The K-discretization and K-incident graphs for discretizable Distance Geometry

Germano Abud[1]**, Jorge Alencar**[2]**,**
Carlile Lavor[3]**, Leo Liberti**[4]**,**
Antonio Mucherino[5]

Abstract The Distance Geometry Problem (DGP) is the problem of determining whether a realization for a simple weighted undirected graph $G = (V, E, d)$ in a given Euclidean space exists so that the distances between pairs of realized vertices $u, v \in V$ correspond to the weights d_{uv}, for each $\{u, v\} \in E$. We focus on a special class of DGP instances, referred to as the Discretizable DGP (DDGP), and we introduce the K-discretization and the K-incident graphs for the DDGP class. The K-discretization graph is independent on the vertex order that can be assigned to V, and can be useful for discovering whether one of such orders actually exists so that the DDGP assumptions are satisfied. The use of a given vertex order allows the definition of another important graph, the K-incident graph, which is potentially useful for performing pre-processing analysis on the solution set of DDGP instances.

1 Introduction

Given a simple weighted undirected graph $G = (V, E, d)$ and a positive integer K, the Distance Geometry Problem (DGP) asks whether a realization

$$x : V \longrightarrow \mathbb{R}^K$$

[1]FAMAT, Federal University of Uberlândia, Minas Gerais, Brazil.
Email: `germano.abud@ufu.br`
[2]Federal Institute of the South of Minas Gerais, Brazil.
Email: `jorge.alencar@ifsuldeminas.edu.br`
[3]IMECC, University of Campinas, Brazil.
Email: `clavor@ime.unicamp.br`
[4]CNRS LIX, École Polytechnique, Palaiseau, France.
Email: `liberti@lix.polytechnique.fr`
[5]IRISA and University of Rennes 1, Rennes, France.
Email: `antonio.mucherino@irisa.fr`

of G in the Euclidean space $\mathbb{R}^K$ exists so that

$$\forall \{u, v\} \in E, \quad ||x_u - x_v|| = d_{uv}, \tag{1}$$

where d_{uv} is the weight on the edge $\{u, v\} \in E$ and $|| \cdot ||$ represents the Euclidean norm. We say that the realization x, satisfying the distance constraints in eq. (1), is a *valid realization* of G [8]. The DGP is NP-hard [13].

This paper concerns a particular class of DGP instances, where the search space of the problem can be discretized and, this way, reduced to a search tree that is binary under the hypothesis that all distances d_{uv} are precise [9,11]. Let n be the number of vertices in V and let $G[\cdot]$ be the subgraph of G induced by the subset of V given as an argument. Given K vertices $u_1, u_2, \ldots, u_K$, let $\mathcal{S}(u_1, u_2, \ldots, u_K)$ be the volume of the simplex that is a valid embedding of such vertices.

Definition 1 THE DISCRETIZABLE DGP (DDGP).
Given a DGP instance (formed by an integer $K > 0$ and a graph $G = (V, E, d)$) and a vertex order on V such that for each $v \in \{K + 1, \ldots, n\}$ there exist K reference vertices $u_1(v), u_2(v), \ldots, u_K(v)$ (simply denoted by $u_1, \ldots, u_K$ for simplicity) with the following properties:

(a) $G[\{u_1, \ldots, u_K\}]$ is a clique;
(b) $u_1 < v, \ldots, u_K < v$ (where $<$ is w.r.t. the vertex order);
(c) $\{\{u_1, v\}, \{u_2, v\}, \ldots, \{u_K, v\}\} \subset E$;
(d) $\mathcal{S}(u_1, u_2, \ldots, u_K) > 0$,

determine whether there exists a realization $x : V \to \mathbb{R}^K$ of the given graph in the K-dimensional Euclidean space such that eq. (1) holds.

We call a graph G satisfying Defn. 1 *discretizable*.

Property (a) in Defn. 1 allows us to fix the coordinate space where all solutions can be constructed, so that no solutions can be obtained from others by applying total translations, rotations and reflections (except the reflection around the (hyper-)plane defined by the vertices of the initial clique). Properties (b)-(d) ensure that, for every vertex v that does not belong to the initial clique, there exist at least K *reference vertices*, with known *reference distances* (i.e. the weights of the edges mentioned in Property (c)), that can be used for finding possible positions for v.

As is well-known, the DDGP can be solved using an algorithm called Branch-and-Prune (BP) [9,8]. The BP algorithm is based on the idea of recursively constructing the search tree by initializing, at each step, two new tree branches, and to verify the feasibility of candidate vertex positions as soon as they are generated [7]. As such, BP generates algorithmic traces that are essentially search trees. The BP algorithm was shown to be very efficient on DDGP instances consisting of exact distances [5], and extensions of this algorithm are under study for the more general case where some distances may be imprecise or represented by suitable real-valued intervals [1,3,6].

The search trees generated by BP are organized so that all possible positions for the same vertex v belong to a common layer of the tree. A solution

can be represented as a path from the root node (the first vertex of the initial clique) to one of its leaf nodes (representing positions for the last vertex in V). We will refer to the associated vertex orders r as *discretization orders*, where $|r|$ indicates their length.

In this paper, we introduce two auxiliary graphs that can be derived from the given graph G: the K-discretization and the K-incidence graph. The *K-discretization graph* can be exploited for verifying whether a DGP instance satisfies the properties in Defn. 1 and to identify a valid discretization order. The *K-incident graph*, which depends on the discretization order, can instead be at the basis of new theoretical results concerning a priori analysis of the cardinality of DDGP solution sets. We present some initial theoretical results related to these two auxiliary graphs.

After providing a list of main notations in Section 1.1, the rest of the paper is organized as follows. In Section 2, we will introduce the K-discretization graph and study its main properties. In Section 3, we will introduce the K-incident graph, which depends instead on a given discretization order r on V. In Section 4, we will provide a detailed example showing how the K-incident graph can come to help in the study of the solution set of DDGP instances. In Section 5, we will rewrite a previously proposed algorithm for the search of discretization orders [4] in terms of K-discretization and K-incident graphs. Section 6 will conclude the paper with some directions for future research involving these newly introduced auxiliary graphs.

1.1 Main notations

- K, embedding dimension;
- $G = (V, E, d)$, the original DGP instance graph;
- r, vertex order associated to the vertex set V of G;
- v, vertex of G (without information about its rank in r);
- u_i, vertex of G (having rank i in r);
- $N(v)$, the set of all vertices that are adjacent to $v \in V$ in G;
- $\mathcal{G} = (V_{\mathcal{G}}, E_{\mathcal{G}})$, the K-discretization graph;
- $\mathfrak{u}$ (or $\mathfrak{v}$), vertex of $\mathcal{G}$ (without information about its rank in r);
- $\mathcal{N}(\mathfrak{v})$, the set of all vertices that are adjacent to $\mathfrak{v} \in V_{\mathcal{G}}$ in $\mathcal{G}$;
- $\mathcal{I} = (V_{\mathcal{I}}, E_{\mathcal{I}})$, the K-incident graph;
- $\underline{\mathfrak{u}}_i$, the set of adjacent predecessors of u_i in $\mathcal{G}$, for a given r;
- $\bar{\mathfrak{u}}_i$, the set of adjacent successors of u_i in $\mathcal{G}$, for a given r;
- $\hat{\mathfrak{u}}_i$, the set of reference vertices of u_i in $\mathcal{G}$ used for the discretization, for a given r.

Notice that we will use the same letters of the alphabet, but in different styles, for making reference to related vertices in graphs G, $\mathcal{G}$ and $\mathcal{I}$. For example, the set $\mathfrak{u} \in V_{\mathcal{G}}$ is the one corresponding to the vertex $u \in V$. The same style is used for the vertices of $\mathcal{G}$ and $\mathcal{I}$, because $\mathcal{I}$ is isomorphic to a subgraph of $\mathcal{G}$.
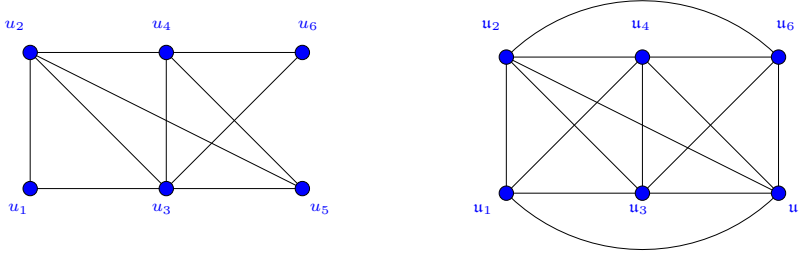
Fig. 1 The original DGP graph G (left) and its 2-discretization graph (right).

2 The K-discretization graph

Let $K > 0$ and $G = (V, E, d)$ be an instance of the DGP with exact distances. Let $N(v)$ be the set containing the adjacent vertices of $v \in V$:

$$N(v) = \{u \in V \mid \{u, v\} \in E\}.$$

Definition 2 A K-*discretization graph* $\mathcal{G} = (V_{\mathcal{G}}, E_{\mathcal{G}})$ of G is a simple undirected graph defined as follows:

– for all $v \in V$, the vertex $\mathfrak{v} \in V_{\mathcal{G}}$ is the set $N(v) \cup \{v\}$;
– $\{\mathfrak{u}, \mathfrak{v}\} \in E_{\mathcal{G}}$ if and only if $|\mathfrak{u} \cap \mathfrak{v}| \geq K$.

By definition, the cardinality of the vertex set $V_{\mathcal{G}}$ of the K-discretization graph corresponds to $|V|$.

Consider for example the following graph, with $n = |V| = 6$ and $K = 2$:

$$E = \{ \{u_1, u_2\}, \{u_1, u_3\}, \{u_2, u_3\}, \{u_2, u_4\}, \{u_2, u_5\}, \{u_3, u_4\}, \{u_3, u_5\},$$
$$\{u_3, u_6\}, \{u_4, u_5\}, \{u_4, u_6\} \}.$$

This graph is shown in Fig. 1, with its 2-discretization graph. Table 1 shows the sets $\mathfrak{u}_i \in V_{\mathcal{G}}$ in correspondence with every vertex u_i of G. By definition, the edges in $\mathcal{G}$ depend upon the cardinality of the set intersections between pairs of vertices: for example, $\{\mathfrak{u}_2, \mathfrak{u}_6\} \in E_{\mathcal{G}}$ because $|\mathfrak{u}_2 \cap \mathfrak{u}_6| = 2$; conversely, $\{\mathfrak{u}_1, \mathfrak{u}_6\} \notin E_{\mathcal{G}}$ because $|\mathfrak{u}_2 \cap \mathfrak{u}_6| = 1$.

Notice that, if the K-discretization graph $\mathcal{G}$ is not connected, then the instance represented by G cannot be discretizable (although each of the connected components might still be discretizable). Moreover, if no subset of $\mathfrak{v} \in V_{\mathcal{G}}$ induces a subgraph containing a K-clique, then the instance represented by G cannot be discretizable either.

Defn. 2 immediately implies Lemma 1.

Lemma 1 *If no K-tuple of vertices of $\mathcal{G}$ induces a K-clique subgraph of G, then the instance represented by G is not discretizable in dimension K.*

u_1	$\mathfrak{u}_1 = \{u_1, u_2, u_3\}$
u_2	$\mathfrak{u}_2 = \{u_1, u_2, u_3, u_4, u_5\}$
u_3	$\mathfrak{u}_3 = \{u_1, u_2, u_3, u_4, u_5, u_6\}$
u_4	$\mathfrak{u}_4 = \{u_2, u_3, u_4, u_5.u_6\}$
u_5	$\mathfrak{u}_5 = \{u_2, u_3, u_4, u_5\}$
u_6	$\mathfrak{u}_6 = \{u_3, u_4, u_6\}$

Table 1 The sets $\mathfrak{u}_i$ associated to each vertex of G.

Proof By contradiction, let us suppose that G is discretizable. In this case, a K-clique C exists in G by Property (a) of Defn. 1. Let us consider the sets $\mathfrak{u} \in V_{\mathcal{G}}$ related to the vertices $u \in C$. Every set $\mathfrak{u}$ contains u and all the other vertices belonging to C, by definition. Therefore the intersection of all these sets $\mathfrak{u}$ needs to contain the K-clique C, against the hypothesis. $\square$

By definition, the K-discretization graph does not depend on the choice of a particular discretization order r. We introduce two new sets that are instead strongly related to the chosen discretization order. We use the subscripts i in u_i and $\mathfrak{u}_i$ to refer to the rank of these vertices in a given discretization order. The definition of a discretization order for the vertex set V induces the definition of an equivalent order for the vertex set $V_{\mathcal{G}}$, and vice versa. The set

$$\underline{\mathfrak{u}}_i = \mathfrak{u}_i \cap \{u_j \in V \mid j \leq i\} \tag{2}$$

contains all the adjacent vertices of $u_i \in V$ that can be used as reference vertices. By contrast, the set

$$\bar{\mathfrak{u}}_i = \mathfrak{u}_i \cap \{u_j \in V \mid j > i\} \tag{3}$$

contains all the adjacent vertices of $u_i \in V$ that do not play the role of reference for u_i in the given choice for the vertex order (the vertices in $\bar{\mathfrak{u}}_i$ can be references for higher-ranked vertices). Since u_i appears in the two sets intersected in eq. (2), the set $\underline{\mathfrak{u}}_i$ always contains the vertex u_i. From a given vertex order r, the two sequences

$$\{\underline{\mathfrak{u}}_i\}_{i \in \{K+1, \ldots, |r|\}}, \quad \{\bar{\mathfrak{u}}_i\}_{i \in \{K+1, \ldots, |r|\}}$$

can be defined. Fig. 2 shows a graphical illustration of the definitions of $\underline{\mathfrak{u}}_i$ and $\bar{\mathfrak{u}}_i$ from $\mathfrak{u}_i$.

In order to apply the BP algorithm to a DDGP instance (see Section 1), it is necessary to identify a particular subset $\hat{\mathfrak{u}}_i$ out of each of the sets $\underline{\mathfrak{u}}_i$, containing the reference vertices. We remark that $\underline{\mathfrak{u}}_i \setminus \hat{\mathfrak{u}}_i$ contains vertices which might, at level i, be incident to edges that are used for pruning, and generally named "pruning edges". Naturally, when the cardinality of $\underline{\mathfrak{u}}_i$ is $K+1$, then all the vertices it contains need to be considered for the discretization process. However, when this cardinality is strictly larger than $K + 1$, then there are many choices for the set $\hat{\mathfrak{u}}_i$. This is likely to have an impact on the performance of the BP algorithm [2].
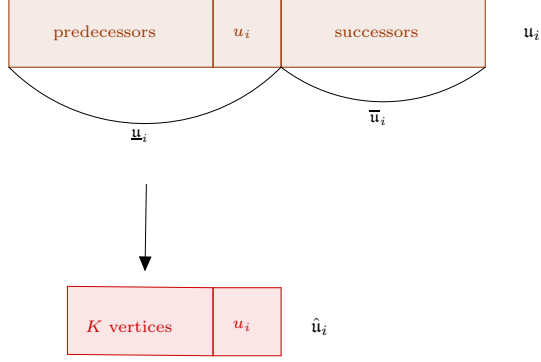
Fig. 2 Definition of $\underline{u}_i$, $\bar{u}_i$ and $\hat{u}_i$.

Definition 3 A sequence $\{\hat{u}_i\}_{i \in \{K+1,\ldots,|r|\}}$ is a *valid list* of reference vertices for G if, for each $i \in \{K+1,\ldots,|r|\}$, we have:

$$\hat{u}_i \subseteq \underline{u}_i, \quad |\hat{u}_i| = K+1, \quad u_i \in \hat{u}_i.$$

In other words, the set $\hat{u}_i$ contains the original vertex u_i of G, as well as the K reference vertices that are supposed to be used for the discretization (see Fig. 2).

3 The K-incident graph

Definition 4 Given a DGP graph G with a vertex order r associated to its vertex set, and given a valid list of reference vertices $\{\hat{u}_i\}_{i \in \{K+1,\ldots,|r|\}}$, the K-*incident graph* $\mathcal{I} = (V_\mathcal{I}, E_\mathcal{I})$ of G is such that

– $V_\mathcal{I} = \{\hat{u}_i \,:\, i \in \{K+1,\ldots,|r|\}\}$;
– $\{\hat{u}_i, \hat{u}_j\} \in E_\mathcal{I}$ if and only if $|\hat{u}_i \cap \hat{u}_j| = K$.

By definition, the K-incident graph is isometric to a subgraph of the K-discretization graph, which can be identified once a valid list of reference vertices is associated to the original graph G. The cardinality of the vertex set $V_\mathcal{I}$ of the K-incident graph is always $|V| - K$.

Lemma 2 *If C is a clique of the K-incident graph $\mathcal{I}$, and $\hat{u}'$, $\hat{u}''$ and $\hat{u}'''$ are three distinct vertices of the clique, then*

$$\hat{u}' \cap \hat{u}'' = \hat{u}' \cap \hat{u}'''.$$

Proof We proceed by induction on the number m of vertices forming the clique. The case $m = 2$ is trivial but it cannot be used as a basis for the induction. We consider therefore the case $m = 3$: C is induced by $\{\hat{u}', \hat{u}'', \hat{u}'''\}$, related to

the vertices u', u'' and u''' of G. We assume WLOG that the superscript order corresponds to the order in r. As a consequence, $u''' \notin \hat{\mathfrak{u}}'$ because u''' appears in the order after u' and it cannot be reference vertex for u'. Similarly, we have that $u''' \notin \hat{\mathfrak{u}}''$. Since, by definition, every vertex $\hat{\mathfrak{u}}$ has cardinality $K + 1$, and every intersection of distinct sets $\hat{\mathfrak{u}}'$, $\hat{\mathfrak{u}}''$ and $\hat{\mathfrak{u}}'''$ has cardinality K (see Defn. 4), we can state that

$$\hat{\mathfrak{u}}' \cap \hat{\mathfrak{u}}''' = \hat{\mathfrak{u}}''' \smallsetminus \{u'''\}, \tag{4}$$

and

$$\hat{\mathfrak{u}}'' \cap \hat{\mathfrak{u}}''' = \hat{\mathfrak{u}}''' \smallsetminus \{u'''\}. \tag{5}$$

Let us consider now the last possible intersection in the clique, which is the one between $\hat{\mathfrak{u}}'$ and $\hat{\mathfrak{u}}''$. Because of eqs. (4) and (5), both $\hat{\mathfrak{u}}'$ and $\hat{\mathfrak{u}}''$ contain the set $\hat{\mathfrak{u}}''' \smallsetminus \{u'''\}$. Since this set has cardinality K, which is also the cardinality of the intersection, we have

$$\hat{\mathfrak{u}}' \cap \hat{\mathfrak{u}}'' = \hat{\mathfrak{u}}''' \smallsetminus \{u'''\},$$

and hence the lemma is proved for $m = 3$.

Let us now suppose that this property is true for m-cliques, with $m \geq 3$. Let $\hat{\mathfrak{u}}'$ and $\hat{\mathfrak{u}}''$ be two distinct vertices of the $(m + 1)$-clique C, appearing in the order r as specified by their superscripts. The two subgraphs C' and C'' obtained by removing either $\hat{\mathfrak{u}}'$ or $\hat{\mathfrak{u}}''$, respectively, from the $(m + 1)$-clique C are cliques formed by m vertices, for which the property is supposed to be true. By the induction hypothesis, the property holds for all triplets in C' and all those in C''; this just leaves the case of triplets containing both $\hat{\mathfrak{u}}'$ and $\hat{\mathfrak{u}}''$. So, let us consider a triplet containing both $\hat{\mathfrak{u}}'$ and $\hat{\mathfrak{u}}''$: let $\hat{\mathfrak{u}}'''$ be the third vertex in the triplet, which belongs to both C' and C''. Since the property is true for C', there exists a triplet in C' containing $\hat{\mathfrak{u}}''$ and $\hat{\mathfrak{u}}'''$ for which all intersections give the same set P having cardinality K. In particular,

$$\hat{\mathfrak{u}}'' \cap \hat{\mathfrak{u}}''' = P.$$

Similarly, by considering the clique C'', we can state that

$$\hat{\mathfrak{u}}' \cap \hat{\mathfrak{u}}''' = P.$$

The property is therefore true also for C, because both $\hat{\mathfrak{u}}'$ and $\hat{\mathfrak{u}}''$ contain P, and their intersection needs to have the same cardinality K:

$$\hat{\mathfrak{u}}' \cap \hat{\mathfrak{u}}'' = P.$$

$$\square$$

Lemma 3 *Every cycle in the K-incident graph $\mathcal{I}$ is contained in a clique of $\mathcal{I}$.*
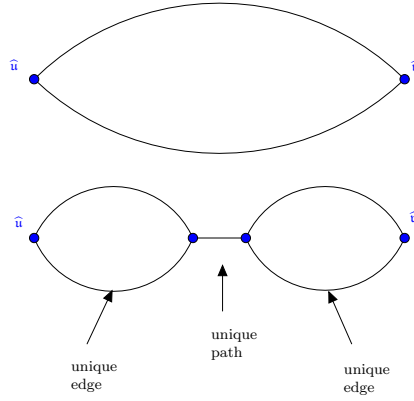
Fig. 3 By contradiction, we suppose that two shortest paths exist. They may intersect or not.

Proof We proceed by induction. For $m = 3$, every 3-cycle is a 3-clique. Let us suppose now that the property is true for size m, and let us consider a cycle $\mathcal{C}$ in the K-incident graph that is incident to $m + 1$ vertices. Let $\hat{\mathfrak{u}}'''$ be the last cycle vertex in the given discretization order, and $\hat{\mathfrak{u}}'$, $\hat{\mathfrak{u}}''$ be the two vertices adjacent to u'' in C. In this situation, both eq. (4) and (5) hold, which implies that

$$\hat{\mathfrak{u}}' \cap \hat{\mathfrak{u}}'' = \hat{\mathfrak{u}}''' \smallsetminus \{u'''\} \text{ and } \{\hat{\mathfrak{u}}', \hat{\mathfrak{u}}''\} \in E_{\mathcal{I}}.$$

For simplicity, let us denote $P = \hat{\mathfrak{u}}''' \smallsetminus \{u'''\}$. We can then identify two cycles on the K-incident graph. One is formed by the subset of vertices $\{\hat{\mathfrak{u}}', \hat{\mathfrak{u}}'', \hat{\mathfrak{u}}'''\}$ with three vertices, and the other is formed by the subset of vertices $V_{\mathcal{C}} \smallsetminus \{\hat{\mathfrak{u}}'''\}$, with m vertices and $\{\hat{\mathfrak{u}}', \hat{\mathfrak{u}}''\}$ as an edge. This last cycle, by hypothesis, is contained in a clique and, by Lemma 2, all the edges of this clique are the set P since $\{\hat{\mathfrak{u}}', \hat{\mathfrak{u}}''\}$ is an edge of the cycle. In other words, all the vertices in the clique has P as subset. Thus, all vertices in this clique are connected to $\hat{\mathfrak{u}}'''$, which means that $\mathcal{C}$ is contained in a clique. $\square$

In the following, we will consider the symbol $P(\hat{\mathfrak{u}}, \hat{\mathfrak{v}})$ to indicate a path between the two vertices $\hat{\mathfrak{u}}$ and $\hat{\mathfrak{v}}$ in the K-incident graph. The subscript in $P_S(\hat{\mathfrak{u}}, \hat{\mathfrak{v}})$ indicates that this is a shortest path, which, by the next proposition, is unique.

Proposition 1 *Every shortest path over the K-incident graph $\mathcal{I}$ is unique.*

Proof By contradiction, let us suppose that two shortest paths between $\hat{\mathfrak{u}}$ and $\hat{\mathfrak{v}}$ exist. If they have only intersection at the two extreme vertices (see top of Fig. 3), then they form a cycle, and, by Lemma 3, a clique containing such a cycle exists in $\mathcal{I}$. Therefore, a path between $\hat{\mathfrak{u}}$ and $\hat{\mathfrak{v}}$ is the unique edge
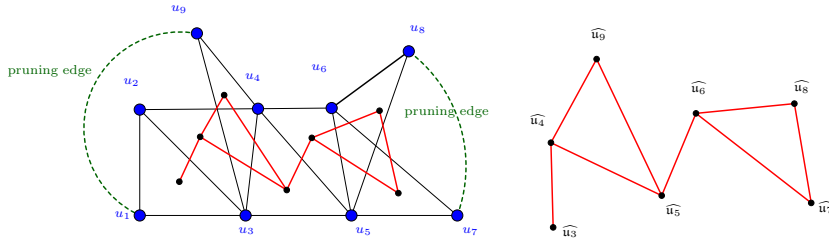
Fig. 4 A graph G representing a DDGP instance (on the left-hand side), with the 2-incident graph (on the right-hand side) corresponding to the vertex order given by the vertex subscripts.

connecting these two vertices in the graph. This path is the unique shortest path: contradiction.

Let us consider now the case where the two shortest paths do have intersections in vertices that do not coincide with neither $\hat{\mathfrak{u}}$ nor $\hat{\mathfrak{v}}$ (see bottom picture in Fig. 3). When the intersection occurs, the sub-path is unique, because it is common to both shortest paths. For the other sub-paths, cycles can be identified, which imply the existence of cliques. As above, therefore, such sub-paths can be replaced with unique and shortest paths formed by only one edge, which is a contradiction. □

4 Potential uses of the K-incident graph

As already mentioned in the Introduction, the K-incident graph can provide very important information about the solution set of DDGP instances. In this section, we will go in details over an example to show this potential use; for lack of space, we will leave the formal study of the content of this section to future works.

Let us consider the DDGP instance in Fig. 4. The instance is composed by 9 vertices and 17 edges, connecting the vertices as shown in Fig 4. We suppose that the realization dimension K is fixed to 2. Moreover, a vertex order is associated to the instance, as indicated by the vertex subscripts. Therefore, from the 2-discretization graph (not shown in Fig 4 for clearness), it is possible to "extract" the 2-incident graph related to this vertex order. Notice that, together with the vertex order, the information about the fact that $\{u_1, u_9\}$ and $\{u_7, u_8\}$ can serve as pruning edges is also considered in the definition of 2-incident graph (in particular, in the definition of the sets $\hat{\mathfrak{u}}_i$, with $i \in \{3, \ldots, 9\}$).

Let us initially consider the sub-instance $G[\{u_1, u_2, u_3, u_4, u_9\}]$. It is evident that it satisfies the DDGP assumptions, and, moreover, this sub-instance has a pruning edge connecting its first and last vertex, in the given order. Following [10], we can state that this sub-instance admits only one realiza-

tion, modulo total translations, rotations and reflections. We can come to the same conclusion while analyzing the sub-instance $G[\{u_5, u_6, u_7, u_8\}]$, because, with the pruning edge $\{u_7, u_8\}$, it forms a 4-clique. Notice however that the "connection bridge" between these two sub-instances, represented by the edge $\{\hat{u}_5, \hat{u}_6\}$ in the K-incident graph, is *flexible*, for the absence of pruning edges. Since this is a DDGP instance, flexibility means that there exist two feasible positions for u_6 w.r.t. its predecessors in $\hat{u}_6$.

The study of the properties of these sub-instances can help discovering, before the explicit computation of the realizations, some properties of the solution set of the original instance. Recall that solutions obtained from others by a total reflection around (a realization of) the initial clique are included in the solution set: pairs of solutions separated by this transformation are said to be *symmetric* [10,12]. Therefore, the realization corresponding to the sub-instance $G[\{u_1, u_2, u_3, u_4, u_9\}]$, together with its symmetric, need to be considered in the construction of the solution set of the original instance. Then, as remarked above, the "connection bridge" $\{\hat{u}_5, \hat{u}_6\}$ is flexible, and therefore 4 total possible solutions can be obtained up to the vertex u_6. Finally, the sub-instance $G[\{u_5, u_6, u_7, u_8\}]$ admits as well one realization, and, by counting its symmetric, we have a total of 8 solutions for the original instance.

The entire discussion above can be deduced, in a much simpler way, by using the K-incident graph associated to the DDGP instance (see Fig. 4). Let us consider the two pruning edges $\{u_1, u_9\}$ and $\{u_7, u_8\}$, and let us compute the shortest paths over the K-incident graph between the corresponding vertices of $\mathcal{I}$ (recall that $u_1 \in \hat{u}_3$). By Prop. 1, these two shortest paths are unique. Moreover, they allow to automatically identify the sub-instances that we have considered above. Instead of studying in details the properties of each sub-instance, we propose to perform the following operation on the K-incident graph: for every shortest path, we replace the vertices it covers with one unique vertex, represented by the set union of the covered vertices (see Fig. 5). Let $V_{\mathcal{I}}^c$ be the vertex set of $\mathcal{I}$ after having performed this transformation: we conjecture that the number of solutions of the original instance is $2^{|V_{\mathcal{I}}^c|-1}$.

5 Extracting the K-incident graph from $\mathcal{G}$

Given a DGP graph G, its K-discretization graph can be immediately constructed. The definition of the K-incident graph, on the other hand, needs a discretization order associated to V. In this section we re-cast in this new abstract setting a greedy algorithm previously proposed in [4] in order to find discretization orders in DDGP instances. We also describe an additional feature designed to select subsets of reference vertices for the BP algorithm (see Section 2).

The basic idea of the greedy algorithm can be summarized in the following two steps. First, an initial K-clique of G is identified and its vertices are placed at the beginning of the new order; then, the rest of the order is constructed by choosing as next one the vertex that maximizes the number of adjacent
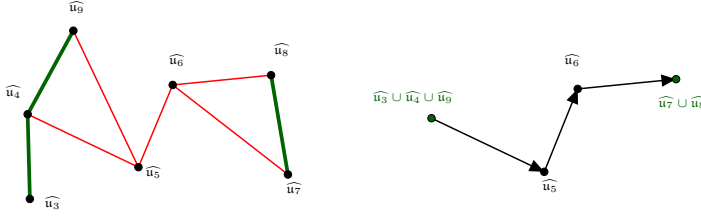
Fig. 5 The 2-incident graph in Fig. 4 where we marked with bold lines the identified shortest paths; on the right-hand side, the same graph after having replaced the vertices covered by the shortest paths with one unique vertex represented by a set union.

vertices already included in the order (for more details, the reader is referred to [4]).

In terms of K-discretization graph, the selection of the initial clique can be performed by selecting K vertices of $\mathcal{G}$ for which the intersection induces a subgraph of G containing a K-clique (which, by Lemma 1, does exist when G is a DDGP graph). All these vertices are placed at the beginning of the order (we can assign to them the ranks from 1 to K, in any internal order).

For constructing the rest of the order, we consider the set

$$\mathcal{N}(\mathfrak{v}) = \{\mathfrak{w} \in V_{\mathcal{G}} \mid \{\mathfrak{w}, \mathfrak{v}\} \in E_{\mathcal{G}}\}$$

of adjacent vertices to a given $\mathfrak{v}$ of $\mathcal{G}$. At each iteration $i = K+1, \ldots, |r|$ of the algorithm, we compute the set $\mathcal{N}(\mathfrak{u}_{i-1})$ and select the $\mathfrak{w} \in \mathcal{N}(\mathfrak{u}_{i-1})$ such that

$$|\mathfrak{w} \cap \mathfrak{u}_{i-1}|$$

is maximized.

The greedy algorithm in [4] can be therefore rewritten in terms of K-discretization graph, as shown in Alg. 1. We point out that, in our notations, the operation of assigning a rank to a vertex $\mathfrak{u}$ is represented by the numerical subscript i associated to it. The theoretical results presented in [4] ensure that, when they exist, the greedy algorithm is able to find discretization orders for G; it provides otherwise a certificate of non-existence. The definition of a vertex order on V is equivalent to defining the two sequences $\{\underline{\mathfrak{u}}_i\}_{i \in \{K+1,\ldots,|r|\}}$ and $\{\bar{\mathfrak{u}}_i\}_{i \in \{K+1,\ldots,|r|\}}$. From the former sequence, moreover, a valid list $\{\hat{\mathfrak{u}}_i\}_{i \in \{K+1,\ldots,|r|\}}$ of reference vertices can be defined, and hence the corresponding K-incident graph.

At iteration i of Alg. 1, if the set $\underline{\mathfrak{u}}_i$ has a cardinality larger than $K+1$, then several subsets $\hat{\mathfrak{u}}_i$ can be defined (see Fig. 6). As remarked above, these subsets can bring to the definition of different K-incident graphs (see Fig. 7). In this work, we simply define every $\hat{\mathfrak{u}}_i$ so that the vertices that are closer in rank to u_i are exploited as reference vertices in the discretization process.
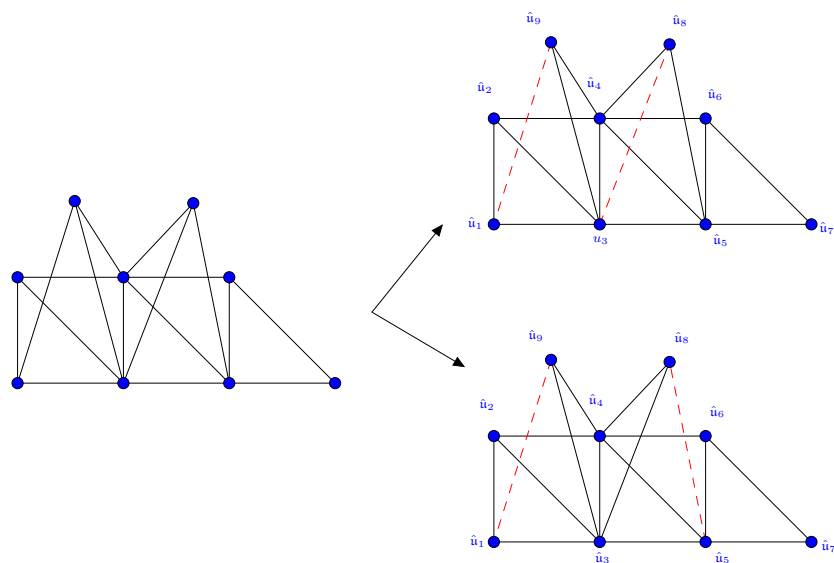
Fig. 6 A 2-discretization graph (left-hand side) with different associated sequences $\{\hat{u}_i\}$ (which imply different sets of pruning edges, in red dashed lines.)
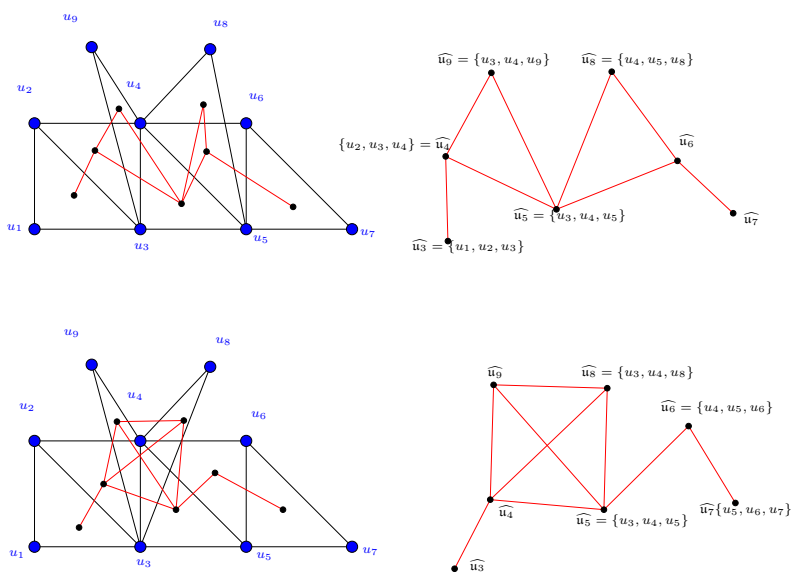

Fig. 7 Two 2-incident graphs related to the two different choices for $\hat{u}_8$ depicted in Fig. 6

Algorithm 1 Extracting a K-incident graph $\mathcal{I}$ from $\mathcal{G}$

1: **input:** graphs G and $\mathcal{G}$
2: **fix** K vertices $\mathfrak{u} \in V_{\mathcal{G}}$ whose intersection gives a K-clique C
3: **assign** the first K ranks to the vertices of C
4: **let** $V_{\mathcal{I}} = \emptyset$
5: **for** $(i = K + 1$ to $|V_{\mathcal{G}}|)$ **do**
6:      *// defining vertex order*
7:      **compute** $\mathcal{N}(\mathfrak{u}_{i-1})$
8:      **choose** $\mathfrak{w} \in \mathcal{N}(\mathfrak{u}_{i-1})$ such that $|\mathfrak{w} \cap \mathfrak{u}_{i-1}|$ is maximized
9:      **let** $\underline{\mathfrak{u}}_i = \mathfrak{w} = \{v_{j(1)}, v_{j(2)}, \ldots, v_{j(K+h)}, u_i\}$;
10:      *// defining reference list*
11:      **if** $(h = 0)$ **then**
12:          **let** $\hat{\mathfrak{u}}_i = \underline{\mathfrak{u}}_i$;
13:      **else**
14:          **let** M be the set of K vertices in $\underline{\mathfrak{u}}_i \setminus \{u_i\}$ such that their ranks j are closer to i;
15:          **let** $\hat{\mathfrak{u}}_i = M \cup \{u_i\}$;
16:      **end if**
17:      **let** $V_{\mathcal{I}} = V_{\mathcal{I}} \cup \{\underline{\mathfrak{u}}_i\}$;
18: **end for**
19: *// defining edge set of K-incident graph*
20: **let** $E_{\mathcal{I}} = \{\{hat\mathfrak{u}_i, \hat{\mathfrak{v}}_j\} \in V_{\mathcal{I}} \times V_{\mathcal{I}} : |\hat{\mathfrak{u}}_i \cap \hat{\mathfrak{v}}_j| = K\}$
21: **output:** graph $\mathcal{I}$

However, future works will be devoted to studying the impact of different reference lists $\{\hat{\mathfrak{u}}_i\}$ on the performances of the BP algorithm, that can be associated to K-incident graphs having different properties.

6 Conclusions

We have introduced the K-discretization graph and the K-incident graph for the DDGP. Future works will be aimed at exploiting these two auxiliary graphs of the DDGP for studying the properties of its solution set, as for example for determining the cardinality of the solution set before the solution of DDGP instances (see our conjecture in Section 4).

References

1. R. Alves, C. Lavor, *Geometric Algebra to Model Uncertainties in the Discretizable Molecular Distance Geometry Problem*, Advances in Applied Clifford Algebras **27**, 439–452, 2017.
2. D. Gonçalves, A. Mucherino, *Optimal Partial Discretization Orders for Discretizable Distance Geometry*, International Transactions in Operational Research **23**, 947–967, 2016.
3. D. Gonçalves, A. Mucherino, C. Lavor, L. Liberti, *Recent Advances on the Interval Distance Geometry Problem*, Journal of Global Optimization **69**, 525–545, 2017.
4. C. Lavor, J. Lee, A. Lee-St.John, L. Liberti, A. Mucherino, M. Sviridenko, *Discretization Orders for Distance Geometry Problems*, Optimization Letters **6**, 783–796, 2012.
5. C. Lavor, L. Liberti, N. Maculan, A. Mucherino, *The Discretizable Molecular Distance Geometry Problem*, Computational Optimization and Applications **52**, 115–146, 2012.

6. 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**, 855–871, 2013.
7. L. Liberti, C. Lavor, N. Maculan, *A Branch-and-Prune Algorithm for the Molecular Distance Geometry Problem*, International Transactions in Operational Research **15**, 1–17, 2008.
8. L. Liberti, C. Lavor, N. Maculan, A. Mucherino, *Euclidean Distance Geometry and Applications*, SIAM Review **56**, 3–69, 2014.
9. L. Liberti, C. Lavor, A. Mucherino, N. Maculan, *Molecular Distance Geometry Methods: from Continuous to Discrete*, International Transactions in Operational Research **18**, 33–51, 2011.
10. L. Liberti, B. Masson, J. Lee, C. Lavor, A. Mucherino, *On the Number of Realizations of Certain Henneberg Graphs arising in Protein Conformation*, Discrete Applied Mathematics **165**, 213–232, 2014.
11. A. Mucherino, C. Lavor, L. Liberti, *The Discretizable Distance Geometry Problem*, Optimization Letters **6**, 1671–1686, 2012.
12. A. Mucherino, C. Lavor, L. Liberti, *Exploiting Symmetry Properties of the Discretizable Molecular Distance Geometry Problem*, Journal of Bioinformatics and Computational Biology **10**(3), 1242009(1–15), 2012.
13. J. Saxe, *Embeddability of Weighted Graphs in k-Space is Strongly NP-hard*, Proceedings of 17th Allerton Conference in Communications, Control and Computing, 480–489, 1979.