On The Complexity of Distance-d Independent Set Reconfiguration

HOANG, Duc Anh

On The Complexity of Distance-d Independent Set Reconfiguration

Duc A. Hoang[1]

[1]Graduate School of Informatics, Kyoto University
Kyoto, Japan
hoang.duc.8r@kyoto-u.ac.jp

Abstract

For a fixed positive integer $d \geq 2$, a *distance-d independent set (DdIS)* of a graph is a vertex subset whose distance between any two members is at least d. Imagine that there is a token placed on each member of a DdIS. Two DdISs are adjacent under Token Sliding (TS) if one can be obtained from the other by moving a token from one vertex to one of its unoccupied adjacent vertices. Under Token Jumping (TJ), the target vertex needs not to be adjacent to the original one. The Distance-d Independent Set Reconfiguration (DdISR) problem under TS/TJ asks if there is a corresponding sequence of adjacent DdISs that transforms one given DdIS into another. The problem for $d = 2$, also known as the Independent Set Reconfiguration problem, has been well-studied in the literature and its computational complexity on several graph classes has been known. In this paper, we study the computational complexity of DdISR on different graphs under TS and TJ for any fixed $d \geq 3$. On chordal graphs, we show that DdISR under TJ is in P when d is even and PSPACE-complete when d is odd. On split graphs, there is an interesting complexity dichotomy: DdISR is PSPACE-complete for $d = 2$ but in P for $d = 3$ under TS, while under TJ it is in P for $d = 2$ but PSPACE-complete for $d = 3$. Additionally, certain well-known hardness results for $d = 2$ on general graphs, perfect graphs, planar graphs of maximum degree three and bounded bandwidth can be extended for $d \geq 3$.

Keywords and phrases: reconfiguration problem, distance-d independent set, computational complexity, token sliding, token jumping

Mathematics Subject Classification (2020): 05C85, 68R10

1. Introduction

Recently, *reconfiguration problems* have attracted the attention from both theoretical and practical viewpoints. The input of a reconfiguration problem consists of two *feasible solutions* of some *source problem* (e.g., Satisfiability, Independent Set, Vertex Cover, Dominating Set, etc.) and a *reconfiguration rule* that describes an adjacency relation between solutions. One of the primary goal is to decide whether one feasible solution can be transformed into the other via a sequence of adjacent feasible solutions where each intermediate member is obtained from its predecessor by applying the given reconfiguration rule exactly once. Such a sequence, if exists, is called a *reconfiguration sequence*. In general, the (huge) set of *all* feasible solutions is *not* given, and one can verify in polynomial time (with respect to the input's size) the feasibility and adjacency of solutions. Another way of looking at reconfiguration problems involves the so-called *reconfiguration graphs*—a graph whose vertices/nodes are feasible solutions and their adjacency relation is defined via the given reconfiguration rule. Naturally, a reconfiguration sequence is nothing but a walk in the corresponding reconfiguration graph. Readers may recall the classic Rubik's cube puzzle as an example of a reconfiguration problem, where each configuration of the Rubik's cube corresponds to a feasible solution, and two configurations (solutions) are adjacent if one can be obtained from the other by rotating a face of the cube by either 90, 180, or 270 degrees. The question is whether one can transform an arbitrary configuration to the one where each face of the cube has only one color. For an overview of this research area, we refer readers to the surveys [1, 8, 11, 25].

Reconfiguration problems involving *vertex subsets* (e.g., clique, independent set, vertex cover, dominating set, etc.) of a graph have been extensively considered in the literature. In such problems, to make it more convenient for describing reconfiguration rules, one usually view a vertex subset of a graph as a set of tokens placed on its vertices. Some of the well-known reconfiguration rules in this setting are:

- Token Sliding (TS): a token can only move to one of the unoccupied adjacent vertices;
- Token Jumping (TJ): a token can move to any unoccupied vertex; and
- Token Addition/Removal (TAR(k)): a token can either be added to an unoccupied vertex or remove from an occupied one, such that the number of tokens is always (upper/lower) bounded by some given positive integer k.

Let G be a simple, undirected graph. An *independent set* of a graph G is a set of pairwise non-adjacent vertices. The Independent Set (MaxIS) problem, which asks if G has an independent set of size at least some given positive integer k, is one of the fundamental NP-complete problems in the computational complexity theory [39]. Given an integer $d \geq 2$, a *distance-d independent set* (also known as *d-scattered set* or sometimes *d-independent set*[1]) of G is a set of vertices whose pairwise distance is at least d. This "distance-d independent set" concept generalizes "independent set": an independent set is also a distance-2 independent set but may not be a distance-d independent set for $d \geq 3$. Given a fixed integer $d \geq 2$, the Distance-d Independent Set (MaxDdIS) problem asks if there is a distance-d independent set of G whose size is at least some given positive integer k. Clearly, MaxD2IS is nothing but MaxIS. It is known that MaxDdIS is NP-complete for every fixed $d \geq 3$ on general graphs [37] and even on regular bipartite graphs when $d \in \{3, 4, 5\}$ [34]. Eto et al. [23] proved that MaxDdIS is NP-complete for every fixed $d \geq 3$ even for bipartite graphs and for planar bipartite graphs of maximum degree three. They also proved that on chordal graphs, MaxDdIS is in P for any even $d \geq 2$ and remains NP-complete for any odd $d \geq 3$. The complexity of MaxDdIS on several other graphs has also been studied [10, 17, 31]. Additionally, MaxDdIS and its variants have been studied extensively from different viewpoints, including exact exponential algorithms [9], approximability [6, 10, 23], and parameterized complexity [5].

In this paper, we take MaxDdIS as the source problem and initiate the study of Distance-d Independent Set Reconfiguration (DdISR) from the computational complexity viewpoint. The problem for $d = 2$, which is usually known as the Independent Set Reconfiguration (ISR) problem, has been well-studied from both classic and parameterized complexity viewpoints. ISR under TS was first introduced by Hearn and Demaine [30]. Ito et al. [27] introduced and studied the problem under TAR along with several other reconfiguration problems. Under TJ, the problem was first studied by Kamiński et al. [26]. Very recently, the study of ISR on directed graphs has been initiated by Ito et al. [2]. Readers are referred to [1, 11] for a complete overview of recent developments regarding ISR. We now briefly mention some known results regarding the computational complexity of ISR on different graph classes. ISR remains PSPACE-complete under any of TS, TJ, TAR on general graphs [27], planar graphs of maximum degree three and bounded bandwidth [13, 21, 30] and perfect graphs [26]. Under TS, the problem is PSPACE-complete even on split graphs [3]. Interestingly, on bipartite graphs, ISR under TS is PSPACE-complete while under any of TJ and TAR it is NP-complete [7]. On the positive side, ISR under any of TJ and TAR is in P on even-hole-free graphs [26] (which also contains chordal graphs, split graphs, interval graphs, trees, etc.), cographs [16], and claw-free graphs [22]. ISR under TS is in P on cographs [26], claw-free graphs [22], trees [18], bipartite permutation graphs and bipartite distance-hereditary graphs [19], and interval graphs [4, 14].

DdISR for $d \geq 3$ was first studied by Siebertz [12] from the parameterized complexity viewpoint. More precisely, in [12], Siebertz proved that DdISR under TAR is in FPT for every $d \geq 2$ on "nowhere dense graphs" (which generalized the previously known result for $d = 2$ of Lokshtanov et al. [20]) and it is W[1]-hard for some value of $d \geq 2$ on "somewhere dense graphs" that are closed under taking subgraphs.

Since the TJ and TAR rules are somewhat equivalent [26], i.e., any TJ-sequence between two size-k token-sets can be converted into a TAR-sequence between them whose members are token-sets of size at

[1]In fact, the terminology *d-independent set* is sometimes used to indicate a vertex subset such that any two members are of distance at least $d + 1$ for $d \geq 1$. We note that sometimes a *d-independent set* is defined as a vertex subset S such that the maximum degree of the subgraph induced by S is at most d. In some other contexts, a *d-independent set* is nothing but an independent set of size d.

least $k-1$ and vice versa[2], in this paper, we consider DdISR ($d \geq 3$) under only TS and TJ. Most of our results are summarized in Table 1. In short, we show the following results.

- It is worth noting that
  - Even though it is well-known that MAXDdIS on G and MAXIS on its $(d-1)$th power (this concept will be defined later) are equivalent, this does *not* necessarily holds for their reconfiguration variants. (Section 3.1.)
  - The definition of DdIS implies the triviality of DdISR for large enough d on graphs whose (connected) components' diameters are bounded by some constant, including cographs and split graphs. (Section 3.2.)
- On chordal graphs and split graphs, there are some interesting complexity dichotomies. (Section 4.)
  - Under TJ on chordal graphs, DdISR is in P for even d and PSPACE-complete for odd d.
  - On split graphs, DdISR under TS is PSPACE-complete for $d = 2$ [3] but in P for $d = 3$. Under TJ, it is in P for $d = 2$ [26] but PSPACE-complete for $d = 3$.
- Several known results for $d = 2$ can be extended for $d \geq 3$. (Section 6.) Additionally, there is an alternative proof of the PSPACE-hardness of DdISR on general graphs under TJ for $d \geq 3$ which is not a direct extension of the corresponding result by Ito et al. [27] for $d = 2$. (Section 5.)

<table>
<tr><th>Graph</th><th>TS</th><th>TJ</th></tr>
<tr><td rowspan="2">chordal</td><td>$d = 2$: PSPACE-complete [3]</td><td>$d = 2$: P [26]</td></tr>
<tr><td>$d \geq 3$: unknown</td><td>even $d \geq 3$: P (Corollary 4.1)
odd $d \geq 3$: PSPACE-complete (Theorem 4.2)</td></tr>
<tr><td rowspan="3">split ($\subseteq$ chordal)</td><td>$d = 2$: PSPACE-complete [3]</td><td>$d = 2$: P [26]</td></tr>
<tr><td>$d = 3$: P (Proposition 4.4)</td><td>$d = 3$: PSPACE-complete (Corollary 4.3)</td></tr>
<tr><td colspan="2">$d \geq 4$: P (Proposition 3.3)</td></tr>
<tr><td rowspan="2">tree ($\subseteq$ chordal)</td><td>$d = 2$: P [18]</td><td>$d = 2$: P [26]</td></tr>
<tr><td>$d \geq 3$: unknown</td><td>$d \geq 3$: P (Corollary 7.1)</td></tr>
<tr><td rowspan="2">perfect</td><td colspan="2">$d = 2$: PSPACE-compelete [26]</td></tr>
<tr><td colspan="2">$d \geq 3$: PSPACE-complete (Theorem 6.2)</td></tr>
<tr><td rowspan="2">planar $\cap$ subcubic
$\cap$ bounded bandwidth</td><td colspan="2">$d = 2$: PSPACE-complete [13, 21, 30]</td></tr>
<tr><td colspan="2">$d \geq 3$: PSPACE-complete (Theorem 6.3)</td></tr>
<tr><td rowspan="2">cograph ($\subseteq$ perfect)</td><td>$d = 2$: P [26]</td><td>$d = 2$: P [16]</td></tr>
<tr><td colspan="2">$d \geq 3$: P (Corollary 3.4)</td></tr>
<tr><td rowspan="2">interval ($\subseteq$ chordal)</td><td>$d = 2$: P [4, 14]</td><td>$d = 2$: P [26]</td></tr>
<tr><td>$d \geq 3$: unknown</td><td>$d \geq 3$: P (Corollary 3.2)</td></tr>
</table>

Table 1: The computational complexity of DdISR ($d \geq 3$) on some graphs considered in this paper. For comparison, we also mention the corresponding known results for $d = 2$.

2. Preliminaries

For terminology and notation not defined here, see [15]. Let G be a simple, undirected graph with vertex-set $V(G)$ and edge-set $E(G)$. For two sets I, J, we sometimes use $I - J$ and $I + J$ to indicate $I \setminus J$ and $I \cup J$, respectively. Additionally, we simply write $I - u$ and $I + u$ instead of $I - \{u\}$ and $I + \{u\}$, respectively. The *neighbors* of a vertex v in G, denoted by $N_G(v)$, is the set $\{w \in V(G) : vw \in E(G)\}$. The *closed neighbors* of v in G, denoted by $N_G[v]$, is simply the set $N_G(v) + v$. Similarly, for a vertex subset $I \subseteq V(G)$, its *neighbor* $N_G(I)$ and *closed neighbor* $N_G[I]$ are respectively $\bigcup_{v \in I} N_G(v)$ and $N_G(I) + I$. The *degree* of a vertex v in G, denoted by $\deg_G(v)$, is $|N_G(v)|$. The *distance* between two vertices u, v in G, denoted by $\mathsf{dist}_G(u, v)$, is the number of edges in a shortest path between them. For convenience, if there is no path between u and v then $\mathsf{dist}_G(u, v) = \infty$. The *diameter* of G, denoted by $\mathsf{diam}(G)$, is the largest distance between any two vertices. A *(connected) component* of G is a maximal subgraph in which there is a path connecting any pair of vertices. An *independent set (IS)* of G is a vertex subset I such that for any $u, v \in I$, we have $uv \notin E(G)$. A *distance-d independent set (DdIS)* of G for an integer $d \geq 2$ is a vertex subset $I \subseteq V(G)$ such that for any $u, v \in I$, $\mathsf{dist}_G(u, v) \geq d$. We use $\alpha_d(G)$ to denote

[2]They proved the result for ISR, but it is not hard to extend it for DdISR.

the *maximum* size of a distance-d independent set of G. When $d = 2$, we use the well-known notation $\alpha(G)$ instead of $\alpha_2(G)$.

Unless otherwise noted, we denote by (G, I, J, R, d) an instance of DdISR under $\mathsf{R} \in \{\mathsf{TS}, \mathsf{TJ}\}$ where I and J are two distinct DdISs of a given graph G, for some fixed $d \geq 2$. Since (G, I, J, R, d) is obviously a no-instance if $|I| \neq |J|$, from now on, we always assume that $|I| = |J|$. Imagine that a token is placed on each vertex in a DdIS of a graph G. A TS-*sequence* in G between two DdISs I and J is the sequence $\mathcal{S} = \langle I = I_0, I_1, \dots, I_q = J\rangle$ such that for $i \in \{0, \dots, q-1\}$, the set I_i is a DdIS of G and there exists a pair $x_i, y_i \in V(G)$ such that $I_i - I_{i+1} = \{x_i\}$, $I_{i+1} - I_i = \{y_i\}$, and $x_i y_i \in E(G)$. By simply removing the restriction $x_i y_i \in E(G)$, we immediately obtain the definition of a TJ-*sequence* in G. Depending on the considered rule $\mathsf{R} \in \{\mathsf{TS}, \mathsf{TJ}\}$, we can also say that I_{i+1} is obtained from I_i by *immediately sliding/jumping* a token from x_i to y_i and write $x_i \xrightarrow{G}_{\mathsf{R}} y_i$. As a result, we can also write $\mathcal{S} = \langle x_0 \xrightarrow{G}_{\mathsf{R}} y_0, \dots, x_{q-1} \xrightarrow{G}_{\mathsf{R}} y_{q-1}\rangle$. In short, $\mathcal{S}$ can be viewed as a (ordered) sequence of either DdISs or token-moves. (Recall that we defined $\mathcal{S}$ as a sequence between I and J. As a result, when regarding $\mathcal{S}$ as a sequence of token-moves, we implicitly assume that the initial DdIS is I.) With respect to the latter viewpoint, we say that $\mathcal{S}$ *slides/jumps a token t from u to v in G* if t is originally placed on $u \in I_0$ and finally on $v \in I_q$ after performing $\mathcal{S}$. The *length* of a R-sequence is simply the number of times the rule R is applied.

We conclude this section with the following simple remark: since MAXDdIS is in NP, DdISR is always in PSPACE [27]. As a result, to show the PSPACE-completeness of DdISR, it is sufficient to construct a polynomial-time reduction from a known PSPACE-hard problem and prove its correctness.

3. Observations

3.1. Graphs and Their Powers

An extremely useful concept for studying distance-d independent sets is the so-called *graph power*. For a graph G and an integer $s \geq 1$, the *sth power of G* is the graph G^s whose vertices are $V(G)$ and two vertices u, v are adjacent in G^s if $\mathsf{dist}_G(u, v) \leq s$. Observe that I is a distance-d independent set of G if and only if I is an independent set of G^{d-1}. Therefore, MAXDdIS in G is equivalent to MAXIS in G^{d-1}.

However, this may *not* apply for their reconfiguration variants. More precisely, the statement "the DdISR's instance (G, I, J, R, d) is a yes-instance if and only if the ISR's instance $(G^{d-1}, I, J, \mathsf{R}, 2)$ is a yes-instance" holds for $\mathsf{R} = \mathsf{TJ}$ but not for $\mathsf{R} = \mathsf{TS}$. The reason is that we do not care about edges when performing token-jumps (as long as they result new DdISs), therefore whatever token-jump we perform in G can also be done in G^{d-1} and vice versa. Therefore, we have

Proposition 3.1. *Let $\mathcal{G}$ and $\mathcal{H}$ be two graph classes and suppose that for every $G \in \mathcal{G}$ we have $G^{d-1} \in \mathcal{H}$ for some fixed integer $d \geq 2$. If* ISR *under* TJ *on $\mathcal{H}$ can be solved in polynomial time, so does* DdISR *under* TJ *on $\mathcal{G}$.*

Recall that the power of any interval graph is also an interval graph [32, 33] and ISR under TJ on even-hole-free graphs (which contains interval graphs) is in P [26]. Along with Proposition 3.1, we immediately obtain the following corollary.

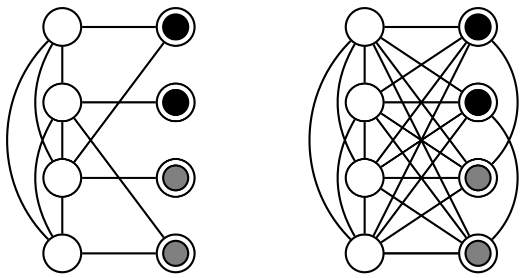
Corollary 3.2. DdISR *under* TJ *is in* P *on interval graphs for any $d \geq 2$.*

On the other hand, when using token-slides, we need to consider which edge can be used for moving tokens, and clearly G^{d-1} has much more edges than G, which means certain token-slides we perform in G^{d-1} cannot be done in G. Figure 1 describes an example of a DdISR's no-instance $(G, I, J, \mathsf{TS}, 3)$ whose corresponding ISR's instance $(G^2, I, J, \mathsf{TS}, 2)$ is a yes-instance. One can verify that in the former instance no token can move without breaking the "distance-3 restriction", while in the latter I can be transformed into J using exactly two token-slides. Moreover, these moves use edges that do not appear in G.

3.2. Graphs With Bounded Diameter Components

The following observation is straightforward.

Proposition 3.3. *Let $\mathcal{G}$ be a graph class such that there is some constant $c > 0$ satisfying $\mathsf{diam}(C_G) \leq c$ for any $G \in \mathcal{G}$ and any component C_G of G. Then,* DdISR *on $\mathcal{G}$ under $\mathsf{R} \in \{\mathsf{TS}, \mathsf{TJ}\}$ is in* P *for every $d \geq c + 1$.*



$(G, I, J, \mathsf{TS}, 3)$ $(G^2, I, J, \mathsf{TS}, 2)$

Figure 1: A DdISR's no-instance $(G, I, J, \mathsf{TS}, 3)$ whose corresponding ISR's instance $(G^2, I, J, \mathsf{TS}, 2)$ is a yes-instance. Tokens in I (resp., J) are marked with black (resp., gray) color.

Proof. When $d \geq c + 1$, any DdIS contains at most one vertex in each component of G, and the problem becomes trivial: under TJ, the answer is always "yes"; under TS, compare the number of tokens in each component. □

As a result, on cographs (a.k.a P_4-free graphs), one can immediately derive the following corollary.

Corollary 3.4. DdISR *on cographs under* $\mathsf{R} \in \{\mathsf{TS}, \mathsf{TJ}\}$ *is in* `P` *for any* $d \geq 2$.

Proof. It is well-known that the problems for $d = 2$ is in `P` [16, 26]. Since a connected cograph has diameter at most two, Proposition 3.3 settles the case $d \geq 3$. □

4. Chordal Graphs and Split Graphs

In this section, we will focus on chordal graphs and split graphs. Recall that the odd power of a chordal graph is also chordal [33, 38] and ISR under TJ on even-hole-free graphs (which contains chordal graphs) is in `P` [26]. Therefore, it follows from Proposition 3.1 that

Corollary 4.1. DdISR *is in* `P` *on chordal graphs under* TJ *for any even* $d \geq 2$.

In contrast, we have the following theorem.

Theorem 4.2. DdISR *is* `PSPACE`*-complete on chordal graphs under* TJ *for any odd* $d \geq 3$.

Proof. We reduce from the ISR problem, which is known to be `PSPACE`-complete under any of TS and TJ [27]. Let $(G, I, J, \mathsf{TJ}, 2)$ be an ISR's instance. We construct a DdISR's instance $(G', I', J', \mathsf{TJ}, d)$ ($d \geq 3$ is odd) as follows. We note that the same reduction was used by Eto et al. [23] for showing the `NP`-completeness of MaxDdIS on chordal graphs for odd $d \geq 3$. We first describe how to construct G' from G. First, for each edge $uv \in V(G)$, add a new vertex x_{uv} and create a new edge between x_{uv} and both u and v. Next, we add an edge in G' between x_{uv} and $x_{u'v'}$ for any pair of distinct edges $uv, u'v' \in E(G)$. Finally, for each $v \in V(G)$ we add a new path P_v on $(d-3)/2$ vertices and then add a new edge between v and one of P_v's endpoints. Let the resulting graph be G'. One can verify that G' is indeed a chordal graph: it is obtained from a split graph by attaching new paths to certain vertices. Clearly this construction can be done in polynomial time. (For example, see Figure 2.)

For each $u \in V(G)$, we define $f(u) \in V(G')$ to be the vertex whose distance in G' from u is largest among all vertices in $V(P_u) + u$. Let $f(X) = \bigcup_{x \in X}\{f(x)\}$ for a vertex subset $X \subseteq V(G)$. From the construction of G', note that if u and v are two vertices of distance 2 in G, one can always find a shortest path Q between $f(u)$ and $f(v)$ whose length is exactly d. Indeed, Q can be obtained by joining the paths from $f(u)$ to u, from u to x_{uw}, from x_{uw} to x_{wv}, from x_{wv} to v, and from v to $f(v)$, where $w \in N_G(u) \cap N_G(v)$. It follows that if I is an independent set of G then $f(I)$ is a distance-d independent set of G'. Therefore, we can set $I' = f(I)$ and $J' = f(J)$.

From the construction of G', note that for each $uv \in E(G)$, x_{uv} is of distance exactly one from each x_{wz} for $wz \in E(G) - uv$, at most two from each $v \in V(G)$, and at most $2 + (d-3)/2 \leq d - 1$ from each vertex in P_v for $v \in V(G)$. It follows that any distance-d independent set of G' of size at least two must not contain any vertex in $\bigcup_{uv \in E(G)}\{x_{uv}\}$.

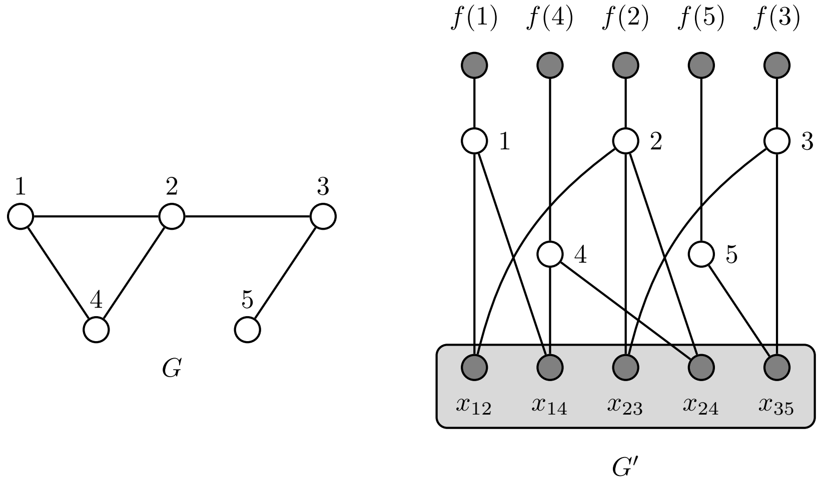


Figure 2: An example of constructing a chordal graph G' from G for $d = 5$. Vertices in a light-gray box form a clique. New vertices in $V(G') - V(G)$ are marked with the gray color.

We now show that there is a TJ-sequence between I and J in G if and only if there is a TJ-sequence between I' and J' in G'. If $|I| = |J| = 1$, the claim is trivial. As a result, we consider the case $|I| = |J| \geq 2$. Since $f(I)$ is a distance-d independent set in G' if I is an independent set in G, the only-if direction is clear. It remains to show the if direction. Let $\mathcal{S}'$ be a TJ-sequence in G' between I' and J'. We modify $\mathcal{S}'$ by repeating the following steps:

- Let $x \xrightarrow{G'}_{\mathsf{TJ}} y$ be the first token-jump that move a token from $x \in f(V(G))$ to some $y \in V(P_u) + u - f(u)$ for some $u \in V(G)$. If no such token-jump exists, we stop. Let I_x and I_y be respectively the distance-d independent sets obtained before and after this token-jump. In particular, $I_y = I_x - x + y$.
- Replace $x \xrightarrow{G'}_{\mathsf{TJ}} y$ by $x \xrightarrow{G'}_{\mathsf{TJ}} f(u)$ and replace the first step after $x \xrightarrow{G'}_{\mathsf{TJ}} y$ of the form $y \xrightarrow{G'}_{\mathsf{TJ}} z$ by $f(u) \xrightarrow{G'}_{\mathsf{TJ}} z$. From the construction of G', note that any path containing $f(u)$ must also contains all vertices in $V(P_u) + u$. Additionally, $I_x \cap (V(P_u) + u) = \emptyset$, otherwise no token in I_x can jump to y. Therefore, the set $I_x - x + f(u)$ is also a distance-d independent set. Moreover, $\mathsf{dist}_{G'}(f(u), z) \geq \mathsf{dist}_{G'}(y, z)$ for any $z \in V(G') - V(P_u)$. Roughly speaking, this implies that no token-jump between $x \xrightarrow{G'}_{\mathsf{TJ}} y$ and $y \xrightarrow{G'}_{\mathsf{TJ}} z$ breaks the "distance-d restriction". Thus, after the above replacements, $\mathcal{S}'$ is still a TJ-sequence in G'.
- Repeat the first step.

After modification, the final resulting TJ-sequence $\mathcal{S}'$ in G' contains only token-jumps between vertices in $f(V(G))$. By definition of f, we can construct a TJ-sequence between I and J in G simply by replacing each step $x \xrightarrow{G'}_{\mathsf{TJ}} y$ in $\mathcal{S}'$ by $f^{-1}(x) \xrightarrow{G}_{\mathsf{TJ}} f^{-1}(y)$. Our proof is complete. ☐

Now, we consider the split graphs. Proposition 3.3 implies that on split graphs (where each component has diameter at most 3), DdISR is in $\mathtt{P}$ under $\mathsf{R} \in \{\mathsf{TS}, \mathsf{TJ}\}$ for any $d \geq 4$. Interestingly, recall that when $d = 2$, the problem under TS is $\mathtt{PSPACE}$-complete even on split graphs [3] while under TJ it is in $\mathtt{P}$ [26]. It remains to consider the case $d = 3$.

Observe that the constructed graph G' in the proof of Theorem 4.2 is indeed a split graph when $d = 3$. Therefore, we have the following corollary.

Corollary 4.3. D3ISR *is* $\mathtt{PSPACE}$*-complete on split graphs under* TJ.

In contrast, under TS, we have the following proposition.

Proposition 4.4. D3ISR *is in* $\mathtt{P}$ *on split graphs under* TS.

Proof. Let $(G, I, J, \mathsf{TS}, 3)$ be an instance of D3ISR and suppose that $V(G)$ can be partitioned into a clique K and an independent set S. One can assume without loss of generality that G is connected, otherwise each component can be solved independently. If $|I| = |J| = 1$, the problem becomes trivial: $(G, I, J, \mathsf{R}, 3)$ is always a yes-instance. Thus, we now consider $|I| = |J| \geq 2$. Observe that for every

$u \in V(G)$ and $v \in K$, we have $\mathsf{dist}_G(u, v) \leq 2$. Therefore, in this case, both I and J are subsets of S. Now, no token in $I \cup J$ can be slid, otherwise such a token must be slid to some vertex in K, and each vertex in K has distance at most two from any other token, which contradicts the restriction that tokens must form a D3IS. Hence, $(G, I, J, \mathsf{TS}, 3)$ is always a no-instance if $|I| = |J| \geq 2$. □

5. An Alternative Reduction under TJ on General Graphs

Recall that Ito et al. [27] proved the PSPACE-completeness of ISR under TJ/TAR by reducing from 3-Satisfiability Reconfiguration (3SAT-R). In this section, we present a simple alternative proof for the PSPACE-hardness of DdISR ($d \geq 3$) on general graphs under TJ by *reducing from* ISR *instead of* 3SAT-R.

Theorem 5.1. DdISR *is* PSPACE*-complete under* TJ *for any* $d \geq 3$.

Proof. We reduce from the ISR problem, which is known to be PSPACE-complete under TJ [27]. Let $(G, I, J, \mathsf{TJ}, 2)$ be an ISR's instance. We construct a DdISR's instance $(G', I, J, \mathsf{TJ}, d)$ ($d \geq 3$) as follows. Unless otherwise noted, we always assume $p = (d-1)/2$ if d is odd and $p = \lfloor (d-1)/2 \rfloor$ if d is even. To construct G' from G, we first replace each edge $uv \in E(G)$ with a new path $P_{uv} = x_0^{uv} \dots x_{d-1}^{uv}$ of length $d-1$, where $x_0^{uv} = u$ and $x_{d-1}^{uv} = v$. We emphasize that the ordering of endpoints is important, i.e., $P_{vu} = x_0^{vu} \dots x_{d-1}^{vu}$ is the path where $x_i^{vu} = x_{d-1-i}^{uv}$ for $0 \leq i \leq d-1$. Basically, P_{uv} and P_{vu} describe the same path with different vertex-labels. If d is odd, we create a new clique K whose vertices are $\bigcup_{uv \in E(G)} \{x_p^{uv}\}$. If d is even, we add a new vertex $x^\star$ and create an edge between it and every vertex in $\bigcup_{uv \in E(G)} \{x_p^{uv}, x_{p+1}^{uv}\}$. The resulting graph is G'. (For example, see Figure 3.) Clearly this construction can be done in polynomial time.

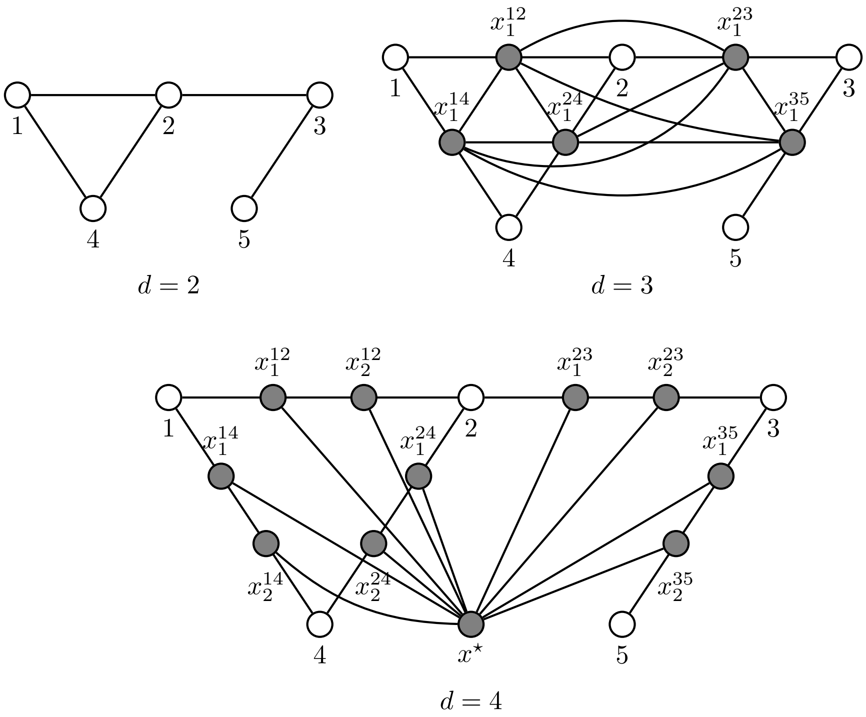


Figure 3: An example of constructing G' from a given graph G for some values of $d \in \{2, 3, 4\}$ (in case $d = 2$, we have $G' = G$). New vertices are marked with the gray color.

Now, we show that an independent set of G of size $k \geq 2$ is also a distance-d independent set of G' and vice versa. From the construction, for every $uv \in E(G)$, it follows that $\mathsf{dist}_{G'}(u, v) \leq d - 1$ and therefore both u and v cannot be in the same distance-d independent set of G'. Additionally, observe that for every $uv \notin E(G)$, any path between u and v in G' must contain x_p^{uw} and x_p^{vz}, for some $w \in N_G(u)$ and $z \in N_G(v)$. Thus, a shortest path between u and v in G' must be of the form $x_0^{uw} \dots x_p^{uw} x_p^{vz} \dots x_0^{vz}$ if d is odd and $x_0^{uw} \dots x_p^{uw} x^\star x_p^{vz} \dots x_0^{vz}$ if d is even. One can verify that such a shortest path has length exactly d. Therefore, $\mathsf{dist}_{G'}(u, v) \geq d$. As a result, if I is an independent set of G (of size $k \geq 2$) then it is also a distance-d independent set of G'. On the other hand, suppose that I' is a distance-d independent set of G' of size $k \geq 2$. We claim that I' does not contain any new vertex. Let $X = \bigcup_{uv \in E(G)} \{x_1^{uv}, \dots, x_{d-2}^{uv}\} \cup \{x^\star\}$ be the set of all new vertices. (Note that $x^\star$ only appears when d is even.) To show that $I' \cap X = \emptyset$, we will show that the distance between a vertex in X and any other vertex in $V(G')$ is at most $d-1$. From the construction of G', the distance between $x^\star$ (if exists) and any other vertex is at most $p+1 = \lfloor (d-1)/2 \rfloor + 1 \leq d-1$. It remains to show that one can always find a path of length at most $d-1$ between a new vertex $x = x_i^{uv} \in X - x^\star$ and a vertex $y = x_j^{wz} \in V(G') - \{x, x^\star\}$, where $1 \leq i \leq d-2$, $0 \leq j \leq d-1$, and $uv, wz \in E(G)$. If d is odd, such a path can be constructed by joining the paths from $x = x_i^{uv}$ to x_p^{uv} (of length $\leq p-1$), from x_p^{uv} to x_p^{wz} (of length ≤ 1), and from x_p^{wz} to $y = x_j^{wz}$ (of length $\leq p$). If d is even, such a path can be constructed by joining the paths from $x = x_i^{uv}$ to the vertex $a \in \{x_p^{uv}, x_{p+1}^{uv}\}$ closest to x (of length $\leq p-1$), from a to $x^\star$ (of length ≤ 1), from $x^\star$ to the vertex $b \in \{x_p^{wz}, x_{p+1}^{wz}\}$ closest to y (of length ≤ 1), and from b to $y = x_j^{wz}$ (of length $\leq p$). As a result, $I' \subseteq V(G)$, and from the construction of G', it follows that I' is an independent set of G of size $k \geq 2$.

We are now ready to show that $(G, I, J, \mathsf{TJ}, 2)$ is a yes-instance if and only if $(G', I, J, \mathsf{TJ}, d)$ is a yes-instance. Suppose that $|I| = |J| = k$. If $k = 1$, the claim is trivial. Therefore, we consider $k \geq 2$. In this case, since any independent set of G is also a distance-d independent set of G' and vice versa, it follows that any TJ-sequence in G is also a TJ-sequence in G' and vice versa. Our proof is complete. □

6. Extending Some Known Results for $d = 2$

In this section, we prove that several known results on the complexity of DdISR for the case $d = 2$ can be extended for $d \geq 3$.

6.1. General Graphs

Ito et al. [27] proved that ISR is PSPACE-complete on general graphs under TJ/TAR. Indeed, their proof uses only maximum independent sets, which implies that any token-jump is also a token-slide [22], and therefore the PSPACE-completeness also holds under TS. We will show that the reduction of Ito et al. [27] can be extended for showing the PSPACE-completeness of DdISR for $d \geq 3$.

Theorem 6.1. DdISR *is* PSPACE*-complete under* $\mathsf{R} \in \{\mathsf{TS}, \mathsf{TJ}\}$ *for any* $d \geq 3$.

Proof. Recall that Ito et al. [27] proved the PSPACE-hardness of the problem for $d = 2$ by reducing from a known PSPACE-complete problem called 3-Satisfiability Reconfiguration (3SAT-R) [28]. A *3SAT formula* φ consists of n variables $x_1, \dots, x_n$ and m clauses $c_1, \dots, c_m$, each clause contains at most three literals (a literal is either x_i or $\overline{x_i}$). A *truth assignment* ϕ is a way of assigning either 1 (true) or 0 (false) to each x_i. A truth assignment ϕ *satisfies a clause* if at least one literal is 1 and it *satisfies the formula* φ if all clauses are satisfied. In a 3SAT-R instance $(\varphi, \phi_s, \phi_t)$, two satisfying assignments ϕ and ϕ' are adjacent if they differ in assigning exactly one variable. The 3SAT-R problem asks whether there is an adjacent satisfying assignments for φ between two given satisfying assignments ϕ_s and ϕ_t. We will slightly modify their reduction to prove the PSPACE-hardness for $d \geq 3$.

We first describe the reduction by Ito et al. [27]. Let φ be a given 3SAT formula. We construct a graph G as follows. For each variable x_i $(1 \leq i \leq n)$ in φ, we add an edge e_{x_i} to the graph; its two endpoints are labeled x_i and $\overline{x_i}$. Then, for each clause c_j $(1 \leq j \leq m)$ in φ, we add a clique whose nodes correspond to literals in c_j. Finally, we add an edge between two nodes in different components if and only if the nodes correspond to opposite literals. Ito et al. [27] proved that there is a one-to-one correspondence between satisfying assignments for φ and maximum independent sets of size $n + m$ in G: n vertices are chosen from the endpoints of edges corresponding to the variables; a literal is 1 (true) if the corresponding endpoint is chosen. They uses this observation to show that there is a sequence of adjacent satisfying assignments between ϕ_s and ϕ_t if and only if there is a TJ-sequence between two corresponding size-$(n + m)$ independent sets of G. (Indeed, they proved under TAR rule. However, since only independent sets of size at least $n + m - 1$ are used, any TAR-sequence in G can be converted into a TJ-sequence [26], the result under TJ also holds.)

Our modification is simple. Instead of joining two nodes in different components by an edge, we join them by a path of length $d - 1$. Let G' be the resulting graph. Figure 4 describes an example of constructing G' from the same 3SAT formula used by Ito et al. [27]. Indeed, when $d = 2$, our constructed

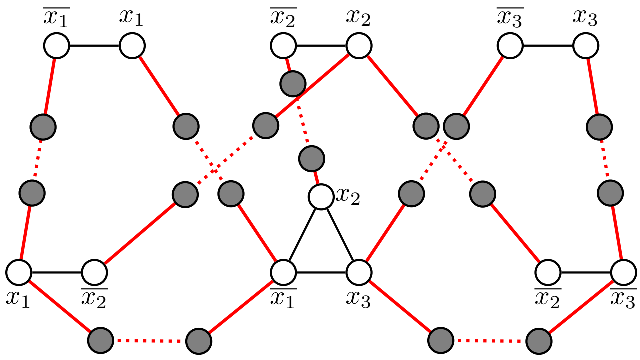


Figure 4: An example of constructing the graphs G and G' from a 3SAT formula φ used in [27] having three variables x_1, x_2, and x_3 and three clauses $c_1 = \{x_1, \overline{x_2}\}$, $c_2 = \{\overline{x_1}, x_2, x_3\}$, and $c_3 = \{\overline{x_2}, \overline{x_3}\}$. The red paths are of length $d-1$. New vertices in $V(G') - V(G)$ are marked with gray color.

graph G' and the graph G constructed by Ito et al. are identical. One can verify that any independent set of G of size $n+m$ is also a DdIS of G'. Additionally, note that no token can ever leave its corresponding component. To see this, choose a token t and assume inductively that the statement holds for all other tokens, then t can never jump/slide to any vertex outside its corresponding component because some other token must be moved first in order to make room for t. Thus, any token-jump is also a token-slide. (Recall that each component is either a single vertex, an edge, or a triangle.) Hence, $(G, I, J, \mathsf{R}, 2)$ is a yes-instance if and only if $(G', I, J, \mathsf{R}, d)$ is a yes-instance, where I and J are size-$(n+m)$ independent sets of G (which are also DdISs of G') and $\mathsf{R} \in \{\mathsf{TS}, \mathsf{TJ}\}$. Our proof is complete. □

6.2. Perfect Graphs

In this section, we will show the PSPACE-completeness of DdISR ($d \geq 3$) on perfect graphs by extending the corresponding known result of Kamiński et al. [26] for ISR.

Theorem 6.2. DdISR *is* PSPACE*-complete on perfect graphs under* $\mathsf{R} \in \{\mathsf{TS}, \mathsf{TJ}\}$ *for any* $d \geq 3$.

Proof. Recall that Kamiński et al. [26] proved the PSPACE-hardness of the problem for $d = 2$ by reducing from a known PSPACE-complete problem called Shortest Path Reconfiguration (SPR) [24]. In a SPR instance (G, u, v, P, Q), two shortest uv-paths are adjacent if one can be obtained from the other by exchanging exactly one vertex. The SPR problem asks whether there is a sequence of adjacent shortest uv-paths between P and Q. We will slightly modify their reduction to prove the PSPACE-hardness for $d \geq 3$.

We first describe the reduction by Kamiński et al. [26]. Let G be a given graph and let $u, v \in V(G)$. We delete all vertices and edges that does not appear in any shortest uv-path and let the resulting graph be $\widetilde{G}$. Let $k = \mathsf{dist}_G(u, v)$ and for each $i \in \{0, \dots, k\}$ let D_i be the set of vertices in $\widetilde{G}$ of distance i from u and $k-i$ from v. Let G' be the graph obtained from $\widetilde{G}$ by turning each D_i into a clique and complementing (i.e., if there is an edge between two vertices, remove it; otherwise, create one) the edges of $\widetilde{G}$ between two consecutive layers D_i and D_{i+1}. Kamiński et al. [26] proved that G' is a perfect graph. They also claimed that there is a one-to-one correspondence between uv-shortest paths of G and independent sets of size $k+1$ of G': the $k+1$ vertices of a shortest uv-path in G become independent in G'. This is used for showing that there is a sequence of adjacent uv-shortest paths in G if and only if there is a R-sequence between size-$(k+1)$ independent sets of G', for $\mathsf{R} \in \{\mathsf{TS}, \mathsf{TJ}, \mathsf{TAR}\}$.

Our modification is simple. We construct a graph G'' from G' as follows. First, we replace each edge between two consecutive layers D_i and D_{i+1} by a new path of length $d-1$, and let the resulting graph be $\widetilde{G'}$. For each $j \in \{1, \dots, d-2\}$, let D_i^j be the set of vertices in $\widetilde{G'}$ of distance j from some vertex in D_i and distance $d-1-j$ from some vertex in D_{i+1}. Let G'' be the graph obtained from $\widetilde{G'}$ by turning each D_i^j into a clique. Since G' is perfect, the above construction implies that G'' is perfect too. Clearly the construction of both G' and G'' can be done in polynomial time. In fact, if $d = 2$, we have $G'' = G'$. (For example, see Figure 5.)

It is sufficient to show that there is a R-sequence of size-$(k+1)$ independent sets in G' if and only if there is a R-sequence of size-$(k+1)$ distance-d independent sets in G'', for $\mathsf{R} \in \{\mathsf{TS}, \mathsf{TJ}\}$. From the

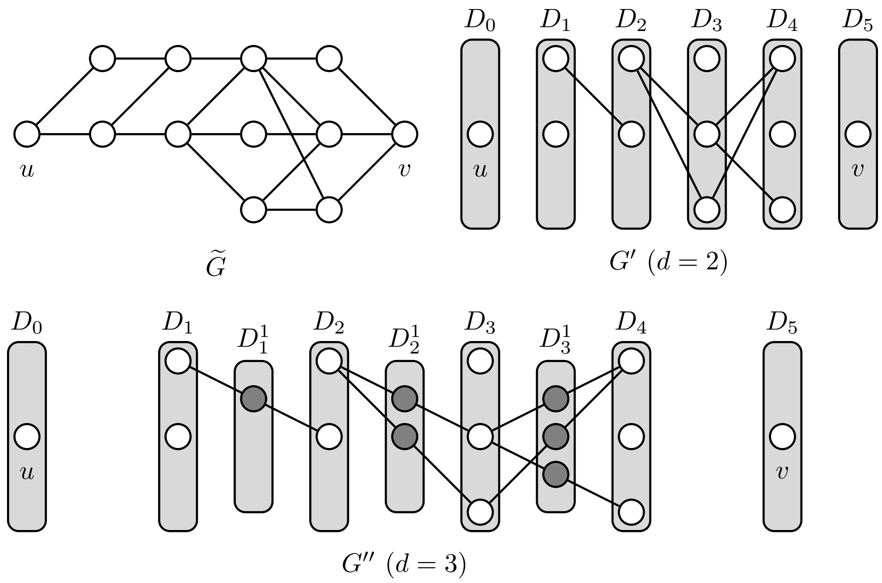


Figure 5: An example of constructing the perfect graphs G' and G'' from $\widetilde{G}$. Vertices in a light-gray box form a clique. New vertices in $V(G'') - V(G')$ are marked with the gray color.

construction of G'', it follows that any independent set of G' of size $k+1$ is also a distance-d independent set of G'', since a shortest path of length 2 in G' between two vertices $x \in D_i$ and $y \in D_{i+1}$ becomes a shortest path of length $(d-1)+1 = d$ in G''. Thus, any TJ-sequence in G' between two size-$(k+1)$ independent sets is also a TJ-sequence in G''. On the other hand, note that a size-$(k+1)$ distance-d independent set of G'' must not contain any new vertex in $V(G'') - V(G')$, otherwise one can verify that the set must be of size at most k, which is a contradiction. (Note that a DdIS must contain at most one vertex in $\bigcup_{j=1}^{d-2} D_i^j$.) As a result, a size-$(k+1)$ distance-d independent set of G'' is also an independent set of G'. Thus, any TJ-sequence in G'' between two size-$(k+1)$ distance-d independent sets is also a TJ-sequence in G''. We note that in both G' and G'', a token can never move out of the layer where it belongs, and therefore any token-jump is also a token-slide. Our proof is complete. ☐

6.3. Planar Graphs

In this section, we claim that the `PSPACE`-hard reduction of Hearn and Demaine [30] for ISR under TS can be extended to DdISR ($d \geq 3$) under $\mathsf{R} \in \{\mathsf{TS}, \mathsf{TJ}\}$. We now briefly introduce the powerful tool Hearn and Demaine [29, 30] used as the source problem for proving several hardness results, including the `PSPACE`-hardness of ISR: the *Nondeterministic Constraint Logic (NCL)* machine. Now we define the NCL

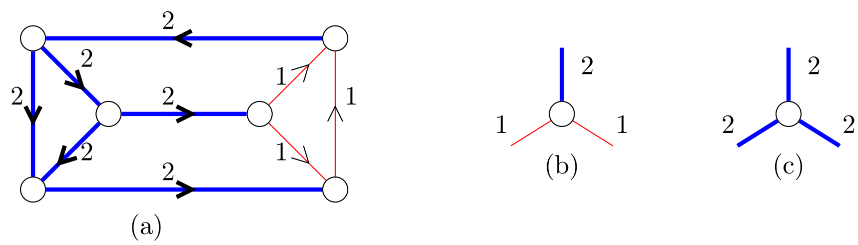


Figure 6: (a) A configuration of an NCL machine, (b) NCL And vertex, and (c) NCL Or vertex.

problem [30]. An NCL "machine" is an undirected graph together with an assignment of weights from $\{1, 2\}$ to each edge of the graph. An *(NCL) configuration* of this machine is an orientation (direction) of the edges such that the sum of weights of in-coming arcs at each vertex is at least two. Figure 6(a) illustrates a configuration of an NCL machine, where each weight-2 edge is depicted by a thick (blue) line and each weight-1 edge by a thin (red) line. Then, two NCL configurations are *adjacent* if they differ in a single edge direction. Given an NCL machine and its two configurations, it is known to be

PSPACE-complete to determine whether there exists a sequence of adjacent NCL configurations which transforms one into the other [30].

We note that the problem indeed remains PSPACE-complete even when restricted to AND/OR *constraint graphs*. An AND/OR *constraint graph* is a NCL machine that consists of only two types of vertices, called "NCL AND vertices" and "NCL OR vertices" defined as follows:

- A vertex of degree three is called an *NCL* AND *vertex* if its three incident edges have weights 1, 1 and 2. (See Figure 6(b).) An NCL AND vertex u behaves as a logical AND, in the following sense: the weight-2 edge can be directed outward for u if and only if both two weight-1 edges are directed inward for u. Note that, however, the weight-2 edge is not necessarily directed outward even when both weight-1 edges are directed inward.
- A vertex of degree three is called an *NCL* OR *vertex* if its three incident edges have weights 2, 2 and 2. (See Figure 6(c).) An NCL OR vertex v behaves as a logical OR: one of the three edges can be directed outward for v if and only if at least one of the other two edges is directed inward for v.

It should be noted that, although it is natural to think of NCL AND/OR vertices as having inputs and outputs, there is nothing enforcing this interpretation; especially for NCL OR vertices, the choice of input and output is entirely arbitrary because an NCL OR vertex is symmetric. For example, the NCL machine in Figure 6(a) is an AND/OR constraint graph.

Theorem 6.3. DdISR *is* PSPACE*-complete under* $\mathsf{R} \in \{\mathsf{TS}, \mathsf{TJ}\}$ *on planar graphs of maximum degree three and bounded bandwidth for any* $d \geq 2$.

Proof. We note that Hearn and Demaine [30]'s proof can be applied with the vertex gadgets shown in Figure 7. Indeed, when $d = 2$, the gadgets in Figure 7 are exactly those used in [30]. We rephrase their proof here for the sake of completeness. From now on we consider only TS rule. We will see later that in the constructed graph, any token-jump is also a token-slide, and therefore our reduction also holds for TJ. The maximum degree will also be clear from the construction of gadgets and how they are joined. The final complexity result follows from the result of van der Zanden [21] which combines [30] and [13]: NCL remains PSPACE-complete even if an input NCL machine is planar and bounded bandwidth.

The NCL AND and OR vertex gadgets are constructed as in Figure 7(a) and (b). The edges that cross the dashed-line gadget borders are "port" edges. A token on an outer port-edge vertex represents an inward-directed NCL edge, and vice versa. Given an AND/OR graph and configuration, we construct a corresponding graph for DdISR under TS, by joining together AND and OR vertex gadgets at their shared port edges, placing the port tokens appropriately.

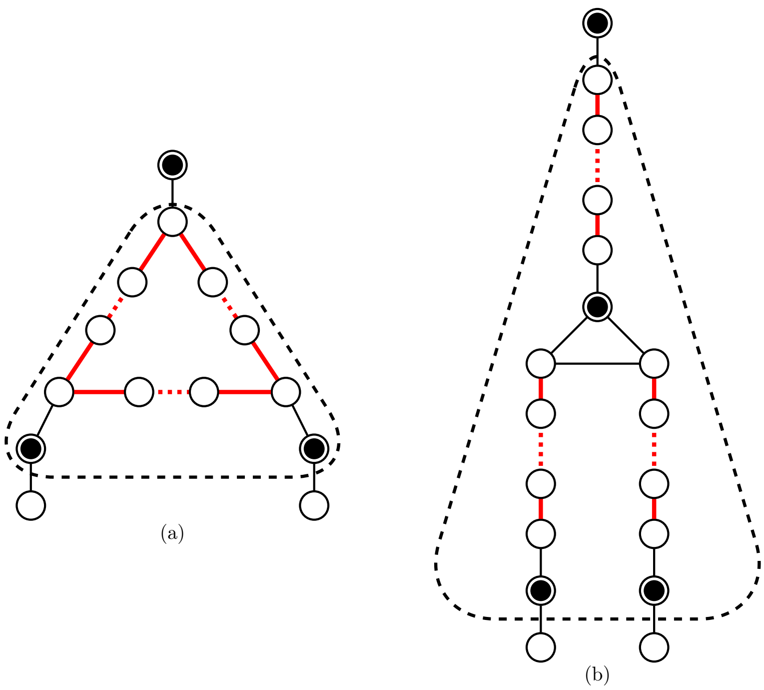


Figure 7: Vertex gadgets for DdISR under TS: (a) AND, (b) OR. The red paths are of length $d - 2$.

First, observe that no port token may ever leave its port edge. Choosing a particular port edge E, if we inductively assume that this condition holds for all other port edges, then there is never a legal move outside E for its token—another port token would have to leave its own edge first.

The AND gadget clearly satisfies the same constraints as an NCL AND vertex; the upper token can slide in just when both lower tokens are slid out. Likewise, the upper token in the OR gadget can slide in when either lower token is slid out—the internal token can then slide to one side or the other to make room. It thus satisfies the same constraints as an NCL OR vertex. As a result, it follows that a sequence of adjacent NCL configurations in a given AND/OR graph can indeed be transformed into a TS-sequence in the constructed graph and vice versa. Since no port token ever leaves its port edge, it follows from the construction of gadgets that in the constructed graph any token-jump is also a token-slide, which means the theorem also holds for TJ. Our proof is complete. □

7. Open Problem: Trees

Since the power of a tree is a (strongly) chordal graph [35, 36] and ISR on chordal graphs under TJ is in P [26], Proposition 3.1 implies that

Corollary 7.1. DdISR *under* TJ *on trees is in* P *for any* $d \geq 3$.

On the other hand, the complexity of DdISR under TS for $d \geq 3$ remains unknown.

Conjecture 7.2. DdISR *under* TS *on trees is in* P *for* $d \geq 3$.

Demaine et al. [18] showed that the problem for $d = 2$ is in P. Their algorithm is based on the so-called *rigid tokens*. Given a tree T and a DdIS I of T ($d \geq 2$), a token t on $u \in I$ is (T, I, d)*-rigid* if there is no TS-sequence that slides t from u to some vertex $v \in N_T(u)$. We denote by $\mathcal{R}(T, I, d)$ the set of all vertices where (T, I, d)-rigid tokens are placed. Demaine et al. proved that $\mathcal{R}(T, I, 2)$ can be found in linear time. (In general, deciding whether a token is $(G, I, 2)$-rigid on some graph G is PSPACE-complete [19].) Clearly it holds for any $d \geq 2$ that every instance $(T, I, J, \mathsf{TS}, d)$ where $\mathcal{R}(T, I, d) \neq \mathcal{R}(T, J, d)$ is a no-instance. When $\mathcal{R}(T, I, 2) = \mathcal{R}(T, J, 2) = \emptyset$, they proved that $(T, I, J, \mathsf{TS}, 2)$ is always a yes-instance. Based on these observations, one can derive a polynomial-time algorithm for solving ISR on trees under TS.

On the other hand, for $d \geq 3$, even when $\mathcal{R}(T, I, d) = \mathcal{R}(T, J, d) = \emptyset$, $(T, I, J, \mathsf{TS}, d)$ may be a no-instance. An example of such instances is described in Figure 8. As a result, Demaine et al.'s strategy

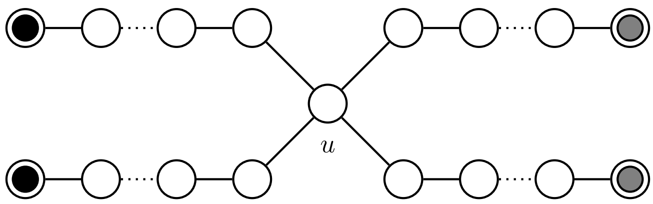


Figure 8: A no-instance $(T, I, J, \mathsf{TS}, d)$ ($d \geq 3$) with $\mathcal{R}(T, I, d) = \mathcal{R}(T, J, d) = \emptyset$. Tokens in I (resp., J) are marked with the black (resp. gray) color. All tokens are of distance $d - 1$ from u.

cannot be directly applied and thus the problem for $d \geq 3$ becomes more challenging.

Acknowledgements

This research is partially supported by the Japan Society for the Promotion of Science (JSPS) KAKENHI Grant Number JP20H05964 (AFSA).

References

[1] Nicolas Bousquet, Amer E. Mouawad, Naomi Nishimura, and Sebastian Siebertz. "A survey on the parameterized complexity of the independent set and (connected) dominating set reconfiguration problems". In: *arXiv preprint* (2022). arXiv: 2204.10526.

[2] Takehiro Ito, Yuni Iwamasa, Yasuaki Kobayashi, Yu Nakahata, Yota Otachi, Masahiro Takahashi, and Kunihiro Wasa. “Independent set reconfiguration on directed graphs”. In: *Proceedings of MFCS 2022*. Ed. by Stefan Szeider, Robert Ganian, and Alexandra Silva. Vol. 241. LIPIcs. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2022, 58:1–58:15. DOI: 10.4230/LIPIcs.MFCS.2022.58.

[3] Rémy Belmonte, Eun Jung Kim, Michael Lampis, Valia Mitsou, Yota Otachi, and Florian Sikora. “Token sliding on split graphs”. In: *Theory of Computing Systems* 65.4 (2021), pp. 662–686. DOI: 10.1007/s00224-020-09967-8.

[4] Marcin Briański, Stefan Felsner, Jędrzej Hodor, and Piotr Micek. “Reconfiguring Independent Sets on Interval Graphs”. In: *Proceedings of MFCS 2021*. Ed. by Filippo Bonchi and Simon J. Puglisi. Vol. 202. LIPIcs. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2021, 23:1–23:14. DOI: 10.4230/LIPIcs.MFCS.2021.23.

[5] Ioannis Katsikarelis, Michael Lampis, and Vangelis Th. Paschos. “Structurally parameterized d-scattered set”. In: *Discrete Applied Mathematics* 308 (2020), pp. 168–186. DOI: 10.1016/j.dam.2020.03.052.

[6] Ioannis Katsikarelis, Michael Lampis, and Vangelis Th. Paschos. “Improved (in-)approximability bounds for d-scattered set”. In: *Proceedings of WAOA 2019*. Ed. by Evripidis Bampis and Nicole Megow. Vol. 11926. LNCS. Springer, 2019, pp. 202–216. DOI: 10.1007/978-3-030-39479-0_14.

[7] Daniel Lokshtanov and Amer E. Mouawad. “The complexity of independent set reconfiguration on bipartite graphs”. In: *ACM Transactions on Algorithms* 15.1 (2019), 7:1–7:19. DOI: 10.1145/3280825.

[8] C.M. Mynhardt and S. Nasserasr. “Reconfiguration of colourings and dominating sets in graphs”. In: *50 years of Combinatorics, Graph Theory, and Computing*. Ed. by Fan Chung, Ron Graham, Frederick Hoffman, Ronald C. Mullin, Leslie Hogben, and Douglas B. West. 1st. CRC Press, 2019, pp. 171–191. DOI: 10.1201/9780429280092-10.

[9] Katsuhisa Yamanaka, Shogo Kawaragi, and Takashi Hirayama. “Exact Exponential Algorithm for Distance-3 Independent Set Problem”. In: *IEICE Transactions on Information and Systems* 102.3 (2019), pp. 499–501. DOI: 10.1587/transinf.2018FCL0002.

[10] Sangram K. Jena, Ramesh K. Jallu, Gautam K. Das, and Subhas C. Nandy. “The maximum distance-d independent set problem on unit disk graphs”. In: *Proceedings of FAW 2018*. Ed. by Jianer Chen and Pinyan Lu. Vol. 10823. LNCS. Springer, 2018, pp. 68–80. DOI: 10.1007/978-3-319-78455-7_6.

[11] Naomi Nishimura. “Introduction to reconfiguration”. In: *Algorithms* 11.4 (2018). (article 52). DOI: 10.3390/a11040052.

[12] Sebastian Siebertz. “Reconfiguration on nowhere dense graph classes”. In: *The Electronic Journal of Combinatorics* 25.3 (2018). (article P3.24). DOI: 10.37236/7458.

[13] Marcin Wrochna. “Reconfiguration in bounded bandwidth and treedepth”. In: *Journal of Computer and System Sciences* 93 (2018), pp. 1–10. DOI: 10.1016/j.jcss.2017.11.003.

[14] Marthe Bonamy and Nicolas Bousquet. “Token sliding on chordal graphs”. In: *Proceedings of WG 2017*. Vol. 10520. LNCS. Springer, 2017, pp. 127–139. DOI: 10.1007/978-3-319-68705-6_10.

[15] Reinhard Diestel. *Graph Theory*. 5th. Vol. 173. Graduate Texts in Mathematics. Springer, 2017. DOI: 10.1007/978-3-662-53622-3.

[16] Paul S. Bonsma. “Independent set reconfiguration in cographs and their generalizations”. In: *Journal of Graph Theory* 83.2 (2016), pp. 164–195. DOI: 10.1002/jgt.21992.

[17] Pedro Montealegre and Ioan Todinca. “On distance-d independent set and other problems in graphs with “few” minimal separators”. In: *Proceedings of WG 2016*. Ed. by Pinar Heggernes. Vol. 9941. LNCS. Springer, 2016, pp. 183–194. DOI: 10.1007/978-3-662-53536-3_16.

[18] Erik D. Demaine, Martin L. Demaine, Eli Fox-Epstein, Duc A. Hoang, Takehiro Ito, Hirotaka Ono, Yota Otachi, Ryuhei Uehara, and Takeshi Yamada. “Linear-time algorithm for sliding tokens on trees”. In: *Theoretical Computer Science* 600 (2015), pp. 132–142. DOI: 10.1016/j.tcs.2015.07.037.

[19] Eli Fox-Epstein, Duc A. Hoang, Yota Otachi, and Ryuhei Uehara. “Sliding token on bipartite permutation graphs”. In: *Proceedings of ISAAC 2015*. Vol. 9472. LNCS. Springer, 2015, pp. 237–247. DOI: 10.1007/978-3-662-48971-0_21.

[20] Daniel Lokshtanov, Amer E. Mouawad, Fahad Panolan, M. S. Ramanujan, and Saket Saurabh. "Reconfiguration on Sparse Graphs". In: *Proceedings of WADS 2015*. Vol. 9214. LNCS. Springer, 2015, pp. 506–517. DOI: 10.1007/978-3-319-21840-3_42.

[21] Tom C. van der Zanden. "Parameterized complexity of graph constraint logic". In: *Proceedings of IPEC 2015*. Vol. 43. LIPIcs. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2015, pp. 282–293. DOI: 10.4230/LIPIcs.IPEC.2015.282.

[22] Paul S. Bonsma, Marcin Kamiński, and Marcin Wrochna. "Reconfiguring independent sets in claw-free graphs". In: *Proceedings of SWAT 2014*. Vol. 8503. LNCS. Springer, 2014, pp. 86–97. DOI: 10.1007/978-3-319-08404-6_8.

[23] Hiroshi Eto, Fengrui Guo, and Eiji Miyano. "Distance-d independent set problems for bipartite and chordal graphs". In: *Journal of Combinatorial Optimization* 27.1 (2014), pp. 88–99. DOI: 10.1007/s10878-012-9594-4.

[24] Paul S. Bonsma. "The complexity of rerouting shortest paths". In: *Theoretical Computer Science* 510 (2013), pp. 1–12. DOI: 10.1016/j.tcs.2013.09.012.

[25] Jan van den Heuvel. "The complexity of change". In: *Surveys in Combinatorics*. Vol. 409. London Math. Soc. Lecture Note Ser. Cambridge University Press, 2013, pp. 127–160. DOI: 10.1017/CBO9781139506748.005.

[26] Marcin Kamiński, Paul Medvedev, and Martin Milanič. "Complexity of independent set reconfigurability problems". In: *Theoretical Computer Science* 439 (2012), pp. 9–15. DOI: 10.1016/j.tcs.2012.03.004.

[27] Takehiro Ito, Erik D. Demaine, Nicholas J. A. Harvey, Christos H. Papadimitriou, Martha Sideri, Ryuhei Uehara, and Yushi Uno. "On the complexity of reconfiguration problems". In: *Theoretical Computer Science* 412.12-14 (2011), pp. 1054–1065. DOI: 10.1016/j.tcs.2010.12.005.

[28] Parikshit Gopalan, Phokion G. Kolaitis, Elitza N. Maneva, and Christos H. Papadimitriou. "The connectivity of Boolean satisfiability: computational and structural dichotomies". In: *SIAM Journal on Computing* 38.6 (2009), pp. 2330–2355. DOI: 10.1137/07070440x.

[29] Robert A. Hearn and Erik D. Demaine. *Games, puzzles, and computation*. CRC Press, 2009.

[30] Robert A. Hearn and Erik D. Demaine. "PSPACE-completeness of sliding-block puzzles and other problems through the nondeterministic constraint logic model of computation". In: *Theoretical Computer Science* 343.1-2 (2005), pp. 72–96. DOI: 10.1016/j.tcs.2005.05.008.

[31] Geir Agnarsson, Peter Damaschke, and Magnús M. Halldórsson. "Powers of geometric intersection graphs and dispersion algorithms". In: *Discrete Applied Mathematics* 132.1-3 (2003), pp. 3–16. DOI: 10.1016/S0166-218X(03)00386-X.

[32] Mingjang Chen and Gerard J. Chang. "Families of graphs closed under taking powers". In: *Graphs and Combinatorics* 17.2 (2001), pp. 207–212. DOI: 10.1007/PL00007241.

[33] Geir Agnarsson, Raymond Greenlaw, and Magnús M. Halldórsson. "On powers of chordal graphs and their colorings". In: *Congressus Numerantium* 144 (2000), pp. 41–65.

[34] M. Kong and Y. Zhao. "Computing k-Independent sets for regular bipartite graphs". In: *Congressus Numerantium* 143 (2000), pp. 65–80.

[35] Paul E. Kearney and Derek G. Corneil. "Tree powers". In: *Journal of Algorithms* 29.1 (1998), pp. 111–131. DOI: 10.1006/jagm.1998.9999.

[36] Yaw-Ling Lin and Steven S Skiena. "Algorithms for square roots of graphs". In: *SIAM Journal on Discrete Mathematics* 8.1 (1995), pp. 99–118. DOI: 10.1137/S089548019120016X.

[37] M. Kong and Y. Zhao. "On computing maximum k-independent sets". In: *Congressus Numerantium* 95 (1993), pp. 47–47.

[38] R. Balakrishnan and P. Paulraja. "Powers of chordal graphs". In: *Journal of the Australian Mathematical Society* 35.2 (1983), pp. 211–217. DOI: 10.1017/S1446788700025696.

[39] Michael R. Garey and David S. Johnson. *Computers and intractability: A guide to the theory of NP-completeness*. W.H. Freeman and Company, 1979.