Constraint Programming Approaches to the Discretizable Molecular Distance Geometry Problem

Moira MacNeil and Merve Bodur

University of Toronto, Toronto, ON, Canada
m.macneil@mail.utoronto.ca, bodur@mie.utoronto.ca

Abstract. The Distance Geometry Problem (DGP) seeks to find positions for a set of points in geometric space when some distances between pairs of these points are known. The so-called discretization assumptions allow to discretize the search space of DGP instances. In this paper, we study the Discretizable Molecular Distance Geometry Problem whose feasible solutions provide a discretization scheme for the DGP. We propose three constraint programming formulations as well as a set of checks for proving infeasibility, domain reduction techniques, symmetry breaking constraints and valid inequalities. Our computational results indicate that our formulations outperform the state-of-the-art integer programming formulations, both for feasible and infeasible instances.

Keywords: Distance geometry · Discretization vertex order · Constraint programming

1 Introduction

In its essence, the Distance Geometry Problem (DGP) seeks to find positions for a set of points in geometric space when some distances between pairs of these points are known [6]. This has many applications, including in protein molecular geometry, wireless sensor localization, graph drawing, and clock synchronization [6]. The input to the DGP can be represented as a graph, say G, where the vertices are the points we would like to position and weighted edges represent known distances between two points. Here the DGP aims to find a realization function mapping each vertex of G to a coordinate in a K-dimensional space such that the distance between pairs of vertex coordinates is equal to the corresponding edge weights in G. The so-called discretization assumptions reduce the search space of the realization from a continuous space to a finite discrete one, which makes the DGP considerably easier to solve. In this paper, we study a key class of Discretizable DGPs, which ensure the discretization assumptions are met, namely the Discretizable Molecular Distance Geometry Problem (DMDGP).

The rest of the paper is organized as follows. In Section 2, we present in detail the DMDGP and review two existing IP formulations from the literature. In Section 3, we introduce three novel CP models for DMDGP. We then present

a series of enhancements which may aid in the solution of DMDGP, in Section 4. Finally, in Section 5, we present a computational study which compares the CP and IP models, and demonstrates the utility of the enhancements.

2 Preliminaries

Notation. All sets are denoted calligraphically. Let $G = (\mathcal{V}, \mathcal{E})$ be an undirected graph, where $\mathcal{V}$ is the set of vertices and $\mathcal{E}$ is the set of edges. Denote the neighbourhood of a vertex v as $\mathcal{N}(v)$, where $v \notin \mathcal{N}(v)$ and the degree of v as $d(v) = |\mathcal{N}(v)|$. We let $G[\mathcal{V}'] = (\mathcal{V}', \mathcal{E}')$ be the induced subgraph of $\mathcal{V}' \subseteq \mathcal{V}$, and $\mathcal{E}' = \{\{u, v\} : u, v \in \mathcal{V}', \{u, v\} \in \mathcal{E}\}$. A clique, $\mathcal{K}$, in G is a set of vertices $\{v_1, \ldots, v_{|\mathcal{K}|}\} \subseteq \mathcal{V}$ such that $\{v_i, v_j\} \in \mathcal{E}\ \forall\ v_i, v_j \in \mathcal{K}$ such that $v_i \neq v_j$. Similarly, a stable set, $\mathcal{SS}$, is a set of vertices $\{u_1, \ldots, u_{|\mathcal{SS}|}\} \subseteq \mathcal{V}$ such that $\{u_i, u_j\} \notin \mathcal{E}\ \forall\ u_i, u_j \in \mathcal{SS}$ with $u_i \neq u_j$.

Indices follow these conventions: indices start at 0, so that the possible positions of an order are $[|\mathcal{V}|]$. We let $|\mathcal{V}| = n$, and use $|\mathcal{V}|$ in relation to vertices and n in relation to ranks. For indices we use the following notation: let $a, b \in \mathbb{Z}_+$, $a \leq b$, $[a] = \{0, 1, \ldots, a-1\}$ and $[a, b] = \{a, a+1, \ldots, b\}$.

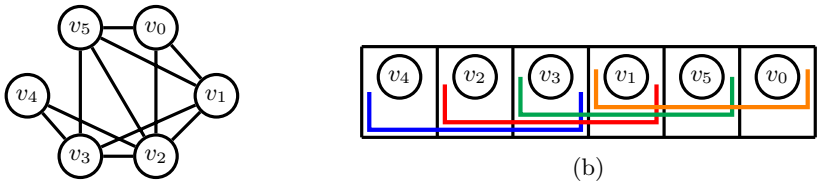
2.1 Problem Definition

The *Discretizable Molecular Distance Geometry Problem* (DMDGP) [1] is the search for a total order of the vertices of a simple, connected, undirected graph $G = (\mathcal{V}, \mathcal{E})$, given an integer dimension K, that satisfies the following:

(i) the first K vertices in the order form a clique in the input graph G, and
(ii) for all vertices with rank $\geq K$, each is adjacent to at least the K vertices that immediately precede it in the order, that is together with its K *adjacent immediate predecessors* it forms a $(K+1)$-clique in the input graph.

Given an order $(v_0, v_1, \ldots, v_{n-1})$, for a vertex v_i, we define its K *immediate predecessors* in the order as $\{v_{i-K}, v_{i-K+1}, \ldots, v_{i-1}\}$. If an immediate predecessor to v_i is also adjacent to v_i in G we call it an *adjacent immediate predecessor*. We refer to a total order that satisfies (i) and (ii) as a *DMDGP order*, and the clique satisfying (i) as the *initial clique*. We say an instance for which a DMDGP order exists is *feasible*, otherwise it is *infeasible*. The problem of determining whether a DMDGP order exists for G is known as the Contiguous Trilateration Ordering Problem (CTOP) [1]. An instance of CTOP, i.e. an integer $K > 0$ and a simple, undirected, connected graph $G = (\mathcal{V}, \mathcal{E})$, will be denoted $(G = (\mathcal{V}, \mathcal{E}), K)$ or simply (G, K). Cassioli et al. [1] proved CTOP is NP-complete.

Example 1 *The graph given in Figure 1a with $K = 2$ is a feasible instance for DMDGP. A possible DMDGP order is $(v_4, v_2, v_3, v_1, v_5, v_0)$. Clearly, since they are adjacent $\{v_4, v_2\}$ form a clique, v_3 is adjacent to both of its immediate predecessors: v_4, and v_2, so $\{v_4, v_2, v_3\}$ form a $(K+1)$-clique in the input graph. Similarly, v_1 is adjacent to v_2, and v_3, forming a $(K+1)$-clique in the input graph and so on.*



(a) (b)

Fig. 1: (a) A graph instance which is feasible for DMDGP with $K = 2$. (b) Overlapping cliques of the order$(v_4, v_2, v_3, v_1, v_5, v_0)$.

By definition of DMDGP and as seen in Example 1, minimally a DMDGP order is a series of $(K+1)$-cliques which overlap by at least K vertices [1], see Figure 1b.

2.2 Existing Integer Programming Models

Prior to this work, Cassioli et al. [1] present two integer programming (IP) formulations for CTOP:

- The vertex-rank formulation ($\mathbb{IP}^{\mathbb{VR}}$): They introduce $|\mathcal{V}| \times n$ binary variables indicating vertex-rank assignment. Then, the model contains $(|\mathcal{V}|+n)$-many 1-1 assignment constraints and $(|\mathcal{V}| \times n)$-many clique constraints.
- The clique digraph formulation ($\mathbb{IP}^{\mathbb{CD}}$): They enumerate all ordered cliques of size $(K+1)$ in G, define a clique digraph D with vertices as those ordered cliques and arcs for pairs of cliques that suitably overlap to follow each other in the order (as in Figure 1b). Then, the DMDPG solution corresponds to a path in D. This IP model has digraph arc variables, first clique and last clique variables, and precedence variables for vertices in G.

3 Constraint Programming Models

Constraint Programming (CP) is a natural approach to distance geometry ordering problems since we wish only to find a feasible solution and not prove optimality. CP has been shown to work well for problems with a permutation structure [4] and allows the leveraging of global constraints such as AllDifferent. To our knowledge, no CP model for CTOP has ever been proposed. The flexibility of CP allows for three possible formulations for CTOP.

The first formulation follows naturally from the ($\mathbb{IP}^{\mathbb{VR}}$) formulation. We define integer variables r_v equal to the rank of vertex $v \in \mathcal{V}$.

$$(\mathbb{CP}^{\mathbb{RANK}}) : \text{AllDifferent}(r_0, r_1, ..., r_{|\mathcal{V}|-1}) \tag{1a}$$
$$|r_u - r_v| \geq K+1 \qquad \forall\, u, v \in \mathcal{V} \text{ s.t. } u \neq v \text{ and } \{u, v\} \notin \mathcal{E} \tag{1b}$$
$$r_v \in [n-1] \qquad \forall\, v \in \mathcal{V} \tag{1c}$$

Using the global constraint AllDifferent [5] (1a), we enforce that each vertex has a unique rank. Together with the domain constraints (1c), we obtain a one-to-one vertex-rank assignment, since each rank has a possible domain of $[n-1]$ and we are enforcing the constraint over all the rank variables which are indexed by the vertices, i.e., $|\mathcal{V}| = n$ variables. To enforce clique constraints (1b), we use the idea that if two vertices do not have an edge between them, they cannot be in the same $(K+1)$-clique. In other words, their ranks must have a difference of at least $K+1$. This constraint completely models the clique constraints and the predecessor constraints since if their rank difference is $\leq K$ then vertices u, and v must be in the same clique which contradicts there being no edge between them.

Secondly, we present what is called a dual formulation in CP [4], here the values and variable meanings are swapped. We define integer variables v_r equal to the vertex in position r of the order, and note that the adjaceny matrix of G is denoted by A, i.e., $A_{v,u} = 1$ if and only if edge $\{u, v\} \in \mathcal{E}$.

$$
\begin{aligned}
(\mathbb{CP}^{\text{VERTEX}}): \ & \text{AllDifferent}(v_0, v_1, .., v_{n-1}) && && (2a)\\
& A_{v_i,v_j} = 1 && \forall\ i \in [0, K-2], j \in [i+1, K-1] && (2b)\\
& A_{v_i,v_j} = 1 && \forall\ i \in [K, n-1], j \in [i-K, i-1] && (2c)\\
& v_r \in [|V|-1] && \forall\ r \in [n-1] && (2d)
\end{aligned}
$$

In (2a) we enforce that each rank has a unique vertex, again using AllDifferent [5]. To enforce the clique and predecessor constraints, we use the CP notion of *element constraints* [3] we enforce that there is an initial clique in Constraints (2b) and that all subsequent vertices have at least K adjacent immediate predecessors in (2c) by ensuring edges exist between the appropriate vertices. Finally, (2d) enforces the domain of the variables.

The last CP model is the result of combining the rank and vertex models into a single model by *channelling* the variables using an *inverse constraint*. It uses the constraints for predecessors and cliques from both formulations. This is useful because redundant constraints may actually help CP solvers perform more inference and discover feasible solutions in a shorter amount of time. Having defined v and r variables as before, the combined model is as follows:

$$
\begin{aligned}
(\mathbb{CP}^{\text{COMBINED}}): \ & |r_u - r_v| \geq K+1 && \forall\ u, v \in \mathcal{V} \text{ s.t. } u \neq v \text{ and } \{u, v\} \notin \mathcal{E} && (3a)\\
& A_{v_i,v_j} = 1 && \forall\ i \in [0, K-2],\ j \in [i+1, K-1] && (3b)\\
& A_{v_i,v_j} = 1 && \forall\ i \in [K, n-1], j \in [i-K, i-1] && (3c)\\
& \text{inverse}(r, v) && && (3d)\\
& r_v \in [n-1] && \forall\ v \in \mathcal{V} && (3e)\\
& v_r \in [|V|-1] && \forall\ r \in [n-1] && (3f)
\end{aligned}
$$

In this formulation, the inverse constraint (3d) enforces the relation $(r_u = j) \equiv (v_j = u)$, which also makes the AllDifferent constraints in the vertex and rank models redundant. The AllDifferent constraints may be included as redundant constraints in the model, however initial computational results showed they were detrimental thus are omitted hereafter.

4 Enhancements

In this section we present a series of enhancements based on the structure of DMDGP orders that can be used to improve the formulations presented in Section 3. (For brevity, we omit the proofs and illustrative examples, which can be found in our full paper [7].) We begin by discussing checks for infeasible instances, followed by procedures for reducing the domains and breaking symmetries in DMDGP orders. Finally, we present a class of valid inequalities.

4.1 Infeasibility Checks

We begin the discussion of enhancements to DMDGP formulations by introducing some simple checks which will immediately indicate if an instance (G, K) is infeasible. The first check arises from the fact that every vertex needs at least K neighbours to be a part of a $(K+1)$-clique.

Infeasibility Check 1 (Minimum Degree) *Given (G, K), if* $\exists\ v \in \mathcal{V}$ *such that* $d(v) < K$ *then* G *does not have a DMDGP order for* K*.*

Similarly, it is possible to determine a lower bound on the number of edges in G.

Infeasibility Check 2 (Minimum Edges) *Given (G, K), if* $|\mathcal{E}| < \left(|\mathcal{V}| - \frac{1}{2}\right) K - \frac{1}{2}K^2$ *then this instance is infeasible.*

Next checks establish an upper and lower bound on the number of vertices with small and large degree, respectively, where we call vertices v with $d(v) < 2K$ *small degree vertices* and vertices v with $d(v) \geq 2K$ *large degree vertices*. We introduce the set $\mathcal{V}^{d[K,K+\delta]} = \{v \in \mathcal{V} |\ d(v) \in [K, K+\delta]\}$ for a given $\delta \in \mathbb{Z}_+$.

Infeasibility Check 3 (Upper Bound on Small Degree Vertices) *Given an instance (G, K), if* $\exists\ \delta \in [K-1]$ *such that* $\left|\mathcal{V}^{d[K,K+\delta]}\right| > 2(\delta+1)+1$ *then this instance is infeasible.*

Infeasibility Check 4 (Lower Bound on Large Degree Vertices) *Given (G, K), with* $n \geq (2K+1)$*, if* $\left|\mathcal{V}^{d[2K,n-1]}\right| \leq n - (2K+1)$ *then the instance is infeasible.*

4.2 Domain Reduction

We are able to exploit some structural characteristics of CTOP to help prune variable domains in the CP formulations. Let the domain of an integer variable x be given by D_x.

First, we extend the lower bounds on the degree of a vertex given by Cassioli et al. [1] to set the domains for rank variables. As observed previously, a DMDGP order is a series of overlapping cliques of size (at least) $K+1$. In the minimal case, the first and last vertices in the order are in exactly one clique, the second

and second to last vertices are in two cliques, and so on. The central $(|\mathcal{V}| - 2K)$ vertices are in at least $2K$ cliques. From this we can infer the minimum number of neighbours required by a vertex at a given rank.

Domain Reduction Rule 1 (Small Degree Vertices) *Given an instance (G, K), we can define the domain for the rank variables as follows:*

$$D_{r_v} = \begin{cases} [d(v) - K] \cup [n - 1 - (d(v) - K), n - 1] & \text{if } d(v) < 2K \\ [n - 1] & \text{otherwise} \end{cases}$$

We are able to extend domain reduction to the vertices that are adjacent to small degree vertices. The intuition is that if a vertex has small degree, the position of its neighbours cannot be too far from that vertex. If the position of a small degree vertex v^* has already been limited, its neighbours must be within the first or the last $d(v^*)$ vertices of the order since they are all connected to v^*.

Domain Reduction Rule 2 (Neighbourhood of Small Degree Vertices) *Given an instance (G, K), with $n \geq (2K + 1)$, for all $v^* \in \mathcal{V}^{d[K,2K-1]}$*

$$D_{r_v} = [d(v^*)] \cup [n - 1 - d(v^*), n - 1] \quad \forall\ v \in \mathcal{N}(v^*).$$

4.3 Symmetry Breaking

As observed in [1], reversing a DMDGP order also gives a DMDGP order. We establish that these are not the only symmetries present in DMDGP orders, and present strategies for breaking these symmetries. We begin by a simple condition to break the reverse symmetry. First, notice that if there is a single vertex that has degree K without loss of generality we can fix its position to 0, if there is a second vertex with degree K we can fix its position to $n - 1$, noting that there are at most two vertices of degree K in a DMDGP order due to Infeasibility Check 3.

Symmetry Breaking Condition 1 (Degree K) *If $\mathcal{V}^{d[K,K]} = \{v_i\}$, then let $r_{v_i} = 0$. If $\mathcal{V}^{d[K,K]} = \{v_i, v_j\}$, then let $r_{v_i} = 0$ and $r_{v_j} = n - 1$.*

Next, we observe that if two vertices have the same neighbourhood excluding each other, they are interchangeable in the DMDGP order since they will have exactly the same adjacent immediate predecessors. This guarantees a DMDGP order, since if we interchange two vertices to ensure the order is preserved we need only ensure those vertices have the appropriate adjacent immediate predecessors. We call this symmetry *pairwise symmetry*, which can be broken by imposing an arbitrary order on the pair of such symmetric vertices. Ideally, we would identify a large set of such vertices and order them. However, identifying such vertex sets can be computationally expensive. We instead identify two types of vertex sets that will allow for easy detection and breaking of pairwise symmetry. Specifically, we consider stable sets and cliques in the input graph.

Symmetry Breaking Condition 2 (Stable Set) *For a stable set* $\mathcal{SS} = \{v_1, v_2, \ldots, v_k\} \subseteq \mathcal{V}$ *such that* $\mathcal{N}(v_i) = \mathcal{N}(v_j)\ \forall\ v_i, v_j \in \mathcal{SS}$ *we enforce that* $r_{v_1} < r_{v_2} < \cdots < r_{v_k}$.

Symmetry Breaking Condition 3 (Clique) *For a clique* $\mathcal{K} = \{v_1, v_2, \ldots, v_k\} \subseteq \mathcal{V}$ *such that* $\mathcal{N}(v_i) \setminus \mathcal{K} = \mathcal{N}(v_j) \setminus \mathcal{K},\ \forall\ v_i, v_j \in \mathcal{K}$ *we enforce that* $r_{v_1} < r_{v_2} < \cdots < r_{v_k}$.

In our experiments, we examine only cliques of size three or less, since we are usually unable to find large cliques satisfying Condition 3. Furthermore, we are able to conditionally extend these symmetry breaking conditions to include more vertices. Consider, for example, two vertices v and u whose neighbourhoods differ only by one vertex $w \in \mathcal{N}(v)$. If in the DMDGP order w is at least $K+1$ away from v, the edge connecting them is not necessary to enforce precedence in the order, that is, w is not an adjacent immediate predecessor of v and vice versa. In this case we can essentially consider u and v as having the same neighbourhood and so can impose symmetry breaking on them. For some set $\mathcal{S} \subseteq \mathcal{V}$ we denote $\mathcal{N}(\mathcal{S}) = \cup_{v \in \mathcal{S}} \mathcal{N}(v) \setminus \mathcal{S}$, the set of all vertices, outside of $\mathcal{S}$ that are adjacent to a vertex in $\mathcal{S}$.

Symmetry Breaking Condition 4 (Extended Stable Set) *Let* $\mathcal{SS}$ *be a stable set meeting Condition 2 or a single vertex not in any stable set meeting Condition 2. For a vertex* $v \in \mathcal{V} \setminus (\mathcal{SS} \cup \mathcal{N}(\mathcal{SS}))$ *such that* $\mathcal{N}(v) \setminus \mathcal{N}(\mathcal{SS}) = \{w\}$ *we enforce the logical constraints:*

$$|r_v - r_w| \geq K + 1 \implies r_v < r_u\ \forall\ u \in \mathcal{SS}.$$

If we have already enforced an ordering for $\mathcal{SS}$ *already, we need only add the constraint*

$$|r_v - r_w| \geq K + 1 \implies r_v < r_{v_1}.$$

Symmetry Breaking Condition 5 (Extended Clique) *Let* $\mathcal{K}$ *be a clique meeting Condition 3 or a single vertex not in any clique meeting Condition 3. For a vertex* $v \in \mathcal{N}(\mathcal{K})$ *such that* $(\mathcal{N}(v) \cup \{v\}) \setminus (\mathcal{N}(\mathcal{K}) \cup \mathcal{K}) = \{w\}$ *we enforce the logical constraints*

$$|r_v - r_w| \geq K + 1 \implies r_v < r_u\ \forall\ u \in \mathcal{K}.$$

Finally, if we have not been able to break any symmetry via any of the previous ways we can arbitrarily choose two vertices and impose an order on them.

Symmetry Breaking Condition 6 (Arbitrary) *For any* $v_1, v_2 \in \mathcal{V}$ *enforce that* $r_{v_1} < r_{v_2}$.

4.4 A Class of Valid Inequalities

Next, we develop some valid inequalities that can allow our models to perform better in the case of infeasible instances. We proceed with the following intuition: if we identify some subset $\mathcal{S} \subseteq \mathcal{V}$ such that the induced subgraph of $\mathcal{S}$ does not have a DMDGP order, the entire set $\mathcal{S}$ cannot appear consecutively in the order.

If for a given instance, (G, K), we are able to identify subsets $\mathcal{S} \subseteq \mathcal{V}$ whose induced graphs, $G[\mathcal{S}]$, do not have DMDGP orders for K, we can add cuts to enforce that the difference between the maximum rank and the minimum rank of any element in $\mathcal{S}$ is at least $|\mathcal{S}|$. Let r_{max}, r_{min} denote the maximum rank and the minimum rank of any vertex in $\mathcal{S}$, respectively. The valid inequality is:

$$r_{max} - r_{min} \geq |\mathcal{S}| \tag{4}$$

We can improve this cut by examining the vertex in $\mathcal{S}$ with the smallest degree in the induced subgraph. Let $\delta_{\mathcal{S}}^{miss}(v) = |\{u \in \mathcal{S} \setminus \{v\} \mid (v,u) \notin \mathcal{E}[\mathcal{S}]\}|$, i.e., the number of edges with one endpoint at $v \in \mathcal{S}$ missing from $G[\mathcal{S}]$ and let $\delta_{\mathcal{S}}^{miss} = \max_{v \in \mathcal{S}} \delta_{\mathcal{S}}^{miss}(v)$. If $K > |\mathcal{S}| - \delta_{\mathcal{S}}^{miss}$, the difference between the maximum rank and the minimum rank must be greater than $\delta_{\mathcal{S}}^{miss} + K$, because the $v \in \mathcal{S}$ which has $\delta_{\mathcal{S}}^{miss}$, cannot be in a clique with $\delta_{\mathcal{S}}^{miss}$ of the vertices in $\mathcal{S}$, so we need at minimum $\delta_{\mathcal{S}}^{miss}$ extra vertices between the vertices of $\mathcal{S}$ in the order. Otherwise, if $K \leq |\mathcal{S}| - \delta_{\mathcal{S}}^{miss}$, the difference in ranks must be greater than $|\mathcal{S}|$ which is the inequality (4). So, the valid inequality is

$$r_{max} - r_{min} \geq \max\{|\mathcal{S}|, \delta_{\mathcal{S}}^{miss} + K\}. \tag{5}$$

The task of finding subsets of vertices $\mathcal{S}$ so that the subgraph induced by $\mathcal{S}$ does not have a DMDGP order is as difficult as determining if the whole graph has a DMDGP order. Thus, we would like to find sets of vertices with the most edges missing in their induced subgraph. As the sets with the most missing edges are stable sets, we can consider stable sets in G as candidate $\mathcal{S}$ sets. For any stable set $\mathcal{SS}$, no pair of vertices can appear in the same $(K+1)$-clique. Thus, each pair of vertices in $\mathcal{SS}$ needs to have a difference in their ranks of at least $K+1$, meaning the minimum rank and maximum rank must have a difference of $(|\mathcal{SS}| - 1)(K+1)$. The inequality becomes

$$r_{max} - r_{min} \geq (|\mathcal{SS}| - 1)(K+1). \tag{6}$$

This observation also yields a simple check for infeasibility.

Infeasibility Check 5 *Given (G, K), if the size of the maximum stable set in G is greater than $\frac{n}{K+1} + 1$, we can immediately say G does not have a DMDGP order with K.*

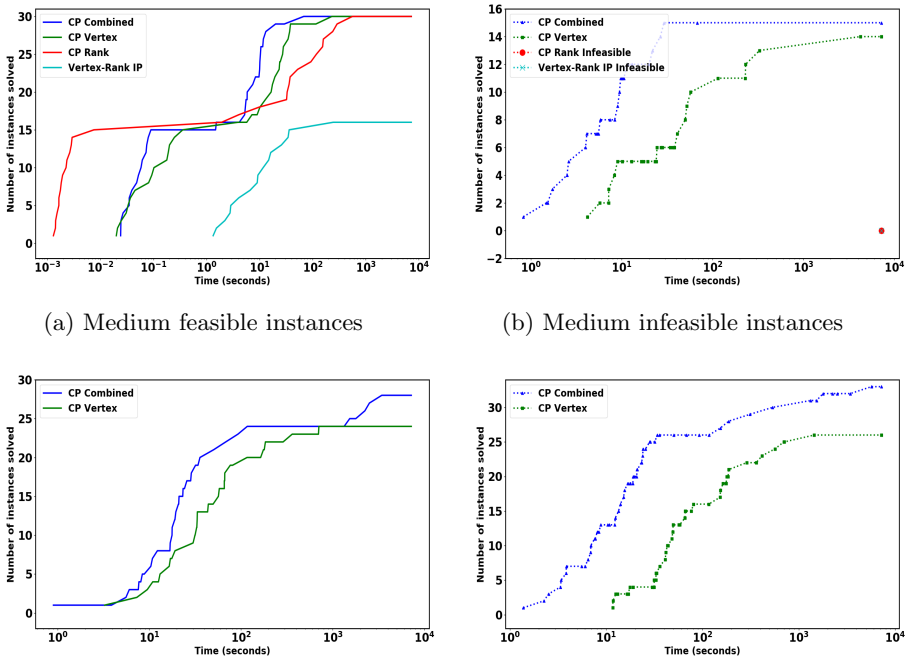
Finally, we define these valid inequalities so that they may be added to the ($\mathbb{CP}^{\mathbb{RANK}}$) and ($\mathbb{CP}^{\mathbb{COMBINED}}$) formulations. Given a stable set $\mathcal{SS} \subseteq \mathcal{V}$, and the rank variables r_v we have

$$\max\{r_v | v \in \mathcal{SS}\} - \min\{r_v | v \in \mathcal{SS}\} \geq (|\mathcal{SS}| - 1)(K+1).$$

5 Computational Results

Instances. We perform our numerical experiments on a test data set consisting of randomly generated graphs with $n \in \{40, 45, \ldots, 100\}$ and the expected edge density (measured as $\frac{2|\mathcal{E}|}{n(n-1)}$) in $\{0.3, 0.5, 0.7\}$. The graphs are generated using the dense_gnm_random_graph() function in the NetworkX package [2]. We refer to the instances with $n \leq 60$ as the *medium instances*, and the rest as the *large instances*.

Experimental Setup. The CPs and IPs are respectively solved using IBM ILOG CP Optimizer and IBM ILOG CPLEX version 12.8.0. All models are implemented in C++ and run on MacOs with 16GB RAM and a 2.3 GHz Intel Core i5 processor, using a single thread. As the most frequently used value in applications, $K = 3$ is used for all experiments. The time limit is set to 7200 seconds.



(a) Medium feasible instances

(b) Medium infeasible instances

(c) Large feasible instances

(d) Large infeasible instances

Fig. 2: Solution times of the models.

Results. We compare our CP formulations with the IP formulations of [1], namely ($\mathbb{IP}^{\mathbb{VR}}$) and ($\mathbb{IP}^{\mathbb{CD}}$)[1], that are briefly described in Section 2.2. We compare

[1] Here, we only provide the results for the former.

all formulations for medium instances and for large instances consider only the best performing formulations as the others reached the time limit on almost all cases. In Figure 2, we show the cumulative number of solved instances with respect to time. Note that solution times are given in a logarithmic scale. We also note that the enhancements are not included in these results and that the complete numerical experiments can be found in [7].

We observe that CP formulations all outperform the IP, except for the medium infeasible case where ($\mathbb{IP}^{\mathbb{VR}}$) and ($\mathbb{CP}^{\mathbb{RANK}}$) solve none of the instances. Overall, ($\mathbb{CP}^{\mathbb{COMBINED}}$) performs the best, solving all the instances in the given time limit, while ($\mathbb{CP}^{\mathbb{VERTEX}}$) is the second-best model. Based on ($\mathbb{CP}^{\mathbb{COMBINED}}$) results, we also observe that low density instances take significantly more time to solve.

6 Conclusion

We propose the first CP formulations for the DMDGP and compare them against two existing IP formulations in the literature. We also introduce three classes of enhancements to help solve the DMDGP; namely infeasibility checks, domain reduction, and symmetry breaking. Our computational results show our models outperform the state-of-the-art IP formulations.We also provide the first class of valid inequalities for DMDGP, whose efficient implementation remains as future work.

References

1. Cassioli, A., Günlük, O., Lavor, C., Liberti, L.: Discretization vertex orders in distance geometry. Discrete Applied Mathematics **197**, 27–41 (2015)
2. Hagberg, A.A., Schult, D.A., Swart, P.J.: Exploring network structure, dynamics, and function using networkx. In: Proceedings of the 7th Python in Science Conference (SciPy2008). pp. 11–15 (2008)
3. Hentenryck, P.V., Carillon, J.P.: Generality versus specificity: An experience with ai and or techniques. In: AAAI (1988)
4. Hnich, B., Smith, B.M., Walsh, T.: Dual modelling of permutation and injection problems. Journal of Artificial Intelligence Research **21**(1), 357–391 (2004)
5. Laurière, J.L.: A language and a program for stating and solving combinatorial problems. Artificial Intelligence **10**(1), 29 – 127 (1978)
6. Lavor, C., Liberti, L., Lodwick, W.A., da Costa, T.M.: An Introduction to Distance Geometry applied to Molecular Geometry. Springer (2017)
7. MacNeil, M., Bodur, M.: Constraint programming approaches to the discretizable molecular distance geometry problem (2019), https://arxiv.org/abs/1908.00048