Geometric Algebra to describe the exact Discretizable Molecular Distance Geometry Problem for an arbitrary dimension

Valter S. Camargo, Emerson V. Castelani,
Leandro A. F. Fernandes and Felipe Fidalgo

To our estimated colleague Waldyr Rodrigues, Jr.

Abstract. The K–Discretizable Molecular Distance Geometry Problem (^{K}DMDGP) is a subclass of the Distance Geometry Problem (DGP), whose complexity is NP-hard, such that the search space is finite. In this work, the authors describe it completely using Conformal Geometric Algebra (CGA), exploring a Minkowski space that provides a natural interpretation of hyperspheres, hyperplanes, points and pair of points as computational primitives, which are largely relevant to the ^{K}DMDGP. It also presents a theoretical approach to solve the ^{K}DMDGP using ideas from classic Branch-and-Prune (BP) algorithm in this new fashion. Time complexity analysis and practical computational results showed that the naive implementation of the CGA is not as efficient as classical formulation. In order to illustrate this, preliminary results are displayed at the end and, also, directions to future developments.

Mathematics Subject Classification (2010). 51K99.

Keywords. Distance Geometry Problem, Conformal Algebra, Branch-and-prune, Symmetry, Minkowski Space, Outer Product.

1. Introduction

Let $(\mathcal{M}, d_{\mathcal{M}})$ be a metric space. Considering a simple and undirected graph $G = (V, E, d)$, where the edges in E are weighted by the distance function $d : E \longrightarrow \mathbb{R}_+$, the exact-distance case of the *Distance Geometry Problem* (DGP) asks whether there exists an embedding $\mathbf{x}$ of G into $\mathcal{M}$ such that

$$\forall \{u, v\} \in E, \qquad \|\mathbf{x}(u) - \mathbf{x}(v)\|_{\mathcal{M}} = d(\{u, v\}) \triangleq d_{u,v}, \tag{1}$$

where $\|\cdot\|_{\mathcal{M}}$ is the norm provided by the metric $d_{\mathcal{M}}$. It has been proved to be **NP**–complete for $K = 1$ and **NP**–hard for $K > 1$ [32].

There are examples of applications of this problem in several areas or dimensions. For lower dimensions ($K \leq 3$), the *Clock Synchronization Problem* (CSyP), the *Sensor Network Localization Problem* (SNLP) and the position-analysis of the Assur Kinematic Chains can be held, respectively, as DGPs in $\mathbb{R}$, $\mathbb{R}^2$ and $\mathbb{R}^3$ [22, 31]. Applications for dimensions greater than $K = 3$ include the *Graph Embedding Subproblem* (GES), which arises in many areas of Machine Learning, including deep neural networks [21].

If $\mathcal{M}$ is the Euclidean space $\mathbb{R}^3$, such problem is named *Molecular Distance Geometry Problem* (MDGP), which has shown to be a suitable model to look for tridimensional molecular conformations, usually solved in terms of Continuous Optimization [7, 27, 28]. Moreover, inspired by the application of the MDGP for proteins, its search space can be discretized so that it can be explored by combinatorial tools [19]. This subclass of the MDGP is the so-called *Discretizable Molecular Distance Geometry Problem* (DMDGP). The main method for solving it computationally is the so-called Branch-and-Prune (BP) algorithm, a combinatorial method that uses sphere intersections recursively [23]. One can set it to look up only for one solution, which is called BP*one*. Furthermore, some existing symmetry equivalence (congruence) relations on V have been discovered, allowing computational improvements on BP, which is summarized in what follows and described in details in [29]. Considering two realizations of G as congruent is equivalent to say that one can be got from the other by the evaluation of partial reflections in the *symmetry vertices*, the ones from the set $S_G = \{v \in V : \nexists \{u, w\} \in E \text{ such that } u + 3 < v \leq w\}$, preserving the distances between all the pair of points for both. This existence proves what has been empirically noticed that the number of solutions for each DMDGP is power of two (since the reference frame is fixed) and also gives the result that finding only one solution is enough, as all the remaining ones can be determined using partial reflections. From this fact, an adaptation for BP is driven and called *sym*BP. It simply follows what has been sketched up in the latter, that is, it takes the solution from BP*one* and finds all the other solutions by applying sequences of partial reflections in the symmetry vertices [29]. Another interesting feature is that such symmetries also have afforded an environment to solve the DMDGP using parallel computing. For inatnce, Fidalgo *et al.* [13] have proposed an efficient manner of splitting an instance depending only on the symmetry vertices, minimizing the number of pruning restraints yet to be checked after the joining of the parts.

Liberti *et al.* [24] generalize the DMDGP for any $\mathbb{R}^K$ defining the ^{K}DMDGP, an acronym for *K-Discretizable Molecular Distance Geometry Problem*, and they also restated *sym*BP as an outline method to handle the K-dimensional case, where they suggest the resolution of the sphere intersection problem following [6]. A complete survey about Euclidean Distance Geometry (with a complete taxonomy) can be found in [24].

There are some works in literature that connect Clifford Algebra and Distance Geometry, taking advantage of the coordinate-free character and the geometrical interpretation of blades and rotors in the *Conformal Geometric Algebra* (CGA), *e.g.*, [1–3,8,16,17,34]. To the best of our knowledge, neither the ^{K}DMDGP nor the *sym*BP are wholly described in terms of CGA in the literature so far. The advantage of having such a description is allow the use, under the same mathematical formalism, of concepts coming from formalisms that are typically treated in a decoupled form, such as matrix algebra, Plucker coordinates, complex numbers, quaternion algebra, intersection of two or more objects (not necessarily rounds or flats, only), among others. Thus, the main goal of the present work is to provide a complete theoretical description of the ^{K}DMDGP with exact distances, for any K, using CGA, as it is a useful domain to treat spheres, planes, lines, and motions in a creative and satisfactory way. Moreover, a first attempt to solve it completely in this environment is represented by an algorithm that mimics *sym*BP, but using the closed-form sphere-intersection representation of CGA. We call it C*sym*BP. In order to test it when applied to artificial instances, we present and discuss some computational issues of the C*sym*BP.

This paper is organized as follows. The ^{K}DMDGP is presented in Section 2 and CGA, together with its proper operations for the ^{K}DMDGP, is shortly presented in Section 3. Section 4 brings the most significant contribution of this work, focusing on the theoretical description of the problem via CGA. Section 5 presents the C*sym*BP algorithm, shows preliminary computational results, and discusses its strength and weakness. Finally, Section 6 draws conclusions and points to promising directions for future works.

2. K–Discretizable Molecular Distance Geometry Problem

Let $G = (V, E, d)$ be an undirected and simple graph, with $n = |V|$ vertices, whose edges are weighted by $d : E \longrightarrow \mathbb{R}_+$, and $N_G(v) = \{u \in V \mid \{u, v\} \in E\}$ be the *Neighbourhood* of v, for any $v \in V$. Moreover, given a total order relation $<$ on V, let $\gamma_G(v) = \{u \in V \mid u < v\}$ be the set of all *predecessors* of v, $\mathcal{U}_v = N_G(v) \cap \gamma_G(v)$ be the set of the *adjacent predecessors* of v, $\rho_G(v) = |\gamma_G(v)| + 1$ be the *rank* of v in G with respect to $<$ and, at last, let Equations. (2) and (3) be, respectively, the *Cayley-Menger Formula*

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

and the *Cayley-Menger Determinant*

$$CM(\mathcal{U}) = \begin{vmatrix} 0 & 1 & 1 & \cdots & 1 \\ 1 & 0 & d_{0,1}^2 & \cdots & d_{0,K}^2 \\ 1 & d_{0,1}^2 & 0 & \cdots & d_{1,K}^2 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & d_{0,K}^2 & d_{1,K}^2 & \cdots & 0 \end{vmatrix}, \tag{3}$$

whose non-zero entries are the Euclidean distances in $\mathbb{R}^K$ between all the pairs of points from a set $\mathcal{U} = \{P_0, P_1, \ldots, P_K\}$ of $K + 1$ point elements.

Suppose that $<$ in V satisfies the following assumptions:

(i) $\mathcal{U}_{v_i}$ contains exactly the K immediate adjacent predecessors to v_i;
(ii) $G[\mathcal{U}_{v_i}]$, *i.e.,* the subgraph induced by $\mathcal{U}_{v_i}$, is a K-clique on G; and
(iii) $\Delta_{K-1}(\mathcal{U}_{v_i}) > 0, \quad \forall i > K$.

The ^{K}DMDGP associated to G is the problem of looking for embeddings $\mathbf{x} : V \longrightarrow \mathbb{R}^K$, which are called *Realizations* of G, such that the distance constraints from Equation (1) are satisfied by d.

Furthermore, the set of all realizations of G in $\mathbb{R}_K$ (modulo translations and rotations) is denoted by $\mathcal{X}_G^K$ and we also say that $x, y \in \mathcal{X}_G^K$ are *congruent* if and only if $\|\mathbf{x}_u - \mathbf{x}_v\| = \|\mathbf{y}_u - \mathbf{y}_v\|$, for all $\{u, v\} \in E$, what is denoted as $\mathbf{x} \equiv \mathbf{y}$ [25]. It is possible to see that $\equiv$ is an equivalence relation which performs a congruence-based partition on $\mathcal{X}_G^K$, from what one can find a finite number of incongruent realizations (representing all of them) [25] and whose set will be denoted in this work as $\mathcal{S}_G^K = \mathcal{X}_G^K / \equiv$.

It is remarkable that determining each vertex is equivalent to the problem of computing the intersection of K distinct hyperspheres in $\mathbb{R}^K$ which has, at most, two positions [6]. Besides, Liberti *et al.* [24] provided a convenient split $E = E_D \cup E_P$, such that $E_D \cap E_P = \emptyset$. $E_D = \{\{u, v\} \in E \mid u \in \mathcal{U}_v\}$ is the set of the *Discretization Edges* (which is complete), as it is directly associated to (i), and $E_P = E \setminus E_D$ is the set of the (additional) *Pruning Edges*, which can be empty, as they are not guaranteed from Assumptions (i)–(iii).

Some remarks can be driven from E_P. First, the occurrence of a pruning edge incident in a vertex implies that it can be positioned uniquely (intersection of $K+1$ hyperspheres). Second, additional edges are responsible for determining the symmetry set $S_G = \{v \in V : \nexists \{u, w\} \in E \text{ such that } u + K < v \leq w\}$, that determines the symmetry hyperplanes which are, respectively, uniquely defined by the positions of the K immediate predecessors in the current realization [25]. Additionally, it provides a deterministic way to know the number of realizations (modulo translations and rotations) for G via the expression $|\mathcal{X}_G^K| = 2^{|\mathcal{S}_G^K|}$ [29], *e.g.*, if $n = 20$ and three symmetry vertices, namely 4, 11, and 16, then it has exactly $|\mathcal{X}_G^K| = 8$ feasible conformations (modulo translations and rotations). Also, they define infeasibilities in the problem, which are important to guarantee that the tree is not going to grow to much, as it increases in power of 2. Finally, this binary structure provides a tree-structure to explore the search space which will be denoted here as $\mathcal{T}_G$. A possible tree associated with the previous example is depicted in Fig. 1.

Liberti *et al.* [24] restated BP and BP*one* for any $\mathbb{R}^K$ (Algorithm 1, where $S^{K-1}(\mathbf{c}, r)$ is the sphere entered at $\mathbf{c}$ with radius $r > 0$). Also, after extracting both possible positions in the branching phase, *sym*BP prunes to choose the right one. *Direct Distance Feasibility* (DDF) is the pruning device used here, by checking if each calculated distance of the q-th vertex is 'close' to the available distance data, that is,

$$|d_{i,q} - \|\mathbf{x}_i - \mathbf{x}_q\|\,| < \varepsilon, \qquad \forall i < q \quad \text{and} \quad \{i, q\} \in E, \quad \text{for} \quad \varepsilon > 0.$$

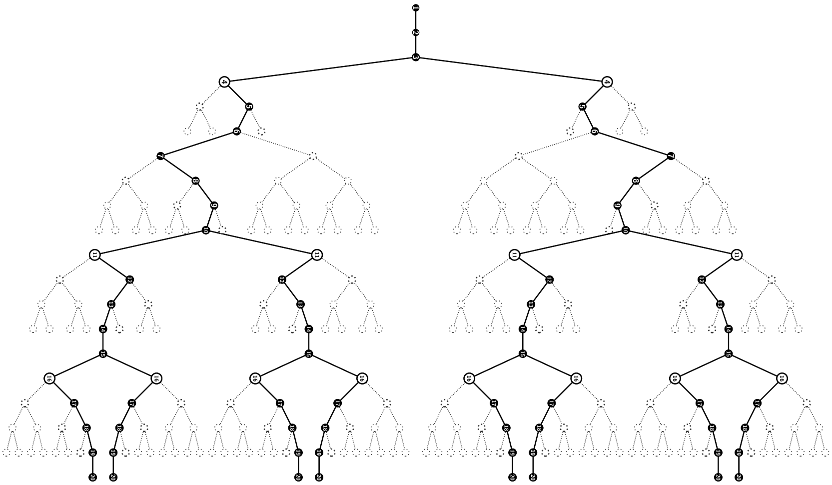

FIGURE 1. An exampling $\mathcal{T}_G$ of a ^{3}DMDGP with 20 vertices and 3 symmetry vertices, whose ranks are 4, 11, and 16 [13].

Algorithm 1 BP*one* algorithm

Require: $v \in V \setminus \{1, \ldots, K\}$ and an embedding $\overline{\mathbf{x}} = \mathbf{x}'$ for $G[\gamma_G(v)]$.

function BPONE(v, $\overline{\mathbf{x}}$)
    $P \leftarrow \bigcap_{u \in N_G(v), u<v} S^{K-1}(\overline{\mathbf{x}}_u, d_{u,v})$
    for $\mathbf{x}_v \in P$ **do**
        $\mathbf{x} \leftarrow (\overline{\mathbf{x}}, \mathbf{x}_v)$
        if $v = n$ **then**
            return *success*, $\mathbf{x}$
        end if
        status, $\mathbf{y} \leftarrow$ BPONE($v+1$, $\mathbf{x}$)
        if status = *success* **then**
            return *success*, $\mathbf{y}$
        end if
    end for
    return *fail*
end function

Other works from literature present alternative approaches to deal with hypersphere intersection and, then, to handle the ^{K}DMDGP. For instance, Coope uses LU factorization [6], Gonçalves adopts a least-squares approach [14], and Maioli *et al.* works with QR factorization [26].

3. Conformal Geometric Algebra of $\mathbb{R}^K$

Geometric Algebra (GA) is the name given by David Hestenes for Clifford Algebra (CA), intending to emphasize that geometric objects in various geometric models can be suitably described by algebraic operations and axioms. CGA represents the Conformal Geometric Model of $\mathbb{R}^K$ in terms of GA, introduced by Friederich Wachter (1792–1817) by noting that there is a surface in the Hyperbolic Space which is metrically equivalent to the Euclidean space, the so-called *Horosphere*. The associated algebra is described as follows, and one can find all details about that in [9, 20, 30].

The basis for the Minkowski Space $\mathbb{R}^2$ with signature $(1,1)$ ($\mathbb{R}^{1,1}$) is given by $\{e_+, e_-\}$, such that $e_+^2 = +1$ and $e_-^2 = -1$. So, the *Conformal Split* $\mathbb{R}^{K+1,1}$ of $\mathbb{R}^{K+2}$ is defined as the direct sum $\mathbb{R}^{K+1,1} = \mathbb{R}^K \oplus \mathbb{R}^{1,1}$, whose basis is $\{e_1, \ldots, e_K, e_+, e_-\}$. A very important structure in $\mathbb{R}^K$ is the *Null Cone* $\mathbb{K}^{K+1}$, defined by the null vectors $X \in \mathbb{R}^{K+1,1}$ such that $X^2 = X \cdot X = 0$, where $\cdot$ denotes the inner product. The two most important null vectors defined here are $e_\infty \triangleq e_+ + e_-$ and $e_o \triangleq \frac{1}{2}(e_+ - e_-)$, where $e_\infty^2 = e_o^2 = 0$ and $e_\infty \cdot e_o = -1$. Furthermore, the *Conformal Model* $\mathbb{H}_a^K$ of $\mathbb{R}^K$ is its embedding into $\mathbb{R}^{K+1,1}$, provided by the operator $\mathcal{C} : \mathbb{R}^K \longrightarrow \mathbb{K}^{K+1}$, established by

$$\mathcal{C}(\mathbf{x}) = \mathbf{x} + \frac{1}{2}\mathbf{x}^2 e_\infty + e_o. \tag{4}$$

Now, let $X = \mathcal{C}(\mathbf{x}), Y = \mathcal{C}(\mathbf{y}) \in \mathbb{H}_a^K$. Considering the Minkowski norm $\|\cdot\|_M$ and the inner product of vectors in $\mathbb{H}_a^K$, given by $X \cdot Y = -\frac{1}{2}\|\mathbf{x} - \mathbf{y}\|^2$, the operator $\tilde{\mathcal{C}} : \mathbb{R}^K \longrightarrow \mathbb{H}_a^K$, set by $\tilde{\mathcal{C}}(\mathbf{x}) = \mathcal{C}(\mathbf{x}) = X$, is an isometry. This is verified, for $\tilde{\mathcal{C}}(\mathbf{x})$ is onto and satisfies $||X - Y||_M^2 = -2X \cdot Y = \|\mathbf{x} - \mathbf{y}\|^2$, what is a very important fact for the connection between CGA and a ^{K}DMDGP. As $\tilde{\mathcal{C}}$ is invertible, we then have that $\tilde{\mathcal{C}}^{-1} : \mathbb{H}_a^K \longrightarrow \mathbb{R}^K$, which is pontually defined by $\tilde{\mathcal{C}}^{-1}(X) = \mathbf{x}$.

Therefore, the Conformal Algebra $\mathcal{C}l_{K+1,1}$ consists on the conformal model $\mathbb{H}_a^K$ with the *Geometric Product* (or *Clifford Product*) of vectors given by

$$XY = X \cdot Y + X \wedge Y, \tag{5}$$

as $\wedge$ denotes the the *Wedge Product* of vectors X and Y (see Grassmann [15]). Further special elements, important operations, and properties of this algebra that will be widely used in this article are described, what is necessary for its interdisciplinary appeal. The wedge product of a finite number of vectors $B = X_1 \wedge \ldots \wedge X_k$ is named a *k-Blade*, where k is the *Grade* of B, denoted by $\mathrm{gr}(B)$. If the vectors are linearly independent, the geometric interpretation of B is a k-dimensional subspace. One interesting example is the unit *Pseudoscalar* I, which represents the attitude of the whole $\mathbb{R}^{K+2}$.

A meaningful operation here is the *Left Contraction* (or simply *Contraction*) of multivectors in $\mathcal{C}l_{K+1,1}$, denoted by $X \rfloor Y$. When applied to blades, the contraction returns a subspace of $\mathbb{R}^{K+2}$, with grade $\mathrm{gr}(Y) - \mathrm{gr}(X)$, that is included in Y and is orthogonal to X. Another important operation is the *Reverse* of the k-blade B, given by $\tilde{B} = X_k \wedge \ldots \wedge X_2 \wedge X_1 = (-1)^{k(k-1)/2} B$. Those operations are employed in the definition of the *Squared Reverse Norm* of a blade A, by $\|A\|^2 = A\tilde{A} = A\rfloor\tilde{A}$, and for the *Inverse* of A, by $A^{-1} = \tilde{A}/\|A\|^2$, where / denotes the *Inverse Geometric Product.*

In order to describe the orthogonal complement of the subspace determined by the K-blade A, a duality relation is established. So, the *Dual* of A is defined by $A^* = A\rfloor I^{-1}$. And, as $A^{**} \neq A$, the *Undual* of A is given by $A^{-*} = A\rfloor I$. It is easy to see that $(A^*)^{-*} = A$ and that $(A \wedge B)^{-*} = A\rfloor B^{-*}$, for another blade B.

In this paper, geometric entities are always represented in the *Inner-Product Null-Space* (IPNS) paradigm, which is important to remark in order to exclude possible ambiguities. Thus, the hypersphere σ with radius r and center X is algebraic represented in $\mathcal{C}l_{K+1,1}$ by $\sigma = X - \frac{1}{2}r^2 e_\infty$ and that $X_1 \wedge X_2 \wedge X_3 \wedge \cdots \wedge X_K \wedge e_\infty$ is the dual representation of the hyperplane π, uniquely determined by the family of points $\{X_i\}$. Moreover, a point $X \in \mathbb{H}_a^K$ is such that $X \in \sigma$ if and only if $X \cdot \sigma = 0$. A crucial and original outcome, then, is stated.

Proposition 3.1. *If $\sigma_1, \cdots, \sigma_K$ are the hyperspheres $\sigma_i = X_i - \frac{1}{2}r_i e_\infty$ in $\mathbb{H}_a^K$, then the hyperplane π, uniquely determined by $\{X_i\}$, can be represented by*

$$\pi = (\sigma_1 \wedge \cdots \wedge \sigma_K)\rfloor\left(e_\infty \rfloor I\right). \tag{6}$$

Proof. Using the wedge product properties, we have

$$\begin{aligned}
\sigma_1 \wedge \sigma_2 \wedge \cdots \wedge \sigma_K = \quad & X_1 \wedge X_2 \wedge X_3 \wedge \cdots \wedge X_K \\
& + (-1)^1 \frac{1}{2} r_1^2 (e_\infty \wedge X_2 \wedge X_3 \wedge \cdots \wedge X_K) \\
& + (-1)^2 \frac{1}{2} r_2^2 (e_\infty \wedge X_1 \wedge X_3 \wedge \cdots \wedge X_K) \\
& \vdots \\
& + (-1)^K \frac{1}{2} r_K^2 (e_\infty \wedge X_1 \wedge X_2 \wedge \cdots \wedge X_{K-1}).
\end{aligned} \tag{7}$$

By applying the wedge product by e_∞ on the right-side of Equation (7), the dual representation of the hyperplane π ends up to be

$$\sigma_1 \wedge \sigma_2 \wedge \cdots \wedge \sigma_K \wedge e_\infty = X_1 \wedge X_2 \wedge X_3 \wedge \cdots \wedge X_K \wedge e_\infty = \pi^*. \tag{8}$$

Finally, using the Equation (8), the undualization with its properties and the blade $D = \sigma_1 \wedge \cdots \wedge \sigma_K$, we can conclude that

$$\pi = (\pi^*)^{-*} = (D \wedge e_\infty)^{-*} = D\rfloor e_\infty^{-*} = D\rfloor(e_\infty \rfloor I), \tag{9}$$

completing the proof. Here, D is a K-blade interpreted as the pair of points resulting from the intersection of K spheres. □

Given vectors $X, Y \in \mathbb{H}_a^K$ interpreted as normalized finite points, written according to Equation (4), their *Perpendicular Bisector* is the hyperplane π that can be computed using $\pi = X - Y$. If one knows π and wants to extract both points X and Y from the pair of points D, one can use $D^* = X \wedge Y$ and the expressions presented by Dorst *et al.* [9, Eq. (14.13), p. 427]

$$X = \frac{D^* + \sqrt{D^* D^*}}{-e_\infty \rfloor D^*} \quad \text{and} \quad Y = \frac{D^* - \sqrt{D^* D^*}}{-e_\infty \rfloor D^*}. \tag{10}$$

4. The CGA Version of the symBP for Solving a ^{K}DMDGP

Let $G = (V, E, d)$ be a ^{K}DMDGP graph with n vertices and $\mathbf{x} = (\mathbf{x}_i) \in \mathcal{X}_G$, an arbitrary realization of G in $\mathbb{R}^K$, for $n > K > 0$. As it is possible to biunivocally identify each $\mathbf{x}_i$ to a $X_i \in \mathbb{H}_a^K$, for $\tilde{\mathcal{C}}(\mathbf{x}_i) = X_i$, the sequence $X = (X_i)$ is defined and, then, the following result can be stated.

Theorem 4.1. *The sequence $X = (X_i)$ is a realization of G in the metric space $(\mathbb{H}_a^K, d_M)$, where d_M is the distance provided by the norm $\|\cdot\|_M$.*

Proof. It follows directly from the facts that, for each i, $X_i = \tilde{\mathcal{C}}(\mathbf{x}_i)$ and that the operator $\tilde{\mathcal{C}}$ is an isometry, preserving the same distances and constraints as the associated Euclidean realization. □

In this section, we present the main original contribution of the present theoretical work, which gives another example of the relation between both GA and DG research areas. In this way, all the ^{K}DMDGP compounds, with exact distances, will be merged entirely into CGA terms, including discretization, pruning, and symmetries. Also, some preliminary computational issues about this are displayed.

4.1. Results of discretization edges and pair of points determination

As discussed in Section 2, discretization edges ensure that there exist two possible positions for the j-th vertex, namely $\mathbf{x}'_j$ and $\mathbf{x}''_j$, since the $j-K$ immediate predecessors are all determined and whose positions are $\mathbf{x}_{j-K}, \ldots, \mathbf{x}_{j-1}$, for $j = K+1, \ldots, n$. In addition, both $\mathbf{x}'_j$ and $\mathbf{x}''_j$ lie in the intersection of the (Euclidean) hyperspheres S_{ij}, centered in $\mathbf{x}_i$ and whose radius is $r_{ij} = d_{i,j}$. Also, they are symmetric w.r.t the hyperplane Π_j, uniquely determined by $\mathbf{x}_i$, for all $i = j-K, \ldots, j-1$ [24]. The conformal version of the hypersphere S_{ij} will be denoted in this article by $\sigma_i^{(j)}$.

Proposition 4.2. *The primal representation of Π_j, embedded in $\mathbb{H}_a^K$, can be written as the odd versor*

$$\pi_j = D_j \rfloor (e_\infty \rfloor I), \tag{11}$$

where $D_j = \sigma_{j-K}^{(j)} \wedge \sigma_{j-K+1}^{(j)} \wedge \ldots \wedge \sigma_{j-1}^{(j)}$.

Proof. It is enough to apply Proposition 3.1 to the hyperspheres $\sigma_{j-K}^{(j)}, \sigma_{j-K+1}^{(j)}, \ldots, \sigma_{j-1}^{(j)}$ and taking D_j as stated, one is able to attain what is desired. □

It is possible, now, to express explicitly both possible positions for the representatives of $\mathbf{x}'_j$ and $\mathbf{x}''_j$ in $\mathbb{H}^K_a$ and affirm what comes next.

Proposition 4.3. *Let $X'_j = \mathcal{C}(\mathbf{x}'_j)$ and $X''_j = \mathcal{C}(\mathbf{x}''_j)$. Both conformal points can be calculated from $\sigma^{(j)}_{j-K}, \sigma^{(j)}_{j-K+1}, \ldots, \sigma^{(j)}_{j-1}$ by the expressions*

$$X'_j = \frac{D^*_j + \sqrt{D^*_j D^*_j}}{-e_\infty \rfloor D^*_j} \qquad \text{and} \qquad X''_j = \frac{D^*_j - \sqrt{D^*_j D^*_j}}{-e_\infty \rfloor D^*_j}, \tag{12}$$

where $D_j = \sigma^{(j)}_{j-K} \wedge \sigma^{(j)}_{j-K+1} \wedge \ldots \wedge \sigma^{(j)}_{j-1}$.

Proof. As in Proposition 4.2, one knows D_j *a priori* and its dual D^*_j can then be computed. Thus, it is sufficient to apply the formulas from Equation (10). □

In this context, as in the Euclidean version of the ^{K}DMDGP, both points are one reflection of the other through the hyperplane Π_j (primarily represented by the versor π_j) which also defines the perpendicular bisector of them. It is asserted as follows.

Proposition 4.4. *Let X'_j and X''_j be the two points in $\mathbb{H}^K_a$ separated by the symmetry plane Π_j, mentioned in the beginning of this section. The odd versor $\pi_j = D_j \rfloor (e_\infty \rfloor I)$ is such that*

$$X'_j = -\pi_j X''_j \pi_j^{-1} \qquad \text{and} \qquad X''_j = -\pi_j X'_j \pi_j^{-1}. \tag{13}$$

Proof. By definition,

$$D^*_j = X'_j \wedge X''_j. \tag{14}$$

Then, the result of e_∞ contracted on D^*_j is exactly π_j, since

$$e_\infty \rfloor D^*_j = e_\infty \rfloor \left(X'_j \wedge X''_j\right) = \left(e_\infty \rfloor X'_j\right) \wedge X''_j - X'_j \wedge \left(e_\infty \rfloor X''_j\right) = X'_j - X''_j.$$

The result of squaring D^*_j is the square of $X'_j \cdot X''_j$ for

$$\begin{aligned} D^*_j D^*_j &= \left(X'_j \wedge X''_j\right)\left(X'_j \wedge X''_j\right) = \left(X'_j \wedge X''_j\right) \rfloor \left(X'_j \wedge X''_j\right) \\ &= X'_j \rfloor \left(X''_j \rfloor \left(X'_j \wedge X''_j\right)\right) = X'_j \rfloor \left(\left(X''_j \rfloor X'_j\right) \wedge X''_j - X'_j \wedge \left(X''_j \rfloor X''_j\right)\right) \\ &= \left(X'_j \cdot X''_j\right)^2. \end{aligned} \tag{15}$$

Finally, we demonstrate that from Equations (12) one can get Equations (13) and vice-versa:

$$\begin{aligned} X'_j = \frac{D^*_j + \sqrt{D^*_j D^*_j}}{-e_\infty \rfloor D^*_j} &= -\left(D^*_j + \sqrt{D^*_j D^*_j}\right)\left(e_\infty \rfloor D^*_j\right)^{-1} \\ &= -\left(D^*_j + \sqrt{D^*_j D^*_j}\right)\pi_j^{-1} \\ &= -\left(X'_j \wedge X''_j + X'_j \cdot X''_j\right)\pi_j^{-1} \\ &= -\left(X'_j X''_j\right)\pi_j^{-1} \\ &= -\left(\left(X'_j - X''_j\right) X''_j\right)\pi_j^{-1} \end{aligned}$$

$$= -\pi_j X_j'' \pi_j^{-1}.$$

The second expression in Equation (12) is completely analogous. □

Next theorem states that symmetric realizations $\mathbf{x}$ and $\mathbf{y}$ of a ^{K}DMDGP in $\mathbb{R}^K$, in terms of partial reflections, are biunivocally related to realizations X and Y in $\mathbb{H}_a^K$ which are also symmetric each other up to partial reflections, preserving all distances.

Theorem 4.5. *Let $G = (V, E, d)$ be a ^{K}DMDGP graph, $\mathbf{x} \in \mathcal{X}_G$, j be a fixed vertex in $V \setminus \{1, \ldots, K\}$, $p, q \in V$ such that $j < p < q$ and π_j the hyperplane associated to $\mathbf{x}_j$. If $\mathbf{x}_p''$ and $\mathbf{x}_q''$ are symmetric to $\mathbf{x}_p'$ and $\mathbf{x}_q'$ w.r.t. π_j, respectively, then*

$$\left\|X_q'' - X_p''\right\|_M^2 = \left\|X_q' - X_p'\right\|_M^2. \tag{16}$$

Proof. The isometry implies that

$$\| X_q'' - X_p'' \|_M^2 = \| \mathbf{x}_q'' - \mathbf{x}_p'' \|^2 = \| \mathbf{x}_q' - \mathbf{x}_p' \|^2 = \| X_q' - X_p' \|_M^2. \tag{17}$$

□

4.2. Results of pruning edges and partial reflections

Also, from Section 2, each additional edge incident on the current vertex guarantees the uniqueness of the position. Thus, any algorithmic method to extract the realizations by sphere intersections must use those data. Therefore, we can model DDF in CGA fashion, as in Algorithm 2.

Let $\mathbf{x}_i$ be a position for the i-th vertex and $\{i, q\} \in E_P$ such that $q > K + 1$ and $|q - i| \geq K + 1$.

Theorem 4.6. *A possible position $\mathbf{x}_q$, from the q-th vertex of G, is feasible w.r.t the given pruning edge $\{p, q\}$ and position $\mathbf{x}_p$ if and only if $X_q \in \sigma_i$, where $\sigma_i = X_i - \frac{1}{2} d_{i,q}^2 e_\infty$.*

Proof. It is clear by definition that $\mathbf{x}_q$ is feasible (w.r.t $\{p, q\}$ and $\mathbf{x}_p$) if and only if $\mathbf{x}_q \in S_p^{K-1}$, where S_p^{K-1} is an hypersphere centered in $\mathbf{x}_p$ with radius $d_{p.q}$. But, S_p^{K-1}, $\mathbf{x}_q$ and $\mathbf{x}_p$ are, respectively, represented in $\mathbb{H}_a^K$ by the vectors σ_p, X_q and X_p. That is, we can restate that $\mathbf{x}_q$ is feasible if and only if $X_q \in \sigma_p$. On the other hand, $X_q \in \sigma_p$ if and only if $X_q \cdot \sigma_p = 0$, which is true, since

$$X_q \cdot \sigma_p = X_q \cdot \left(X_p - \frac{1}{2} d_{pq}^2 e_\infty \right) = X_q \cdot X_p - \frac{1}{2} d_{pq}^2 \left(X_q \cdot e_\infty \right) \tag{18}$$

and $X_q \cdot X_p = -\frac{1}{2} d_{pq}^2$ and $X_q \cdot e_\infty = -1$, completing the result. □

It implies that DDF is reduced to $X_q \cdot \sigma_p = 0$, a dot product evaluation. A corollary follows directly from Theorem 4.6.

Corollary 4.7. *Let $\{i_1, q\}, \ldots, \{i_w, q\}$ be all the pruning edges inciding into the q-th vertex. If $\mathbf{x}_q$ is a feasible position, then $X_q \in \sigma_{i_1}^{(q)} \wedge \ldots \wedge \sigma_{i_w}^{(q)}$.*

Algorithm 2 CGA Direct Distance Feasibility

```
function CDDF(q, E_P^q, X)
    if E_P^q = ∅ then
        return success
    else
        S ← ⋀_{p<q} σ_p^{(q)}
        Y ← X · S
        if Y = 0 then
            return success
        else
            return fail
        end if
    end if
end function
```

5. Computational Issues

Algorithm 3 outlines the C*sym*BP procedure, an algorithm that mimics *sym*BP *ipsis litteris*, all fashioned using CGA.

In Section 5.1, we compare the computational cost of C*sym*BP against the combination of the *sym*BP algorithm [29] with the computation of the points of intersection of K hyperspheres using the QR matrix decomposition-based approach proposed by Coope [6]. More specifically, we present a theoretical comparison in terms of the number of arithmetic operations (*i.e.,* additions/subtractions, multiplications, and divisions) necessary to evaluate the wedge products in line 2 of Algorithm 3 and the number of operations evaluated by Coope's approach. In addition, Section 5.2 presents a comparison of executions times of the implementation of both algorithms in Julia Programming Language [4], using the Liga library [11].

Our implementation of both algorithms first looks for one solution without exploiting redundant paths in the implicit tree-structure that explores the search space (*e.g.,* a dark path in Figure 1) and then apply the symmetries to determine all solutions (*e.g.,* all other valid paths in Figure 1). We have implemented the algorithms in an iterative fashion rather than by using their recursive counterpart in other to avoid overflow in the system's stack. These implementation cancels previous unexplored BP calls on the moment that a solution is discovered. The experiments were performed on an Intel® Pentium® CPU G3240 with 3.10 GHz×2 and 4 Gb of RAM, running Ubuntu 16.04 operating system.

5.1. Computational Cost

In the following derivations we assume that blades are encoded by multivectors rather than by collections of vector factors. Thus, the K-blade D_j in Algorithm 3 (line 2) has $\binom{K+2}{K} = (K+2)(K+1)/2$ components, written as:

$$D_j = \sigma_{j-K}^{(j)} \wedge \ldots \wedge \sigma_{j-1}^{(j)}$$

$$
\begin{aligned}
&= \left(\mathbf{x}_{j-K} + e_o + \beta^{(j)}_{j-K} e_\infty\right) \wedge \ldots \wedge \left(\mathbf{x}_{j-1} + e_o + \beta^{(j)}_{j-1} e_\infty\right)\\
&= \quad \mathbf{x}_{j-K} \wedge \ldots \wedge \mathbf{x}_{j-1}\\
&\quad + \gamma_K\, \mathbf{x}_{j-K+1} \wedge \ldots \wedge \mathbf{x}_{j-1} \wedge e_o\\
&\qquad\qquad \vdots\\
&\quad + \gamma_1\, \mathbf{x}_{j-K} \wedge \ldots \wedge \mathbf{x}_{j-2} \wedge e_o\\
&\quad + \gamma_K \beta^{(j)}_{j-K}\, \mathbf{x}_{j-K+1} \wedge \ldots \wedge \mathbf{x}_{j-1} \wedge e_\infty\\
&\qquad\qquad \vdots\\
&\quad + \gamma_1 \beta^{(j)}_{j-1}\, \mathbf{x}_{j-K} \wedge \ldots \wedge \mathbf{x}_{j-2} \wedge e_\infty\\
&\quad + \gamma_{K,K-1} \left(\beta^{(j)}_{j-K} - \beta^{(j)}_{j-K+1}\right) \mathbf{x}_{j-K+2} \wedge \ldots \wedge \mathbf{x}_{j-1} \wedge e_o \wedge e_\infty\\
&\qquad\qquad \vdots\\
&\quad + \gamma_{2,1} \left(\beta^{(j)}_{j-2} - \beta^{(j)}_{j-1}\right) \mathbf{x}_{j-K} \wedge \ldots \wedge \mathbf{x}_{j-3} \wedge e_o \wedge e_\infty,
\end{aligned}
\tag{19}
$$

where $\mathbf{x}_i = \alpha_{i,1} e_1 + \alpha_{i,2} e_2 + \cdots + \alpha_{i,K} e_K$ are the (Euclidean) support vectors of the centers of the hypersheres σ_i, $\beta^{(j)}_i = (\mathbf{x}_i \cdot \mathbf{x}_i - (r^{(j)}_i)^2)/2$ are scalar values computed once for each hypersphere, and γ_k and $\gamma_{k,l}$ are constant values that assume ± 1 according to the application of the antisymmetry property of the wedge product. In Equation (19), the $\mathbf{x}_{j-K} \wedge \ldots \wedge \mathbf{x}_{j-1}$ term produces the component of D_j having the Euclidean pseudoscalar $e_1 \wedge \ldots \wedge e_K$ as basis blade. The terms $\gamma_k\, \mathbf{x}_{j-K} \wedge \ldots \wedge \breve{\mathbf{x}}_{j-k} \wedge \ldots \wedge \mathbf{x}_{j-1} \wedge e_o$ and $\gamma_k \beta^{(j)}_{j-k}\, \mathbf{x}_{j-K} \wedge \ldots \wedge \breve{\mathbf{x}}_{j-k} \wedge \ldots \wedge \mathbf{x}_{j-1} \wedge e_\infty$ produce the $2K$ components of D_j whose basis blades are spanned by $K-1$ Euclidean basis vectors and by, respectively, e_o or e_∞. Here, $\breve{\mathbf{x}}_{j-k}$ denote that $\mathbf{x}_{j-k}$ was removed from the product. Finally, the $\gamma_{k,l}(\beta^{(j)}_{j-k} - \beta^{(j)}_{j-l})\, \mathbf{x}_{j-K} \wedge \ldots \wedge \breve{\mathbf{x}}_{j-k} \wedge \ldots \wedge \breve{\mathbf{x}}_{j-l} \wedge \ldots \wedge \mathbf{x}_{j-1} \wedge e_o \wedge e_\infty$ terms result in $K(K-1)/2$ components of the multivector after evaluating the wedge products and collecting the basis blades spanned by $K-2$ Euclidean basis vector and by $e_o \wedge e_\infty$.

Existing Geometric Algebra source code optimizers and libraries that implement the lazy-evaluation strategy [33], like, respectively, the Geometric Algebra ALgorithms OPtimizer (GAALOP) [5] and the Geometric Algebra Template Library (GATL) [12], evaluate the wedge product in such a way that the computation naturally reduces to the same operations than the evaluation of the minors of a matrix whose rows are the vector factors of the blade. As a result, by using such existing computational solutions, the number of arithmetic operations required to evaluate $\mathbf{x}_{j-K} \wedge \ldots \wedge \mathbf{x}_{j-1}$ in Equation (19) is the same of calculate a $K \times K$ determinant by way of cofactor expansion [10]:

$$
N_{det} = K! \left(1 + \sum_{t=1}^{K} \frac{1}{t!}\right) - 2. \tag{20}
$$

Algorithm 3 C*sym*BP algorithm

Require: $v \in V \setminus \{1, \ldots, K\}$, an embedding $\overline{X} = X'$ for $G[\gamma_G(v)]$, and a set E_P^v with the edges incident in v.

function C*sym*BP(v, $\overline{X}$, E_P^v)
    $D_j \leftarrow \sigma_{j-K}^{(j)} \wedge \sigma_{j-K+1}^{(j)} \wedge \ldots \wedge \sigma_{j-1}^{(j)}$
    for $X_v \in D_j$ **do**
        if CDDF(v, E_P^v, X_v) = *success* **then**
            $X \leftarrow (\overline{X}, X_v)$
            if $v = n$ **then**
                return *success*, X
            end if
            status, $Y \leftarrow$ C*sym*BP($v+1$, X, E_P^{v+1})
            if status = *success* **then**
                return *success*, Y
            end if
        end if
    end for
    return *fail*
end function

Given the recursive nature of cofactor expansion, some of the minors computed during the process are precisely the scalar values resulting from the wedge product of Euclidean vectors in the terms $\gamma_k \, \mathbf{x}_{j-K} \wedge \ldots \wedge \check{\mathbf{x}}_{j-k} \wedge \ldots \wedge \mathbf{x}_{j-1} \wedge e_o$, $\gamma_k \beta_{j-k}^{(j)} \, \mathbf{x}_{j-K} \wedge \ldots \wedge \check{\mathbf{x}}_{j-k} \wedge \ldots \wedge \mathbf{x}_{j-1} \wedge e_\infty$, and $\gamma_{k,l}(\beta_{j-k}^{(j)} - \beta_{j-l}^{(j)}) \, \mathbf{x}_{j-K} \wedge \ldots \wedge \check{\mathbf{x}}_{j-k} \wedge \ldots \wedge \check{\mathbf{x}}_{j-l} \wedge \ldots \wedge \mathbf{x}_{j-1} \wedge e_o \wedge e_\infty$ of Equation (19). As a result, the evaluation of those terms does not impose additional complexity to the computation of D_j, except by multiplications and additions/subtractions required while scaling by $\beta_{j-k}^{(j)}$ and $(\beta_{j-k}^{(j)} - \beta_{j-l}^{(j)})$, and collecting the basis blades of the K-vector space. It is important to notice that γ_k and $\gamma_{k,l}$ does not impose the evaluation of multiplications. Those constant terms only switch from addition to subtraction and vice-versa according to rules that can be deduced in function of the indices of vectors $\mathbf{x}_k$ and $\mathbf{x}_l$, suppressed from the products.

For the $\gamma_k \, \mathbf{x}_{j-K} \wedge \ldots \wedge \check{\mathbf{x}}_{j-k} \wedge \ldots \wedge \mathbf{x}_{j-1} \wedge e_o$ terms,

$$N_o = K^2 - K \tag{21}$$

additions/subtractions are required while collecting the basis blades. The $\gamma_k \beta_{j-k}^{(j)} \, \mathbf{x}_{j-K} \wedge \ldots \wedge \check{\mathbf{x}}_{j-k} \wedge \ldots \wedge \mathbf{x}_{j-1} \wedge e_\infty$ terms also require $K\,(K-1)$ additions/subtractions, plus K^2 multiplications for scaling by $\beta_{j-k}^{(j)}$, leading to:

$$N_\infty = 2K^2 - K \tag{22}$$

arithmetic operations. The $\gamma_{k,l}(\beta_{j-k}^{(j)} - \beta_{j-l}^{(j)}) \, \mathbf{x}_{j-K} \wedge \ldots \wedge \check{\mathbf{x}}_{j-k} \wedge \ldots \wedge \check{\mathbf{x}}_{j-l} \wedge \ldots \wedge \mathbf{x}_{j-1} \wedge e_o \wedge e_\infty$ terms impose $K^2\,(K-1)^2/4$ additions or subtractions and $K\,(K-1)/2$ multiplications to the process, totaling

$$N_{o,\infty} = \frac{1}{4}K^4 + \frac{3}{4}K^2 - \frac{1}{2}K^3 - \frac{1}{2}K \tag{23}$$

arithmetic operations.

Putting all together, the total number of arithmetic operations required for computing D_j using existing GA libraries is:

$$\begin{aligned} N_{total} &= N_{det} + N_o + N_\infty + N_{o,\infty} \\ &= \Gamma\,(K+1,1)\,\mathrm{e} + \frac{1}{4}K^4 + \frac{15}{4}K^2 - \frac{1}{2}K^3 - \frac{5}{2}K - 2 \end{aligned} \tag{24}$$

additions/subtractions and multiplications. In Equation (24), $\Gamma\,(a,z)$ is the incomplete gamma function and e denotes the Euler number. Unfortunately, this amount of operations is much larger than the $2/3\,K^3 + 5/2\,K^2$ additions/subtractions and multiplications and K squared roots required by Coope's approach if, for example, the QR factors are calculated by Householder transformations [6]. Figure 2 illustrates the growth in the number of operations as a function of K. However, it is essential to emphasize that this complexity issue is not intrinsic to the GA. It is related to the way existing libraries implement GA operations. Therefore, the search for other ways of processing operations without compromising the high level of abstraction provided by GA is an interesting direction of investigation in Computer Science.

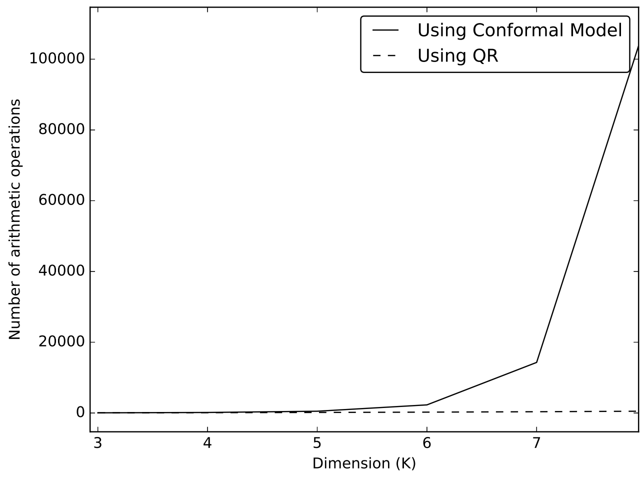

FIGURE 2. Number or arithmetic operations evaluated while computing the intersection of K hyperspheres using the wedge product (thick) and Coope's approach (dashed).

5.2. Numerical Experiments

On this spot, numerical results w.r.t the implementation of C*sym*BP and *sym*BP are presented. The artificially-generated instances which are considered here are the so-called *Lavor Instances*, completely described in [18].

Two aspects are compared for both methods and for $K = 3$, 4, and 5 in Table 1. We apply the C*sym*BP and *sym*BP implementations and highlight the number of points in each problem ($|V|$), the cardinality ($|E|$), the Mean Distance Error (MDE), given by

$$MDE = \frac{1}{|E|} \sum_{(i,j)\in E} \frac{|2X_i \cdot X_j - d_{i,j}|}{d_{i,j}},$$

and the processing time (t_P), in seconds, considering smaller instances. On the other hand, the comparisons displayed in Table 2 regards only dimension $K = 3$ for larger instances.

TABLE 1. Data for ^{K}DMDGP with $K = 3$, 4, and 5, respectively, for both algorithms.

		$K=3$			$K=4$			$K=5$		
Algorithm	$\|V\|$	$\|E\|$	MDE	t_P	$\|E\|$	MDE	t_P	$\|E\|$	MDE	t_P
C*sym*BP	10	45	$5.3e^{-15}$	0.0013	35	$1.5e^{-15}$	0.0046	42	$8.4e^{-15}$	0.0055
*sym*BP	10	45	$1.6e^{-9}$	0.0001	35	$5.9e^{-16}$	0.0002	42	$7.0e^{-15}$	0.0003
C*sym*BP	20	126	$9.8e^{-13}$	0.0024	124	$8.2e^{-13}$	0.0057	99	$2.1e^{-15}$	0.0077
*sym*BP	20	126	$2.8e^{-15}$	0.0003	124	$6.2e^{-16}$	0.0006	99	$6.7e^{-16}$	0.0003
C*sym*BP	30	195	$3.2e^{-13}$	0.0099	204	$5.0e^{-15}$	0.0155	225	$8.4e^{-15}$	0.0199
*sym*BP	30	195	$3.3e^{-15}$	0.0003	204	$2.2e^{-15}$	0.0004	225	$3.1e^{-15}$	0.0006
C*sym*BP	50	438	$1.2e^{-12}$	0.0100	847	$8.9e^{-14}$	0.0143	322	$5.8e^{-14}$	0.0292
*sym*BP	50	438	$1.0e^{-8}$	0.0008	847	$3.6e^{-14}$	0.0007	322	$1.2e^{-14}$	0.0009
C*sym*BP	70	1103	$2.2e^{-12}$	0.0182	2310	$1.4e^{-13}$	0.0280	875	$5.1e^{-14}$	0.0465
*sym*BP	70	1103	$1.5e^{-8}$	0.0010	2310	$6.7e^{-14}$	0.0015	875	$1.5e^{-14}$	0.0016
C*sym*BP	100	1291	$7.6e^{-12}$	0.0241	4779	$2.9e^{-12}$	0.0340	4170	$2.8e^{-13}$	0.0642
*sym*BP	100	1291	$2.9e^{-11}$	0.0015	4779	$1.4e^{-12}$	0.0024	4170	$1.9e^{-13}$	0.0035

TABLE 2. Data for ^{K}DMDGP with $K = 3$ for both algorithms in instances with $n = 500, \ldots, 2000$ atoms.

Algorithm	$\|V\|$	$\|E\|$	MDE	t_P
C*sym*BP	500	8051	$1.847398e^{-8}$	0.1118
*sym*BP	500	8051	$5.774382e^{-9}$	0.0361
C*sym*BP	700	11171	$1.938936e^{-8}$	0.1625
*sym*BP	700	11171	$1.444638e^{-8}$	0.0164
C*sym*BP	1000	14093	$3.909181e^{-7}$	0.2315
*sym*BP	1000	14093	$1.716101e^{-8}$	0.0263
C*sym*BP	2000	32802	$4.328062e^{-6}$	0.4898
*sym*BP	2000	32802	$2.402945e^{-8}$	0.0786

It is worth noting that as the dimension of the space grows, the number of operations in the calculation of the sphere intersections will impact substantially on the total time, as highlighted in Section 5.1. In one hand, CGA provides an elegant framework to express the geometric reasoning that leads to problem-solving. On the other hand, existing computational tools implementing Geometric Algebra are still not able to produce systems with the same computational performance than traditional linear algebra tools without the user having to intervene directly in the programming. It would be helpful if, for example, geometric algebra libraries were able to identify code snippets that represent expressions that could be evaluated by more robust algorithms, such as the outer product of hyperspheres, even if they will require mapping to matrices or tensors.

The global view of tests is presented by Figure 3.

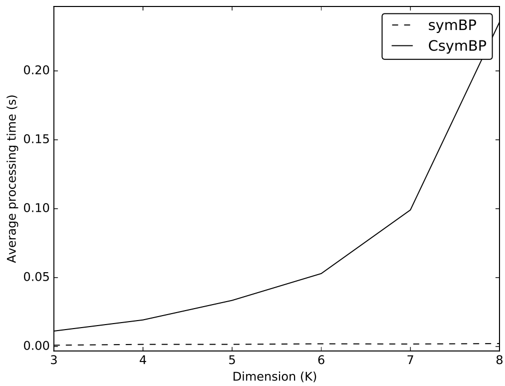

FIGURE 3. Comparing average processing time between *sym*BP and C*sym*BP in several dimensions.

6. Concluding Remarks and Future Works

In the present work, the Discretizable Molecular Distance Geometry Problem for any finite dimension K with exact distance values is completely described in Conformal Geometric Algebra terms, free from internal coordinates such as angles between edges and between hyperplanes. It presented formulas that translate the geometric meanings directly to algebraic expressions for branching, pruning, and for finding other solutions using symmetry skills.

In addition, an adaptation of the symmetry-driven Branch-and-Prune (*sym*BP) algorithm is proposed in order to check what would happen if all characteristics of it could resemble in CGA. It was named C*sym*BP and proved not to be as computationally efficient as the combination of the classic *sym*BP with the QR matrix-decomposition from Coope's work [6].

Guiding the future-work track, authors would like to describe the interval-distance case treatment using CGA, now, for a finite arbitrary dimension by relaxing some constraints to have imprecisions. The idea is to attempt to make what Alves and Dorst performed in the case $K = 3$ [8, 16]. Also, a numerical improvement for the C*sym*BP in the exact-case is pursued and taking hybrid strategies into account is not discarded.

Acknowledgment

The authors would like to thank the CNPq, FAPERJ, UFSC, UFF, and UEM for financial support and Leo Liberti precious research information.

References

[1] Alves, R., and Lavor, C. Geometric algebra to model uncertainties in the discretizable molecular distance geometry problem. *Advances in Applied Clifford Algebra 27* (2017), 439–452.

[2] Alves, R., Lavor, C., Souza, C., and Souza, M. Clifford algebra and discretizable distance geometry. *Mathematical Methods in the Applied Sciences 41* (2018), 4063–4073.

[3] Andreas, W. M. D., and Havel, T. F. Distance geometry and geometric algebra. *Foundations of Physics 23*, 10 (1993), 1357–1374.

[4] Bezanson, J., Edelman, A., Karpinski, S., and Shah, V. B. Julia: A fresh approach to numerical computing. *SIAM review 59*, 1 (2017), 65–98.

[5] Charrier, P., Klimek, M., Steinmetz, C., and Hildenbrand, D. Geometric algebra enhanced precompiler for C++, OpenCL and Mathematica's OpenCLLink. *Advances in Applied Clifford Algebras 24*, 2 (2014), 613–630.

[6] Coope, I. Reliable computation of the points of intersection of n spheres in n-space. *ANZIAM J.* (2000), 461–477. 42.

[7] Crippen, G., and Havel, T. *Distance Geometry and Molecular Conformation*. Research Studies Press Ltda., 1988. New York.

[8] Dorst, L. Boolean combination of circular arcs using orthogonal spheres. *Advances in Applied Clifford Algebra 29* (2019), online.

[9] Dorst, L., Fontijne, D., and Mann, S. *Geometric Algebra for Computer Science: An Object- Oriented Approach to Geometry (The Morgan Kaufmann Series in Computer Graphics)*. Morgan Kaufmann Publishers Inc., 2007.

[10] Dubbs, C., and Siegel, D. Computing determinants. *The College Mathematics Journal 18*, 1 (1987), 48–50.

[11] Faria, V. R., Castelani, E. V., da Silva, J., Camargo, V. S., and Shirabayashi, W. V. I. evcastelani/liga.jl dev version. GitHub Repository: `https://github.com/evcastelani/Liga.jl`, November 2018.

[12] Fernandes, L. A. F. Geometric Algebra Template Library. GitHub Repository: `https://github.com/laffernandes/gatl`, April 2017.

[13] Fidalgo, F., Gonçalves, D., Lavor, C., Liberti, L., and Mucherino, A. A symmetry-based splitting strategy for discretizable distance geometry problems. *Journal of Global Optimization 71* (2018), 717–733.

[14] Gonçalves, D. S. A least-squares approach for discretizable distance geometry problems with inexact distances. *Optimization Letters to appear* (2017).

[15] Grassmann, H. *Die Lineale Ausdehnungslehre.* Verlag von Otto Wigand, Leipzig, 1878.

[16] Lavor, C., and Alves, R. Oriented conformal geometric algebra and the molecular distance geometry problem. *Advances in Applied Clifford Algebra 29* (2019), 439–452.

[17] Lavor, C., Alves, R., Figueiredo, W., Petraglia, A., and Maculan, N. Clifford algebra and the discretizable molecular distance geometry problem. *Advances in Applied Clifford Algebra 25* (2015), 925–942.

[18] Lavor, C., Liberti, L., and Maculan, N. A note on a branch-and-prune algorithm for the molecular distance geometry problem. *International Transactions in Operational Research 18*, 6 (2011), 751–752.

[19] Lavor, C., Liberti, L., Maculan, N., and Mucherino, A. The discretizable molecular distance geometry problem. *Computational Optimization and Applications 52* (2012), 115–146.

[20] Li, H., Hestenes, D., and Rockwood, A. Generalized homogeneous coordinates for computational geometry. In *Geometric Computing with Clifford Algebras*, G. Sommer, Ed. Springer, 2001, pp. 27–59.

[21] Liberti, L. Private communication, may 2019.

[22] Liberti, L., and Lavor, C. *Euclidean Distance Geometry: an introduction.* Springer, Cham, Switzerland, 2017.

[23] Liberti, L., Lavor, C., and Maculan, N. A branch-and-prune algorithm for the molecular distance geometry problem. *International Transactions in Operational Research 15*, 1 (2008), 1–17. Janeiro.

[24] Liberti, L., Lavor, C., Maculan, N., and Mucherino, A. Euclidean distance geometry and applications. *SIAM Review 56*, 1 (2014), 3–69.

[25] Liberti, L., Masson, B., Lee, J., Lavor, C., and Mucherino, A. On the number of realizations of certain henneberg graphs arising in protein conformation. *Discrete Applied Mathematics* (2013). Article in Press.

[26] Maioli, D. S., Lavor, C., and Gonçalves, D. S. A note on computing the intersection of spheres in $\mathbb{R}^n$. *The ANZIAM Journal 59*, 2 (2017), 271–279.

[27] Moré, J. J., W. Z. Global continuation for distance geometry problems. *SIAM Journal on Computing 7*, 3 (1997), 814–836.

[28] Moré, J. J., W. Z. Distance geometry optimization for protein structures. *Journal of Global Optimization 15*, 3 (1999), 219–234.

[29] Mucherino, A., Lavor, C., and Liberti, L. Exploiting symmetry properties of the discretizable molecular distance geometry problem. *Journal of Bioinformatics and Computational Biology 10*, 3 (2012).

[30] Perwass, C. B. U. *Geometric Algebra with Applications in Engineering.* Springer-Verlag, 2009.

[31] Rojas, N., and Thomas, F. Distance-based position analysis of the three seven-link assur kinematic chains. *Mechanism and Machine Theory 46*, 2 (2011), 112–126.

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

[33] Watt, D. A., and Findlay, W. *Programming language design concepts.* John Wiley and Sons, 2004.

[34] Worley, B., Delhommel, F., Cordier, F., Malliavin, T. E., Bardiaux, B., Wolff, N. andNilges, M., Lavor, C., and Liberti, L. Tuning interval branch-and-prune for protein structure determination. *Journal of Global Optimization 72* (2018), 109–127.

Valter S. Camargo
Department of Mathematics, State University of Paraná (UNESPAR)
Paranavaí, PR, Brazil
e-mail: valter.camargo@unespar.edu.br

Emerson V. Castelani
Department of Mathematics, State University of Maringá (UEM)
Maringá, PR, Brazil
e-mail: evcastelani@uem.br

Leandro A. F. Fernandes
Instituto de Computação, Universidade Federal Fluminense (UFF),
Niterói, RJ, Brazil
e-mail: laffernandes@ic.uff.br

Felipe Fidalgo
Department of Mathematics, Universidade Federal de Santa Catarina (UFSC),
Blumenau, SC, Brazil
e-mail: felipe.fidalgo@ufsc.br