Refuting FPT Algorithms for Some Parameterized Problems Under Gap-ETH

Akanksha Agrawal
Indian Institute of Technology Madras, Chennai, India

Ravi Kiran Allumalla
Indian Institute of Technology Madras, Chennai, India

Varun Teja Dhanekula
Indian Institute of Technology Madras, Chennai, India

Abstract

In this article we study a well-known problem, called Bipartite Token Jumping and not-so-well known problem(s), which we call, Half (Induced-) Subgraph, and show that under Gap-ETH, these problems do not admit FPT algorithms. The problem Bipartite Token Jumping takes as input a bipartite graph G and two independent sets S, T in G, where $|S| = |T| = k$, and the objective is to test if there is a sequence of exactly k-sized independent sets $\langle I_0, I_1, \cdots, I_\ell \rangle$ in G, such that: i) $I_0 = S$ and $I_\ell = T$, and ii) for every $j \in [\ell]$, I_j is obtained from I_{j-1} by replacing a vertex in I_{j-1} by a vertex in $V(G) \setminus I_{j-1}$. We show that, assuming Gap-ETH, Bipartite Token Jumping does not admit an FPT algorithm. We note that this result resolves one of the (two) open problems posed by Bartier et al. (ISAAC 2020), under Gap-ETH. Most of the known reductions related to Token Jumping exploit the property given by triangles (i.e., C_3s), to obtain the correctness, and our results refutes FPT algorithm for Bipartite Token Jumping, where the input graph cannot have any triangles.

For an integer $k \in \mathbb{N}$, the half graph $S_{k,k}$ is the graph with vertex set $V(S_{k,k}) = A_k \cup B_k$, where $A_k = \{a_1, a_2, \cdots, a_k\}$ and $B_k = \{b_1, b_2, \cdots, b_k\}$, and for $i, j \in [k]$, $\{a_i, b_j\} \in E(T_{k,k})$ if and only if $j \geq i$. We also study the Half (Induced-)Subgraph problem where we are given a graph G and an integer k, and the goal is to check if G contains $S_{k,k}$ as an (induced-)subgraph. Again under Gap-ETH, we show that Half (Induced-)Subgraph does not admit an FPT algorithm, even when the input is a bipartite graph. We believe that the above problem (and its negative) result maybe of independent interest and could be useful obtaining new fixed parameter intractability results.

There are very few reductions known in the literature which refute FPT algorithms for a parameterized problem based on assumptions like Gap-ETH. Thus our technique (and simple reductions) exhibits the potential of such conjectures in obtaining new (and possibly easier) proofs for refuting FPT algorithms for parameterized problems.

2012 ACM Subject Classification Theory of computation → Parameterized complexity and exact algorithms

Keywords and phrases Token Jumping, Bipartite Graphs, Fixed Parameter Intractability, Half Graphs, Gap-Exponential Time Hypothesis

Digital Object Identifier 10.4230/LIPIcs.IPEC.2021.2

1 Introduction

In this article we study two graph problems and obtain intractability results under Gap-Exponential Time Hypothesis (Gap-ETH), which was conjectured by Dinur [13] and Marx [33]. Gap-ETH conjectures that for some constant $\epsilon > 0$, even to distinguish whether all clauses or at most $(1 - \epsilon)$ fraction of the clauses in the given CNF-SAT formula can be satisfied by an assignment, requires exponential amount of time. Most of the known (fixed parameter) intractability results in the field of parameterized complexity are based on conjectures like FPT $\neq$ W[t], where t is a positive integer. While conjectures like Gap-ETH are stronger than the more accepted conjectures like the one stated above, they can be used to explain the hardness of a problem by connecting them to SAT, particularly in the absence of insights into the (in)tractability/hardness of a problem. The main message of this article is that, we can (in certain situtation) harness the "approximation gap" provided by the Gap-ETH in obtaining new and simple intractability results. We would like to remark that Gap-ETH has been also previously used by Bhattacharyya et al. [4] for obtaining fixed parameter intractability result for Even Set (among other problems), whose complexity status, prior to their result, was an infamous open problem in the field (see for example, the book of Downey and Fellow [15]).

Reconfigurations and Token Jumping. The first problem that we study is a reconfiguration problem, where in a typical such problem, we have an underlying problem, an instance of the underlying problem, solutions for the instances, and a transformation function from one solution to the other, and the objective is to use check existence of a sequence of transformation from a given solution to another (see, for example, [35] for an introduction to reconfigurations). We study the reconfiguration of independent sets on a graph, where the initial independent set S and final independent set T are of the same size k. The transformations we allow is called token jumping, which was first introduced by Kamiński et al. [22]. We can view a transformation of an independent set to another, in our setting, as jumping a token from one vertex to some other vertex, while maintaining the independence property, where all the tokens are initially placed in the former independent set S. The objective of Token Jumping for a given graph G and independent sets S and T of size k is to test existence of a sequence of transformations from S to T via jumping tokens (one at-a-time). For two sets X and Y, $X\Delta Y$ denotes the set $(X \setminus Y) \cup (Y \setminus X)$. Next we formally define the problem Bipartite Token Jumping, which is Token Jumping when the input is a bipartite graph.

Bipartite Token Jumping **Parameter:** k
Input: A bipartite graph G and two independent sets S and T in G, where $|S| = |T| = k$.
Question: Determine if there a sequence of exactly k-sized independent sets $\langle I_0, I_1, \cdots, I_\ell \rangle$, in G, such that the following holds:
1. $I_0 = S$ and $I_\ell = T$, and
2. for each $j \in [\ell]$, $|I_{j-i} \Delta I_j| = 2$.

Some Known Results on Token Jumping. We note that Token Jumping problem is known to be polynomial time solvable, when the input is restricted to special graph classes like trees [11], interval graphs [5], bipartite permutation and bipartite distance-hereditary graphs [17], line graphs [19] and even-hole-free graphs [22].

Although Independent Set on bipartite graphs is one of the classic polynomial time solvable problem (see, for instance, [24] and [18]), Lokshtanov and Mouawad [31] showed that the problem Bipartite Token Jumping is NP-hard. Lokshtanov et al. [32] showed that Token Jumping is W[1]-hard, when parameterized by the sizes of the given independent sets. They also designed an FPT algorithm for the problem when the input graph has bounded degeneracy. We remark that prior to the above result, Ito et al. [21] had obtained an FPT algorithm for the problem when parameterized by k and the maximum degree of the input graph. Token Jumping is also known to be FPT on strongly $K_{\ell,\ell}$-free graphs [6, 20], where a graph is said to be strongly $K_{\ell,\ell}$-free if it does not contain the complete bipartite graph $K_{\ell,\ell}$ as a subgraph. Bartier et al. [2, 3] designed an FPT algorithm for Token Jumping on C_4-free bipartite graphs when parameterized by the sizes of the independent sets. They also showed that the problem has no FPT algorithm when the input is restricted to C_4-free graphs. The status of TOKEN JUMPING on bipartite graphs, i.e., the problem BIPARTITE TOKEN JUMPING, was left as an open problem by Bartier et al. [2].

Our Result on Token Jumping. We resolve one of the open problems of Bartier et al. [2], under Gap-ETH, by proving the following theorem.

▶ **Theorem 1.** *Assuming Gap-ETH, BIPARTITE TOKEN JUMPING does not admit an FPT algorithm, when parameterized by k.*

We obtain the above result by via a simple reduction from the MAXIMUM BALANCED BICLIQUE problem, and then use its FPT-inapproximability result by Chalermsook et al. [9] under Gap-ETH, to show that BIPARTITE TOKEN JUMPING cannot have an FPT algorithm. Roughly speaking, by exploiting the Pigeonhole principle in our proof, if the input bipartite graph has a biclique $K_{k,k}$ as an (induced-)subgraph with bipartition A and B, then we are able to argue that there will be a point in the transformation from one of the appropriately constructed independent set to the another, which must contain at least half of vertices from A and B each. Using the above we are able to exploit the known FPT inapproximability of MAXIMUM BALANCED BICLIQUE to obtain our result.

(Induced-)Subgraphs and Half Graphs. The SUBGRAPH ISOMORPHISM and INDUCED-SUBGRAPH ISOMORPHISM are among the fundamental problems in algorithmic graph theory. (INDUCED-)SUBGRAPH ISOMORPHISM takes as input two graphs G and H, and the objective is to test if (an isomorphic copy of) H is contained as an (induced-)subgraph in G. For a family of graphs Π, a natural generalization of (INDUCED-)SUBGRAPH ISOMORPHISM is the problem Π (INDUCED-)SUBGRAPH, where we are given a graph G and an integer k, and the objective is to test if G contains an (induced-)subgraph isomorphic to a graph on k vertices in the family Π. The problems SUBGRAPH ISOMORPHISM and INDUCED-SUBGRAPH ISOMORPHISM are among the very well-studied problems in Computer Science, which are NP-hard, as they (both) encapsulate problems like CLIQUE (see, [10] and [23]). The fixed parameter intractability of CLIQUE, parameterized by the solution size [14], also implies that both of the above problems do not admit any FPT algorithm when parameterized by $|V(H)|$.

We study Π (INDUCED-)SUBGRAPH, when Π is a family of structured graphs, called *half* graphs (see, [16]) and the corresponding problem(s) HALF (INDUCED-)SUBGRAPH.[1] For an integer $k \in \mathbb{N}$, the half graph $S_{k,k}$ is the graph with vertex set $V(S_{k,k}) = A_k \cup B_k$, where $A_k = \{a_1, a_2, \cdots, a_k\}$ and $B_k = \{b_1, b_2, \cdots, b_k\}$, and for $i, j \in [k]$, $\{a_i, b_j\} \in E(S_{k,k})$ if and only if $j \geq i$ (see Figure 1, for an illustration).

Our result will particularly focus on the instances where the input graph is bipartite, and thus we formally define the problems BIPARTITE HALF SUBGRAPH and BIPARTITE HALF INDUCED-SUBGRAPH below.

BIPARTITE HALF (INDUCED-)SUBGRAPH	**Parameter:** k
Input: A bipartite graph G and an integer k.	
Question: Does G contain a graph isomorphic to $S_{k,k}$ as an (induced-)subgraph?	

[1] We note that the term “skew” was used in the context of SKEW MULTICUT and DIRECTED FEEDBACK ARC SET for a graph class closely related to half graphs (see [26], for more details). Another closely related graph class is the family of bipartite chain graphs (or bisplits), please see [7].

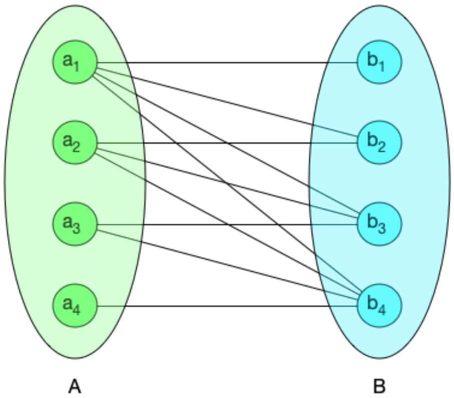

Figure 1 The half graph $S_{4,4}$.

Some Known Results on Π (Induced-)Subgraph. The problems Π (Induced-)Subgraph have been studied for several structured families of graphs, most notable being Clique and Independent Set. Although, when Π is among the many well-known and highly structured families like cliques, independent sets, and bicliques, the problem seem to be "inherently-enumerative", and thus, fixed parameter intractable [9], we do know several instances where the problem is FPT. Khot and Raman [25] obtained a complete fixed parameter tractability dichotomy for Π Subgraph, for hereditary families, by showing that if Π includes all graphs with no edges but not all complete graphs, or vice versa, then the problem is fixed parameter intractable, and otherwise, it admits an FPT algorithm. Also, Kratsch et al. [27] studied the kernelization complexity for hereditary families of graphs for such problems. Lin and Chen [30] obtained an FPT algorithm for Π Induced-Subgraph, when Π is the family of all graphs with exactly k edges.[2]

One of the notorious parameterized problems, whose status of fixed parameter tractability was open until 2015 is the Biclique problem. Despite it's deceptive similarity with Clique, the problem was not known to be fixed parameter intractable, until such a result was obtained by Lin [28, 29]. Later, Chalermsook et al. [8, 9] showed that, for any function $f \in o(k)$, Biclique does not admit a $k/f(k)$-approximation algorithm, running in FPT time, assuming Gap-ETH.

Our Contribution on Π (Induced-)Subgraph. We study the Bipartite Half (Induced-) Subgraph problem(s), and obtain the following result.

▶ **Theorem 2.** *Assuming Gap-ETH, Bipartite Half (Induced-)Subgraph does not admit an FPT algorithm, when parameterized by k.*

We obtain the above result by a (Turing) reduction starting from the problem Maximum Balanced Biclique. For the given instance of Maximum Balanced Biclique, we begin by "color-coding" a copy of $K_{k,k}$ in the input graph (if it exist). We note that color-coding was introduced by Alon et al. [1], for designing FPT algorithms for parameterized by the solution size, for detecting cycles and paths of a given length. After this we create two copies of each of the color classes obtained using the color-coding to "complete" an $S_{2k,2k}$ using the

[2] Here, k used in the definition of Π is the same as the number k which is provided as input in the Π Induced-Subgraph instance.

presence of the biclique. We note that directly using Multicolored Biclique leads to issues in the one of direction of the proof, relating to the stricter condition of picking vertices from each of the color classes. We note that approximation in the solution that we are able to output for the Maximum Balanced Biclique instance arises due to the possibility of picking multiple copies of the same vertex.

We would like to note that, we came across the problem Bipartite Half (Induced-)Subgraph, while studying Bipartite Token Jumping. Although, later we obtained a simple proof for our negative result regarding Bipartite Token Jumping, eliminating the need to go via the route of Bipartite Half (Induced-)Subgraph. Nevertheless we believe due to the "sequence-like" structure it offers, it may turn out to be a useful starting point for obtaining new intractability results, similar to its cousins like Clique and Biclique.

2 Preliminaries

Sets and Functions. We denote the set of natural numbers (including 0) by $\mathbb{N}$. For $n \in \mathbb{N}$, we use $[n]$ and $[n]_0$ to denote the sets $\{1, 2, \cdots, n\}$ and $\{0, 1, 2, \cdots, n\}$, respectively. For a set X, we denote its power set by $2^X = \{X' \mid X' \subseteq X\}$. For a set Y, by $Y \times Y$, we denote the set $\{(y, y') \mid y, y' \in Y\}$. For a function $f : X \to Y$ and an element $y \in Y$, $f^{-1}(y)$ denotes the set $\{x \in X \mid f(x) = y\}$. For a non-empty set X, a family $\mathcal{F} \subseteq 2^X$ is a *partition* of X, if i) for each $Y \in \mathcal{F}$, $Y \neq \emptyset$, ii) for distinct $Y, Z \in \mathcal{F}$, we have $Y \cap Z = \emptyset$, and iii) $\cup_{Y \in \mathcal{F}} Y = X$.

Graph Theory. We refer to the book of [12] for standard graph terminology. Given a graph G, we denote its vertex set and edge set by $V(G)$ and $E(G)$, respectively. Whenever the context is clear, we use n and m to denote the number of vertices and the number of edges in the input graph, respectively. Consider a graph G. For $X \subseteq V(G)$, $G[X]$ denotes the subgraph of G induced by X, i.e. $G[X]$ is the graph with vertex set X and edge set $\{\{x, y\} \in E(G) \mid x, y \in X\}$.

For a graph G, a set $S \subseteq V(G)$ is an *independent set* in G if for every $u, v \in S$, we have $\{u, v\} \notin E(G)$. A graph G is *bipartite* if its vertex set can be partitioned into two independent sets X and Y. In the above, X, Y is called a *bipartition* of G.

For an integer $k \in \mathbb{N}$, $K_{k,k}$ denotes the bipartite graph with a bipartition A and B, each with k vertices, so that for each $a \in A$ and $b \in B$, we have $\{a, b\} \in E(K_{k,k})$. In the above, $K_{k,k}$ is called a *biclique*. For a bipartite graph G, by $\mathsf{opt\text{-}biclq}(G)$ we denote the largest integer k^*, such that G has a K_{k^*,k^*}-biclique as an induced subgraph.

Satisfiability and Gap-ETH. For $q \in \mathbb{N}$, the q-SAT problem takes as input a q-CNF formula ϕ with m clauses on a variable set X of size n, and the objective is to test if there is an assignment $\mathsf{asg} : X \to \{0, 1\}$ that satisfies ϕ.

Next we define the maximization version of q-SAT, called Max q-SAT. For $q \in \mathbb{N}$, the problem Max q-SAT takes as input a q-CNF formula ϕ with m clauses on a variable set X of size n, and the objective is to compute, $\mathsf{opt\text{-}SAT}(\phi)$, which is the maximum number of clauses in ϕ that can be simultaneously satisfied by an assignment $\mathsf{asg} : X \to \{0, 1\}$.

We next state the statement of Gap-ETH.

▶ **Conjecture 3** (Gap-Exponential Time Hypothesis (Gap-ETH), [13, 33])**.** *For some constant $\delta, \epsilon > 0$, no algorithm can, given a 3-CNF formula ϕ on n variables and $m = \mathcal{O}(n)$ clauses, distinguish between the following cases correctly with probability at least $2/3$ in time $\mathcal{O}(2^{\delta n})$: i) $\mathsf{opt\text{-}SAT}(\phi) = m$ and ii) $\mathsf{opt\text{-}SAT}(\phi) < (1 - \epsilon) \cdot m$.*

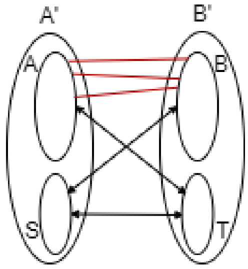

Figure 2 Illustration of construction of G'. Double arrows show all possible edges between the sets, and set lines between A and B shows complementation of edges between them from G.

Some Results on Maximum Balanced Biclique. In the MAXIMUM BALANCED BICLIQUE problem we are given a bipartite graph G and an integer k, and the objective is to test if $\mathsf{opt\text{-}biclq}(G) \geq k$.[3]

In the following we state an inapproximability result regarding MAXIMUM BALANCED BICLIQUE, which follows as a corollary from Definition 2.1 and 3.2, Proposition 3.3 and Corollary 5.16 of [9].

▶ **Proposition 4.** *Assuming Gap-ETH, there is no algorithm, which for all bipartite graphs G on n vertices and an integer k, runs in time $f(k) \cdot n^{\mathcal{O}(1)}$, and does one of the following i) outputs* 1 *if* $\mathsf{opt\text{-}biclq}(G) \geq k$*, or ii) outputs* 0 *if* $\mathsf{opt\text{-}biclq}(G) < k/2$*; where f is some computable function. That is,* MAXIMUM BALANCED BICLIQUE *does not admit a factor* 2*-approximation algorithm running in FPT time.*

We remark that the above proposition is a very special instantiation, which is enough for our purpose, of the more general result of Corollary 5.16 of [9].

3 Refuting FPT Algorithms for Bipartite Token Jumping

The objective of this section is to prove Theorem 1. We obtain the above by exhibiting an appropriate reduction from MAXIMUM BALANCED BICLIQUE. We begin by explaining the intuition behind our reduction. Consider an instance (G, k) of MAXIMUM BALANCED BICLIQUE, where G is a bipartite graph with vertex bipartition A and B. We will create an instance (G', S, T) of BIPARTITE TOKEN JUMPING as follows. Intuitively speaking, G' will be obtained from G by adding two new sets of vertices S and T, each of size k. The construction will ensure that in any transformation from S to T, there must exist a consecutive pair of independent sets, which places $k/2$ tokens in A and B, each, which will induce a complete bipartite graph in G (via complementing edges). We will now formally describe our construction.

Construction of (G', S, T). Initialize $V(G') = V(G) = A \cup B$ and $E(G') = \{\{a, b\} \mid a \in A, b \in B$, and $\{a, b\} \notin E(G)\}$ (see Figure 2). We add two sets of ($k + 1$ new vertices), $S = \{s_1, s_2, \cdots, s_k, s_{k+1}\}$ and $T = \{t_1, t_2, \cdots, t_k, t_{k+1}\}$ to $V(G')$. We add all the edges between $S \cup A$ and T, i.e, we add all the edges in $\{\{u, t\} \mid u \in S \cup A$ and $t \in T\}$ to $E(G')$. Also,

[3] The definition of MAXIMUM BALANCED BICLIQUE given in [9], is a purely optimization version, where k is not part of the input. We state the problem this way, as this helps us get around unnecessary technicalities, without compromising on the technical merit of the result.

we add all the edges between vertices in S and B, i.e., the edges in $\{\{s,b\} \mid s \in S \text{ and } b \in B\}$ to $E(G')$. Clearly, by the construction, G' is a bipartite graph, with bipartition $A' = S \cup A$ and $B' = T \cup B$. The above concludes the construction of the Bipartite Token Jumping instance (G', S, T).

In the following lemmata we will establish some useful properties that will help us prove the theorem.

▶ **Lemma 5.** *If (G,k) is a yes-instance of* Maximum Balanced Biclique*, then (G', S, T) is a yes-instance of* Bipartite Token Jumping*.*

Proof. Suppose that $\mathsf{opt\text{-}biclq}(G) \geq k$, and consider $X = \{x_1, x_2, \cdots, x_k\} \subseteq A$ and $Y = \{y_1, y_2, \cdots, y_k\} \subseteq B$, such that $G[X \cup Y]$ is isomorphic to $K_{k,k}$. We will construct a sequence $\langle S = I_0, I_1, \cdots, I_k, I'_1, I'_2, \cdots, I'_k, \widehat{I}_1, \widehat{I}_2, \cdots, \widehat{I}_{k+1} = T\rangle$, of independent sets of size $k+1$ in G', to conclude that (G', S, T) is a yes-instance of Bipartite Token Jumping.

For $j \in [k]$, let $I_j = (I_{j-1} \setminus \{s_j\}) \cup \{x_j\}$. As $X \subseteq A$ and A is an independent set in G', it follows that, for each $j \in [k]$, I_j is an independent set in G'. Recall that S is an independent set in G' of size exactly $k+1$, $S \cap A = \emptyset$, and $X \subseteq A$. Thus we can obtain that, for each $j \in [k]$, I_j is an independent set in G' of size exactly $k+1$ in G'.

Let $I'_1 = (I_k \setminus \{s_{k+1}\}) \cup \{y_1\}$. For $j \in [k] \setminus \{1\}$, $I'_j = (I'_{j-1} \setminus \{x_{j-1}\}) \cup \{y_j\}$. Note that for each $x \in X$ and $y \in Y$, we have $\{x,y\} \in E(G)$, and thus $\{x,y\} \notin E(G')$. The above together with the fact that $X \cap Y = \emptyset$, implies that for each $j \in [k]$, I'_j is an independent set in G' of size exactly $k+1$.

Finally, let $\widehat{I}_1 = (I'_k \setminus \{x_k\}) \cup \{t_1\}$, and for $j \in [k+1] \setminus \{1\}$, let $\widehat{I}_j = (\widehat{I}_{j-1} \setminus \{y_{j-1}\}) \cup \{t_j\}$. As $B \cup T$ is an independent set in G' and I'_k is an independent set of size exactly $k+1$ in G', it follows that for each $j \in [k+1]$, $\widehat{I}_j$ is an independent set of exactly size $k+1$ in G'. Also, by the construction of $\widehat{I}_{k+1}$ it follows that $\widehat{I}_{k+1} = T$.

From the construction of $\langle S = I_0, I_1, \cdots, I_k, I'_1, I'_2, \cdots, I'_k, \widehat{I}_1, \widehat{I}_2, \cdots, \widehat{I}_{k+1} = T\rangle$ it follows that it is a solution for the Bipartite Token Jumping instance (G, S, T). This concludes the proof. ◀

In the next lemma we exploit a solution for the Bipartite Token Jumping instance (if it exists), to obtain an approximation for the Maximum Balanced Biclique instance.

▶ **Lemma 6.** *If (G', S, T) is a yes-instance of* Bipartite Token Jumping*, then* $\mathsf{opt\text{-}biclq}(G) \geq k/2$*.*

Proof. Suppose that (G', S, T) is a yes-instance of Bipartite Token Jumping, and let $\langle S = I_0, I_1, \cdots, I_\ell = T\rangle$ be a solution for it. Recall that for each $s \in S$ and $u \in T \cup B$, we have $\{s,u\} \in E(G')$, $|S| = |T| = k+1$, and $S \cap T = \emptyset$. Thus, there must exist $q \in [\ell]$ such that $|I_q \cap A| = k$. Let q_{lrg} be the largest integer in $[\ell]$, such that $|I_{q_{\mathsf{lrg}}} \cap A| = k$. Let r_{sml} be the smallest integer in $r \in [\ell]$, such that $r_{\mathsf{sml}} > q_{\mathsf{lrg}}$ and $|I_r \cap (T \cup B)| \geq k$. Note that r_{sml} exists as $T = I_\ell$. As $|I_{q_{\mathsf{lrg}}} \cap A| = k$, $|I_{r_{\mathsf{sml}}} \cap (T \cup B)| = k$, and $A \cap (T \cup B) = \emptyset$, there must exist $j \in [\ell]$, such that $q_{\mathsf{lrg}} < j < r_{\mathsf{sml}}$ and $|I_j \cap (T \cup B)| = \lceil k/2 \rceil$. As $k \geq 2$, $I_j \cap (T \cup B) \neq \emptyset$, and thus there is some $v \in I_j \cap (T \cup B)$. Also, $|I_j| = k+1$, and thus, I_j must contain at least $k + 1 - \lceil k/2 \rceil \geq k/2$ vertices from $S \cup A$. Recall that by the construction of G', $\{s,v\} \in E(G')$, for each $s \in S$. From the above we can obtain that $I_j \cap S = \emptyset$. Recall that each vertex in T is adjacent to every vertex in $S \cup A$. From the above discussions we can obtain that $X = I_j \cap A$ has at least $k/2$ vertices and $Y = I_j \cap B$ has at least $\lceil k/2 \rceil$ vertices. By the construction of G', we can obtain that $G[X \cup Y]$ is a biclique in G, and thus we can obtain that $\mathsf{opt\text{-}biclq}(G) \geq k/2$. ◀

We are now ready to prove Theorem 1.

Proof of Theorem 1. Towards a contradiction, suppose that there is an algorithm $\mathcal{A}$, for Bipartite Token Jumping running in time $f(k)n^{\mathcal{O}(1)}$. Using $\mathcal{A}$, we design an algorithm $\mathcal{B}$, running in time $f(k)n^{\mathcal{O}(1)}$, which given a bipartite graph G and an integer k, returns 1 if $\mathsf{opt\text{-}biclq}(G) \geq k$, and returns 0 if $\mathsf{opt\text{-}biclq}(G) < k/2$, thus contradicting Proposition 4.

Our algorithm $\mathcal{B}$ is as follows. Given an instance (G, k) of Maximum Balanced Biclique, $\mathcal{B}$ constructs (in polynomial time) an instance (G', S, T) of Bipartite Token Jumping, as discussed previously. The algorithm $\mathcal{B}$ calls $\mathcal{A}$ on the input (G', S, T), and returns 1 if and only if $\mathcal{A}$ returns 1 on the Bipartite Token Jumping instance.

We now argue the correctness of our algorithm. We first argue that whenever $\mathsf{opt\text{-}biclq}(G) \geq k$, then $\mathcal{B}$ returns 1. From Lemma 5, whenever $\mathsf{opt\text{-}biclq}(G) \geq k$, then (G', S, T) is a yes-instance of Bipartite Token Jumping. Thus $\mathcal{A}$, and hence $\mathcal{B}$, must return 1. Notice that the contrapositive of Lemma 6 gives us that whenever $\mathsf{opt\text{-}biclq}(G) < k/2$, then (G', S, T) is a no-instance of Bipartite Token Jumping. Thus, $\mathcal{A}$ and hence, $\mathcal{B}$ must return 0 for the above case. This concludes the proof. ◀

4 Refuting FPT Algorithms for Bipartite Half (Induced-)Subgraph

The objective of this section is to prove Theorem 2. We begin with the following simple observation that will be useful in "encoding" an instance Maximum Balanced Biclique as instances of Bipartite Half Induced-Subgraph (resp. Bipartite Half Subgraph).

▶ **Observation 7.** *For any $k \in \mathbb{N}$, the half graph $T_{2k,2k}$ contains $K_{k,k}$ as an induced subgraph.*

Proof. Recall that, for any $k \in \mathbb{N}$, $T_{2k,2k}$ is the graph with vertex set $V(T_{2k,2k}) = A_{2k} \cup B_{2k}$, where $A_{2k} = \{a_1, a_2, \cdots, a_{2k}\}$ and $B_{2k} = \{b_1, b_2, \cdots, b_{2k}\}$, and for $i, j \in [2k]$, $\{a_i, b_j\} \in E(T_{2k,2k})$ if and only if $j \geq i$. From the above we can deduce that, for any $i \in [k]$ and $j \in [2k] \setminus [k]$, we have $\{a_i, b_j\} \in E(T_{k,k})$. The above implies that $T_{2k,2k}[\{a_i \mid i \in [k]\} \cup \{b_j \mid j \in [2k] \setminus [k]\}]$ is isomorphic to $K_{k,k}$. ◀

We will use the above observation to encode (approximately) the instance of Maximum Balanced Biclique into an instance of Bipartite Half (Induced-)Subgraph. We remark that the same construction will work for both Bipartite Half Subgraph and Bipartite Half Induced-Subgraph. Before moving further we introduce a definition and an algorithmic result about it, that will be used in our reduction.

▶ **Definition 8.** For $n, \ell \in \mathbb{N}$, a family $\mathcal{F}$ of functions from $[n]$ to $[\ell]$ is an (n, ℓ)-*perfect hash family* if for every $S \subseteq [n]$ of size ℓ, there exists $f \in \mathcal{F}$, such that for each $i \in [\ell]$, $|S \cap \{j \in [n] \mid f(j) = i\}| = 1$.

▶ **Proposition 9** ([34]). *For any $n, \ell \in \mathbb{N}$, we can construct an (n, ℓ)-prefect hash family of size $e^{\ell} \cdot \ell^{\mathcal{O}(\log \ell)} \cdot \log n$ in time $e^{\ell} \cdot \ell^{\mathcal{O}(\log \ell)} \cdot n \log n$.*

We will next intuitively explain our (Turing) reduction below. Consider an instance (G, k) of Maximum Balanced Biclique, where G is a bipartite graph with bipartition A and B. We will begin by "color-coding" a copy of $K_{k,k}$ in G (if it exists). We remark that the technique of color-coding was introduced by Alon et al. [1], for designing FPT algorithms parameterized by the solution size, for detecting cycles and paths of a given length. Also, the above was derandomized (in the same paper), by using the construction of perfect-hash families based on the result of Schmidt and Siegel [36], and an improved algorithm for computing such families was obtained by Naor et al. [34] (see Proposition 9). The idea of color coding is to distinctly color the vertices in the copy of $K_{k,k}$ present (as a subgraph/induced subgraph) in G. Using the above, we obtain a partition of A into k sets $A_1, A_2, \cdots, A_k$, and also a partition of B into k sets, $B_1, B_2, \cdots, B_k$, so that the vertices of a fixed $K_{k,k}$ in G picks exactly one vertex from each A_is and B_is. After this, using one more copy of the sets A_is and B_is, to be denoted by A'_is and B'_is, respectively, we create an instance of Bipartite Half (Induced-)Subgraph, by providing extra vertices/edges using the copies. We remark that the "loss" in our reduction comes from the situation where a solution for the Bipartite Half (Induced-)Subgraph instance may pick two copies of the same vertex.

We will now move to the formal description of the reduction. Consider an instance (G, k) of Maximum Balanced Biclique, where G is a bipartite graph with bipartition $A = \{a_1, a_2, \cdots, a_{n_1}\}$ and $B = \{b_1, b_2, \cdots, b_{n_2}\}$, and let $n = n_1 + n_2$. We let $A' = \{a'_i \mid i \in [n_1]\}$ and $B' = \{b'_i \mid i \in [n_2]\}$, copies of A and B, respectively. Using Proposition 9, we begin by computing (n_1, k)- and (n_2, k)-prefect hash families, $\mathcal{F}_1$ and $\mathcal{F}_2$, respectively. For $f \in \mathcal{F}_1$ and $i \in [k]$, we let $A[f, i] = \{a_j \in A \mid f(j) = i\}$ and $A'[f, i] = \{a'_j \in A' \mid f(j) = i\}$. Similarly, for $f \in \mathcal{F}_2$ and $i \in [k]$, we let $B[f, i] = \{b_j \in B \mid f(j) = i\}$ and $B'[f, i] = \{b'_j \in B' \mid f(j) = i\}$. We will use the convention that, for a vertex $v \in A \cup B$, its corresponding copy in $A' \cup B'$ will be denoted by v'.

For each pair of functions $f_1 \in \mathcal{F}_1$ and $f_2 \in \mathcal{F}_2$, we create a Bipartite Half (Induced-)Subgraph instance $I_{f_1,f_2} = (G_{f_1,f_2}, 2k)$, where G_{f_1,f_2} is a bipartite graph as follows. We let $V(G_{f_1,f_2}) = A \cup B \cup A' \cup B'$. We will now describe the edge set of G_{f_1,f_2}. First we add all the edges in G, between A and B, to G_{f_1,f_2}, i.e., for each $a \in A$ and $b \in B$, such that $\{a, b\} \in E(G)$, we add $\{a, b\}$ to $E(G_{f_1,f_2})$. (Intuitively speaking, the objective of the above edges will be to directly preserve, in G_{f_1,f_2}, the $K_{k,k}$s of G, if they exist.) Roughly speaking, next we will add edges, that will be useful in completing the missing vertices/edges in a half graph (using the copies A' and B'), apart from those that are already present in the biclique. For each $i \in [k]$ and $j \in [k]$, where $j \geq i$, and for each $a \in A[f_1, i]$ and $b \in B[f_2, j]$, we add $\{a, b'\}$ to $E(G_{f_1,f_2})$ if and only if $\{a, b\} \in E(G)$. Similarly, for each $i \in [k]$ and $j \in [k]$, where $j \geq i$, and for each $a \in A[f_1, i]$ and $b \in B[f_2, j]$, we add $\{a', b\}$ to $E(G_{f_1,f_2})$ if and only if $\{a, b\} \in E(G)$. This completes the description of G_{f_1,f_2}, and thus the Bipartite Half (Induced-)Subgraph instance $I_{f_1,f_2} = (G_{f_1,f_2}, 2k)$. We let $\mathcal{I} = \{I_{f_1,f_2} \mid f_1 \in \mathcal{F}_1 \text{ and } f_2 \in \mathcal{F}_2\}$.

In the following two lemmata we will establish some useful relationships between the Maximum Balanced Biclique instance (G, k) and the instances of Bipartite Half (Induced-)Subgraph that we have created.

▶ **Lemma 10.** *If (G, k) is a yes-instance of Maximum Balanced Biclique, then there are $f_1 \in \mathcal{F}_1$ and $f_2 \in \mathcal{F}_2$, such that I_{f_1,f_2} is a yes-instance of Bipartite Half (Induced-)Subgraph.*

Proof. Suppose that $\mathsf{opt\text{-}biclq}(G) \geq k$, and consider $X = \{x_1, x_2, \cdots, x_k\} \subseteq A$ and $Y = \{y_1, y_2, \cdots, y_k\} \subseteq B$, such that $G[X \cup Y]$ is isomorphic to $K_{k,k}$. As $\mathcal{F}_1$ is an (n_1, k)-perfect hash family, there must exist $f_1^* \in \mathcal{F}_1$, such that for each $i \in [k]$, $|A[f_1^*, i] \cap X| = |A'[f_1^*, i] \cap X| = 1$. Similarly, there exists $f_2^* \in \mathcal{F}_2$, such that for each $i \in [k]$, $|B[f_2^*, i] \cap Y| = |B'[f_2^*, i] \cap Y| = 1$. Without loss of generality, we will assume that for $i \in [k]$, we have $A[f_1^*, i] \cap X = \{x_i\}$ and $B[f_2^*, i] \cap X = \{y_i\}$. (Notice that the above can be achieved by renaming of the vertices.) Let $X^* = \{x_i \mid i \in [k]\} \cup \{x'_i \mid i \in [k]\}$ and $Y^* = \{y_i \mid i \in [k]\} \cup \{y'_i \mid i \in [k]\}$, $G_{f_1^*,f_2^*}[X^* \cup Y^*]$. The construction of $G_{f_1^*,f_2^*}$ and the assumption that $G[X \cup Y]$ is isomorphic to $K_{k,k}$, implies that $G_{f_1^*,f_2^*}[X^* \cup Y^*]$ is isomorphic to $T_{2k,2k}$. This concludes the proof. ◀

▶ **Lemma 11.** *If there are functions $f_1 \in \mathcal{F}_1$ and $f_2 \in \mathcal{F}_2$, such that $(G_{f_1,f_2}, 2k)$ is a yes-instance of* Bipartite Half (Induced-)Subgraph*, then* $\mathsf{opt\text{-}biclq}(G) \geq k/2$.

Proof. Consider $f_1 \in \mathcal{F}_1$ and $f_2 \in \mathcal{F}_2$, such that $(G_{f_1,f_2}, 2k)$ is a yes-instance of Bipartite Half (Induced-)Subgraph. Let $X^* \subseteq A \cup A'$ and $Y^* \subseteq B \cup B'$ be each of size $2k$, such that $G_{f_1,f_2}[X^* \cup Y^*]$ contains $T_{2k,2k}$ as an (induced) subgraph.[4] From Observation 7, we can obtain that there is $\widehat{X} \subseteq X^* \subseteq A \cup A'$ and $\widehat{Y} \subseteq Y^* \subseteq B \cup B'$, each of size k, such that $G_{f_1,f_2}[\widehat{X} \cup \widehat{Y}]$ is isomorphic to $K_{k,k}$. We construct two sets $X \subseteq A$ and $Y \subseteq B$ as follows, each initialised to $\emptyset$. For each $a \in A$, we add a to X, if $\{a, a'\} \cap \widehat{X} \neq \emptyset$. Similarly, for each $b \in B$, we add b to Y, if $\{b, b'\} \cap \widehat{Y} \neq \emptyset$. Note that the sizes of X and Y, each, must be at least $k/2$. Using the above together with the construction of G_{f_1,f_2} can be used to obtain that $G[X \cup Y]$ is isomorphic to $K_{\lceil k/2 \rceil, \lceil k/2 \rceil}$. Thus we conclude that $\mathsf{opt\text{-}biclq}(G) \geq k/2$. ◀

Now we are ready to prove Theorem 2.

Proof of Theorem 2. Towards a contradiction, suppose that there is an algorithm $\mathcal{A}$, for Bipartite Half (Induced-)Subgraph running in time $f(k)n^{\mathcal{O}(1)}$, where n is the number of vertices in the input graph. Using $\mathcal{A}$, we design an algorithm $\mathcal{B}$, running in time $g(k) \cdot n^{\mathcal{O}(1)}$ (where g is a computable function), which given a bipartite graph G on n vertices and an integer k, returns 1 if $\mathsf{opt\text{-}biclq}(G) \geq k$, and returns 0 if $\mathsf{opt\text{-}biclq}(G) < k/2$, thus contradicting Proposition 4.

Our algorithm $\mathcal{B}$ is as follows. Given an instance (G, k) of Maximum Balanced Biclique, the algorithm $\mathcal{B}$ starts by constructing the family of Bipartite Half (Induced-)Subgraph instance $\mathcal{I}$, as discussed previously, in time bounded by $2^{\mathcal{O}(k \log k)} \cdot n^{\mathcal{O}(1)}$ time (see Proposition 9). Next, for each $I \in \mathcal{I}$, $\mathcal{B}$ calls $\mathcal{A}$ for the input I of Bipartite Half (Induced-)Subgraph. The algorithm $\mathcal{B}$ return 1 if and only if $\mathcal{A}$ returns 1 for at least one instance $I \in \mathcal{I}$.

The construction of the set $\mathcal{I}$, together with the assumed running time of $\mathcal{A}$ implies that the running time of $\mathcal{B}$ can be bounded by $f(k) \cdot 2^{\mathcal{O}(k \log k)} \cdot n^{\mathcal{O}(1)}$. We now argue the correctness of our algorithm $\mathcal{B}$.

We first argue that whenever $\mathsf{opt\text{-}biclq}(G) \geq k$, then $\mathcal{B}$ returns 1. From Lemma 10, whenever $\mathsf{opt\text{-}biclq}(G) \geq k$, then there must exist a yes-instance of Bipartite Half (Induced-)Subgraph $I \in \mathcal{I}$. Thus, in the above case, $\mathcal{B}$ always returns 1, as required. Notice that the contrapositive of Lemma 11 gives us that whenever $\mathsf{opt\text{-}biclq}(G) < k/2$, then there is no $I \in \mathcal{I}$, such that I is a yes-instance of Bipartite Half (Induced-)Subgraph, and thus, $\mathcal{B}$ will necessarily output 0 for this case. This concludes the proof. ◀

5 Conclusion

Assuming Gap-ETH, we showed that the Bipartite Token Jumping and Bipartite Half (Induced-)Subgraph problems do not admit FPT algorithms. Our results are obtained by appropriate reductions from the Maximum Balanced Biclique problem, and then exploiting the known FPT-inapproximability result of [9] under Gap-ETH, to show that our problems cannot have an FPT algorithms. The natural open problems that arise from our results (and also from the previous works) is obtaining fixed parameter intractability results for these problems under standard and more well-believed conjectures like FPT $\neq$ W[t], where $t \in \mathbb{N} \setminus \{0\}$.

[4] We assume $T_{2k,2k}$ is an induced subgraph of $G_{f_1,f_2}[X^* \cup Y^*]$ when we are considering the problem Bipartite Half Induced-Subgraph, whereas, we assume that it is a subgraph when we are considering the problem Bipartite Half Subgraph.

We believe that our (negative) result regarding Bipartite Half (Induced-)Subgraph maybe a useful starting point for obtaining newer results, similar to its counterparts, like Clique and Biclique. Again we note that not many (parameterized) reductions are known, which refute FPT algorithms under conjectures like Gap-ETH, one of the exceptions to the above is the result of [4]. We believe that newer conjectures like Gap-ETH maybe useful in obtaining newer (and perhaps easier) fixed parameter intractability results, and our reduction serves as a (yet another) evidence for it.

References

1 Noga Alon, Raphael Yuster, and Uri Zwick. Color-coding. *Journal of the ACM*, 42(4):844–856, 1995.

2 Valentin Bartier, Nicolas Bousquet, Clément Dallard, Kyle Lomer, and Amer E. Mouawad. On girth and the parameterized complexity of token sliding and token jumping. In *31st International Symposium on Algorithms and Computation (ISAAC)*, volume 181, pages 44:1–44:17, 2020.

3 Valentin Bartier, Nicolas Bousquet, Clément Dallard, Kyle Lomer, and Amer E. Mouawad. On girth and the parameterized complexity of token sliding and token jumping. *Algorithmica*, 83(9):2914–2951, 2021.

4 Arnab Bhattacharyya, Suprovat Ghoshal, Karthik C. S., and Pasin Manurangsi. Parameterized intractability of even set and shortest vector problem from gap-eth. In *45th International Colloquium on Automata, Languages, and Programming (ICALP)*, volume 107, pages 17:1–17:15, 2018.

5 Marthe Bonamy and Nicolas Bousquet. Token sliding on chordal graphs. In *International Workshop on Graph-Theoretic Concepts in Computer Science*, pages 127–139. Springer, 2017.

6 Nicolas Bousquet, Arnaud Mary, and Aline Parreau. Token jumping in minor-closed classes. In *International Symposium on Fundamentals of Computation Theory*, pages 136–149. Springer, 2017.

7 Andreas Brandstädt, Peter L Hammer, Vadim V Lozin, et al. Bisplit graphs. *Discrete Mathematics*, 299(1-3):11–32, 2005.

8 Parinya Chalermsook, Marek Cygan, Guy Kortsarz, Bundit Laekhanukit, Pasin Manurangsi, Danupon Nanongkai, and Luca Trevisan. From gap-eth to fpt-inapproximability: Clique, dominating set, and more. In *58th IEEE Annual Symposium on Foundations of Computer Science (FOCS)*, pages 743–754, 2017.

9 Parinya Chalermsook, Marek Cygan, Guy Kortsarz, Bundit Laekhanukit, Pasin Manurangsi, Danupon Nanongkai, and Luca Trevisan. From gap-exponential time hypothesis to fixed parameter tractable inapproximability: Clique, dominating set, and more. *SIAM Journal on Computing*, 49(4):772–810, 2020.

10 Stephen A Cook. The complexity of theorem-proving procedures. In *Proceedings of the third annual ACM symposium on Theory of computing (STOC)*, pages 151–158, 1971.

11 Erik D Demaine, Martin L Demaine, Eli Fox-Epstein, Duc A Hoang, Takehiro Ito, Hirotaka Ono, Yota Otachi, Ryuhei Uehara, and Takeshi Yamada. Polynomial-time algorithm for sliding tokens on trees. In *International Symposium on Algorithms and Computation*, pages 389–400. Springer, 2014.

12 Reinhard Diestel. *Graph Theory, 4th Edition*, volume 173 of *Graduate texts in mathematics*. Springer, 2012.

13 Irit Dinur. Mildly exponential reduction from gap 3sat to polynomial-gap label-cover. *ECCC*, 23:128, 2016.

14 Rodney G Downey and Michael R Fellows. Fixed-parameter intractability. In *1992 Seventh Annual Structure in Complexity Theory Conference*, pages 36–37. IEEE Computer Society, 1992.

15 Rodney G Downey and Michael R Fellows. *Fundamentals of parameterized complexity*, volume 4. Springer, 2013.

16 Paul Erdös and András Hajnal. Chromatic number of finite and infinite graphs and hypergraphs. *Discrete Mathematics*, 53:281–285, 1985.

17 Eli Fox-Epstein, Duc A Hoang, Yota Otachi, and Ryuhei Uehara. Sliding token on bipartite permutation graphs. In *International Symposium on Algorithms and Computation*, pages 237–247. Springer, 2015.

18 John E Hopcroft and Richard M Karp. An $n^{5/2}$ algorithm for maximum matchings in bipartite graphs. *SIAM Journal on computing*, 2(4):225–231, 1973.

19 Takehiro Ito, Erik D Demaine, Nicholas JA Harvey, Christos H Papadimitriou, Martha Sideri, Ryuhei Uehara, and Yushi Uno. On the complexity of reconfiguration problems. *Theoretical Computer Science*, 412(12-14):1054–1065, 2011.

20 Takehiro Ito, Marcin Kamiński, and Hirotaka Ono. Fixed-parameter tractability of token jumping on planar graphs. In *International Symposium on Algorithms and Computation*, pages 208–219. Springer, 2014.

21 Takehiro Ito, Marcin Kaminski, Hirotaka Ono, Akira Suzuki, Ryuhei Uehara, and Katsuhisa Yamanaka. On the parameterized complexity for token jumping on graphs. In *Theory and Applications of Models of Computation - 11th Annual Conference (TAMC)*, volume 8402, pages 341–351, 2014.

22 Marcin Kamiński, Paul Medvedev, and Martin Milanič. Complexity of independent set reconfigurability problems. *Theoretical computer science*, 439:9–15, 2012.

23 Richard M Karp. Reducibility among combinatorial problems. In *Complexity of computer computations*, pages 85–103. Springer, 1972.

24 Dénes Kőnig. Graphok és alkalmazásuk a determinánsok és a halmazok elméletére. *Mathematikai és Természettudományi Ertesito*, 34:104–119, 1916.

25 Subhash Khot and Venkatesh Raman. Parameterized complexity of finding subgraphs with hereditary properties. *Theoretical Computer Science*, 289(2):997–1008, 2002.

26 Stefan Kratsch, Shaohua Li, Dániel Marx, Marcin Pilipczuk, and Magnus Wahlström. Multi-budgeted directed cuts. In *13th International Symposium on Parameterized and Exact Computation (IPEC)*, volume 115, pages 18:1–18:14, 2018.

27 Stefan Kratsch, Marcin Pilipczuk, Ashutosh Rai, and Venkatesh Raman. Kernel lower bounds using co-nondeterminism: Finding induced hereditary subgraphs. In *Algorithm Theory - SWAT*, volume 7357, pages 364–375, 2012.

28 Bingkai Lin. The parameterized complexity of k-biclique. In *Proceedings of the Twenty-Sixth Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 605–615, 2015.

29 Bingkai Lin. The parameterized complexity of the k-biclique problem. *Journal of the ACM (JACM)*, 65(5):1–23, 2018.

30 Bingkai Lin and Yijia Chen. The parameterized complexity of k-edge induced subgraphs. In *International Colloquium on Automata, Languages, and Programming*, pages 641–652. Springer, 2012.

31 Daniel Lokshtanov and Amer E Mouawad. The complexity of independent set reconfiguration on bipartite graphs. *ACM Transactions on Algorithms (TALG)*, 15(1):1–19, 2018.

32 Daniel Lokshtanov, Amer E Mouawad, Fahad Panolan, MS Ramanujan, and Saket Saurabh. Reconfiguration on sparse graphs. *Journal of Computer and System Sciences*, 95:122–131, 2018.

33 Dániel Marx. Parameterized complexity and approximation algorithms. *Computer Journal*, 51(1):60–78, 2008.

34 Moni Naor, Leonard J. Schulman, and Aravind Srinivasan. Splitters and near-optimal derandomization. In *36th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 182–191, 1995.

35 Naomi Nishimura. Introduction to reconfiguration. *Algorithms*, 11(4):52, 2018.

36 Jeanette P Schmidt and Alan Siegel. The spatial complexity of oblivious k-probe hash functions. *SIAM Journal on Computing*, 19(5):775–786, 1990.