simple undirected graph G, whether it is generically rigid. Notice that the input, in this case, does not involve edge weights. For example, any graph is almost always flexible for large enough values of K unless it is a clique [6, Cor. 4].

We remark as an aside that, although genericity is required for laying the theoretical foundations of graph rigidity (see the proof of [81, Thm. 6.1]), in practice it is too strong. For an edge weighting to be algebraically independent over $\mathbb{Q}$, at most one edge weight can be rational (or even algebraic). Since computers are usually programmed to only represent rational (or at best algebraic) numbers, no generic realization can be treated exactly in any practical algorithmic implementation. The conceptual requirement that genericity is really meant to convey is that an infinitesimally rigid generic realization will stay rigid even though the edge weighting is perturbed slightly [201]. The definition given by Graver in [87] is more explicit in this sense: a realization is generic if all the nontrivial minors of the complete rigidity matrix have nonzero value. Specifically, notice that the polynomials induced by each minor are algebraic relations between the values of the components of each vector in the realization. Naturally, asking for full algebraic independence with respect to any polynomial in $\mathbb{Q}$ guarantees Graver's definition, but in fact, as Graver points out [88], it is sufficient to enforce algebraic independence with respect to the system of polynomials induced by the nontrivial minors of the rigidity matrix (see also section 3.3.2).

Generic graph rigidity can also be described using the graphic matroid $M(G)$ on G: a set of edges is independent if it does not contain simple cycles. The closure of an edge subset $F \subseteq E$ contains F and all edges that form simple cycles with edges of F. We call the edge set F *rigid* if its closure is the clique on the vertices incident on F. A graphical matroid $M(G)$ is an *abstract rigidity matroid* if it satisfies two requirements: (i) if two edge sets are incident to fewer than K common vertices, the closure of their union should be the union of their closures; and (ii) if two edge sets are incident to at least K common vertices, their union should be a rigid edge set [201]. Condition (i) loosely says that if the two edge sets are not "connected enough," then their union should give rise to flexible frameworks in $\mathbb{R}^K$, as the common vertices can be used as a "hinge" in $\mathbb{R}^K$ around which the two edge sets can rotate. Condition (ii) says that when no such hinges can be found, the union of the two edge sets gives rise to rigid graphs. If the only resolution to the zero equilibrium force is the zero vector, then the complete rigidity matrix has maximum rank (i.e., it has the maximum possible rank over all embeddings in $\mathbb{R}^{nK}$), and its rows naturally induce a matroid on the complete set of edges $\{\{u,v\} \mid u \neq v \in V\}$, called the *rigidity matroid* of the framework (G,x). It was shown in [87] that if x is generic, then the rigidity matroid is abstract.

4.2.3. Some Classes of Rigid Graphs. Euler conjectured in 1766 that all graphs given by the edge incidence of any triangulated polyhedral surface are rigid in $\mathbb{R}^3$. This conjecture was proven true for special cases but eventually disproved in general. Cauchy proved in 1813 that the conjecture holds for strictly convex polyhedra [37], Alexandrov proved in 1950 that it holds for convex polyhedra [1], and Gluck proved in 1975 that it also almost always holds for any triangulation of a topological sphere [81]. The general conjecture was finally disproved by Connelly in 1977 [43] using a skew octahedron.

This does not mean that there are no purely topological characterizations of rigid graphs. In 1911, Henneberg described two local procedures (or "steps") to construct new, larger rigid graphs from given rigid graphs [97] (if a given graph can be "deconstructed" by using the same procedures backwards, then the graph is rigid). The Henneberg type I step is as follows: start with a K-clique and add new vertices adjacent to at least K existing vertices. This defines a vertex order known as Henneberg type I order (see section 1.1.2). The Henneberg type II step is somewhat more involved, and we refer the interested reader to the extensive account of Henneberg and Henneberg-like procedures found in [218]. Here follows a philological note on Henneberg type I orders: although they are always referred to [97], they were actually first defined in a previous book by Henneberg [96, p. 267]. However, a picture with a Henneberg type I order in $\mathbb{R}^2$ appeared one year earlier, in 1885, in [195, Fig. 30, Pl. XV].

Limited to $\mathbb{R}^2$, a characterization of all rigid graphs G in $\mathbb{R}^2$ was described by Laman in 1970 [116]: $|E| = 2|V| - 3$ and for every subgraph (V', E') of G, $|E'| \leq 2|V'| - 3$. Equivalent but more easily verifiable conditions were proposed in [154, 188, 217]. Unluckily, such conditions do not hold for $\mathbb{R}^3$. For $K > 2$, no such complete characterization is known as yet; an account of the current conjectures can be found in [234, 104], and a heuristic method was introduced in [210].

4.3. Other Applications. DG is not limited to these applications. For example, an application to the synchronization of clocks from the measure of time offsets between pairs of clocks is discussed in [205]. This, incidentally, is the only engineering application of the DGP$_1$ we are aware of. The solution method involves maximizing a quadratic form subject to normalization constraints; this is relaxed to the maximization of the same quadratic form over a sphere, which is solved by the normalized eigenvector corresponding to the largest eigenvalue. Another application is the localization and control of fleets of autonomous underwater vehicles (AUVs) [11]. This is essentially a time-dependent DGP, as the delays in sound measurements provide an estimate of AUV-to-AUV distance and an indication of how it varies in time. We remark that GPS cannot be used underwater, so AUVs must resurface in order to determine their positions precisely. A third application to the quantitative analysis of music and rhythm is discussed in [59].

In the following sections, we briefly discuss two other important engineering applications of DG: dimensionality reduction by means of MDS and robotics, specifically inverse kinematic calculations. In the former, we aim to find a projection in the plane or the space which renders the graph visually as close as possible to the higher-dimensional picture (see section 4.3.1). In the latter, the main issue is to study how a robotic arm (or system of robotic arms) moves in space in order to perform certain tasks. Known distances include those from a joint to its neighboring joints. The main problem is that of assigning coordinate values to the position vector of the farthest joint (see section 4.3.2).

4.3.1. Dimensionality Reduction. Multidimensional Scaling (MDS) [31, 74, 68] is a dimensionality reduction tool in data analysis for representing measurements of dissimilarity among pairs of objects as distances between points in a low-dimensional space in such a way that the given dissimilarities are well approximated by the distances in that space. The choice of dimension is arbitrary, but the most frequently used dimensions are 2 and 3. MDS methods differ mainly according to the distance model, but the most usual model is the Euclidean one (in order to represent correlation measurements, a spherical model can also be used). Other distances, such as the ℓ_1 norm (also called the Manhattan distance) are used [5, 246]. The output of MDS provides graphical displays that allow decision makers to discover hidden structures in complex data sets.

MDS techniques have been used primarily in psychology. According to [109], the first important contributions to the theory of MDS are probably [213, 214], but they did not lead to practical methods. The contributions to the MDS methods are due to the Thurstonian approach, summarized in Chapter 11 of [223], although the real computational breakthrough was due to Shepard [202, 203, 204]. The next important step was taken by Kruskal [112, 113], who put Shepard's ideas into a formal setting in terms of optimization of a least squares function. Two important contributions following the Shepard and Kruskal works are [35] and [216].

Measurements of dissimilarity among n objects can be represented by a dissimilarity matrix $D = (d_{ij})$ [69]. The goal of MDS is to construct a set of points $x_i \in \mathbb{R}^K$ (for $i \leq n$ and K low, typically $K \in \{2, 3\}$) corresponding to those n objects such that pairwise distances approximate pairwise object dissimilarities (also see the APA method in section 3.4.4). MDS is complementary to principal component analysis (PCA) [107, 84] in the following sense. Given a set X of n points in $\mathbb{R}^H$ (with H "high"), PCA finds a K-dimensional subspace of $\mathbb{R}^H$ (with K "small") on which to project X in such a way that the variance of the projection is maximum (essentially, PCA attempts to avoid projections where two distant points are projected to be very close). PCA might lose some distance information in the projection, but the remaining information is not distorted. MDS identifies a K-dimensional subspace τ of $\mathbb{R}^H$ which minimizes the discrepancy between the original dissimilarity matrix D of the points in X and the dissimilarity matrix D' obtained by the projection on τ of the points in X. In other words, MDS attempts to represent all distance information in the projection, even if this might mean that the information is distorted.

MDS and PCA methods can be considered classical approaches to dimensionality reduction [138] in the domains of computational topology and geometry. However, the nonlinear structures presented in many complex data sets are invisible to MDS and PCA. Two different methods that are able to discover such nonlinearities are Isomap [219] and Laplacian eigenmaps [13]. The Isomap, motivated by manifold learning [155], tries to preserve the intrinsic geometry of the data by exploring geodesic distances, and the Laplacian eigenmaps, motivated by spectral graph theory [19], are based on the Laplacian matrix of the graph associated with the problem.

4.3.2. Robotics. Kinematics is the branch of mechanics concerning the geometric analysis of motion. The kinematic analysis of rigid bodies connected by flexible joints has many similarities with the geometric analysis of molecules, when the force effects are ignored.

The fundamental DG problem in robotics is known as the inverse kinematic problem (IKP—see item 15 in the list of section 1.2). Geometric constructive methods can be applied to solve the IKP [78], but algebraic techniques are more suitable to handle more general instances. Reviews of these techniques in the context of robotics and molecular conformation can be found, for example, in [179, 72, 189]. There are three main classes of methods in this category: those that use algebraic geometry, those based on continuation techniques, and those based on interval analysis.

In general, the solution of the IKP leads to a system of polynomial equations. The methods based on algebraic geometry reduce the polynomial system to a univariate polynomial, whose roots yield all solutions of the original system [158, 34]. Continuation methods, originally developed in [192], start with an initial system, whose solutions are known, and transform it into the system of interest, whose solutions are sought. In [224], using continuation methods, it was shown that the inverse kinematics of the general 6R manipulator (an arm system with six rotatable bonds with fixed lengths and angles [101]) has 16 solutions; more information can be found in [230].

One type of interval method applied to IKP is related to the interval version of the Newton method [186], and others are based on the iterative division of the distance space of the problem [144]. An interesting method in the latter class [220] essentially consists in solving a EDMCP whose entries are intervals (see sections 2.6 and 2.6.2). When the distance matrix is complete, the realization of the selected points can be carried out in polynomial time (see, e.g., [209, 65]). In order to determine the values for the unknown distances, in [185] a range is initially assigned to the unknowns and their bounds are reduced using a BP technique, which iteratively eliminates from the distance space entire regions which cannot contain any solution. This elimination is accomplished by applying conditions derived from the theory of DG. This BP technique is different from the BP algorithm discussed in sections 3.3 and 3.5, as the search space is continuous in the former and discrete in the latter. Another BP scheme for searching continuous space is described in [244]. This is applied to molecular conformational calculations related to computer-assisted drug design.

5. Conclusion.

Euclidean distance geometry is an extensive field with major biological, statistics, and engineering applications. The foundation of its theory was laid around a century ago by mathematicians such as Cayley, Menger, Schoenberg, Blumenthal, and Gödel. Recent extensions, targeting the inverse problem of determining a distance space given a partial distance function, contribute further mathematical as well as applied interest to the field. Because of the breadth and maturity of this field, our survey makes no claim to completeness; furthermore, we admit to a personal bias toward applications to molecular conformation. We have striven, however, to give the reader a sufficiently informative account of the most useful, interesting, and beautiful results of Euclidean DG. Obviously, scientific progress was made while this survey was being written. We refer the reader to [173] for very recent advances.

Acknowledgments. We are grateful to Andrea Cassioli, Jon Lee, Audrey Lee-St. John, Therese Malliavin, Benoît Masson, Michael Nilges, and Maxim Sviridenko for coauthoring some of the papers we wrote on different facets of this topic. We are grateful to Leandro Martinez for useful discussions, and to three anonymous referees for carefully checking and improving this paper. We also wish to thank Chiara Bellasio for providing inspiring dishes, a pleasant atmosphere, and lots of patience and support during many working sessions in Paris.

REFERENCES

[1] A. Alexandrov, *Convex Polyhedra*, Gosudarstv. Izdat. Tekhn.-Theor. Lit., Moscow, 1950 (in Russian).

[2] A. Alfakih, A. Khandani, and H. Wolkowicz, *Solving Euclidean distance matrix completion problems via semidefinite programming*, Comput. Optim. Appl., 12 (1999), pp. 13–30.

[3] R. Alves, A. Cassioli, A. Mucherino, C. Lavor, and L. Liberti, *Adaptive branching in iBP with Clifford algebra*, in Proceedings of the Workshop on Distance Geometry and Applications, A. Andrioni, C. Lavor, L. Liberti, A. Mucherino, N. Maculan, and R. Rodriguez, eds., Universidade Federal do Amazonas, Manaus, 2013, pp. 65–69.

[4] B. Anderson, P. Belhumeur, T. Eren, D. Goldenberg, S. Morse, W. Whiteley, and R. Yang, *Graphical properties of easily localizable sensor networks*, Wireless Networks, 15 (2009), pp. 177–191.

[5] P. Arabie, *Was Euclid an unnecessarily sophisticated psychologist?*, Psychometrika, 56 (1991), pp. 567–587.

[6] L. Asimow and B. Roth, *The rigidity of graphs*, Trans. Amer. Math. Soc., 245 (1978), pp. 279–289.

[7] L. Asimow and B. Roth, *The rigidity of graphs* II, J. Math. Anal. Appl., 68 (1979), pp. 171–190.

[8] J. Aspnes, T. Eren, D. Goldenberg, S. Morse, W. Whiteley, R. Yang, B. Anderson, and P. Belhumeur, *A theory of network localization*, IEEE Trans. Mobile Comput., 5 (2006), pp. 1663–1678.

[9] J. Aspnes, D. Goldenberg, and R. Yang, *On the computational complexity of sensor network localization*, in Algorithmic Aspects of Wireless Sensor Networks, S. Nikoletseas and J. Rolim, eds., Lecture Notes in Comput. Sci. 3121, Springer, Berlin, 2004, pp. 32–44.

[10] L. Auslander and R. MacKenzie, *Introduction to Differentiable Manifolds*, Dover, New York, 1977.

[11] A. Bahr, J. Leonard, and M. Fallon, *Cooperative localization for autonomous underwater vehicles*, Internat. J. Robotics Res., 28 (2009), pp. 714–728.

[12] A. Barvinok, *Problems of distance geometry and convex properties of quadratic maps*, Discrete Comput. Geom., 13 (1995), pp. 189–202.

[13] M. Belkin and P. Niyogi, *Laplacian eigenmaps for dimensionality reduction and data representation*, Neural Comput., 15 (2003), pp. 1373–1396.

[14] P. Belotti, J. Lee, L. Liberti, F. Margot, and A. Wächter, *Branching and bounds tightening techniques for non-convex MINLP*, Optim. Methods Softw., 24 (2009), pp. 597–634.

[15] A. Ben-Israel and B. Mond, *What is invexity?*, J. Aust. Math. Soc. Ser. B, 28 (1986), pp. 1–9.

[16] R. Benedetti and J.-J. Risler, *Real Algebraic and Semi-algebraic Sets*, Hermann, Paris, 1990.

[17] B. Berger, J. Kleinberg, and T. Leighton, *Reconstructing a three-dimensional model with arbitrary errors*, J. ACM, 46 (1999), pp. 212–235.

[18] H. Berman, J. Westbrook, Z. Feng, G. Gilliland, T. Bhat, H. Weissig, I. Shindyalov, and P. Bourne, *The protein data bank*, Nucleic Acid Res., 28 (2000), pp. 235–242.

[19] N. Biggs, *Algebraic Graph Theory*, Cambridge University Press, Cambridge, UK, 1974.

[20] N. Biggs, E. Lloyd, and R. Wilson, *Graph Theory* 1736–1936, Oxford University Press, Oxford, 1976.

[21] P. Biswas, *Semidefinite Programming Approaches to Distance Geometry Problems*, Ph.D. thesis, Stanford University, Stanford, CA, 2007.

[22] P. Biswas, T. Lian, T. Wang, and Y. Ye, *Semidefinite programming based algorithms for sensor network localization*, ACM Trans. Sensor Networks, 2 (2006), pp. 188–220.

[23] P. Biswas, T.-C. Liang, K.-C. Toh, T.-C. Wang, and Y. Ye, *Semidefinite programming approaches for sensor network localization with noisy distance measurements*, IEEE Trans. Automation Sci. Engrg., 3 (2006), pp. 360–371.

[24] P. Biswas, K.-C. Toh, and Y. Ye, *A distributed SDP approach for large-scale noisy anchor-free graph realization with applications to molecular conformation*, SIAM J. Sci. Comput., 30 (2008), pp. 1251–1277.

[25] P. Biswas and Y. Ye, *Semidefinite programming for ad hoc wireless sensor network localization*, in Proceedings of the 3rd International Symposium on Information Processing in Sensor Networks (IPSN04), ACM, New York, 2004, pp. 46–54.

[26] P. Biswas and Y. Ye, *A distributed method for solving semidefinite programs arising from ad hoc wireless sensor network localization*, in Multiscale Optimization Methods and

Applications, W. Hager et al., eds., Nonconvex Optim. Appl. 82, Springer, New York, 2006, pp. 69–84.

[27] A. Björner, M. Las Vergnas, B. Sturmfels, N. White, and G. Ziegler, *Oriented Matroids*, Cambridge University Press, Cambridge, UK, 1993.

[28] L. Blumenthal, *Theory and Applications of Distance Geometry*, Oxford University Press, Oxford, 1953.

[29] H. Bohr and S. Brunak, eds., *Protein Folds: A Distance Based Approach*, CRC Press, Boca Raton, FL, 1996.

[30] J. Bokowski and B. Sturmfels, *On the coordinatization of oriented matroids*, Discrete Comput. Geom., 1 (1986), pp. 293–306.

[31] I. Borg and P. Groenen, *Modern Multidimensional Scaling*, 2nd ed., Springer, New York, 2010.

[32] S. Boyd, L. El Ghaoui, E. Feron, and V. Balakrishnan, *Linear Matrix Inequalities in System and Control Theory*, SIAM, Philadelphia, 1994.

[33] H. Breu and D. Kirkpatrick, *Unit disk graph recognition is* NP*-hard*, Comput. Geom., 9 (1998), pp. 3–24.

[34] J. Canny and I. Emiris, *A subdivision-based algorithm for the sparse resultant*, J. ACM, 47 (2000), pp. 417–451.

[35] J. Carroll and J. Chang, *Analysis of individual differences in multidimensional scaling via an n-way generalization of "Eckart-Young" decomposition*, Psychometrika, 35 (1970), pp. 283–319.

[36] R. Carvalho, C. Lavor, and F. Protti, *Extending the geometric build-up algorithm for the molecular distance geometry problem*, Inform. Process. Lett., 108 (2008), pp. 234–237.

[37] A.-L. Cauchy, *Sur les polygones et les polyèdres*, J. Ecole Polytechnique, 16 (1813), pp. 87–99.

[38] A. Cayley, *A theorem in the geometry of position*, Cambridge Math. J., 2 (1841), pp. 267–271.

[39] H. Chen, *Distance Geometry for Kissing Balls*, preprint, arXiv:1203.2131v2, 2012.

[40] C. Chevalley, *The Construction and Study of Certain Important Algebras*, The Mathematical Society of Japan, Tokyo, 1955.

[41] B. Clark, C. Colburn, and D. Johnson, *Unit disk graphs*, Discrete Math., 86 (1990), pp. 165–177.

[42] G. Clore and A. Gronenborn, *Determination of three-dimensional structures of proteins and nucleic acids in solution by nuclear magnetic resonance spectroscopy*, Critical Reviews in Biochemistry and Molecular Biology, 24 (1989), pp. 479–564.

[43] R. Connelly, *A counterexample to the rigidity conjecture for polyhedra*, Inst. Hautes Études Sci. Publ. Math., 47 (1978), pp. 333–338.

[44] R. Connelly, *On generic global rigidity*, in Applied Geometry and Discrete Mathematics, DIMACS Ser. Discrete Math. Theoret. Comput. Sci. 4, AMS, Providence, RI, 1991, pp. 147–155.

[45] R. Connelly, *Generic global rigidity*, Discrete Comput. Geom., 33 (2005), pp. 549–563.

[46] J. Conway and N. Sloane, *Sphere Packings, Lattices and Groups*, Springer, Berlin, 1993.

[47] I. Coope, *Reliable computation of the points of intersection of n spheres in* $\mathbb{R}^n$, Aust. N. Z. Indust. Appl. Math. J., 42 (2000), pp. C461–C477.

[48] V. Costa, C. Lavor, A. Mucherino, A. Cassioli, L. Carvalho, and N. Maculan, *Discretization orders for protein side chains*, J. Global Optim., to appear.

[49] L. Cremona, *Le figure reciproche nella statica grafica*, G. Bernardoni, Milano, 1872.

[50] L. Cremona, *Elementi di calcolo grafico*, Paravia, Torino, 1874.

[51] G. Crippen, *Distance geometry for realistic molecular conformations*, in Distance Geometry: Theory, Methods, and Applications, A. Mucherino, C. Lavor, L. Liberti, and N. Maculan, eds., Springer, New York, 2013, pp. 315–328.

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

[53] A. Crum Brown, *On the theory of isomeric compounds*, Trans. Roy. Soc. Edinburgh, 23 (1864), pp. 707–719.

[54] M. Cucuringu, Y. Lipman, and A. Singer, *Sensor network localization by eigenvector synchronization over the Euclidean group*, ACM Trans. Sensor Networks, 8 (2012), pp. 1–42.

[55] M. Cucuringu, A. Singer, and D. Cowburn, *Eigenvector synchronization, graph rigidity and the molecule problem*, Inform. Inference, 1 (2012), pp. 21–67.

[56] J. Dattorro, *Convex Optimization and Euclidean Distance Geometry*, $\mathcal{M}\epsilon\beta oo$, Palo Alto, 2005.

[57] J. Dattorro, *Equality relating Euclidean distance cone to positive semidefinite cone*, Linear Algebra Appl., 428 (2008), pp. 2597–2600.

[58] J. de Leeuw and W. Heiser, *Theory of multidimensional scaling*, in Classification Pattern Recognition and Reduction of Dimensionality, P. Krishnaiah and L. Kanal, eds., Handbook of Statist. 2, Elsevier, 1982, pp. 285–316.

[59] E. Demaine, F. Gomez-Martin, H. Meijer, D. Rappaport, P. Taslakian, G. Toussaint, T. Winograd, and D. Wood, *The distance geometry of music*, Comput. Geom., 42 (2009), pp. 429–454.

[60] M. Deza and E. Deza, *Encyclopedia of Distances*, Springer, Berlin, 2009.

[61] R. Diestel, *Graph Theory*, Springer, New York, 2005.

[62] G. Dirac, *On rigid circuit graphs*, Abh. Math. Sem. Univ. Hamburg, 25 (1961), pp. 71–76.

[63] L. Doherty, K. Pister, and L. El Ghaoui, *Convex position estimation in wireless sensor networks*, in Twentieth Annual Joint Conference of the IEEE Computer and Communications Societies, Vol. 3 of INFOCOM, IEEE, 2001, pp. 1655–1663.

[64] B. Donald, *Algorithms in Structural Molecular Biology*, MIT Press, Boston, 2011.

[65] Q. Dong and Z. Wu, *A linear-time algorithm for solving the molecular distance geometry problem with exact inter-atomic distances*, J. Global Optim., 22 (2002), pp. 365–375.

[66] Q. Dong and Z. Wu, *A geometric build-up algorithm for solving the molecular distance geometry problem with sparse distance data*, J. Global Optim., 26 (2003), pp. 321–333.

[67] A. Dress and T. Havel, *Distance geometry and geometric algebra*, Found. Phys., 23 (1993), pp. 1357–1374.

[68] G. Dzemyda, O. Kurasova, and J. Žilinskas, *Multidimensional Data Visualiation: Methods and Applications*, Springer, New York, 2013.

[69] E. Dzhafarov and H. Colonius, *Reconstructing distances among objects from their discriminability*, Psychometrika, 71 (2006), pp. 365–386.

[70] J. Eaton, *GNU Octave Manual*, Network Theory Limited, 2002.

[71] C. Eckart and G. Young, *The approximation of one matrix by another of lower rank*, Psychometrika, 1 (1936), pp. 211–218.

[72] I. Emiris and B. Mourrain, *Computer algebra methods for studying and computing molecular conformations*, Algorithmica, 25 (1999), pp. 372–402.

[73] T. Eren, D. Goldenberg, W. Whiteley, Y. Yang, A. Morse, B. Anderson, and P. Belhumeur, *Rigidity, computation, and randomization in network localization*, IEEE Infocom Proc., 4 (2004), pp. 2673–2684.

[74] B. Everitt and S. Rabe-Hesketh, *The Analysis of Proximity Data*, Arnold, London, 1997.

[75] S. Feferman, J. Dawson, S. Kleene, G. Moore, R. Solovay, and J. van Heijenoort, eds., *Kurt Gödel: Collected Works*, Vol. I, Oxford University Press, Oxford, 1986.

[76] Z. Fekete and T. Jordán, *Uniquely localizable networks with few anchors*, in Algorithmic Aspects of Wireless Sensor Networks, S. Nikoletseas and J. Rolim, eds., Lecture Notes in Comput. Sci. 4240, Springer, Berlin, 2006, pp. 176–183.

[77] G. Forman and J. Zahorjan, *The challenges of mobile computing*, IEEE Comput., 27 (1994), pp. 38–47.

[78] I. Fudos and C. Hoffmann, *A graph-constructive approach to solving systems of geometric constraints*, ACM Trans. Graphics, 16 (1997), pp. 179–216.

[79] M. Garey and D. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*, Freeman and Company, New York, 1979.

[80] K. Gibson and H. Scheraga, *Energy minimization of rigid-geometry polypeptides with exactly closed disulfide loops*, J. Comput. Chem., 18 (1997), pp. 403–415.

[81] H. Gluck, *Almost all simply connected closed surfaces are rigid*, in Geometric Topology, A. Dold and B. Eckmann, eds., Lecture Notes in Math. 438, Springer, Berlin, 1975, pp. 225–239.

[82] W. Glunt, T. L. Hayden, S. Hong, and J. Wells, *An alternating projection algorithm for computing the nearest Euclidean distance matrix*, SIAM J. Matrix Anal. Appl., 11 (1990), pp. 589–600.

[83] S. Gortler, A. Healy, and D. Thurston, *Characterizing generic global rigidity*, Amer. J. Math., 132 (2010), pp. 897–939.

[84] J. Gower, *Some distance properties of latent root and vector methods in multivariate analysis*, Biometrika, 53 (1966), pp. 325–338.

[85] J. Gower, *Euclidean distance geometry*, Math. Sci., 7 (1982), pp. 1–14.

[86] W. Gramacho, A. Mucherino, C. Lavor, and N. Maculan, *A parallel BP algorithm for the discretizable distance geometry problem*, in Proceedings of the Workshop on Parallel Computing and Optimization, Shanghai, 2012, IEEE, pp. 1756–1762.

[87] J. Graver, *Rigidity matroids*, SIAM J. Discrete Math., 4 (1991), pp. 355–368.

[88] J. Graver, B. Servatius, and H. Servatius, *Combinatorial Rigidity*, AMS, Providence, RI, 1993.

[89] L. Grippo and M. Sciandrone, *On the convergence of the block nonlinear Gauss-Seidel method under convex constraints*, Oper. Res. Lett., 26 (2000), pp. 127–136.

[90] R. Grone, C. Johnson, E. de Sá, and H. Wolkowicz, *Positive definite completions of partial Hermitian matrices*, Linear Algebra Appl., 58 (1984), pp. 109–124.

[91] A. Grosso, M. Locatelli, and F. Schoen, *Solving molecular distance geometry problems by global optimization algorithms*, Comput. Optim. Appl., 43 (2009), pp. 23–27.

[92] T. Havel, *Metric matrix embedding in protein structure calculations*, Magnetic Resonance Chem., 41 (2003), pp. 537–550.

[93] T. Havel, I. Kuntz, and G. Crippen, *The theory and practice of distance geometry*, Bull. Math. Biol., 45 (1983), pp. 665–720.

[94] B. Hendrickson, *Conditions for unique graph realizations*, SIAM J. Comput., 21 (1992), pp. 65–84.

[95] B. Hendrickson, *The molecule problem: Exploiting structure in global optimization*, SIAM J. Optim., 5 (1995), pp. 835–857.

[96] L. Henneberg, *Statik der starren Systeme*, Bergstræsser, Darmstadt, 1886.

[97] L. Henneberg, *Die Graphische Statik der starren Systeme*, Teubner, Leipzig, 1911.

[98] L. Hoai An, *Solving large scale molecular distance geometry problems by a smoothing technique via the Gaussian transform and D.C. programming*, J. Global Optim., 27 (2003), pp. 375–397.

[99] L. T. Hoai An and P. Dinh Tao, *Large-scale molecular optimization from distance matrices by a d.c. optimization approach*, SIAM J. Optim., 14 (2003), pp. 77–114.

[100] H.-X. Huang, Z.-A. Liang, and P. Pardalos, *Some properties for the Euclidean distance matrix and positive semidefinite matrix completion problems*, J. Global Optim., 25 (2003), pp. 3–21.

[101] K. Hunt, *Kinematic Geometry of Mechanisms*, Oxford University Press, Oxford, 1990.

[102] S. Izrailev, F. Zhu, and D. Agrafiotis, *A distance geometry heuristic for expanding the range of geometries sampled during conformational search*, J. Comput. Chem., 26 (2006), pp. 1962–1969.

[103] B. Jackson and T. Jordán, *Connected rigidity matroids and unique realization of graphs*, J. Combin. Theory Ser. B, 94 (2005), pp. 1–29.

[104] B. Jackson and T. Jordán, *On the rigidity of molecular graphs*, Combinatorica, 28 (2008), pp. 645–658.

[105] B. Jackson and T. Jordán, *Graph theoretic techniques in the analysis of uniquely localizable sensor networks*, in Localization Algorithms and Strategies for Wireless Sensor Networks: Monitoring and Surveillance Techniques for Target Tracking, G. Mao and B. Fidan, eds., IGI Global, 2009, pp. 146–173.

[106] C. Johnson, B. Kroschel, and H. Wolkowicz, *An interior-point method for approximate positive semidefinite completions*, Comput. Optim. Appl., 9 (1998), pp. 175–190.

[107] I. Jolliffe, *Principal Component Analysis*, 2nd ed., Springer, Berlin, 2010.

[108] J. Kostrowicki and L. Piela, *Diffusion equation method of global minimization: Performance for standard test functions*, J. Optim. Theory Appl., 69 (1991), pp. 269–284.

[109] P. Krishnaiah and L. Kanal, eds., *Theory of Multidimensional Scaling*, Vol. 2, North-Holland, 1982.

[110] N. Krislock, *Semidefinite Facial Reduction for Low-Rank Euclidean Distance Matrix Completion*, Ph.D. thesis, University of Waterloo, 2010.

[111] N. Krislock and H. Wolkowicz, *Explicit sensor network localization using semidefinite representations and facial reductions*, SIAM J. Optim., 20 (2010), pp. 2679–2708.

[112] J. Kruskal, *Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis*, Psychometrika, 29 (1964), pp. 1–27.

[113] J. Kruskal, *Nonmetric multidimensional scaling: A numerical method*, Psychometrika, 29 (1964), pp. 115–129.

[114] S. Kucherenko, P. Belotti, L. Liberti, and N. Maculan, *New formulations for the kissing number problem*, Discrete Appl. Math., 155 (2007), pp. 1837–1841.

[115] S. Kucherenko and Y. Sytsko, *Application of deterministic low-discrepancy sequences in global optimization*, Comput. Optim. Appl., 30 (2004), pp. 297–318.

[116] G. Laman, *On graphs and rigidity of plane skeletal structures*, J. Engrg. Math., 4 (1970), pp. 331–340.

[117] M. Laurent, *Cuts, matrix completions and graph rigidity*, Math. Program., 79 (1997), pp. 255–283.

[118] M. Laurent, *Polynomial instances of the positive semidefinite and Euclidean distance matrix completion problems*, SIAM J. Matrix Anal. Appl., 22 (2000), pp. 874–894.

[119] M. Laurent, *Matrix completion problems*, in Encyclopedia of Optimization, 2nd ed., C. Floudas and P. Pardalos, eds., Springer, New York, 2009, pp. 1967–1975.

[120] C. Lavor, *On generating instances for the molecular distance geometry problem*, in Global Optimization: From Theory to Implementation, L. Liberti and N. Maculan, eds., Springer, Berlin, 2006, pp. 405–414.

[121] C. Lavor, J. Lee, A. Lee-St. John, L. Liberti, A. Mucherino, and M. Sviridenko, *Discretization orders for distance geometry problems*, Optim. Lett., 6 (2012), pp. 783–796.

[122] C. Lavor, L. Liberti, and N. Maculan, *Grover's algorithm applied to the molecular distance geometry problem*, in Proceedings of the 7th Brazilian Congress of Neural Networks, Natal, Brazil, 2005.

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

[124] C. Lavor, L. Liberti, and N. Maculan, *The Discretizable Molecular Distance Geometry Problem*, preprint, arXiv:q-bio/0608012, 2006.

[125] C. Lavor, L. Liberti, and N. Maculan, *Molecular distance geometry problem*, in Encyclopedia of Optimization, 2nd ed., C. Floudas and P. Pardalos, eds., Springer, New York, 2009, pp. 2305–2311.

[126] C. Lavor, L. Liberti, and N. Maculan, *A note on "A Branch-and-Prune Algorithm for the Molecular Distance Geometry Problem,"* Internat. Trans. Oper. Res., 18 (2011), pp. 751–752.

[127] C. Lavor, L. Liberti, N. Maculan, and A. Mucherino, *The discretizable molecular distance geometry problem*, Comput. Optim. Appl., 52 (2012), pp. 115–146.

[128] C. Lavor, L. Liberti, N. Maculan, and A. Mucherino, *Recent advances on the discretizable molecular distance geometry problem*, European J. Oper. Res., 219 (2012), pp. 698–706.

[129] C. Lavor, L. Liberti, and A. Mucherino, *The interval branch-and-prune algorithm for the discretizable molecular distance geometry problem with inexact distances*, J. Global Optim., 56 (2013), pp. 855–871.

[130] C. Lavor, L. Liberti, A. Mucherino, and N. Maculan, *On a discretizable subclass of instances of the molecular distance geometry problem*, in Proceedings of the 24th Annual ACM Symposium on Applied Computing, D. Shin, ed., ACM, New York, 2009, pp. 804–805.

[131] C. Lavor, A. Mucherino, L. Liberti, and N. Maculan, *An artificial backbone of hydrogens for finding the conformation of protein molecules*, in Proceedings of the Computational Structural Bioinformatics Workshop, Washington D.C., IEEE, 2009, pp. 152–155.

[132] C. Lavor, A. Mucherino, L. Liberti, and N. Maculan, *Computing artificial backbones of hydrogen atoms in order to discover protein backbones*, in Proceedings of the International Multiconference on Computer Science and Information Technology, Mragowo, Poland, IEEE, 2009, pp. 751–756.

[133] C. Lavor, A. Mucherino, L. Liberti, and N. Maculan, *Discrete approaches for solving molecular distance geometry problems using NMR data*, Internat. J. Comput. Biosci., 1 (2010), pp. 88–94.

[134] C. Lavor, A. Mucherino, L. Liberti, and N. Maculan, *On the solution of molecular distance geometry problems with interval data*, in Proceedings of the International Workshop on Computational Proteomics, Hong Kong, IEEE, 2010, pp. 77–82.

[135] C. Lavor, A. Mucherino, L. Liberti, and N. Maculan, *On the computation of protein backbones by using artificial backbones of hydrogens*, J. Global Optim., 50 (2011), pp. 329–344.

[136] C. Lavor, A. Mucherino, L. Liberti, and N. Maculan, *Finding low-energy homopolymer conformations by a discrete approach*, in Proceedings of the Global Optimization Workshop, D. Aloise, P. Hansen, and C. Rocha, eds., Universidade Federal do Rio Grande do Norte, Natal, 2012.

[137] S. Le Grand, A. Elofsson, and D. Eisenberg, *The effect of distance-cutoff on the performance of the distance matrix error when used as a potential function to drive conformational search*, in Protein Folds: A Distance Based Approach, H. Bohr and S. Brunak, eds., CRC Press, Boca Raton, FL, 1996, pp. 105–113.

[138] J. Lee and M. Verleysen, *Nonlinear Dimensionality Reduction*, Springer, Berlin, 2010.

[139] N.-H. Z. Leung and K.-C. Toh, *An SDP-based divide-and-conquer algorithm for large-scale noisy anchor-free graph realization*, SIAM J. Sci. Comput., 31 (2009), pp. 4351–4372.

[140] L. Liberti, *Reformulation and Convex Relaxation Techniques for Global Optimization*, Ph.D. thesis, Imperial College London, London, 2004.

[141] L. Liberti, *Reformulations in mathematical programming: Definitions and systematics*, RAIRO Oper. Res., 43 (2009), pp. 55–85.

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

[143] L. Liberti and S. Kucherenko, *Comparison of Deterministic and Stochastic Approaches to Global Optimization*, Tech. Rep. 2004.25, DEI, Politecnico di Milano, 2004.

[144] L. Liberti and C. Lavor, *On a relationship between graph realizability and distance matrix completion*, in Optimization Theory, Decision Making, and Operational Research Applications, A. Migdalas, A. Sifaleras, C. Georgiadis, J. Papathanaiou, and E. Stiakakis, eds., Proc. Math. Statist. 31, Springer, Berlin, 2013, pp. 39–48.

[145] L. Liberti, C. Lavor, and N. Maculan, *A branch-and-prune algorithm for the molecular distance geometry problem*, Internat. Trans. Oper. Res., 15 (2008), pp. 1–17.

[146] L. Liberti, C. Lavor, N. Maculan, and F. Marinelli, *Double variable neighbourhood search with smoothing for the molecular distance geometry problem*, J. Global Optim., 43 (2009), pp. 207–218.

[147] L. Liberti, C. Lavor, and A. Mucherino, *The discretizable molecular distance geometry problem seems easier on proteins*, in Distance Geometry: Theory, Methods, and Applications, A. Mucherino, C. Lavor, L. Liberti, and N. Maculan, eds., Springer, New York, 2013, pp. 47–60.

[148] L. Liberti, C. Lavor, A. Mucherino, and N. Maculan, *Molecular distance geometry methods: From continuous to discrete*, Internat. Trans. Oper. Res., 18 (2010), pp. 33–51.

[149] L. Liberti, B. Masson, C. Lavor, J. Lee, and A. Mucherino, *On the Number of Solutions of the Discretizable Molecular Distance Geometry Problem*, preprint, arXiv:1010.1834v1[cs.DM], 2010.

[150] L. Liberti, B. Masson, C. Lavor, and A. Mucherino, *Branch-and-prune trees with bounded width*, in Proceedings of Cologne/Twente Workshop, G. Nicosia and A. Pacifici, eds., Università di Roma 2–Tor Vergata, Rome, 2011.

[151] L. Liberti, B. Masson, J. Lee, C. Lavor, and A. Mucherino, *On the number of solutions of the discretizable molecular distance geometry problem*, in Combinatorial Optimization, Constraints and Applications (COCOA11), Lecture Notes in Comput. Sci. 6831, Springer, New York, 2011, pp. 322–342.

[152] L. Liberti, B. Masson, J. Lee, C. Lavor, and A. Mucherino, *On the number of realizations of certain Henneberg graphs arising in protein conformation*, Discrete Appl. Math., to appear.

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

[154] L. Lovász and Y. Yemini, *On generic rigidity in the plane*, SIAM J. Algebraic Discrete Methods, 3 (1982), pp. 91–98.

[155] Y. Ma and Y. Fu, eds., *Manifold Learning Theory and Applications*, CRC Press, Boca Raton, FL, 2012.

[156] T. Malliavin, A. Mucherino, and M. Nilges, *Distance geometry in structural biology*, in Distance Geometry: Theory, Methods, and Applications, A. Mucherino, C. Lavor, L. Liberti, and N. Maculan, eds., Springer, New York, 2013, pp. 329–350.

[157] A. Man-Cho So and Y. Ye, *Theory of semidefinite programming for sensor network localization*, Math. Program. Ser. B, 109 (2007), pp. 367–384.

[158] D. Manocha and J. Canny, *Efficient inverse kinematics for general 6R manipulators*, IEEE Trans. Robotics Automation, 10 (1994), pp. 648–657.

[159] J. Maxwell, *On reciprocal figures and diagrams of forces*, Philos. Mag., 27 (1864), pp. 250–261.

[160] J. Maxwell, *On the calculation of the equilibrium and stiffness of frames*, Philos. Mag., 27 (1864), pp. 294–299.

[161] K. Menger, *Untersuchungen über allgemeine Metrik*, Math. Ann., 100 (1928), pp. 75–163.

[162] K. Menger, *New foundation of Euclidean geometry*, Amer. J. Math., 53 (1931), pp. 721–745.

[163] B. Mishra, *Computational real algebraic geometry*, in Handbook of Discrete and Computational Geometry, 2nd ed., J. Goodman and J. O'Rourke, eds., CRC Press, Boca Raton, FL, 2004, pp. 743–764.

[164] J. Moré and Z. Wu, *Global continuation for distance geometry problems*, SIAM J. Optim., 7 (1997), pp. 814–836.

[165] J. Moré and Z. Wu, *Distance geometry optimization for protein structures*, J. Global Optim., 15 (1999), pp. 219–234.

[166] A. Mucherino and C. Lavor, *The branch and prune algorithm for the molecular distance geometry problem with inexact distances*, in Proceedings of the International Conference on Computational Biology, Vol. 58, World Academy of Science, Engineering and Technology, 2009, pp. 349–353.

[167] A. Mucherino, C. Lavor, and L. Liberti, *A symmetry-driven BP algorithm for the discretizable molecular distance geometry problem*, in Proceedings of Computational Structural Bioinformatics Workshop, IEEE, 2011, pp. 390–395.

[168] A. Mucherino, C. Lavor, and L. Liberti, *The discretizable distance geometry problem*, Optim. Lett., 6 (2012), pp. 1671–1686.

[169] A. Mucherino, C. Lavor, and L. Liberti, *Exploiting symmetry properties of the discretizable molecular distance geometry problem*, J. Bioinform. Comput. Biol., 10 (2012), 1242009.

[170] A. Mucherino, C. Lavor, L. Liberti, and N. Maculan, *On the definition of artificial backbones for the discretizable molecular distance geometry problem*, Math. Balkanica (N.S.), 23 (2009), pp. 289–302.

[171] A. Mucherino, C. Lavor, L. Liberti, and N. Maculan, *Strategies for solving distance geometry problems with inexact distances by discrete approaches*, in Proceedings of the Toulouse Global Optimization Workshop, S. Cafieri, E. Hendrix, L. Liberti, and F. Messine, eds., Toulouse, 2010, pp. 93–96.

[172] A. Mucherino, C. Lavor, L. Liberti, and N. Maculan, *On the discretization of distance geometry problems*, in Proceedings of the Conference on Mathematics of Distances and Applications, M. Deza, M. Petitjean, and K. Markov, eds., ITHEA, Varna, 2012, pp. 160–168.

[173] A. Mucherino, C. Lavor, L. Liberti, and N. Maculan, eds., *Distance Geometry: Theory, Methods, and Applications*, Springer, New York, 2013.

[174] A. Mucherino, C. Lavor, L. Liberti, and E.-G. Talbi, *A parallel version of the branch & prune algorithm for the molecular distance geometry problem*, in ACS/IEEE International Conference on Computer Systems and Applications (AICCSA10), Hammamet, Tunisia, IEEE, 2010, pp. 1–6.

[175] A. Mucherino, C. Lavor, and N. Maculan, *The molecular distance geometry problem applied to protein conformations*, in Proceedings of the 8th Cologne-Twente Workshop on Graphs and Combinatorial Optimization, S. Cafieri, A. Mucherino, G. Nannicini, F. Tarissan, and L. Liberti, eds., École Polytechnique, Paris, 2009, pp. 337–340.

[176] A. Mucherino, C. Lavor, T. Malliavin, L. Liberti, M. Nilges, and N. Maculan, *Influence of pruning devices on the solution of molecular distance geometry problems*, in Experimental Algorithms, P. Pardalos and S. Rebennack, eds., Lecture Notes in Comput. Sci. 6630, Springer, Berlin, 2011, pp. 206–217.

[177] A. Mucherino, L. Liberti, and C. Lavor, `MD-jeep`*: An implementation of a branch-and-prune algorithm for distance geometry problems*, in Mathematical Software, K. Fukuda, J. van der Hoeven, M. Joswig, and N. Takayama, eds., Lecture Notes in Comput. Sci. 6327, Springer, New York, 2010, pp. 186–197.

[178] A. Mucherino, L. Liberti, C. Lavor, and N. Maculan, *Comparisons between an exact and a metaheuristic algorithm for the molecular distance geometry problem*, in Proceedings of the Genetic and Evolutionary Computation Conference, F. Rothlauf, ed., Montreal, ACM, New York, 2009, pp. 333–340.

[179] J. Nielsen and B. Roth, *On the kinematic analysis of robotic mechanisms*, Internat. J. Robotics Res., 18 (1999), pp. 1147–1160.

[180] M. Nilges, M. Macias, S. O'Donoghue, and H. Oschkinat, *Automated NOESY interpretation with ambiguous distance restraints: The refined NMR solution structure of the Pleckstrin homology domain from β-spectrin*, J. Molecular Biol., 269 (1997), pp. 408–422.

[181] P. Nucci, L. Nogueira, and C. Lavor, *Solving the discretizable molecular distance geometry problem by multiple realization trees*, in Distance Geometry: Theory, Methods, and Applications, A. Mucherino, C. Lavor, L. Liberti, and N. Maculan, eds., Springer, New York, 2013, pp. 161–176.

[182] M. Petitjean, *Sphere unions and intersections and some of their applications in molecular modeling*, in Distance Geometry: Theory, Methods, and Applications, A. Mucherino, C. Lavor, L. Liberti, and N. Maculan, eds., Springer, New York, 2013, pp. 61–83.

[183] T. Pong and P. Tseng, *(Robust) edge-based semidefinite programming relaxation of sensor network localization*, Math. Program. Ser. A, 130 (2011), pp. 321–358.

[184] J. Porta, L. Ros, and F. Thomas, *Inverse kinematics by distance matrix completion*, in Proceedings of the 12th International Workshop on Computational Kinematics, 2005, pp. 1–9.

[185] J. Porta, L. Ros, F. Thomas, and C. Torras, *A branch-and-prune solver for distance constraints*, IEEE Trans. Robotics, 21 (2005), pp. 176–187.

[186] R. Rao, A. Asaithambi, and S. Agrawal, *Inverse kinematic solution of robot manipulators using interval analysis*, ASME J. Mech. Design, 120 (1998), pp. 147–150.

[187] R. Reams, G. Chatham, W. Glunt, D. McDonald, and T. Hayden, *Determining protein structure using the distance geometry program APA*, Computers and Chemistry, 23 (1999), pp. 153–163.

[188] A. Recski, *A network theory approach to the rigidity of skeletal structures. Part* 2. *Laman's theorem and topological formulae*, Discrete Appl. Math., 8 (1984), pp. 63–68.

[189] N. Rojas, *Distance-Based Formulations for the Position Analysis of Kinematic Chains*, Ph.D. thesis, Universitat Politecnica de Catalunya, 1989.

[190] D. J. Rose, R. E. Tarjan, and G. S. Lueker, *Algorithmic aspects of vertex elimination on graphs*, SIAM J. Comput., 5 (1976), pp. 266–283.

[191] B. Roth, *Rigid and flexible frameworks*, Amer. Math. Monthly, 88 (1981), pp. 6–21.

[192] B. Roth and F. Freudenstein, *Synthesis of path-generating mechanisms by numerical methods*, J. Engrg. Industry, 85 (1963), pp. 298–307.

[193] S. Sallaume, S. Martins, L. Ochi, W. Gramacho, C. Lavor, and L. Liberti, *A discrete search algorithm for finding the structure of protein backbones and side chains*, Internat. J. Bioinform. Res. Appl., 9 (2013), pp. 261–270.

[194] R. Santana, P. Larrañaga, and J. Lozano, *Side chain placement using estimation of distribution algorithms*, Art. Intell. Med., 39 (2007), pp. 49–63.

[195] C. Saviotti, *Nouvelles méthodes pour le calcul des travures réticulaires*, in Appendix to L. Cremona, "Les figures réciproques en statique graphique," Gauthier-Villars, Paris, 1885, pp. 37–100.

[196] C. Saviotti, *La statica grafica: Lezioni*, U. Hoepli, Milano, 1888.

[197] A. Savvides, C.-C. Han, and M. Strivastava, *Dynamic fine-grained localization in ad-hoc networks of sensors*, in Proceedings of the 7th Annual International Conference on Mobile Computing and Networking, MobiCom '01, ACM, New York, 2001, pp. 166–179.

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

[199] T. Schlick, *Molecular Modelling and Simulation: An Interdisciplinary Guide*, Springer, New York, 2002.

[200] I. Schoenberg, *Remarks to Maurice Fréchet's article "Sur la définition axiomatique d'une classe d'espaces distanciés vectoriellement applicable sur l'espace de Hilbert,"* Ann. of Math., 36 (1935), pp. 724–732.

[201] B. Servatius and H. Servatius, *Generic and abstract rigidity*, in Rigidity Theory and Applications, M. Thorpe and P. Duxbury, eds., Fundamental Materials Research, Springer, New York, 2002, pp. 1–19.

[202] R. Shepard, *The analysis of proximities: Multidimensional scaling with an unknown distance function, Part* I, Psychometrika, 27 (1962), pp. 125–140.

[203] R. Shepard, *The analysis of proximities: Multidimensional scaling with an unknown distance function, Part* II, Psychometrika, 27 (1962), pp. 219–246.

[204] R. Shepard, *Metric structures in ordinal data*, J. Math. Psych., 3 (1966), pp. 287–315.

[205] A. Singer, *Angular synchronization by eigenvectors and semidefinite programming*, Appl. Comput. Harmon. Anal., 30 (2011), pp. 20–36.

[206] A. Singer and M. Cucuringu, *Uniqueness of low-rank matrix completion by rigidity theory*, SIAM J. Matrix Anal. Appl., 31 (2010), pp. 1621–1641.

[207] A. Singer, Z. Zhao, Y. Shkolnisky, and R. Hadani, *Viewing angle classification of cryo-electron microscopy images using eigenvectors*, SIAM J. Imaging Sci., 4 (2011), pp. 723–759.

[208] M. Sippl and H. Scheraga, *Solution of the embedding problem and decomposition of symmetric matrices*, Proc. Natl. Acad. Sci. USA, 82 (1985), pp. 2197–2201.

[209] M. Sippl and H. Scheraga, *Cayley-Menger coordinates*, Proc. Natl. Acad. Sci. USA, 83 (1986), pp. 2283–2287.

[210] M. Sitharam and Y. Zhou, *A tractable, approximate, combinatorial* 3*D rigidity characterization*, in the Fifth Workshop on Automated Deduction in Geometry, 2004.

[211] M. Souza, C. Lavor, A. Muritiba, and N. Maculan, *Solving the molecular distance geometry problem with inaccurate distance data*, BMC Bioinform., 14 (2013), pp. S71–S76.

[212] M. Souza, A. Xavier, C. Lavor, and N. Maculan, *Hyperbolic smoothing and penalty techniques applied to molecular structure determination*, Oper. Res. Lett., 39 (2011), pp. 461–465.

[213] C. Stumpf, *Tonpsychologie*, Vol. I, Hirzel, Leipzig, 1883.

[214] C. Stumpf, *Tonpsychologie*, Vol. II, Hirzel, Leipzig, 1890.

[215] J. Sylvester, *Chemistry and algebra*, Nature, 17 (1877), pp. 284–284.

[216] Y. Takane, F. Young, and J. De Leeuw, *Nonmetric individual differences in multidimensional scaling: An alternating least squares method with optimal scaling features*, Psychometrika, 42 (1977), pp. 7–67.

[217] T.-S. Tay, *On the generic rigidity of bar-frameworks*, Adv. Appl. Math., 23 (1999), pp. 14–28.

[218] T.-S. Tay and W. Whiteley, *Generating isostatic frameworks*, Struct. Topology, 11 (1985), pp. 21–69.

[219] J. Tenenbaum, V. de Silva, and J. Langford, *A global geometric framework for nonlinear dimensionality reduction*, Science, 290 (2000), pp. 2319–2322.

[220] F. Thomas, J. Porta, and L. Ros, *Distance constraints solved geometrically*, in Advances in Robot Kinematics, G. Galletti and J. Lenarcic, eds., Kluwer, Dordrecht, 2004, pp. 123–132.

[221] C. Thomassen, *The graph genus problem is NP-complete*, J. Algorithms, 10 (1989), pp. 568–576.

[222] D. Tolani, A. Goswami, and N. Badler, *Real-time inverse kinematics techniques for anthropomorphic limbs*, Graphical Models, 62 (2000), pp. 353–388.

[223] W. Torgerson, *Theory and Methods of Scaling*, Wiley, New York, 1958.

[224] L. Tsai and A. Morgan, *Solving the kinematics of the most general six- and five-degree-of-freedom manipulators by continuation methods*, J. Mech. Transmissions Automation in Design, 107 (1985), pp. 189–200.

[225] P. Tseng, *Convergence of a block coordinate descent method for nondifferentiable minimization*, J. Optim. Theory Appl., 109 (2001), pp. 475–494.

[226] P. Tseng, *Second-order cone programming relaxations of sensor network localization*, SIAM J. Optim., 18 (2007), pp. 156–185.

[227] P. Varignon, *Nouvelle Mécanique*, Claude Jombert, Paris, 1725.

[228] Z. Voller and Z. Wu, *Distance geometry methods for protein structure determination*, in Distance Geometry: Theory, Methods, and Applications, A. Mucherino, C. Lavor, L. Liberti, and N. Maculan, eds., Springer, New York, 2013, pp. 139–159.

[229] P. von Rague, P. Schreiner, N. Allinger, T. Clark, J. Gasteiger, P. Kollman, and H. Schaefer, eds., *Distance Geometry: Theory, Algorithms, and Chemical Applications*, Wiley, 1988.

[230] C. Wampler, A. Morgan, and A. Sommese, *Numerical continuation methods for solving polynomial systems arising in kinematics*, J. Mech. Design, 112 (1990), pp. 59–68.

[231] Z. Wang, S. Zheng, Y. Ye, and S. Boyd, *Further relaxations of the semidefinite programming approach to sensor network localization*, SIAM J. Optim., 19 (2008), pp. 655–673.

[232] M. Weiser, *Some computer science issues in ubiquitous computing*, Comm. ACM, 36 (1993), pp. 75–84.

[233] W. Whiteley, *Infinitesimally rigid polyhedra.* I. *Statics of frameworks*, Trans. Amer. Math. Soc., 285 (1984), pp. 431–465.

[234] W. Whiteley, *Rigidity and scene analysis*, in Handbook of Discrete and Computational Geometry, J. Goodman and J. O'Rourke, eds., CRC Press, Boca Raton, FL, 2004.

[235] G. Williams, J. Dugan, and R. Altman, *Constrained global optimization for estimating molecular structure from atomic distances*, J. Comput. Biol., 8 (2001), pp. 523–547.

[236] D. Wu and Z. Wu, *An updated geometric build-up algorithm for solving the molecular distance geometry problem with sparse distance data*, J. Global Optim., 37 (2007), pp. 661–673.

[237] D. Wu, Z. Wu, and Y. Yuan, *Rigid versus unique determination of protein structures with geometric buildup*, Optim. Lett., 2 (2008), pp. 319–331.

[238] K. Wüthrich, *Protein structure determination in solution by nuclear magnetic resonance spectroscopy*, Science, 243 (1989), pp. 45–50.

[239] K. Wüthrich, M. Billeter, and W. Braun, *Pseudo-structures for the* 20 *common amino acids for use in studies of protein conformations by measurements of intramolecular proton-proton distance constraints with nuclear magnetic resonance*, J. Molecular Biol., 169 (1983), pp. 949–961.

[240] H. Xu, S. Izrailev, and D. Agrafiotis, *Conformational sampling by self-organization*, J. Chem. Inform. Comput. Sci., 43 (2003), pp. 1186–1191.

[241] L. Yang, *Solving spatial constraints with global distance coordinate system*, J. Comput. Geom. Appl., 16 (2006), pp. 533–547.

[242] Y. Yemini, *The positioning problem—a draft of an intermediate summary*, in Proceedings of the Conference on Distributed Sensor Networks, Carnegie-Mellon University, Pittsburgh, 1978, pp. 137–145.

[243] Y. Yemini, *Some theoretical aspects of position-location problems*, in Proceedings of the 20th Annual Symposium on the Foundations of Computer Science, IEEE, 1979, pp. 1–8.

[244] M. Zhang, R. White, L. Wang, R. Goldman, L. Kavraki, and B. Hassett, *Improving conformational searches by geometric screening*, Bioinform., 21 (2005), pp. 624–630.

[245] Z. Zhu, A. Man-Cho So, and Y. Ye, *Universal rigidity and edge sparsification for sensor network localization*, SIAM J. Optim., 20 (2010), pp. 3059–3081.

[246] A. Žilinskas and J. Žilinskas, *Branch and bound algorithm for multidimensional scaling with city-block metric*, J. Global Optim., 43 (2009), pp. 357–372.