USING QUATERNION GEOMETRIC ALGEBRA FOR EFFICIENT ROTATIONS IN THE BRANCH-AND-PRUNE ALGORITHM TO SOLVE THE DISCRETIZABLE MOLECULAR DISTANCE GEOMETRY PROBLEM

F. Fidalgo [a]

[a] Department of Mathematics
Federal University of Santa Catarina (UFSC), Blumenau, Brazil
felipe.fidalgo@ufsc.br [presenter, corresponding]

ABSTRACT. The even-grade 3D Euclidean geometric algebra $\mathscr{G}_{3,0,0}^{+}$ provides a valuable device to perform rotations of three-dimensional vectors, by means of unit quaternions, which has been called *a rotor*. The aim of this paper is to present a theoretical study with the use of quaternion rotors replacing the homogeneous-orthogonal-matrix-product structure of the Branch-and-Prune algorithm when tackling the so-called Discretizable Molecular Distance Geometry Problem with exact distances, a subclass of distance geometry problems, which determines structures in $\mathbb{R}^d$ only subjected to exact-distance and chirality constraints. The proposed idea shows to be productive by reducing around a half of the number of numerical operations to find a solution and by demanding less space of storage.

1. INTRODUCTION

Quaternions are known as an efficient tool to rotate 3D vectors. Such discussion of the Hamilton algebra [11, 12] started with Arthur Cayley around 1855 [2] and he was followed by Clifford, Klein, Hurwitz, Hathaway and others [6]. What comes next is based on [2, 5, 8, 14, 15].

A *quaternion* is a hyper-complex number $q = q_0 + \mathbf{q}_v \in \mathbb{H}$, where the scalar part q_0 is real and $\mathbf{q}_v = q_1\mathbf{i} + q_2\mathbf{j} + q_3\mathbf{k}$ is the complex vector part [8]. If the scalar part is null, such number is called a *pure quaternion* whose subset is denoted by $\mathbb{H}_0$. It was already proved that $\mathbb{H}$ is isomorphic to $\mathbb{R}^4$ and that the subset $\mathbb{H}_0$ is isomorphic to $\mathbb{R}^3$, which is fundamental to the use of quaternions for rotations of 3D vectors. Furthermore, the *conjugate* of q is defined as $q^* = q_0 - \mathbf{q}_v \in \mathbb{H}$.

$\mathbb{H}$ is a non-commutative algebra with the usual addition and the multiplication defined by the equations $\mathbf{i}^2 = \mathbf{j}^2 = \mathbf{k}^2 = \mathbf{ijk} = -1$, $\mathbf{jk} = \mathbf{i} = -\mathbf{kj}$, $\mathbf{ki} = \mathbf{j} = -\mathbf{ik}$ and $\mathbf{ij} = \mathbf{k} = -\mathbf{ji}$. Therefore, given two quaternions $p = p_0 + \mathbf{p}_v$ and $q = q_0 + \mathbf{q}_v$, the latter equations gives

$$pq = (p_0q_0 - \mathbf{p}_v \cdot \mathbf{q}_v) + (p_0\mathbf{q}_v + q_0\mathbf{p}_v + \mathbf{p}_v \times \mathbf{q}_v), \tag{1}$$

which demands 28 numerical operations to be performed.

The *norm* of q is defined by $Nq = qq^* = q^*q$. We call q a *unit quaternion* when $Nq = 1$.

Cayley, then, noticed that the mapping in $\mathbb{R}^4$ given by $p \longmapsto qpq^*$, with $q \in \mathbb{H}$, represents a four-dimensional rotation, taking advantage of such isomorfism $\mathbb{R}^4 \sim \mathbb{H}$ [5].

Now, in Geometric Algebra (geometric approach of Clifford Algebra [3] pionereed by D. Hestenes [13]), a *rotor* in the 3D Euclidean space is an even-grade element

$$R_{\varphi,\hat{\mathbf{r}}} = \mathrm{e}^{-\varphi\hat{\mathbf{r}}I} = \cos(\varphi) - \sin(\varphi)\hat{r}I \qquad \text{(polar representation)} \tag{2}$$

of the 3D Euclidean algebra $\mathscr{G}_{3,0,0}$ which rotates vectors in 2φ about the axis spanned by dual of the unit vector $\hat{\mathbf{r}}$ by means of the mapping $\mathbf{v} \mapsto R_{\varphi,\hat{\mathbf{r}}}\mathbf{v}\tilde{R}_{\varphi,\hat{\mathbf{r}}}$, where $\tilde{R}_{\varphi,\hat{\mathbf{r}}}$ is the reversal of $R_{\varphi,\hat{\mathbf{r}}}$ and $I = \mathbf{e}_1\mathbf{e}_2\mathbf{e}_3$ is the pseudoscalar of $\mathscr{G}_{3,0,0}$, where $\{\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3\}$ is the canonical base of $\mathbb{R}^3$. [8]. Moreover, rotors define their own algebraic structure called *Rotor Algebra*, which is denoted by $\mathscr{G}_{3,0,0}^{+}$ and consists of a subalgebra of $\mathscr{G}_{3,0,0}$.

Next proposition states that unit quaternions from $\mathbb{H}$ can be binunivocally identified with rotors in $\mathscr{G}^+_{3,0,0}$ through the relations $\mathbf{i} \leftrightarrow -\mathbf{e}_1 I, \mathbf{j} \leftrightarrow -\mathbf{e}_2 I$ and $\mathbf{k} \leftrightarrow -\mathbf{e}_3 I$. We show no proof, as it is fully based on [5, 8, 15].

Proposition 1. *The unit quaternion* $q = q_0 + \mathbf{q}_v = \cos\left(\frac{\theta}{2}\right) + \sin\left(\frac{\theta}{2}\right)\hat{\mathbf{r}}$ *can be identified with the rotor* $R = R_{\frac{\theta}{2},\hat{\mathbf{r}}} \in \mathscr{G}^+_{3,0,0}$*, which rotates vectors around the subspace spanned by the unit vector* $\hat{\mathbf{r}}$ *about the angle* θ *in the right-hand orientation. The reverse is given by the conjugate quaternion in this case, that is,* $\tilde{R} = \tilde{R}_{-\theta,\mathbf{r}} = e^{\frac{\theta}{2}\hat{\mathbf{r}}I}$*. Moreover, the result of rotating* $\mathbf{v}$ *by* R *can be associated with a linear combination in a local base* $\{\mathbf{v}, \mathbf{q}_v, \mathbf{q}_v \times \mathbf{v}\}$ *as*

$$R\mathbf{v}\tilde{R} \leftrightarrow (q_0^2 - \mathbf{q}_v \cdot \mathbf{q}_v)\mathbf{v} + 2(\mathbf{q}_v \cdot \mathbf{v})\mathbf{q}_v + 2q_0(\mathbf{q}_v \times \mathbf{v}). \tag{3}$$

It is easy to see, by Equation (3), that R demands 32 numerical operations to rotate $\mathbf{v}$.

In this work, we use such rotors to implement the branching device of the so-caled Branch-and-Prune (BP) algorithm with less numerical operations than the usual homogeneous rotation matrix approach when exploiting the search space of solutions for the Discretizable Molecular Distance Geometry Problem (DMDGP).

It is organized as follows. In Section 2, we motivate the use of the DMDGP in the modelling of a protein-structure-determination problem and discuss its definition using graph theory. Section 3 focuses on the Branch-and-Prune algorithm and how it takes advantages of the combinatorial approach of DMDGP by using a product of matrices. In Section 4, we present the original contribution of this paper which consists of using rotors of Quaternion Geometric Algebra in the kernel of BP instead of matrices. Comparison between classical and quaternion approaches are discussed in Section 5. Finally, Section 6 concludes the paper and present some directions for future works.

2. The Discretizable Molecular Distance Geometry Problem

Proteins consist of chains of amino acids, which are chemically bound forming a larger molecule. The structure of each amino acid is given by atoms of hydrogen, nitrogen, carbon, oxygen and a residue, which varies from one to another and specifies the aminoacid uniquely - the side chain. It can be represented by the graph $G = (V,E)$ in Figure 1 (a), where V is the set of atoms (vertices of the graph) and E is the set of available bonds between the atoms (edges of the graph), and G_{SC} is a subgraph of G which represents the side chain.

A widely important problem in structural and computational Biology is the calculation of the three-dimensional structure of a protein, which gives lots of information about its functions. For this purpose, some interatomic-distance and chirality data can be provided by Ramachandran peptide-unit-mean data [24] or even by physical-chemical experiments such as cristallography and Nuclear Magnetic Resonance (RMN) [7, 25, 28]. Such data can be used to model an inverse problem in order to determine protein tridimensional structure by removing all the side chains and considering only the *backbone* of the protein, represented by the graph G_{PB} with a hand-crafted-virtual order for the atoms Figure 1 (b) [17].

We, then, proceed to formalize the problem. Given an integer $K > 0$ and a simple undirected graph $G = (V,E)$ whose edges are weighted by an distance function $d : E \longrightarrow \mathbb{R}_+$, the Distance Geometry Problem (DGP) with exact distances is a decision problem which asks if there exists or not an embedding $\mathbf{x}$ of V in $\mathbb{R}^K$ such that it does not violate the edge-weight constraints, i.e.,

$$\|\mathbf{x}(u) - \mathbf{x}(v)\| = d(\{u,v\}), \quad \forall \{u,v\} \in E, \tag{4}$$

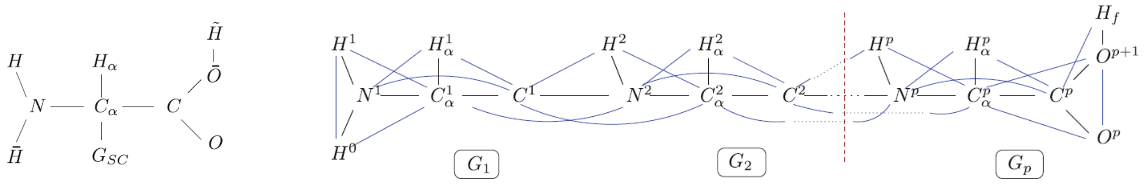

FIGURE 1. (a) A basic graph structure of an aminoacid. (b) A hand-crafted-virtual order of protein backbone. [17]

where $||\cdot||$ is the Euclidean norm. E indicates which exact distances between pairs of objects (represented by the vertices in V) are available and that are given by the edge-weighting function d. A solution $\mathbf{x}$ to the DGP is called a *realization* and the search space of this problem is continuous. From here on, we adopt the short notation $d_{u,v}$ for $d(\{u,v\})$ and $\mathbf{x}_v$ for $\mathbf{x}(v)$.

The Molecular Distance Geometry Problem (MDGP) with exact distances is the 3-dimensional DGP and such name is inspired in the pioneer work of Crippen and Havel [4] which deals with such problem to find tridimensional conformations for molecules by using distances among atoms and chirality information.

Under particular assumptions, the MDGP can be discretized such that the search space assumes a combinatorial fashion which can be represented by a binary tree. It is strongly based on the existence of a vertex total order $(\leq)$ in V, whose position of each vertex in the order is named its *rank*, which gives us a particular MDGP problem, object of our interest in this paper. As the rank identification is injective, we will abuse on denoting the vertex u with rank i by "vertex i" and accept all the implications of this.

The Discretizable Molecular Distance Geometry Problem (DMDGP) is a MDGP with a vertex order such that

(1) (*discretization*)
for every pair of vertices $i, j \in V$ with $1 \leq |i-j| \leq 3$, we have $\{i, j\} \in E$ and

(2) (*non-collinearity*)
the distance values among each triplet of consecutive vertices $i-2, i-1$ and i in the order satisfy the strict triangular inequalities

$$d_{i-2,i} < d_{i-2,i-1} + d_{i-1,i},$$

for all $i \geq 3$.

Assumption (1) ensures that the first three vertices in the order induce a 3-clique and guarantees that, for every $i \geq 4$, the set $\{i-3, i-2, i-1, i\} \subset V$ induces a 4-clique. It means that a set of feasible positions for vertex i, w.r.t. the considered distances, lie in the intersection I of three spheres S_1, S_2 and S_3 with centers in the positions of $i-3, i-2$ and $i-1$ and radii $d_{i-3,i}, d_{i-2,i}$ and $d_{i-1,i}$, respectively, which can have zero, one or two positions (discretization) [16]. Liberti *et al.* [19] studies such intersection and guarantees that it has either no point or two points, assertion which occurs with probability 1, as the probability of $|I| = 1$ is 0 (zero Lebesgue measure). Assumption (2), in turn, ensures that collinearity of the three centers does not hold and, therefore, it is not possible to have infinitely many points in the intersection.

Finally, the search space for solutions is designed as a binary tree. The set of edges can be partioned into two disjoint sets $E = E_d \cup E_p$: E_d has the *discretization edges* and is always non-empty by Assumption (1) and E_p has the *pruning edges* which can be empty and whose weights will be used by the BP algorithm to prune away the infeasible positions determined by the discretization edges [10, 19, 16]. This dynamics guarantees that the tree doesn not grow too much and what makes efficient a depth-first search [20, 23]. After removing all possible, but infeasible, positions, each path from the root node to the leaf node is a solution for the

DMDGP [16]. Moreover, there are vertices which are called *symmetry vertices* that makes possible to transform one solution to another by performing partial reflections on such vertices [20, 23]. Thus, it is enough to determine one solution only. And, the number of solutions is deterministically known *a priori* by $2^{n-|S|-3}$, where $n = |V|$ and S is the set of symmetry vertices which has, at least, the fourth one always [10, 16].

3. CLASSICAL BP ALGORITHM WITH ROTATION MATRICES

Lavor *et al.* [16] designed a three-stage combinatorial algorithm which solves the DMDGP by performing a depth-first search in a binary tree, which is called *Branch-and-Prune* (BP). Such tree is named *Solution Tree* and turns out to be the output of BP.

The input data is a DMDGP instance $G = (V, E, d)$ whose vertices are numbered from 1 to n, where $n = |V|$. Such distances can provide a special coordinate set for all the vertices which is called *internal molecular coordinate* set and that consists of triples of *bond lenghts*, *bond angles* and *torsion angles* (see Figure 2), widely used in Molecular Geometry [9, 21, 27].

For each vertex $i > 2$, the *bond angle* $\theta_{i-2,i}$ consists on the angle between the bonds $\{i-2, i-1\}$ and $\{i-1, i\}$. It can be calculated using the regular cosine law in constant time by

$$\theta_{i-2,i} = \cos^{-1}\left(\frac{d^2_{i-2,i-1} + d^2_{i-i,i} - d^2_{i-2,i}}{2d_{i-2,i-1}d_{i-1,i}}\right). \tag{5}$$

Also, the *torsion angle* $\omega_{i-3,i}$ is defined as the dihedral angle bewteen the planes $\pi_{i-3,i-1}$ and $\pi_{i-2,i}$, which are respectively and uniquely defined by the positions of the vertices $i-3, i-2, i-1$ and $i-2, i-1, i$. It can also be computed in constant time, now using the dihedral cosine law [1, 18], by

$$\omega_{i-3,i} = \cos^{-1}\left(\frac{2d^2_{i-2,i-1}\left(d^2_{i-3,i-2} + d^2_{i-2,i} - d^2_{i-3,i}\right) - (d_{i-3,i-2,i-1})(d_{i-2,i-1,i})}{\sqrt{\left(4d^2_{i-3,i-2}d^2_{i-2,i-1} - d^2_{i-3,i-2,i-1}\right)\left(4d^2_{i-2,i-1}d^2_{i-1,i} - d^2_{i-2,i-1,i}\right)}}\right), \tag{6}$$

where

$$d_{i-3,i-2,i-1} = d^2_{i-3,i-2} + d^2_{i-2,i-1} - d^2_{i-3,i-1} \qquad \text{and}$$
$$d_{i-2,i-1,i} = d^2_{i-2,i-1} + d^2_{i-i,i} - d^2_{i-2,i}.$$

All distance values involved in Equations 5 and 6 are available by the DMDGP definition.

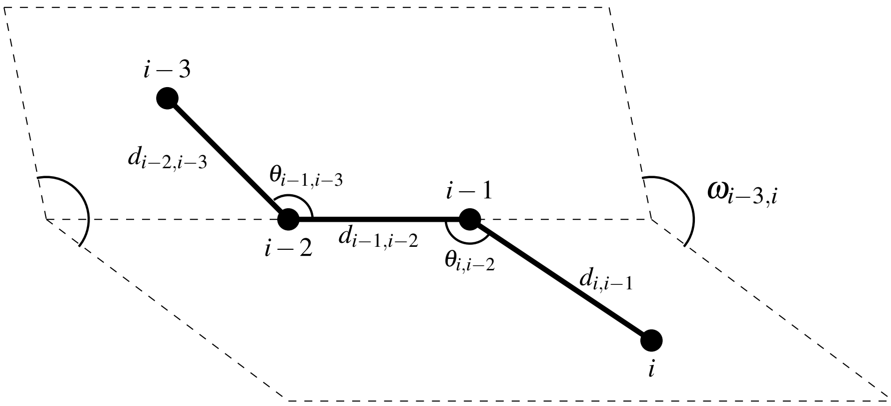

FIGURE 2. Fundamental quatruplet of DMDGP and internal coordinates.

The algorithm is initiallized by positioning the first three vertices in order to fix the base plane $\pi_{1,3}$ [27] (see Figure 3). Vertex 1 is considered as the origin of a local Cartesian frame $x_1y_1z_1$

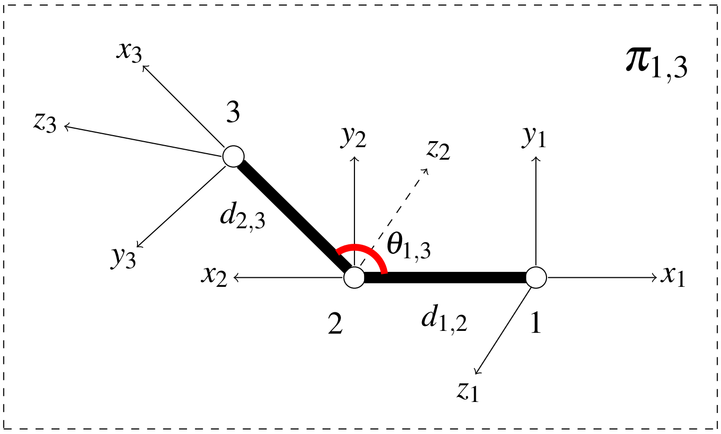

FIGURE 3. Initialization of BP: placing the base plane.

such that vertex 2 is defined in the negative x_1-axis and vertex 3 lies in the first or second quadrant of the x_1y_1-plane. Vertex 2 is also the origin of a coordinate system $x_2y_2z_2$ whose negative x_2-axis passes through vertex 1 and, again, vertex 3 lies in the first or second quadrant of the x_2y_2-plane. At last, vertex 3 is the origin of a coordinate frame $x_3y_3z_3$, whose negative x_3-axis passes through vertex 2 and vertex 1 lies on the third or fourth quadrant of x_3y_3-plane.

So, every conformation $\mathbf{x} : V \to \mathbb{R}^3$ for G is based in this plane, starting from the origin of $x_1y_1z_1$ with $\mathbf{x}_1 = \begin{pmatrix} 0 & 0 & 0 \end{pmatrix}^T$. Such origin can be mapped into the second position with three transformations from the first frame into the $x_2y_2z_2$ system: a translation by the vector $\mathbf{t}_2 = \begin{pmatrix} d_{1,2} & 0 & 0 \end{pmatrix}^T$, a rotation in $\theta = \pi$ around the axis spanned by the canonical vector $\mathbf{k}$ and a rotation in $\omega = \pi$ about the axis spanned by the canonical vector $\mathbf{i}$, i.e., $\mathbf{x}_2 = \begin{pmatrix} -d_{1,2} & 0 & 0 \end{pmatrix}^T$. Analogously, the origin of $x_1y_1z_1$ can be mapped into $x_2y_2z_2$ by means of the latter transformation and, then, into $x_3y_3z_3$ by a translation with the vector $\mathbf{t}_3 = \begin{pmatrix} d_{2,3} & 0 & 0 \end{pmatrix}^T$, a rotation in $\theta = \pi - \theta_{1,3}$ around the axis spanned by $\mathbf{k}$ and a rotation in $\omega = 0$ around the axis spanned by $\mathbf{i}$, which determines the position $\mathbf{x}_3 = \begin{pmatrix} -d_{1,2} + d_{2,3}\cos(\theta_{1,3}) & d_{2,3}\sin(\theta_{1,3}) & 0 \end{pmatrix}^T$.

From the fourth on, we have the *branching* stage: given a vertex i, the discretization assumption guarantees that there are two possibilities x_i^1 and x_i^2 for its position, as we previsouly saw, that lie in the intersection of three spheres with centers in the positions of $i-3, i-2$ and $i-1$ and radii $d_{i-3,i}, d_{i-2,i}$ and $d_{i-1,i}$, respectively. They are symmetric by the plane determined by the x_{i-3}, x_{i-2} and x_{i-1}: one is generated by a rotation by $\omega = \omega_{i-3,i}$ and the other by a rotation by $\omega = -\omega_{i-3,i}$, both about the axis $span\{\mathbf{i}\}$.

Both positions can be determined by solving the quadratic system

$$\|\mathbf{x}_i - \mathbf{x}_{i-3}\|^2 = d_{i-3,i}^2 \tag{7}$$

$$\|\mathbf{x}_i - \mathbf{x}_{i-2}\|^2 = d_{i-2,i}^2 \tag{8}$$

$$\|\mathbf{x}_i - \mathbf{x}_{i-1}\|^2 = d_{i-1,i}^2 \tag{9}$$

Some mentions about solutions to this can be found in [19].

Lavor *et al.* [16], on the other hand, took advantage in the recursive structure of the DMDGP to find both positions by a composition of a translation, a planar rotation and a spatial rotation, as some authors had already done for other applications [9, 27]. After finding one of the solutions from the pair, the other can be found just by changing the signs for the torsion angles [16].

Below, we describe in details such recursive transformations using a product of matrices. It will be of great importance when defining the quaternion rotors, as we need to know precisely the angle and the axis that the rotation will occur.

For each $i \geq 4$, it is defined a $x_i y_i z_i$ frame such that vertex $i+1$ lies in the negative x_i-axis and $i+2$ lies in the the third or fourth quadrant of the $x_i y_i$-plane.

For all the cases, we can transform a position $\mathbf{x}_{i-1}$ in the frame $x_{i-1}y_{i-1}z_{i-1}$ to another position $\mathbf{x}_i$ in the frame $x_i y_i z_i$ by a translation by the vector $\mathbf{t}_i = \begin{bmatrix} d_{i-1,i} & 0 & 0 \end{bmatrix}^T$, a rotation in $\theta = \pi - \theta_{i-2,i}$ about $span\{\mathbf{k}\}$ and a rotation in $\omega = \omega_{i-3,i}$ about $span\{\mathbf{i}\}$. Thus,

$$\mathbf{x}_i = \begin{bmatrix} x_i \\ y_i \\ z_i \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\omega) & -\sin(\omega) \\ 0 & \sin(\omega) & \cos(\omega) \end{bmatrix} \begin{bmatrix} \cos(\theta) & -\sin(\theta) & 0 \\ \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \left(\begin{bmatrix} x_{i-1} \\ y_{i-1} \\ z_{i-1} \end{bmatrix} + \begin{bmatrix} d_{i-1,i} \\ 0 \\ 0 \end{bmatrix} \right) \tag{10}$$

Using the fact that $\sin(\pi - \theta_{i-2,i}) = \sin(\theta_{i-2,i})$ and $\cos(\pi - \theta_{i-2,i}) = -\cos(\theta_{i-2,i})$ and the convenient homogeneous representation of a translation as a matrix [27], the position $\mathbf{x}_i$ as the result of Equation (10) can be represented in the homogeneous space as a product of two orthogonal 4×4 matrices E_i and P_i and a homogenous-translational matrix T_i by the position of $\mathbf{x}_{i-1}$ in the homogeneous space as

$$\mathbf{x}_i^h = \begin{bmatrix} x_i \\ y_i \\ z_i \\ 1 \end{bmatrix} = E_i P_i T_i \begin{bmatrix} x_{i-1} \\ y_{i-1} \\ z_{i-1} \\ 1 \end{bmatrix} = E_i P_i T_i \mathbf{x}_{i-1}^h, \tag{11}$$

where $\mathbf{x}_{i-1}^h = \begin{pmatrix} \mathbf{x}_{i-1}^T & 1 \end{pmatrix}^T$, $\mathbf{x}_i^h = \begin{pmatrix} \mathbf{x}_i^T & 1 \end{pmatrix}^T$,

$$E_i \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(\omega_{i-3,i}) & -\sin(\omega_{i-3,i}) & 0 \\ 0 & \sin(\omega_{i-3,i}) & \cos(\omega_{i-3,i}) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad P_i = \begin{bmatrix} -\cos(\theta_{i-2,i}) & -\sin(\theta_{i-2,i}) & 0 & 0 \\ \sin(\theta_{i-2,i}) & -\cos(\theta_{i-2,i}) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \text{and} \quad T_i = \begin{bmatrix} 1 & 0 & 0 & d_{i-1,i} \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

After the products, we then get to the widely known matrix transformation [9, 16, 21, 26, 27]

$$\mathbf{x}_i^h = B_i \mathbf{x}_{i-1}^h, \tag{12}$$

which is given by

$$\begin{bmatrix} x_i \\ y_i \\ z_i \\ 1 \end{bmatrix} = \begin{bmatrix} -\cos(\theta_{i-2,i}) & -\sin(\theta_{i-2,i}) & 0 & -d_{i-1,i}\cos(\theta_{i-2,i}) \\ \sin(\theta_{i-2,i})\cos(\omega_{i-3,i}) & -\cos(\theta_{i-2,i})\cos(\omega_{i-3,i}) & -\sin(\omega_{i-3,i}) & d_{i-1,i}\sin(\theta_{i-2,i})\cos(\omega_{i-3,i}) \\ \sin(\theta_{i-2,i})\sin(\omega_{i-3,i}) & -\cos(\theta_{i-2,i})\sin(\omega_{i-3,i}) & \cos(\omega_{i-3,i}) & d_{i-1,i}\sin(\theta_{i-2,i})\sin(\omega_{i-3,i}) \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_{i-1} \\ y_{i-1} \\ z_{i-1} \\ 1 \end{bmatrix}$$

This product is efficient to find both possibilities for the position of a vertex, as we can get the other position from the same matrix structure just by turning the sign of the sine, since

$$\sin(-\omega_{i-3,i}) = -\sin(\omega_{i-3,i}). \tag{13}$$

So, there is a biunivocal correspondence between the set of positions for all possible depth-first paths in the binary tree and the set of all possible sequences of internal coordinates, ordered following the DMDGP ordered set V. That is, as in [16, 27], for each choice of sequence of internal coordinates from 1 to n, one can position vertex i uniquely, by transforming the origin of the $x_1 y_1 z_1$ frame into a point in the $x_i y_i z_i$ frame through the product

$$\mathbf{x}_i^h = B_1 B_2 B_3 B_4 \cdots B_{i-1} B_i \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}. \tag{14}$$

For $i = 1, 2, 3$, we have no branchings. So, it is easy to determine the first three transformation matrices are fixed: B_1 is the 4×4 identity matrix,

$$B_2 = \begin{bmatrix} -1 & 0 & 0 - d_{1,2} & \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \text{and} \quad B_3 = \begin{bmatrix} -\cos(\theta_{1,3}) & -\sin(\theta_{1,3}) & 0 & -d_{2,3}\cos(\theta_{1,3}) \\ \sin(\theta_{1,3}) & -\cos(\theta_{1,3}) & 0 & d_{2,3}\sin(\theta_{1,3}) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

Finally, after positioning a vertex i in $\mathbf{x}_i$, we ought to test its feasibility according to all distance values involving such vertex. This is the third stage called *pruning*. When $\mathbf{x}_i$ is found, a pruning device is implemented in order to check feasibility.

Many different kinds of pruning devices can be developed and used at this stage [22], but a very efficient one is the Direct Distance Feasibility (DDF) check.

Alg. 1 is an outline of the BP algorithm, where $v \in V$ is the current vertex one wants to position, $n = |V|$ and d represents the weights for the edges.

Algorithm 1 The BP algorithm

```
BP$(v, n, d)$
**compute** $x'_v$;
**if** ($x'_v$ is feasible) **then**
    **if** $(v = n)$ **then**
        **let** $nsols = nsols + 1$;
    **else**
        BP$(v + 1, n, d)$;
    **end if**
**end if**
**compute** $x''_v$;
**if** ($x''_v$ is feasible) **then**
    **if** $(v = n)$ **then**
        **let** $nsols = nsols + 1$;
    **else**
        BP$(v + 1, n, d)$;
    **end if**
**end if**
```

If it runs until termination, all possible realizations of G are found. However, it can be stopped after the first leaf node when level n is reached. This way, only one realization of G is determined, which is called BP-one.

4. BP ALGORITHM WITH QUATERNION GEOMETRIC ALGEBRA

Instead of using homogenous 4×4 orthogonal matrices to perform rotations of $3D$ vectors when transform a vector from $x_{i-1}y_{i-1}z_{i-1}$ frame into another in $x_iy_iz_i$ frame, one can make use of convenient unit quaternions at each step of BP.

As we know the possible internal coordinates $(d_i, \theta_i, \omega_i)$ for each vertex $i \in V$, we define the spatial and planar rotational unit quaternions, respectively, as

$$q_i^{\varepsilon} = \cos\left(\frac{\omega_i}{2}\right) + \sin\left(\frac{\omega_i}{2}\right)\mathbf{k} \leftrightarrow \mathrm{e}^{-\frac{\omega_i}{2}\mathbf{k}I} \text{ and } q_i^{\pi} = \cos\left(\frac{\pi - \theta_i}{2}\right) + \sin\left(\frac{\pi - \theta_i}{2}\right)\mathbf{i} \leftrightarrow \mathrm{e}^{-\frac{\pi-\theta_i}{2}\mathbf{i}I}.$$

Then, Equation (11) can be translated into

$$\mathbf{x}_i = (q_i^{\varepsilon} q_i^{\pi})\,(\mathbf{x}_{i-1} + \mathbf{t}_i)\,(q_i^{\varepsilon} q_i^{\pi})^*\,. \tag{15}$$

We can simplify such transformation by making

$$q_i = q_i^{\varepsilon} q_i^{\pi} = \sin\left(\frac{\theta_i}{2}\right)\cos\left(\frac{\omega_i}{2}\right) + \sin\left(\frac{\theta_i}{2}\right)\sin\left(\frac{\omega_i}{2}\right)\mathbf{i} - \cos\left(\frac{\theta_i}{2}\right)\sin\left(\frac{\omega_i}{2}\right)\mathbf{j} + \cos\left(\frac{\theta_i}{2}\right)\cos\left(\frac{\omega_i}{2}\right)\mathbf{k}. \tag{16}$$

Thus, Equation (15) can be written as

$$\mathbf{x}_i = q_i\,(\mathbf{x}_{i-1} + \mathbf{t}_i)\,q_i^*\,. \tag{17}$$

As the internal coordinates for the first three vertices are set as $(0,0,0)$, $(d_{1,2}, 0, \pi)$ and $(d_{2,3}, \theta_{1,3}, 0)$, we then have

$$q_1 = 1, \quad q_2 = -\mathbf{j} \quad \text{and} \quad q_3 = \sin\left(\frac{\theta_{1,3}}{2}\right) + \cos\left(\frac{\theta_{1,3}}{2}\right)\mathbf{k}.$$

Therefore, Equation (14) can be written in quaternion terms as

$$\mathbf{x}_i = q_1\left(q_2\left(q_3\left(\cdots\left(q_{i-1}\left(q_i(e+\mathbf{t}_i)q_i^* + \mathbf{t}_{i-1}\right)q_{i-1}^* + \mathbf{t}_{i-2}\right)\cdots + \mathbf{t}_3\right)q_3^* + \mathbf{t}_2\right)q_2^* + \mathbf{t}_1\right)q_1^*, \tag{18}$$

where $\mathbf{e} = \begin{pmatrix}0 & 0 & 0\end{pmatrix}^T$. If we list each one and perform the distributions, we can notice the recursion calling the last position which is already determined as part of the solution, that is,

$$\begin{aligned}
&\mathbf{x}_1 = q_1(\mathbf{e}+\mathbf{t}_1)q_{1^*} = q_1\mathbf{t}_1 q_1^* = \mathbf{e},\\
&\mathbf{x}_2 = q_1(q_2(\mathbf{e}+\mathbf{t}_2)q_2^* + \mathbf{t}_1)q_1^* = (q_1q_2)\,\mathbf{t}_2\,(q_1q_2)^* + q_1\mathbf{t}_1q_1^* = (q_1q_2)\,\mathbf{t}_2\,(q_1q_2)^* + \mathbf{x}_1,\\
&\mathbf{x}_3 = q_1(q_2(q_3(\mathbf{e}+\mathbf{t}_3)q_3^* + \mathbf{t}_2)q_2^* + \mathbf{t}_1)q_1^* = (q_1q_2q_3)\,\mathbf{t}_3\,(q_1q_2q_3)^* + \mathbf{x}_2,\\
&\vdots\\
&\mathbf{x}_i = (q_1q_2\ldots q_i)\mathbf{t}_i(q_1q_2\ldots q_i)^* + \mathbf{x}_{i-1},\\
&\vdots\\
&\mathbf{x}_n = (q_1q_2\ldots q_n)\mathbf{t}_n(q_1q_2\ldots q_n)^* + \mathbf{x}_{n-1}.
\end{aligned}$$

Thus, the general step of BP, from Equation (14), can be stated using quaternion rotors as

$$\mathbf{x}_i = (q_1q_2\ldots q_i)\mathbf{t}_i(q_1q_2\ldots q_i)^* + \mathbf{x}_{i-1}, \quad i = 2,\ldots,n. \tag{19}$$

If we need to backtrack and get the second position, we just change the sign of the sine of the torsion angle (Equation (13)) and the new position is given by $\tilde{R}\mathbf{x}R$, i.e.,

$$\mathbf{x}_i = (q_1q_2\ldots q_i^*)\mathbf{t}_i(q_1q_2\ldots q_i^*)^* + \mathbf{x}_{i-1}, \quad i = 2,\ldots,n. \tag{20}$$

5. Comparisons: orthogonal matrix algebra × quaternion GA for BP

This section brings a comparison between homogeneous rotation matrix product and quaternion rotor product approaches.

For the first (Equation(14)), to find the $\mathbf{x}_1$, we need 28 operations to $\mathbf{x}_1^h = B_1\begin{bmatrix}0 & 0 & 0 & 1\end{bmatrix}^T$. Then, at each step $i \geq 2$, having already positioned $\mathbf{x}_{i-1}$, we make the product

$$\mathbf{x}_i^h = Q_{i-1}B_i\begin{bmatrix}0 & 0 & 0 & 1\end{bmatrix}^T, \tag{21}$$

where $Q_{i-1} = B_1B_2B_3B_4\cdots B_{i-1}$. To do so, it is necessary to perform 112 operations to build the matrix $Q_{i-1}B_i$ and more 28 operations to make the product of the resulting matrix and the vector $\begin{bmatrix}\mathbf{e}^t & 1\end{bmatrix}^T$. In summary, to build a solution for an instance with n vertices, we need

$$\mathcal{O}_M(n) = 140n - 112 \tag{22}$$

operations.

Yet, for the second approach (Equation(19)), we need 32 operations to find the first position by $\mathbf{x}_1 = q_1\mathbf{t}_1q_1^*$. After that, at each step $i \geq 2$, having already positioned $\mathbf{x}_{i-1}$, we make the transformation

$$\mathbf{x}_i = (q_{i-1}^\rho q_i)\mathbf{t}_i(q_{i-1}^\rho q_i)^* + \mathbf{x}_{i-1}, \tag{23}$$

where $q_{i-1}^\rho = q_1q_2\ldots q_{i-1}$. To do so, we need 28 operations to find the composed quaternion $q_{i-1}^\rho q_i$, more 32 operations to rotate $\mathbf{t}_i$ by $(q_{i-1}^\rho q_i)\mathbf{t}_i(q_{i-1}^\rho q_i)^*$ and more 3 operations to sum with the previous position already determined. As a result, to build the same solution as with the matrix product with n vertices, we need

$$\mathcal{O}_q(n) = 65n - 30 \tag{24}$$

numerical operations.

We used ten instance sizes to analyze the numerical counting of operations for both Matrices and Quaternions. The sizes are displayed in Table 1 and the numerical operation numbers and ratio between them are, respectively, plotted in Figure 4(i) and Figure 4(ii).

TABLE 1. Instance Size Table

Instance	1	2	3	4	5	6	7	8	9	10
n	1	10	100	500	700	1000	5000	7000	10000	20000

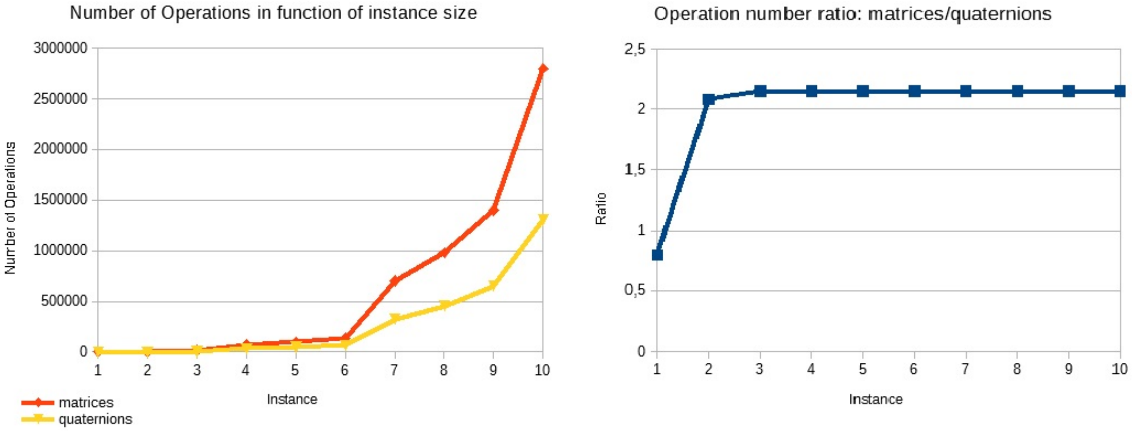

FIGURE 4. (i) In red, the values for $\mathcal{O}_M(n)$ and, in yellow, the values for $\mathcal{O}_q(n)$. (ii) In blue, the ratio between $\mathcal{O}_M(n)$ and $\mathcal{O}_q(n)$

We can see that as the size of the instance grows, the difference between the number of operations of both approaches grows considerably. Such number for the structure with matrices (red line in Figure 4(i)) increases around twice each time more than the number of operations to make the same determination with quaternions (yellow line in Figure 4(i)). The second plotted curve (Figure 4 (ii)) shows the ratio $\dfrac{\mathcal{O}_M(n)}{\mathcal{O}_q(n)}$, which is stable around two since when dealing with a few number of vertices, validating that.

At last, each matrix needs 16 positions and each quaternion need 4 ones. That is, additionally to make less operations, it needs less space to store the internal-coordinate data.

6. CONCLUSIONS AND FUTURE WORK

In this paper, we have shown that the transformation device in the kernel of the Branch-and-Prune algorithm, in order to solve a DMDGP, can be made using either homogeneous rotation-translation matrix product or a product of rotors in the Geometric Algebra of Quaternions, which is the aim of this work.

One can see that using quaternions is not only a good choice in terms of saving memory, but also in terms of saving numerical operations to perform the rigid movements, when compared to the classical orthogonal matrix approach. It possibly, then, can prevent us from numerical round-off errors better than the first approach. The latter numerical experiments was not studied in this theoretical work, standing for a future step.

Some more future challenges are to deal with numerical tests using real protein instances, for showing advantages and drawbacks in setting quaternion approach as the main one. To handle imprecise experimental data (which can be provided by crystallography or NMR) [17], Alves *et al.* have used conformal Geometric Algebra in [1]. Therefore, we also want to evaluate the possibilities of using the idea presented in this paper for doing so, as we chose to work only with precise distance values.

References

[1] R. Alves and C. Lavor, *Geometric Algebra to model uncertainties in the Discretizable Molecular Distance Geometry Problem*, Adv. Appl. Clifford Algebras **27** (2017), 439–452.

[2] A. Cayley, *Recherches ultérieurs sur les déterminants gauches*,Journal für die reine und angewandte Mathematik **50** (1855), 299–313. Springer, London, 2010.

[3] W.K. Clifford, *Applications of Grassmann's extensive algebra*, Am. J. Math. **1** (1878), 350–358.

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

[5] E. Bayro-Corrochano, *Geometric Computing: for wavelets transforms, robot vision, learning, control and action*, Springer, London, 2010.

[6] H.S.M. Coxeter, *Quaternions and reflextions*, The Amer. Math. Month. **53**(3) (1946), 136–146.

[7] T. Creighton, *Proteins: structures and molecular properties*, Freeman, New York, 1993.

[8] L. Dorst, D. Fontijne, S. Mann, *Geometric Algebra for Computer Science (Revised Edition)*, Morgan Kaufmann, Boston, 2009.

[9] H. Eyring, *The resultant electric moment of complex molecules*, Phys. Review **39** (1932), 746–748.

[10] F. Fidalgo, D. S. Gonçalves, C. Lavor, L. Liberti and A. Mucherino, *A symmetry-based splitting strategy for discretizable distance geometry problem*, Journ. Glob. Optm. (2018), *to appear*.

[11] W.R. Hamilton, *Lectures on Quaternions*, Hodges and Smith, Dublin, 1853.

[12] W.R. Hamilton, *Elements of Quaternions*, vol. 1, Longmans, Green and Co., London, 1866.

[13] D. Hestenes, *Space-Time Algebra*, Gordon and Breach, London, 1966.

[14] B. K. P. Horn, *Closed-form solution of absolute orientation using unit quaternions*, Journ. Glob. Optm. **56** (2013), 855–871.

[15] J.B. Kuipers, *Quaternions and Rotation Sequences: a primer with applications to orbits, aerospáce and virtual reality*, Princeton University Press: Princeton, New Jersey, 2002.

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

[17] C. Lavor, L. Liberti and A. Mucherino, *The interval branch-and-prune algorithm for the discretizable molecular distance geometry problem with inexact distances*, Journ. Glob. Optm. **56** (2013), 855–871.

[18] C. Lavor, R. Alves, W. Figueiredo, A. Petraglia and N. Maculan, *Clifford Algebra and the discretizable molecular distance geometry problem*, Adv. Appl. Clifford Algebras **25** (2015), 925–942.

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

[20] L. Liberti, B. Masson, J. Lee, C. Lavor and A. Mucherino, *on the number of realizations of certain hennenberg graphs arising in protein conformation*, Discr. Appl. Mathem. **165** (2014), 213–232.

[21] R.L. McCullough and P.E. McMahon, *Contributions to conformational energy from interactions between nonbonded atoms and groups. Part1 - General formulation*, Trans.Farady Soc. **60** (1964), 2089–2096.

[22] 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: P.M Pardalos and S. Rebennack (Eds.), Proceedings of the 10th International Symposium on Experimental Algorithms (SEA11), Crete, Greece, Lecture Notes in Computer Science **6630** (2011), 206–217.

[23] A. Mucherino, C. Lavor and L. Liberti, *Exploiting symmetry properties of the discretizable molecular distance geometry problem*, Journ. Bioinf. Comput. Biol. **10** (2012), 1–15.

[24] G.N. Ramachandran, A.S. Kolaskar, C. Ramakrishnan and V. Sasisekharan, *The mean geometry of the peptide unit from crystal structure data*, Biochimica et Biophysica Acta - Protein Structure **359**(2) (1974), 298–302.

[25] T. Schlick, *Molecular modeling and simulation: an interdisciplinary guide*, Springer, New York, 2002.

[26] T. Shimanouchi and S-i. Mizushima, *On the helycal configuration of a polymer chain*, Jour. Chem. Physics **23** (1955), 707–711.

[27] B. Thompson, *Calculation of Cartesian cordinates and their derivatives from internal molecular coordinates*, Jour. Chem. Physics **47**(9) (1967), 3407–3410.

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