Proceedings of the 2014 Federated Conference on Computer Science and Information Systems pp. 457–463
DOI: 10.15439/2014F92
ACSIS, Vol. 2

An adaptive branching scheme for the Branch & Prune algorithm applied to Distance Geometry

Douglas Gonçalves*, Antonio Mucherino*, Carlile Lavor†

*IRISA, University of Rennes 1, Rennes, France.
{douglas.goncalves,antonio.mucherino}@irisa.fr

†IMECC-UNICAMP, Campinas-SP, Brazil.
clavor@ime.unicamp.br

***Abstract*—The Molecular Distance Geometry Problem (MDGP) is the one of finding molecular conformations that satisfy a set of distance constraints obtained through experimental techniques such as Nuclear Magnetic Resonance (NMR). We consider a subclass of MDGP instances that can be discretized, where the search domain has the structure of a tree, which can be explored by using an *interval* Branch & Prune (*i*BP) algorithm. When all available distances are exact, all candidate positions for a given molecular conformation can be enumerated. This is however not possible in presence of interval distances, because a continuous subset of positions can actually be computed for some atoms. The focus of this work is on a new scheme for an adaptive generation of a discrete subset of candidate positions from this continuous subset. Our generated candidate positions do not only satisfy the distances employed in the discretization process, but also additional distances that might be available (the so-called pruning distances). Therefore, this new scheme is able to guide more efficiently the search in the feasible regions of the search domain. In this work, we motivate the development and formally introduce this new adaptive scheme. Presented computational experiments show that *i*BP, integrated with our new scheme, outperforms the standard *i*BP on a set of NMR-like instances.**

I. Introduction

LET $G = (V, E, d)$ be a simple weighted undirected graph where the vertices V represent the points of a Euclidean space and where $d : E \to \mathbb{R}_+$ assigns positive weights d_{uv} to edges (u, v) when the distance between u and v is available. The Distance Geometry Problem (DGP) [9] asks to find an embedding $x : V \to \mathbb{R}^3$ satisfying constraints based on the available edge weights, i.e. to find a conformation x in the Euclidean space such that:

$$\underline{d}_{uv} \leq \|x(u) - x(v)\| \leq \overline{d}_{uv}, \quad \forall (u, v) \in E, \tag{1}$$

where $\underline{d}_{uv}$ and $\overline{d}_{uv}$ denote, respectively, the lower and upper bounds for the distance d_{uv} ($\underline{d}_{uv} = \overline{d}_{uv}$ if d_{uv} is an exact distance).

One of the most interesting applications of the DGP arises in biology, where vertices of G represent atoms of a given molecule, and weighted edges provide the relative distances between some atom pairs. When molecules are concerned, the DGP is generally referred to as the Molecular DGP (MDGP) [3], [5]. The interested reader can make reference to a recent survey [9] and to an edited book [13] for additional information about the MDGP and methods for its solution.

The MDGP is generally formulated as a continuous optimization problem where the objective function is a penalty function capable of measuring the violation of the constraints. Under certain assumptions, the domain of this optimization problem can be discretized, so that it becomes combinatorial [7], [12]. The discrete search domain has the structure of a tree, where the candidate positions for a given atom of the molecule belong to the same layer of the tree. We employ an *interval* Branch & Prune (*i*BP) algorithm [8] for exploring such a tree with the aim of finding solutions to discretizable MDGPs. The reader is referred to Section II for more details about the discretization.

The basic idea behind the *i*BP algorithm is to construct the search domain of the optimization problem branch by branch (*branching phase*), and to verify, every time a new branch is added, whether it is feasible or not (*pruning phase*). Atomic positions are generated by intersecting 3 Euclidean objects (spheres and spherical shells), which we can define on each layer of the tree because of the discretization assumptions. When discovered, infeasible positions are pruned away, so that the search can be focused on the parts of the tree where there are feasible solutions. Only a subset of available distances is employed in the discretization process (the *discretization distances*), while others can be exploited for pruning purposes (the *pruning distances*).

In the discretization process, if all considered distances are exact, there can be at most two feasible positions for the current atom [7]. If some distances are represented by intervals, the feasible positions belong to a continuous Euclidean object, that can be discretized by sampling D candidate positions [8]. In this phase, the number D of chosen sample positions plays a very important role.

Experiments reported in previous publications (see for example [2], [8]) show in fact that the obtained results can be strongly influenced by the choice of D. If D is too small, only

978-83-60810-58-3/$25.00

infeasible branches may be generated, so that the whole tree is pruned and no solutions are found. On the other hand, if D is too large, the consequent combinatorial explosion might make the experiments too expensive. Finding a trade-off D value is not an easy task in general.

This paper presents a new scheme for an adaptive branching during the execution of the *i*BP algorithm, which is based on the idea of including, during the intersection of the Euclidean objects related to the known discretization distances, other objects, related to pruning distances, that might be available at the current layer. This way, it is possible to generate branches that are feasible, with respect to the pruning distances, up to the current layer.

The rest of the paper is organized as follows. In Section II, we will briefly discuss the discretization assumptions, present the *i*BP algorithm, and give some details about the generation of the coordinates of candidate positions at each iteration of *i*BP. In Section III, we will propose a new scheme, based on the intersection of several Euclidean objects, for the computation of candidate positions that are *all feasible* at the current layer. Section IV will show some experiments on artificially generated instances, while conclusions will be drawn in Section V.

II. The *i*BP algorithm

Let $G = (V,E,d)$ be an MDGP instance. The subclass of MDGP instances that we consider in this paper is defined as follows. Let $E' \subset E$ be the subset of edges for which their weights d are exact distances.

The interval *Discretizable DGP in dimension 3 (iDDGP$_3$).* Given a simple weighted undirected graph $G = (V,E,d)$, we say that G represents an instance of the *i*DDGP$_3$ if and only if there exists an order on the vertices of V verifying the following conditions:

(a) $G_C = (V_C, E_C) \equiv G[\{1,2,3\}]$ is a clique and $E_C \subset E'$;
(b) $\forall i \in \{4,\ldots,|V|\}$, there exists $\{i',i'',i'''\}$ such that
  1) $i''' < i$, $i'' < i$, $i' < i$;
  2) $\{(i'',i),(i',i)\} \subset E'$ and $(i''',i) \in E$;
  3) $d_{i',i'''} < d_{i',i''} + d_{i'',i'''}$.

Orders satisfying (a) and (b) are named "Discretization Orders". We refer to $\{i''',i'',i'\}$ as *reference atoms*, and to $d_{i''',i}$, $d_{i'',i}$ and $d_{i',i}$ as *reference distances*.

Notice that assumption (a) allows us to place the first 3 atoms uniquely, avoiding to consider congruent solutions that can be obtained by rotations and translations [7]. Assumption (b1) ensures the existence of three reference atoms for every $i > 3$, and assumption (b2) ensures that at most one of the three reference distances may be represented by an interval [12]. Finally, assumption (b3) avoids the reference atoms to be collinear. We remark that assumption (b3) cannot always be verified before the solution of an instance, because some of the necessary distances may not be available (the corresponding edges may not be in E). However, this assumption can fail to be satisfied with probability 0, and therefore we do not really need to verify it in advance [6]. Under these assumptions, the MDGP can be discretized, i.e. the instance at hand belongs to the *i*DDGP$_3$ class. In this case, the search domain becomes a tree, where nodes contain candidate atomic positions, organized layer by layer.

Algorithm 1 The *i*BP algorithm.

```
1:  iBP(i,n,d,D)
2:  if (i > n) then
3:    // one solution is found
4:    print current conformation;
5:  else
6:    // coordinate computation
7:    if (d_{i''',i} is an interval) then
8:      compute the two candidate arcs;
9:      add them to the list L;
10:   else
11:     compute the two candidate positions;
12:     add them to the list L;
13:   end if
14:   for h = 1,...,|L| do
15:     if (L(h) is an arc) then
16:       take D samples from the arc; set N = D;
17:     else
18:       set N = 1;
19:     end if
20:     // verifying the feasibility of the computed positions
21:     for k = 1,...,N do
22:       if (x_i^{h,k} is feasible) then
23:         iBP(i+1,n,d,D);
24:       end if
25:     end for
26:   end for
27: end if
```

We employ an *interval* Branch & Prune (*i*BP) algorithm [8] for the solution of discretizable instances. Alg. 1 is a sketch of this algorithm. The *i*BP algorithm performs a recursive search on the tree which represents the search domain. At each recursive call, candidate positions for the current atom are computed by exploiting the coordinates of previously placed atoms and the distance information ensured by the discretization assumptions. When all reference distances are exact, then two candidate positions are computed. When one of the references is an interval, two *feasible arcs* are rather identified (see Fig. 1).

In the algorithm call, i is the current atom for which candidate positions are currently searched, n is the total number of atoms forming the considered molecule, d is the list of available distances (exact and interval distances), and D is the discretization factor, i.e. the number of sample points that are taken from the arcs when the distance $d_{i''',i}$ is represented by an interval (see assumption (b2)). In the algorithm (see lines 9 and 12), we make use of a list L of positions and arcs, from which candidate positions are extracted.

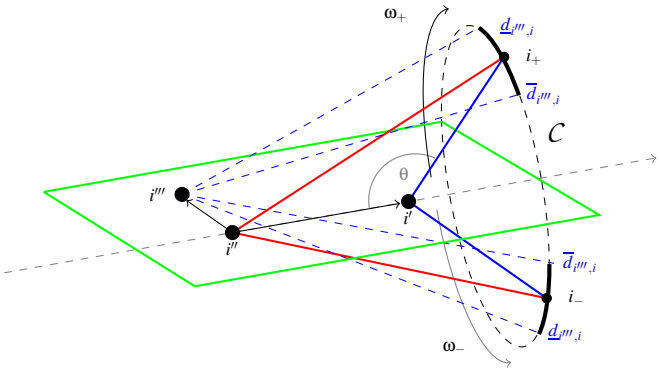

Fig. 1. The two feasible arcs (in bold, black) obtained by intersecting two spheres and one spherical shell.

When working on the atom i, feasible positions for its three reference atoms $\{i''', i'', i'\}$, on the current tree branch, are already available. These reference atoms define a local coordinate system centered at i' [4], [14]. The possible positions for the atom i verifying $d_{i',i}$ and $d_{i'',i}$ can be described by two angles θ_i and ω_i. Using $d_{i'',i'}$ and the cosine law, we can obtain a value for $\theta_i \in [0,\pi]$. Thus, the circle $\mathcal{C}$ of possible positions for atom i (see Fig. 1) can be described in terms of ω_i:

$$x_i(\omega_i) = x_{i'} + U_{i'} w_i, \tag{2}$$

where

$$w_i = \begin{bmatrix} -d_{i',i}\cos\theta_i \\ d_{i',i}\sin\theta_i\cos\omega_i \\ d_{i',i}\sin\theta_i\sin\omega_i \end{bmatrix},$$

$\omega_i \in [0, 2\pi]$, and $U_{i'}$ is the rotation (change of basis) matrix from the local system at i' to the canonical system of coordinates [4].

If $d_{i''',i}$ is exact, at most two values for ω_i, say $\{\omega_i^+, \omega_i^-\}$, can be computed. Two possible positions x_i^+ and x_i^- can be therefore identified for the atom i. These positions are symmetric with respect to the plane defined by the reference atoms. If $d_{i''',i}$ is instead an interval, then two disjoint and symmetric candidate arcs are obtained, as shown in Fig. 1. They correspond to two intervals, $[\underline{\omega}_i^+, \overline{\omega}_i^+]$ and $[\underline{\omega}_i^-, \overline{\omega}_i^-]$, for the angle ω_i. By selecting D equidistant angles in $[\underline{\omega}_i^+, \overline{\omega}_i^+]$ and other D equidistant angles in $[\underline{\omega}_i^-, \overline{\omega}_i^-]$, $2 \times D$ atomic positions for the current atom i can be computed.

In the standard *i*BP, the feasibility of these candidate atomic positions is verified by exploiting the so-called pruning distances. The Direct Distance Feasibility (DDF) is the pruning device that, for each candidate position related to the current atom i, verifies whether the inequality

$$\underline{d}_{ij} - \varepsilon \leq ||\mathrm{x}_i - \mathrm{x}_j|| \leq \overline{d}_{ij} + \varepsilon, \tag{3}$$

is satisfied for each atom $j < i$ that is not involved in the discretization, where $\varepsilon > 0$ is a given tolerance. This way, however, a large number of generated positions may be pruned and only a few of them may be actually feasible. The scheme we propose in this paper aims at overcoming this issue.

Finally, we remark that an essential pre-processing step, before applying the *i*BP, is to find a discretization order for the vertices of the graph G that allow to satisfy the assumptions in the *i*DDGP$_3$ definition. This preprocessing step can be performed efficiently, in polynomial time [11], so that the necessary assumptions can be fulfilled by graphs related to proteins.

III. Adaptive branching in *i*BP

The discretization of the two candidate arcs, used in the standard *i*BP algorithm when interval data are available, represents the simplest way to deal with imprecise information about the distances [8]. The candidate arcs are discretized by considering a finite number of samples in the two intervals $[\underline{\omega}_i^+, \overline{\omega}_i^+]$ and $[\underline{\omega}_i^-, \overline{\omega}_i^-]$, and then a new branch is created for each of them. If D is the discretization factor (see Alg. 1), $2 \times D$ positions are generated, and $2 \times D$ new branches are added to the tree at the current layer. When considering this approach, it is expected that at least one of such samples is able to fulfill the pruning distance constraints at the current layer.

The value given to D plays a critical role. On the one hand, too small values can generate trees where no solutions can be found (all branches are pruned, because no positions are compatible to the pruning distances). On the other hand, too large D values can drastically increase the width of the tree. Unfortunately, no upper bound on D can theoretically be defined: in the case only one specific singleton in the given arcs is actually feasible, only an infinite number of samples could guarantee that this singleton can be discovered. However, this is the worst case scenario: nondegenerate subarcs generally result to be feasible w.r.t. the available pruning distances.

In the standard *i*BP, after the generation of candidate atomic positions, their feasibility is verified by employing pruning devices, such as DDF (see Section II). There are two extreme situations:

1) all positions are feasible: this suggests that we could consider a smaller D value without harming the computations;
2) all positions are infeasible: since a finite number of samples on the two arcs are taken, this information does not allow us to discriminate between "the two arcs are infeasible" and "the chosen samples are infeasible".

The adaptive scheme that we propose was conceived for tailoring the branching phase of the *i*BP algorithm so that all computed candidate positions are feasible at the current layer. The basic idea is to identify, before the branching phase of the algorithm, the subset of positions (if it exists) on the two candidate arcs that is feasible with respect to all pruning distances to be verified at the current layer.

Let us consider expression (2), which is able to give the Cartesian coordinates of the atom i as a function of the torsion angle ω_i. For simplifying the notations, we will omit, in the following, the subscripts of the angles θ_i and ω_i.

In case the distance $d_{i''',i}$ is represented by an interval, i.e. $d_{i''',i} \in [\underline{d}_{i''',i}, \overline{d}_{i''',i}]$, two candidate arcs can be computed (see

Section II). These two arcs correspond to the two interval torsion angles $[\underline{\omega}^+,\overline{\omega}^+] \subset [0,\pi)$ and $[\underline{\omega}^-,\overline{\omega}^-] \subset [\pi,2\pi)$. All points in those two arcs satisfy therefore the interval distance $[\underline{d}_{i''',i},\overline{d}_{i''',i}]$, as well as the two exact distances $d_{i'',i}$ and $d_{i',i}$. However, there can be pruning distances (between already placed atoms and i) that we could exploit for tightening these two arcs. Therefore, instead of using these distances for pruning pre-computed positions, our idea is to exploit pruning distances for tightening the two arcs before sampling, so that all generated positions can be feasible (at least at the current layer).

Tightening the feasible arcs

Let us suppose there is an $h \in \{j < i \mid j \notin \{i''',i'',i'\}\}$, such that the pruning distance $d_{h,i}$ is known. Solutions to the equation

$$d_{h,i} = \|x_h - x_i(\omega)\| \tag{4}$$

give the values for the angle ω that are compatible with the distance $d_{h,i}$. By squaring equation (4), and by using (2), we obtain

$$\begin{aligned} d_{h,i}^2 &= \|x_h - x_i(\omega)\|^2 \\ &= \|x_h - (x_{i'} + U_{i'} w_i)\|^2 \\ &= \|x_h - x_{i'}\|^2 - 2\langle x_h - x_{i'},\ U_{i'} w_i\rangle + \|U_{i'} w_i\|^2, \end{aligned}$$

where $\langle\cdot,\cdot\rangle$ denotes the inner product between two vectors. Since $U_{i'}$ is an orthogonal matrix, we have

$$d_{h,i}^2 = \|x_h - x_{i'}\|^2 - 2\langle x_h - x_{i'},\ U_{i'} w_i\rangle + d_{i',i}^2.$$

Let $v = x_h - x_{i'}$ and let $\hat{x},\hat{y},\hat{z}$ be the columns of $U_{i'}$. Then:

$$\begin{aligned} d_{h,i}^2 &= \|v\|^2 - 2\langle v,\ U_{i'} w_i\rangle + d_{i',i}^2 \\ &= \|v\|^2 + d_{i',i}^2 - 2\langle v, (-d_{i',i}\cos\theta)\hat{x} + \\ &\quad (d_{i',i}\sin\theta\cos\omega)\hat{y} + (d_{i',i}\sin\theta\sin\omega)\hat{z}\rangle \\ &= \|v\|^2 + d_{i',i}^2 - 2\left(\langle v,\hat{x}\rangle(-d_{i',i}\cos\theta) + \right. \\ &\quad \left. \langle v,\hat{y}\rangle(d_{i',i}\sin\theta)\cos\omega + \langle v,\hat{z}\rangle(d_{i',i}\sin\theta)\sin\omega\right). \end{aligned}$$

If we set

$$A = 2\langle v,\hat{y}\rangle(d_{i',i}\sin\theta), \tag{5}$$

$$B = 2\langle v,\hat{z}\rangle(d_{i',i}\sin\theta),$$

$$\Delta = \|v\|^2 + d_{i',i}^2 + 2\langle v,\hat{x}\rangle(d_{i',i}\cos\theta),$$

and

$$C = \Delta - d_{h,i}^2,$$

we obtain the following equation:

$$A\cos\omega + B\sin\omega = C. \tag{6}$$

Solving $A\cos\omega + B\sin\omega = C$

In order to solve equation (6), we consider the following approach. We set

$$A = R\cos\alpha, \tag{7}$$

$$B = R\sin\alpha, \tag{8}$$

and, in order to obtain R, we square and sum the two equations (7) and (8):

$$A^2 + B^2 = R^2\cos^2\alpha + R^2\sin^2\alpha = R^2(\cos^2\alpha + \sin^2\alpha) = R^2.$$

If we consider the positive square root (R can be seen as the length of a triangle side), we have

$$R = \sqrt{A^2 + B^2}.$$

If $A \neq 0$, we can divide (8) by (7), and obtain

$$\frac{B}{A} = \frac{\sin\alpha}{\cos\alpha} = \tan\alpha,$$

or, equivalently

$$\alpha = \tan^{-1}\left(\frac{B}{A}\right).$$

The correct quadrant for α can be identified by checking the signs of $\cos\alpha$ and $\sin\alpha$.

Notice that, when both A and B are zero, we can have either no solutions or an infinite number of solutions. When $A = B = 0$, then $v = x_h - x_{i'}$ is on the $\hat{x}$ axis, because $\sin\theta \neq 0$ (assumption (b3)) and $d_{i',i} > 0$ (see equation (5)). Atoms h,i'',i' are therefore aligned and the sphere centered in x_h does match with the whole dashed circle $\mathcal{C}$ (when there are infinite solutions) or does not (when there are no solutions). If $A = 0$ and $B \neq 0$, then $\cos\alpha = 0$ and α is either $\pi/2$ or $-\pi/2$, depending on the sign of B.

When $A \neq 0$, from equations (6), (7) and (8), we can obtain

$$\begin{aligned} A\cos\omega + B\cos\omega &= R\cos\alpha\cos\omega + R\sin\alpha\sin\omega \\ &= R\cos(\omega - \alpha), \end{aligned}$$

and hence

$$R\cos(\omega - \alpha) = C,$$

which is

$$\omega = \alpha \pm \cos^{-1}\left(\frac{C}{R}\right). \tag{9}$$

Therefore, we usually have two solutions for equation (6) in $[0,2\pi)$. There are two exceptions. When $C = R$, we have only one solution; when $C/R \notin [-1,1]$, there are no intersection points.

Solutions to equation (6) (and therefore to equation (4)) provide the points where the sphere, centered at x_h and with radius $d_{h,i}$, intersects the circle $\mathcal{C}$ in Fig. 1. Those points are the extreme points of the feasible arcs: they define feasible intervals for the angle ω. Fig. 2 shows some possible intersections between the spherical shell centered in x_h (having minimum radius $\underline{d}_{h,i}$ and maximum radius $\overline{d}_{h,i}$) with the dashed circle $\mathcal{C}$.

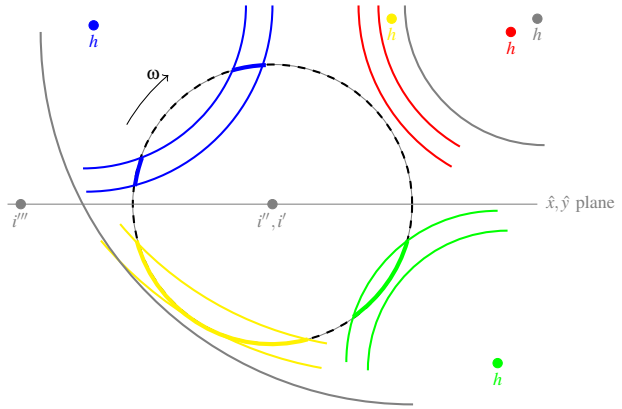

Fig. 2. Possible intersections between the spherical shell related to the distance $d_{h,i}$ and the circle of candidate positions related to $d_{i',i}$ and $d_{i'',i}$.

Managing different scenarios

The feasible positions for the current atom i can be obtained by intersecting the two arcs (computed by using the discretization distances, in bold in Fig. 1) and several spherical shells, each of them defined by considering a pruning distance between $h < i$ and i. In order to perform this intersection, the following two equations need to be solved

$$A\cos\omega + B\sin\omega = \Delta - \underline{d}^2_{h,i}, \tag{10}$$

$$A\cos\omega + B\sin\omega = \Delta - \overline{d}^2_{h,i}, \tag{11}$$

for every pruning distance $d_{h,i}$. There are three situations that can occur while performing the intersections (i.e. while solving equations (10) and (11)).

Both equations have no solutions: If both equations (10) and (11) have no solutions, then the entire candidate circle is either completely valid, or completely invalid. If we can find at least one value for ω such that

$$\underline{d}^2_{h,i} < \|x_h - x_i(\omega)\|^2 < \overline{d}^2_{h,i},$$

then the entire circle $\mathcal{C}$ is feasible w.r.t. the distance $d_{h,i}$. If not, it is sufficient to verify whether one of these 2 equations is satisfied

$$\max_{\omega\in[0,2\pi]} \|x_h - x_i(\omega)\|^2 < \underline{d}^2_{h,i},$$

$$\min_{\omega\in[0,2\pi]} \|x_h - x_i(\omega)\|^2 > \overline{d}^2_{h,i},$$

for stating that the entire circle is infeasible.

Only one equation has solutions: Let us suppose that only equation (10) has solutions. In this case, the resulting intersection is an interval $[\underline{\omega}, \overline{\omega}]$ whose extreme points are the solutions of equation (10). In order to find the right orientation of the arc on the circle $\mathcal{C}$, we define the function

$$F(\omega) = \|x_h - x_i(\omega)\|^2 = \Delta - A\cos\omega - B\sin\omega,$$

and we consider its derivative

$$F'(\omega) = A\sin\omega - B\cos\omega. \tag{12}$$

The orientation at an extreme point (solution of (10)) is the one for which $F(\omega)$ increases, and this information is given by (12) evaluated in such an extreme point. Notice that we might need to add 2π to one of the extreme points in order to have $\overline{\omega} > \underline{\omega}$. The analysis in the case in which only equation (11) has solutions is analogous.

Let $[\underline{\omega^*}, \overline{\omega^*}]$ be the obtained interval for ω. If this interval has an empty intersection with the two initial arcs in $\mathcal{C}$, then there are no feasible positions, and the current branch of the search domain can be pruned. If this intersection is instead non-empty, then the result provides the interval for ω that is feasible w.r.t the discretization distances, as well as the pruning distance $d_{h,i}$. Notice that, when more than one pruning distance is available, the same procedure can be repeated as many times as the number of available pruning distances.

Both equations have solutions: When both equations (10) and (11) have solutions, we obtain four values for ω: two from equation (10) and other two from equation (11). Two intervals can be therefore defined for ω, related to two arcs in $\mathcal{C}$. Both arcs need to be intersected with the initial arcs. The procedure to apply is analogous to the one presented for the previous case.

iBP and the new adaptive scheme

After considering all pruning distances, after performing all intersections, the final result provides a list of arcs on $\mathcal{C}$ that are feasible with all the distances that can be verified at the current layer of the *i*BP tree. All positions that can be taken from these arcs are feasible at the current layer: all of them generate a new branch and may serve as a reference for computing new candidate positions on deeper layers of the tree. In order to integrate the *i*BP algorithm with this adaptive scheme, there are two main changes to be performed on Alg. 1. On line 8 and 11, the adaptive scheme needs to be invoked for taking into consideration the information about the pruning distances. Moreover, line 22 needs to be removed, because this verification is not necessary anymore (unless other pruning devices rather than DDF are employed).

IV. Computational Experiments

Experiments of Nuclear Magnetic Resonance (NMR) [10] are able to provide estimates of some relative distances between pairs of atoms of a molecule. We present in this section some computational experiments on artificially generated NMR instances, where we compare the standard *i*BP algorithm to the new *i*BP integrated with our adaptive scheme for the generation of feasible atomic positions (accordingly to all available distances at the current tree layer). In this work, we do not consider real NMR data because the experiments here presented have the only aim of showing the advantages in using this new adaptive scheme. Later on, this scheme will be integrated in a more general framework capable of dealing with real NMR data. All codes were written in C programming language and all the experiments were carried out on an Intel Core 2 Duo @ 2.4 GHz with 2GB RAM, running Mac OS

Instance			*i*BP w/out adaptive scheme			*i*BP with adaptive scheme		
name	$\|V\|$	$\|E\|$	D	*i*BP calls	Time	D	*i*BP calls	Time
1niz	68	328	5	7668930	9.93	5	105543	0.15
2jnr	96	443	5	17410	0.02	5	16989	0.02
2pv6	110	558	7	174651	0.27	5	181020	0.24
1zec	122	622	6	1194478	1.92	5	932428	1.53
2mla	130	681	5	323354	0.54	5	136547	0.25
2me1	135	687	6	2813983	4.30	5	1415331	2.35
2me4	135	681	5	1533970	2.36	5	249096	0.40
1dsk	140	733	6	3746764	5.34	6	1091745	1.52

TABLE I
EXPERIMENTS ON OUR ARTIFICIALLY GENERATED NMR INSTANCES.

X. The codes have been compiled by the GNU C compiler v.4.0.1 with the `-O3` flag.

The instances that we consider in the experiments have been generated as follows. We consider a subset of proteins from the Protein Data Bank (PDB) [1] that are related to human immunodeficiency. Together with the coordinates of the atoms available on the PDB, we suppose having the chemical structure of the protein, i.e. information about bond lengths and angles. Once the coordinates are loaded from the PDB files, we compute all distances between atom pairs belonging to the protein backbone, and we add a distance in our instances if the computed distance is between:

1) two bonded atoms (considered as exact);
2) two atoms that are bonded to a common atom (considered as exact);
3) two atoms belonging to a quadruplet of bonded atoms forming a torsion angle (considered as an interval);
4) two hydrogen atoms (considered as an interval, if the distance belongs the interval [2.5,5] Å).

We remark that the first 3 items are related to the chemical structure of the molecule; only the last item concerns distances that simulate NMR data. The distances that are derived from the information mentioned in item 3 are generally intervals; however, when one of the possible torsion angles is related to the peptide bond (that connects pairs of consecutive amino acids), the distance is considered as exact, because the peptide bond forces all atoms to lie on the same plane. Interval distances coming from torsion angles are computed so that all possible values for the torsion angle are allowed. The interval distances related to item 4 have instead length equal to 2Å, and their bounds were generated so that the *true* distance is randomly placed inside the interval. After the computation of the distance information, the atoms in every instance have been reordered by considering the discretization order published in [11], which is valid for every protein backbone.

In Table I we compare the performance of the previous version of *i*BP [8] with our new one, where the adaptive branching scheme presented in Section III is implemented. For each instance, we report the label of the corresponding file on the PDB, the total number $|V|$ of atoms and the number $|E|$ of available distances. Moreover, for each *i*BP version, we report the number D of samples to be taken from each candidate arc, the number of *i*BP calls and the CPU time in seconds, that are necessary to find one solution. In the DDF pruning device (equation (3)), the used tolerance ε is 10^{-3}.

The D values in Table I are actually the smallest ones for which *i*BP could find at least one solution in a given time limit (10 seconds in these experiments). When using our adaptive branching scheme, the D value never increased and it was reduced in some cases. This was expected because our adaptive scheme is able to guide the sample points in the feasible regions of the candidate arcs. Even if the computation of the intersections may increase the computational cost for single *i*BP recursive calls, the overall CPU time for each experiment is lower when the adaptive scheme is employed. This is due to the fact that, when the branching phase in *i*BP is adaptive, only feasible coordinates are generated: there are no useless computations (i.e. computed positions that are immediately discarded).

V. CONCLUSIONS

We proposed a new adaptive branching scheme that was integrated in the *i*BP algorithm to solve discretizable MDGPs with interval data. When interval data are used in the discretization process, candidate positions for the current atom are generally represented by two candidate arcs. By exploiting the interval pruning distances that can be verified at the current layer, we can guide the branching phase of the *i*BP algorithm to take samples only on the feasible regions of the candidate arcs.

As it was assessed by our computational experiments, this approach improves the overall performances of the *i*BP algorithm, thereby improving its robustness. Using the intersections of the spherical shells defined by the pruning distances with the candidate arcs provided by the discretization, we avoid the generation of useless samples in infeasible portions of the candidate arcs.

However, it is important to mention that, as in the previous *i*BP version, the presented scheme does not guarantee that the chosen sample positions can lead to feasible positions at further layers: our scheme ensures the feasibility only up to the current layer. Predicting the compatibility of sample positions with the atoms that *follow* the current one is a topic of future research.

VI. ACKNOWLEDGMENTS

We are thankful to Brittany Region (France) which funded a 1-year postdoc for DG at IRISA, University of Rennes 1

(stratégie d'attractivité durable). This work is partially supported by the ANR project ANR-10-BINF-03-01 "Bip:Bip". CL is also thankful to FAPESP and CNPq for financial support.

REFERENCES

[1] H.M. Berman, J. Westbrook, Z. Feng, G. Gilliland, T.N. Bhat, H. Weissig, I.N. Shindyalov, P.E. Bourne, *The Protein Data Bank*, Nucleic Acid Research **28**, 235–242, 2000.

[2] V. Costa, A. Mucherino, C. Lavor, A. Cassioli, L.M. Carvalho, N. Maculan, *Discretization Orders for Protein Side Chains*, to appear in Journal of Global Optimization, 2014.

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

[4] D.S. Gonçalves, A. Mucherino, *Discretization Orders and Efficient Computation of Cartesian Coordinates for Distance Geometry*, to appear in Optimization Letters, 2014.

[5] T.F. Havel, *Distance Geometry*, D.M. Grant and R.K. Harris (Eds.), Encyclopedia of Nuclear Magnetic Resonance, Wiley, New York, 1701-1710, 1995.

[6] C. Lavor, J. Lee, A. Lee-St.John, L. Liberti, A. Mucherino, M. Sviridenko, *Discretization Orders for Distance Geometry Problems*, Optimization Letters **6**(4), 783–796, 2012.

[7] C. Lavor, L. Liberti, N. Maculan, A. Mucherino, *The Discretizable Molecular Distance Geometry Problem*, Computational Optimization and Applications **52**, 115–146, 2012.

[8] C. Lavor, L. Liberti, A. Mucherino, *The interval Branch-and-Prune Algorithm for the Discretizable Molecular Distance Geometry Problem with Inexact Distances*, Journal of Global Optimization **56**(3), 855–871, 2013.

[9] L. Liberti, C. Lavor, N. Maculan, A. Mucherino, *Euclidean Distance Geometry and Applications*, SIAM Review **56**(1), 3–69, 2014.

[10] T.E. Malliavin, A. Mucherino, M. Nilges, *Distance Geometry in Structural Biology: New Perspectives*. In: [13], 329–350, 2013.

[11] A. Mucherino, *On the Identification of Discretization Orders for Distance Geometry with Intervals*, Lecture Notes in Computer Science **8085**, F. Nielsen and F. Barbaresco (Eds.), Proceedings of Geometric Science of Information (GSI13), Paris, France, 231–238, 2013.

[12] A. Mucherino, C. Lavor, L. Liberti, *The Discretizable Distance Geometry Problem*, Optimization Letters **6**(8), 1671–1686, 2012.

[13] A. Mucherino, C. Lavor, L. Liberti, N. Maculan (Eds.), *Distance Geometry: Theory, Methods and Applications*, Springer, 2013.

[14] H.B. Thompson, *Calculation of Cartesian Coordinates and their Derivatives from Internal Molecular Coordinates*, Journal of Chemical Physics **47**, 3407, 1967.