Branch-and-Prune trees with bounded width

Leo Liberti [a,1] Benoît Masson [b,2] Carlile Lavor [c,3]
Antonio Mucherino [d,4]

[a] *LIX, École Polytechnique, 91128 Palaiseau, France*

[b] *IRISA, INRIA, Campus de Beaulieu, 35042 Rennes, France*

[c] *Dept. of Applied Maths (IMECC-UNICAMP), State Univ. of Campinas, C.P. 6065, 13081-970, Campinas - SP, Brazil*

[d] *INRIA Futurs, Lille, France*

Key words: DMDGP, distance geometry, order, reflection, symmetry.

1 Introduction

The Molecular Distance Geometry Problem, which asks to find the embedding in $\mathbb{R}^3$ of a given weighted undirected graph, is a good model for determining the structure of proteins given a set of inter-atomic distances [6,4]. Its generalization to $\mathbb{R}^K$ is called Distance Geometry Problem (DGP), which has applications in wireless sensor networks [2] and graph drawing. In general, the MDGP and DGP implicitly require a search in a continuous Euclidean space. Proteins, however, have further structural properties that can be exploited to define subclasses of instances of the MDGP and DGP whose solution set is finite [5]. These instances can be solved with an algorithmic framework called Branch-and-Prune (BP) [3,5]: this is an iterative algorithm where the i-th atom of the protein can be embedded in $\mathbb{R}^3$ using distances to at least three preceding atoms. Since the intersection of three 3D spheres contains in general two points, the BP gives rise to a binary search tree. In the worst case, the BP is an exponential time algorithm, which is fitting because the MDGP and DGP are **NP**-hard [9].

[1] Corresponding author; e-mail:liberti@lix.polytechnique.fr.
[2] E-mail:benoit.masson@inria.fr.
[3] E-mail:clavor@ime.unicamp.br.
[4] E-mail:antonio.mucherino@inria.fr.

Compared to continuous search algorithms, the performance of the BP algorithm is impressive from the point of view of both efficiency and reliability. In this paper we try to explain why the BP algorithm is so much faster than other approaches notwithstanding its worst-case exponential running time. Specifically, using the particular structure of the protein graph, we argue that it is reasonable to expect that the BP will yield a search tree of bounded width.

2 Discretizable instances and the BP algorithm

For all integers $n > 0$, we let $[n] = \{1, \dots, n\}$. Given an undirected graph $G = (V, E)$ with $|V| = n$, for all $v \in V$ we let $N(v) = \{u \in V \mid \{u, v\} \in E\}$ be the set of vertices *adjacent* to v. Given a positive integer K, an *embedding* of G in $\mathbb{R}^K$ is a function $x : V \to \mathbb{R}^K$. If $d : E \to \mathbb{R}_+$ is a given edge weight function on $G = (V, E, d)$, an embedding is *valid* for G if $\forall \{u, v\} \in E \; \|x_u - x_v\| = d_{uv}$, where $x_v = x(v)$ for all $v \in V$ and $d_{uv} = d(\{u, v\})$ for all $\{u, v\} \in E$. For any $U \subseteq V$, an embedding of $G[U]$ (i.e. the subgraph of G induced by U) is a *partial embedding* of G. If x is a partial embedding of G and y is an embedding of G such that $\forall u \in U \; (x_u = y_u)$ then y is an *extension* of x. For a total order $<$ on V and for each $v \in V$, let $\rho(v) = |\{u \in V \mid u \leq v\}|$ be the *rank* of v in V with respect to $<$. The rank is a bijection between V and $[n]$, so we can identify v with its rank and extend arithmetic notation to V so that for $i \in \mathbb{Z}$, $v + i$ denotes the vertex $u \in V$ with $\rho(u) = \rho(v) + i$. For all $v \in V$ and $\ell < \rho(v)$ we denote by $\gamma_\ell(v)$ the set of ℓ immediate predecessors of v. If $U \subseteq V$ with $|U| = h$ such that $G[U]$ is a clique, let $D'(U)$ be the symmetric matrix whose (u, v)-th component is d_{uv}^2 for $u, v \in U$, and let $D(U)$ be $D'(U)$ bordered by a left $(0, 1, \dots, 1)^\top$ column and a top $(0, 1 \dots, 1)$ row (both of size $h+1$). Then the Cayley-Menger formula [1] states that the volume in $\mathbb{R}^{h-1}$ of the h-simplex defined by $G[U]$ is given by $\Delta_{h-1}(U) = \sqrt{\frac{(-1)^h}{2^{h-1}((h-1)!)^2}|D(U)|}$.

Generalized Discretizable Molecular Distance Geometry Problem ($^{\mathsf{K}}$DMDGP). Given an integer $K > 0$, a weighted undirected graph $G = (V, E, d)$ with $d : E \to \mathbb{Q}_+$, a total order $<$ on V and an embedding $x' : [K] \to \mathbb{R}^K$ such that:
(1) x' is a valid partial embedding of $G[[K]]$ (Start)
(2) G contains all $(K+1)$-cliques of $<$-consecutive vertices as induced subgraphs (Discretization)
(3) $\forall v \in V$ with $v > K$, $\Delta_{K-1}(\gamma_K(v)) > 0$ (Strict Simplex Inequalities),
is there a valid embedding x of G in $\mathbb{R}^K$ extending x'?

We denote by X the set of embeddings solving a $^{\mathsf{K}}$DMDGP instance. The $^{\mathsf{K}}$DMDGP generalizes the Discretizable Molecular Distance Geom-

etry Problem (DMDGP) [3] from $\mathbb{R}^3$ to $\mathbb{R}^K$. Furthermore, it is a subclass of the Discretizable Distance Geometry Problem (DDGP) [8] given by all DDGP instances where the K adjacent predecessors used to determine the two positions for the current vertex are immediate. Since for the DDGP $|X|$ is finite [8], this also holds for the $^{\mathsf{K}}$DMDGP; and since the DMDGP is **NP**-hard [3], the same is true for the $^{\mathsf{K}}$DMDGP. For a partial embedding x of G and $\{u, v\} \in E$ let S^x_{uv} be the sphere centered at x_u with radius d_{uv}. The BP

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

Require: A vtx. $v \in V \setminus [K]$, a partial emb. $\bar{x} = (x_1, \ldots, x_{v-1})$, a set X.

1: $P = \bigcap_{\substack{u \in N(v) \\ u < v}} S^{\bar{x}}_{uv}$;

2: $\forall p \in P$ ($(x \leftarrow (\bar{x}, p))$; **if** $(\rho(v) = n)$ $X \leftarrow X \cup \{x\}$ **else** BP($v+1$, x, X)).

algorithm, used for solving the DDGP and its restrictions, is BP($K+1$, x', $\emptyset$) (see Alg. 1). By Strict Simplex Inequalities, $|P| \leq 2$. At termination, X contains all embeddings extending x' [3,5].

3 BP tree geometry

Since the definition of the $^{\mathsf{K}}$DMDGP requires G to have at least those edges used to satisfy the Discretization axiom, we partition E into the sets $E_D = \{\{u, v\} \mid |\rho(v) - \rho(u)| \leq K\}$ and $E_P = E \setminus E_D$. With a slight abuse of notation we call E_D the *discretization distances* and E_P the *pruning distances*. Discretization distances guarantee that a DGP instance is in the $^{\mathsf{K}}$DMDGP. Pruning distances are used to reduce the BP search space by pruning its tree. In practice, pruning distances might make the set P in Alg. 1 have cardinality 0 or 1 instead of 2. We assume G is a feasible instance of the $^{\mathsf{K}}$DMDGP.

Let $G_D = (V, E_D, d)$ and X_D be the set of embeddings of G_D; since G_D has no pruning distances, the BP search tree for G_D is a full binary tree and $|X_D| = 2^{n-K}$. The discretization distances arrange the embeddings so that, at level ℓ, there are $2^{\ell-K}$ possible embeddings x_v for the vertex v with rank ℓ. Furthermore, when $P = \{x_v, x'_v\}$ and the discretization distances to v only involve the K immediate predecessors of v, we have that $x'_v = R^v_x(x_v)$ [7], the reflection of x_v w.r.t. the hyperplane through $x_{v-K}, \ldots, x_{v-1}$. This also implies that the partial embeddings encoded in two BP subtrees rooted at reflected nodes ν, ν' are reflections of each other.

Theorem 1 ([7]) *With probability 1:* $\forall v > K, u < v - K \; \exists \; H^{uv} \subseteq \mathbb{R}$ *s.t.* $|H^{uv}| = 2^{v-u-K}$ *and* $\forall x \in X \; \|x_v - x_u\| \in H^{uv}$*; also* $\forall x \in X \; \|x_v - x_u\| = \|R^{u+K}_x(x_v) - x_u\|$ *and* $\forall x' \in X \; (x'_v \notin \{x_v, R^{u+K}_x(x_v)\} \rightarrow \|x_v - x_u\| \neq \|x'_v - x_u\|)$.

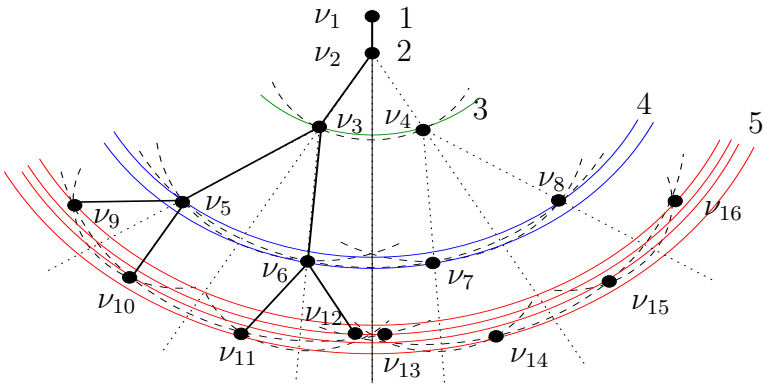

Fig. 1. A pruning distance $\{1, 4\}$ prunes either ν_6, ν_7 or ν_5, ν_8.

Proof. Sketched in Fig. 1; the circles mark distances to vertex 1. □

4 BP search trees with bounded width

Consider the BP tree for G_D and assume that there is a pruning distance $\{u, v\} \in E_P$; at level u there are $\max(2^{u-K}, 1)$ nodes, each of which is the root of a subtree with $2^{v-\max(u,K)}$ nodes at level v. By Thm. 1, for each such subtree only two nodes will encode a valid embedding for v (we call such nodes *valid*). Thus the number of valid nodes at level $v > K$ is $2^{\max(u-K+1,1)}$.

Fig. 2 shows a Directed Acyclic Graph (DAG) $\mathcal{D}_{uv}$ that we use to compute the number of valid nodes in function of pruning distances between two vertices $u, v \in V$ such that $v > K$ and $u < v - K$. The first line shows different values

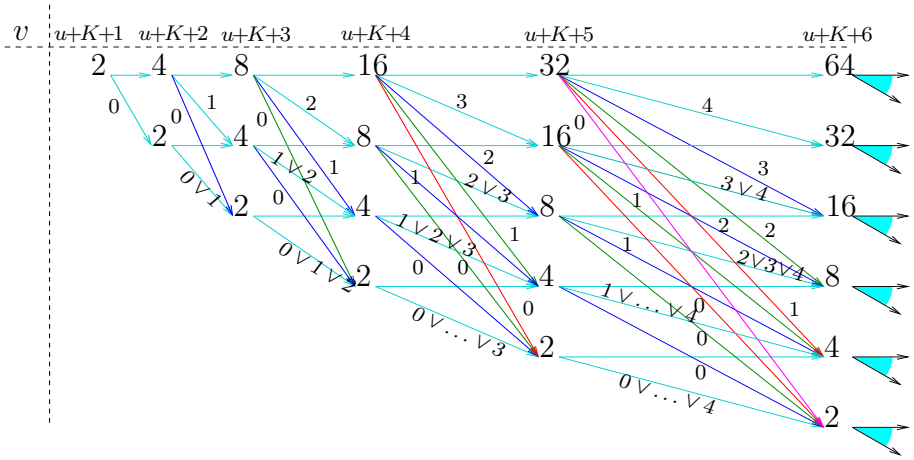

Fig. 2. Number of valid nodes in function of the pruning distances.

for the rank of v w.r.t. u; an arc labelled with an integer i implies the existence of a pruning distance $\{u + i, v\}$ (arcs with $\vee$-expressions replace parallel arcs

with different labels). An arc is unlabelled if there is no pruning distance $\{w, v\}$ for any $w \in \{u, \ldots, v-K-1\}$. The nodes of the DAG are arranged vertically by BP search tree level. A path p in this DAG represents the set of pruning distances between u and v: p_ℓ is the number of valid nodes in the BP search tree at level ℓ. For example, following unlabelled arcs corresponds to no pruning distance between u and v and leads to a full binary BP search tree with 2^{v-K} nodes at level v.

Each E_P corresponds to a longest path in $\mathcal{D}_{1n}$; BP trees have bounded width when these paths are below a diagonal with constant node labels.

Proposition 2 *If* $\exists v_0 \in V \setminus [K]$ *s.t.* $\forall v > v_0\ \exists! u < v - K$ *with* $\{u, v\} \in E_P$ *then the BP search tree width is bounded by* 2^{v_0-K}.

Proof. This corresponds to a path $\mathsf{p}_0 = (2, 4, \ldots, 2^{v_0-K}, \ldots, 2^{v_0-K})$ that follows unlabelled arcs up to level v_0 and then arcs labelled $v_0 - K - 1$, $v_0 - K - 1 \vee v_0 - K$, and so on, leading to nodes that are all labelled with 2^{v_0-K} (see Fig. 3, left). □

Proposition 3 *If* $\exists v_0 \in V \setminus [K]$ *such that every subsequence* s *of consecutive vertices* $>v_0$ *with no incident pruning distance is preceded by a vertex* v_s *such that* $\exists u_s < v_s\ (\rho(v_s) - \rho(u_s) \geq |s| \wedge \{u_s, v_s\} \in E_P)$, *then the BP search tree width is bounded by* 2^{v_0-K}.

Proof. Such instances yield paths that are below the path p_0 described in the proof of Prop. 2 (Fig. 3, right). □

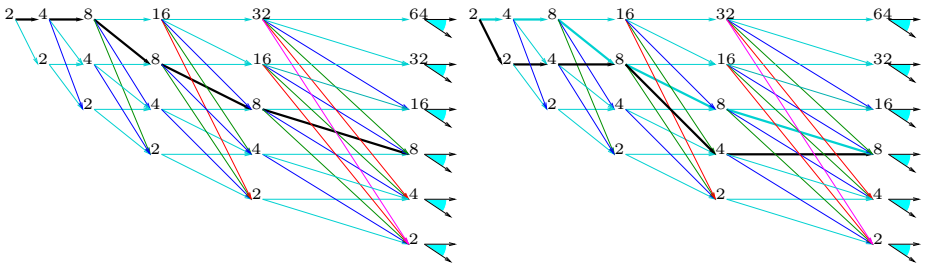

Fig. 3. A path p_0 with treewidth 8 (left) and another path below p_0 (right).

Moreover, For those instances for which the BP search tree width has a $O(\log n)$ bound, the BP has a polynomial worst-case running time $O(L2^{\log n}) = O(Ln)$, where L is the complexity of computing P.

On a set of 16 protein instances from the Protein Data Bank (PDB), twelve satisfy Prop. 4.1, and four Prop. 4.2, all with $v_0 = 4$. This validates the expectation that BP has polynomial complexity on real proteins.

References

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

[2] T. Eren, D.K. Goldenberg, W. Whiteley, Y.R. Yang, A.S. Morse, B.D.O. Anderson, and P.N. Belhumeur. Rigidity, computation, and randomization in network localization. *IEEE Infocom Proceedings*, pages 2673–2684, 2004.

[3] C. Lavor, L. Liberti, and N. Maculan. The discretizable molecular distance geometry problem. Technical Report q-bio/0608012, arXiv, 2006.

[4] C. Lavor, A. Mucherino, L. Liberti, and N. Maculan. On the computation of protein backbones by using artificial backbones of hydrogens. *Journal of Global Optimization*, accepted.

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

[6] L. Liberti, C. Lavor, A. Mucherino, and N. Maculan. Molecular distance geometry methods: from continuous to discrete. *International Transactions in Operational Research*, 18:33–51, 2010.

[7] L. Liberti, B. Masson, C. Lavor, J. Lee, and A. Mucherino. On the number of solutions of the discretizable molecular distance geometry problem. Technical Report 1010.1834v1[cs.DM], arXiv, 2010.

[8] A. Mucherino, C. Lavor, and L. Liberti. The discretizable distance geometry problem. *Optimization Letters*, in revision.

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