A Linear Kernel for Independent Set Reconfiguration in Planar Graphs

Nicolas Bousquet
Université Claude Bernard Lyon 1, France
CNRS – Université de Montréal CRM – CNRS, Canada

Daniel W. Cranston
Department of Computer Science, Virginia Commonwealth University, Richmond, VA, USA

Abstract

Fix a positive integer r, and a graph G that is $K_{3,r}$-minor-free. Let I_s and I_t be two independent sets in G, each of size k. We begin with a "token" on each vertex of I_s and seek to move all tokens to I_t, by repeated "token jumping", removing a single token from one vertex and placing it on another vertex. We require that each intermediate arrangement of tokens again specifies an independent set of size k. Given G, I_s, and I_t, we ask whether there exists a sequence of token jumps that transforms I_s into I_t. When k is part of the input, this problem is known to be PSPACE-complete. But it was shown by Ito, Kamiński, and Ono [20] to be fixed-parameter tractable. That is, the problem can be solved in time $f(k) \cdot P(n)$, for some function f and polynomial P, where n denotes the order of G.

Here we strengthen the upper bound on the running time in terms of k by showing that the problem has a kernel of size linear in k. More precisely, we transform an arbitrary input problem on a $K_{3,r}$-minor-free graph (for some fixed positive integer r) into an equivalent problem on a ($K_{3,r}$-minor-free) graph with order $O(k)$. This answers positively a question of Bousquet, Mouawad, Nishimura, and Siebertz [13] and improves the recent quadratic kernel of Cranston, Mühlenthaler, and Peyrille [14]. For planar graphs, we further strengthen this upper bound to get a kernel of size at most $42k$.

2012 ACM Subject Classification Mathematics of computing → Discrete mathematics

Keywords and phrases Reconfiguration, Independent Set, Kernel, Planar graphs

Digital Object Identifier 10.4230/LIPIcs.STACS.2026.19

Related Version *Full Version*: `https://arxiv.org/abs/2506.03319` [10]

Funding *Nicolas Bousquet*: The author has been partly supported by ANR project ENEDISC (ANR-24-CE48-7768-01).

1 Introduction

The *combinatorial reconfiguration* framework aims at investigating algorithmic and structural aspects of the solution space of an underlying *base problem*. For example, given an instance of some problem Π along with two feasible solutions I_s and I_t, called the *source* and *target* feasible solutions, our goal is to determine if (and in how few steps) we can transform the source into the target via a sequence of adjacent feasible solutions. Such a sequence is called a *reconfiguration sequence* and every step in the sequence (going from one solution to an adjacent one) is called a *reconfiguration step*. Reconfiguration problems arise in various fields such as combinatorial games, motion of robots, random sampling, and enumeration. This framework has been extensively studied for various rules and types of problems in the last twenty years. The surveys [28, 29, 13] give a more complete overview of the field. In this paper we focus on transformation between independent sets; this study was initiated in [18] and motivated by planning motion of robots [19].

43rd International Symposium on Theoretical Aspects of Computer Science (STACS 2026).
Editors: Meena Mahajan, Florin Manea, Annabelle McIver, and Nguyễn Kim Thắng
Article No. 19; pp. 19:1–19:18
Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

Given a simple undirected graph G, a subset of $V(G)$ is *independent* if it induces no edges. Finding an independent set of maximum cardinality, i.e., the Independent Set problem, is a fundamental problem in algorithmic graph theory; it is known to be NP-hard. Furthermore, this problem is not approximable within a factor of $O(n^{1-\varepsilon})$, for any $\varepsilon > 0$, unless P = NP [31].

We view an independent set as a collection of tokens placed on vertices, where no two tokens are adjacent. We can thus naturally define adjacency relations between independent sets, also called *reconfiguration steps*. In this paper, we focus on the Token Jumping (TJ) model, introduced by Kamiński et al. [23], where a single reconfiguration step consists of first removing a token on some vertex v and then immediately adding it back on any other vertex w, as long as no two tokens become adjacent. The token *jumps* from vertex v to vertex w. In the Token Jumping Independent Set Reconfiguration problem (abbreviated as ISR-TJ), we are given a graph G and two independent sets I_s and I_t of G. Our goal is to determine whether there exists a sequence of reconfiguration steps (called a *reconfiguration sequence*) that transforms I_s into I_t.

The ISR-TJ problem has received considerable attention. Hearn and Demaine proved that the problem is PSPACE-complete even restricted to planar graphs of maximum degree 3 [18] and Wrochna proved that it remains PSPACE-complete even restricted to graphs of bounded bandwidth [30]. On the positive side, this problem can be decided in polynomial time in certain restricted graph classes, such as line graphs and some of their extensions [9, 6] and even-hole free graphs (for this latter class, [23] gave a linear-time algorithm). The hardness of the problem motivates studying the problem from a parameterized perspective.

Parameterized Algorithms

A problem Π is *FPT* (fixed-parameter tractable) parameterized by a parameter x if there exists a function f and a polynomial P such that for every instance $\mathcal{I}$ of Π of size n for which parameter x has value k, the problem can be decided in time $f(k) \cdot P(n)$. One can easily prove that the existence of an FPT algorithm is equivalent to the existence of a *kernel* of size $f(k)$ (for a function f), which is an algorithm that provides in polynomial time an *equivalent instance*[1] of size $f(k)$. A kernel is *polynomial* if f is a polynomial function.

Independent Set is known to be $W[1]$-complete [15] but admits FPT algorithms on $K_{r,r}$-free graphs and even semi-ladder-free graphs [16]. Its reconfiguration counterpart, ISR-TJ also is W[1]-hard[2] parameterized by the size k of the independent sets [21]. One can naturally wonder what is happening when we restrict to structured graph classes. In [20], Ito et al. showed that the ISR-TJ problem is FPT parameterized by k on planar graphs, and even on $K_{3,r}$-free graphs (for each fixed value of r), i.e., graphs containing no copy of $K_{3,r}$ as a subgraph. This result has been generalized to nowhere dense graphs [26] and $K_{r,r}$-free graphs [12]. For a thorough history of parameterized aspects of independent set reconfiguration, we refer the reader to [13].

Although the parameterized behavior of ISR-TJ is now well understood, its kernelization counterpart remains largely unexplored. Even beyond ISR-TJ, a deep understanding of kernelization aspects of reconfiguration problems remains elusive (see the end of the present section). Although it was not mentioned in [20], their proof can be easily adapted to get a polynomial size kernel; for instance, see [12]. In a recent survey paper, Bousquet, Mouawad,

---

[1] That is an instance which is positive if and only if the original instance is positive.

[2] Under some standard algorithmic assumptions, a W[1]-hard problem does not admit any FPT algorithm.

Nishimura, and Siebertz [13] asked whether ISR-TJ admits a linear kernel for all planar graphs. Cranston, Mühlenthaler, and Peyrille [14] proposed a quadratic kernel, even for graphs embedded on surfaces. In this paper, we answer affirmatively the question of [13] in a strong sense: we prove that ISR-TJ admits a linear kernel when, for any positive integer r, we restrict to $K_{3,r}$-minor-free graphs.

▶ **Theorem 1.** *For each positive integer r,* ISR-TJ *admits a kernel of size linear in k on $K_{3,r}$-minor-free graphs.*

The special case of planar graphs is of particular interest, so there we make the bound more precise.

▶ **Theorem 2.** *For planar graphs,* ISR-TJ *admits a kernel of size at most $42k$.*

The general approach that we use to prove each of these theorems is the same. But for planar graphs we work harder to optimize the multiplicative constant. We note that the kernel we output is actually a subgraph of the initial graph. This is, as far as we know, the first linear kernel for a reconfiguration problem. As a direct byproduct, we obtain an algorithm running in $2^{O(k)} \cdot Poly(n)$ to decide ISR-TJ. As far as we know, it is the first non-trivial single exponential algorithm for a reconfiguration problem. (At the end of this section, we discuss in more detail kernelization algorithms for other reconfiguration problems.)

To obtain a kernel that is linear, our argument must be more global than for the kernels in [20, 14]. Their proofs are based on a neighborhood decomposition argument. Let I_s and I_t be the source and target independent sets and let $X := I_s \cup I_t$. They prove in [14] that the number of classes of neighborhoods in X is linear using the neighborhood diversity of planar graphs. It is easy to show that classes with at most 1 or at least 3 neighbors in X have linear size in total. The hardest part of the proof consists in reducing the size of classes with 2 neighbors in X. In [12, 20, 14], the proofs show that each class can be individually reduced to a linear number of vertices (in k). To get a linear kernel, we instead use a more subtle global approach, which is different from previous methods. We prove that we can keep only a constant number of vertices in each class and a linear number (in k) of well-chosen vertices in *the union of the* 2*-classes* and still preserve the existence of a solution. As far as we know, this is the first time that such a global argument has been used for reconfiguration.

In Sections 2.1 and 2.2, we explain in more detail the key ingredients needed to prove our Main General Theorem. And in Section 2.3 we complete its proof.

In the second part of this paper (Section 3), we explain how we can improve the kernel size in the specific case of planar graphs, to get a kernel of size $42k$. This bound is probably still far from optimal, so we tried to compromise between a simple analysis and a small kernel size. As far as we know, no lower bound on kernel sizes has been proposed for reconfiguration. In the ISR-TJ PSPACE-hardness proof of Hearn and Demaine [18, Theorem 23], the graph obtained in the reduction has size $3k$ and it seems very complicated to compact in polynomial time, so $3k$ appears to be a natural candidate for a lower bound. (In fact, their proof is written for ISR-TS, but because their reduction uses only *maximum* independent sets, it works equally well for ISR-TJ.) Proving a non-trivial lower bound, or improving further the upper bound to a single digit constant, remains a challenging open problem.

The existence of a polynmial kernel for ISR-TJ on graph classes beyond $K_{3,r}$-minor free graphs remains wide open. The proof of Bousquet, Mary, and Parreau [12] for $K_{r,r}$-free graphs generalizes the method of [20], but yields a kernel of size $k^{f(r)}$ where f is an exponential function. We thus ask the following question.

▶ **Question 3.** *Does* ISR-TJ *admit a kernel of size $f(r) \cdot Poly(k)$ on bounded treewidth graphs? K_r-minor free graphs? On $K_{r,r}$-free graphs?*

Related work: Kernelization and Reconfiguration

While the existence (and non-existence) of FPT algorithms for reconfiguration problems has recently been widely studied, almost no polynomial kernels have been proposed. Mouawad et al. [27] proved that TJ-Vertex Cover Reconfiguration and TJ-Feedback Vertex Set Reconfiguration both admit quadratic kernels. But the existence of linear kernels for these problems is still open. In the specific case of vertex cover, we note that this contrasts with the optimization setting, where in recent decades numerous classical linear kernels for vertex cover have been discovered; see e.g. [1].

As far as we know, our result gives the first non-trivial linear kernel for a reconfiguration problem. While the existence of a linear kernel for independent set is trivial in the optimization setting (every planar graph on at least $4k$ vertices is a yes-instance), the proof of Section 2 requires significant work. Many meta-kernelization algorithms guarantee the existence of linear kernels on planar graphs for optimization problems [2, 17], e.g., for dominating sets. While it remains open to determine whether TJ-DOMINATING SET RECONFIGURATION admits a linear kernel, several reduction rules of [2] cannot be adapted directly for reconfiguration.

Important machinery has been developed to prove that problems do not admit polynomial kernels, even if they admit FPT algorithms (AND and OR compositions for instance). As far as we know, this framework has never been used for reconfiguration problems. In particular, no problem is known to be FPT and also to not admit polynomial kernels. Finding such a problem, or finding a problem that admits a polynomial kernel for the optimization setting but not for its reconfiguration counterpart, remains interesting and open.

Finally, in this paper we focus on Token Jumping. Another model, called Token Sliding (TS), has been studied. In the *TS model*, tokens can only move along edges of the graph. Both problems remain PSPACE-complete on planar graphs and on graphs of bounded bandwidth [18, 30], but their complexities differ on many graph classes, such as chordal graphs [8] and bipartite graphs [25], where the sliding model is harder than its jumping counterpart. From a parameterized viewpoint, very little is known for the problem ISR-TS. While ISR-TJ is known to be FPT even on $K_{r,r}$-free graphs, the parameterized complexity of ISR-TS is open even on graphs of bounded treewidth. ISR-TS is known to be FPT on planar graphs [5] and on graphs with constraints on the girth [3, 4]. But the existence of polynomial kernels for ISR-TS remains wide open. For dominating sets, the sliding version is much harder than its jumping counterpart since DSR-TS is XL-complete even on bounded treewidth graphs, but an FPT algorithm exists for DSR-TJ on planar graphs [11].

2 Main General Theorem

2.1 Proof Overview and Key Ideas

This section is devoted to proving our Main General Theorem. Fix a positive integer r. Fix an input graph G with no $K_{3,r}$-minor, along with source and target independent sets, I_s and I_t, each of size k. We will show that either ISR-TJ(G, I_s, I_t) is a trivial yes-instance, or else ISR-TJ(G, I_s, I_t) is equivalent to a problem ISR-TJ(G', I_s, I_t), where G' is a subgraph of G and $|V(G')| = O(k)$. Let $X := I_s \cup I_t$ and note that $|X| \leq 2k$. The set X is called the set of *key vertices*. For each $Y \subseteq X$, the *X-projection class* $\mathcal{C}_Y$ is defined by $\mathcal{C}_Y := \{v \in V(G) \setminus X \text{ s.t. } N(v) \cap X = Y\}$; the vertices of Y are the *key vertices of* $\mathcal{C}_Y$. Let

$$\mathcal{C}_1 := \bigcup_{\substack{Y \subseteq X \\ |Y| \leq 1}} \mathcal{C}_Y \quad \text{and} \quad \mathcal{C}_2 := \bigcup_{\substack{Y \subseteq X \\ |Y| = 2}} \mathcal{C}_Y \quad \text{and} \quad \mathcal{C}_3 := \bigcup_{\substack{Y \subseteq X \\ |Y| \geq 3}} \mathcal{C}_Y.$$

For each integer k, we say that the X-projection class is a *k-class* if $|Y| = k$. In other words, $\mathcal{C}_1$ and $\mathcal{C}_2$ are the unions respectively of the 1-classes (and 0-class) and of the 2-classes, and $\mathcal{C}_3$ consists of all the other classes. So $V(G) = X \cup \mathcal{C}_1 \cup \mathcal{C}_2 \cup \mathcal{C}_3$. Recall that $|X| \leq 2k$.

We will first bound the sizes of $\mathcal{C}_1$ and $\mathcal{C}_3$, in Section 2.2. As already observed in [20], if $|\mathcal{C}_1| \geq \chi(G) \cdot k$, then ISR-TJ$(G, I_s, I_t)$ is a yes-instance. However, computing $\chi(G)$ is hard. So we will work with a parameter $\chi_{up}(G)$ such that always $\chi(G) \leq \chi_{up}(G)$ and we can efficiently construct a $\chi_{up}(G)$-coloring of G. (For example, for planar graphs we let $\chi_{up}(G) := 4$ and for general $K_{3,r}$-minor-free graphs, we let $\chi_{up}(G) := r + 2$.) Thus, we assume $|\mathcal{C}_1| \leq \chi_{up}(G) \cdot k$. Let $N_2(G) := |\{Y \subseteq X : |Y| = 2 \text{ and } \mathcal{C}_Y \neq \emptyset\}|$, and let $N_3(G) := |\{Y \subseteq X : |Y| \geq 3 \text{ and } \mathcal{C}_Y \neq \emptyset\}|$. Since G has no $K_{3,r}$-minor, for each Y with $|Y| \geq 3$, we have $|\mathcal{C}_Y| \leq r - 1$. So $|\mathcal{C}_3| \leq (r-1)N_3(G)$. Thus, we aim to show that $N_3(G) = O(k)$. Proving this is easy for all planar graphs (and, more generally, for all graphs on each fixed surface), as we will show in Lemma 8. The proof for all $K_{3,r}$-minor-free graphs is more challenging [22, 7]; but the result still holds (see Corollary 10).

So the core of the proof consists in showing that we can reduce the size of $\mathcal{C}_2$. If we could show that $|\mathcal{C}_2| = O(k)$, then we could take G as its own kernel. We cannot prove this directly, although it is true that $N_2(G)$ has size $O(k)$. Yet, unsurprisingly, the size of a class $\mathcal{C}_Y$ with $|Y| = 2$ is in general not bounded by a constant. But can a 2-class be very large, say $\omega(k^2)$?

To motivate our approach in the remainder of the paper, we now sketch a crucial idea. Suppose that a 2-class $\mathcal{C}_Y$ is very large for some 2-element set Y (large enough to necessarily contain an independent set of size kr). If it is impossible to ever, eventually, move a token to $\mathcal{C}_Y$ starting from I_s, then we can delete all of $\mathcal{C}_Y$ without changing whether we can reconfigure I_s into I_t, since no vertices of $\mathcal{C}_Y$ can be used in a reconfiguration sequence. The same is true if it is impossible to ever eventually move a token to $\mathcal{C}_Y$ starting from I_t. So we assume that neither of these is impossible.

Since $\mathcal{C}_Y$ is very large, it contains an independent set I_Y of size kr. Since G is $K_{3,r}$-minor-free, each vertex other than the two key vertices of $\mathcal{C}_Y$ has at most $r-1$ neighbors in $\mathcal{C}_Y$ and, in particular, in I_Y. By assumption, starting from I_s we can eventually move some token to $\mathcal{C}_Y$. In the resulting independent set I', both tokens have moved off the vertices of Y (moving the first of these tokens off is called *unlocking* $\mathcal{C}_Y$), so each of the k vertices with a token has at most $r-1$ neighbors in I_Y. Thus, I_Y contains at least $|I_Y| - k(r-1) = k$ vertices that are not adjacent to any vertex in I'. So we can move all tokens on vertices in I' to these available vertices of I_Y (in arbitrary order). The same is true starting from I_t. Finally, we can move tokens freely within I_Y, since it is an independent set. So we can reconfigure I_s to I_t. This argument succeeds whenever $|\mathcal{C}_Y| \geq \chi_{up}(G)kr$, since that guarantees an independent set I_Y of size kr (the largest color class in a $\chi_{up}(G)$-coloring of $G[\mathcal{C}_Y]$). Thus, whenever $|\mathcal{C}_Y| > \chi_{up}(G)kr$, we can delete arbitrary vertices of $\mathcal{C}_Y$.

Following this approach ensures[3] that $|\mathcal{C}'_2| \leq \chi_{up}(G)krN_2(G) = O(k^2)$. Below we adapt this idea to ensure that $|\mathcal{C}'_2| \leq \max\{O(\chi_{up}(G)kr), O(N_2(G))\}$. The main new idea is that the independent set I_Y above can be spread over multiple 2-classes. So if we unlock a 2-class containing vertices of I_Y, then we use its available vertices to receive tokens from the *next* 2-class containing vertices of I_Y, unlocking that one and proceeding by induction. Informally, G' is formed from G by deleting some vertices of certain "big" 2-classes. Formally, we defer constructing G' (and defining "big" 2-classes) to Construction 13. But once we define G' we can prove the next lemma, which is the core of proving our Main General Theorem.

---

[3] When G is $K_{3,r}$-minor-free, it is straightforward to show that $\chi(G) = O(r)$. In fact, when $r \geq 6300$ Kostochka and Prince [24] showed that G is $(r+2)$-degenerate; thus, $\chi(G) \leq r + 3$. See Lemma 5.

▶ **Lemma 4.** *If in G we can (a) start from I_s and unlock some big 2-class and also (b) start from I_t and unlock some big 2-class, then* $\mathrm{ISR\text{-}TJ}(G', I_s, I_t)$ *is equivalent to* $\mathrm{ISR\text{-}TJ}(G, I_s, I_t)$*, and both of them are* yes*-instances.*

With Lemma 4 (formalized in Lemma 18), we prove our Main General Theorem as follows.

Proof of the Main General Theorem. We show $\mathrm{ISR\text{-}TJ}(G', I_s, I_t)$ and $\mathrm{ISR\text{-}TJ}(G, I_s, I_t)$ are equivalent. Since $G' \subseteq G$, if the latter is a no-instance, then so is the former. So assume instead that $\mathrm{ISR\text{-}TJ}(G, I_s, I_t)$ is a yes-instance. First, suppose that it is impossible, starting from I_s, to ever unlock a big 2-class. So all big 2-classes will always remain locked, and it is impossible to ever move a token to a vertex of a big 2-class. Thus, since every vertex of $V(G) \setminus V(G')$ is in a big 2-class, every reconfiguration sequence in G, starting from I_s is also valid in G'. This proves the desired result. The argument is identical if it is imposible to unlock a big 2-class, starting from I_t. Thus, we assume instead that, starting from both I_s and I_t, it is possible to unlock some big 2-class. Now we are done by Lemma 4. ◀

2.2 Dealing with $\mathcal{C}_1$ and $\mathcal{C}_3$

In this subsection, we determine a function f such that if G is $K_{3,r}$-minor-free, then we can assume that $|\mathcal{C}_1| + |\mathcal{C}_3| \leq f(r)k$. And when G is planar, we can improve our bound on f. Bounding $|\mathcal{C}_1|$ is easy, both in the planar case and in the more general case. But bounding $|\mathcal{C}_3|$ is more work. For this we use the observation (Lemma 7) that $|\mathcal{C}_3| \leq (r-1)N_3(G)$; recall here that $N_3(G)$ denotes the number of sets $Y \subseteq X$ with $|Y| \geq 3$ and $\mathcal{C}_Y \neq \emptyset$. To bound $N_3(G)$ in the general (non-planar) case, we use a powerful result (Lemma 9) from [7].

It is straightforward to prove that if G is $K_{3,r}$-minor-free, then $\chi(G) = O(r)$. But determining the right multiplicative (and additive) constant is more work. This is done by the following lemma, which is sharp.

▶ **Lemma 5** (Kostochka–Prince [24])**.** *Fix $r \geq 6300$. If G is an n-vertex graph with $n \geq r+3$ and G has no $K_{3,r}$-minor, then $2|E(G)| \leq (r+3)(n-2)+2$. Thus, G is $(r+2)$-degenerate and $\chi(G) \leq r+3$.*

▶ **Lemma 6.** *If $|\mathcal{C}_1| \geq \chi_{up}(G)k$, then* $\mathrm{ISR\text{-}TJ}(G, I_s, I_t)$ *is a* yes*-instance. In particular, since G is $K_{3,r}$-minor-free, this is true whenever $r \geq 6300$ and $|\mathcal{C}_1| \geq k(r+3)$.*

Proof. The second statement follows from the first by Lemma 5; thus, we prove the first.

Assume $|\mathcal{C}_1| \geq \chi_{up}(G)k$. By definition, G is $\chi_{up}(G)$-colorable, and we can compute such a coloring efficiently. By Pigeonhole, $\mathcal{C}_1$ contains an independent set I_m (for middle) of size $\chi_{up}(G)k/\chi_{up}(G) = k$. Starting with tokens on I_s, for each $v \in I_s$ with a neighbor $w_v \in I_m$, move the token on v to some such w_v. Now move all remaining tokens (in an arbitrary order) to the unoccupied vertices of I_m. By symmetry, we can also move all tokens from I_t to I_m. Thus, we have a yes-instance of $\mathrm{ISR\text{-}TJ}(G, I_s, I_t)$, as claimed. ◀

Henceforth we assume $|\mathcal{C}_1| < \chi_{up}(G)k$. Recall from above that $N_3(G)$ denotes the number of sets $Y \subseteq X$ with $|Y| \geq 3$ and $\mathcal{C}_Y \neq \emptyset$. The next lemma follows directly from the fact that G is $K_{3,r}$-minor-free.

▶ **Lemma 7.** $|\mathcal{C}_3| \leq (r-1)N_3(G)$*.*

Proof. Suppose the lemma is false. By Pigeonhole, there exists $Y \subseteq X$ with $|\mathcal{C}_Y| \geq \lceil |\mathcal{C}_3|/N_3(G) \rceil > (r-1)N_3(G)/N_3(G)$. That is, $|\mathcal{C}_Y| \geq r$. But now G contains the subgraph $K_{3,r}$ with the vertices in the part of size 3 in Y and those in the part of size r in $\mathcal{C}_Y$. This contradicts that G is $K_{3,r}$-minor-free. ◀

When G is planar, we can use Euler's formula to improve the bound above.

▶ **Lemma 8.** *If G is planar, then $N_2(G) \leq 3|X| \leq 6k$ and $N_3(G) \leq 2|X| \leq 4k$.*

Proof. We draw a plane graph G_X with vertex set X where each set $Y \subseteq X$ with $|Y| = 2$ and $C_Y \neq \emptyset$ corresponds to an edge of G_X. Think of restricting G to X and one vertex v_Y in C_Y for each such Y with $|C_Y| = 2$ (deleting any edges among vertices of X). For each v_Y, we now contract exactly one of its two incident edges. Note that for the resulting plane graph G_X its number of edges is precisely $|N_2(G)|$, the number of 2-classes of G. By Euler's Formula, G_X has at most $3|X| - 6$ edges, so $N_2(G) \leq 3|X| \leq 6k$.

For every class C in $\mathcal{C}_3$, we choose one representative x_C (recall that $x_C \notin X$). We denote by p the number of such classes, i.e., $p := |N_3(X)|$. We now construct the bipartite graph with parts X and $\{x_C : C \in \mathcal{C}_3\}$; that is, we consider the subgraph induced by these vertices, but with all edges removed that have both endpoints inside X or have both endpoints inside $\{x_C : C \in \mathcal{C}_3\}$. The resulting graph is indeed a bipartite planar graph. By Euler's formula, the number of edges of a bipartite planar graph is at most twice its number of vertices. So we have $3p < 2(p + |X|) \leq 2(p + 2k)$, which implies that $p < 2|X| \leq 4k$, as claimed. ◀

The *neighborhood complexity* of a graph class $\mathcal{G}$ is the smallest function f, if it exists, such that for all $G \in \mathcal{G}$, nonempty $A \subseteq V(G)$, and nonnegative integers r, we have the bound $|\{N^s[v] \cap A : v \in V(G)\}| \leq f(s)|A|$. Here $N^s[v]$ is the set of vertices at distance at most s from v. We need the following result.

▶ **Lemma 9** ([7, Theorem 18])**.** *For all positive integers s, t with $t \geq 4$, for every K_t-minor-free graph G, for every set A of vertices of G,*

$$|\{N^s[v] \cap A : v \in V(G)\}| \leq 4^t (t-3) t^{2(t-1)} (s+1)^{3(t-1)} |A|.$$

Since we are interested only in neighborhoods (that is, distance 1), we let $s := 1$. Since G is $K_{3,r}$-minor-free, it is also K_{3+r}-minor-free. So we let $t := r + 3$. Finally, we let $A := X$ and recall that $|A| = |X| \leq 2k$.

▶ **Corollary 10.** $N_2(G) + N_3(G) \leq 4^{r+3} r (r+3)^{2r+4} 2^{3r+6} (2k) \leq 2^{5r+13} (r+3)^{2r+5} k.$

Combining the results in this subsection, we get the following.

▶ **Lemma 11.** $|\mathcal{C}_1| + |\mathcal{C}_3| \leq k(\max\{r, 6300\} + 3 + (r-1)(2^{5r+13}(r+3)^{2r+5}))$. *If G is planar, then $|\mathcal{C}_1| + |\mathcal{C}_3| \leq 12k$.*

Proof. We start with the first statement. If G is $K_{3,r}$-minor-free, with $r \leq 6300$, then also G is $K_{3,6300}$-minor-free. So the bound on $|\mathcal{C}_1|$ follows from Lemma 6, and the bound on $|\mathcal{C}_3|$ follows from Lemma 7 and Corollary 10. Summing these bounds gives the first statement.

Now we prove the second statement. By Lemma 6, we assume that $|\mathcal{C}_1| \leq 4k$. By Lemma 7 (with $r := 3$) and Lemma 8 we get that $|\mathcal{C}_3| \leq (3-1)(4k) = 8k$. Summing these bounds gives the second statement. ◀

2.3 Bounding the size of $\mathcal{C}_2$

The rest of the proof consists in showing that the following lemma holds.

▶ **Lemma 12.** *We can find in polynomial time an equivalent instance, formed by possibly deleting some vertices of $\mathcal{C}_2$, to get a subset $\mathcal{C}_2'$ for which $|\mathcal{C}_2'| \leq \chi_{up}(G)(N_2(G)(4r-1)+k)$.*

First we construct our graph G', by (possibly) deleting some vertices of $\mathcal{C}_2$, and we show that $|\mathcal{C}_2'| \leq \chi_{up}(G)(N_2(G)(4r-1)+k)$. This is fairly straightforward. Afterwards, we show that ISR-TJ(G', I_s, I_t) is equivalent to ISR-TJ(G, I_s, I_t). We sketched this latter step above. So all that remains for us is to handle the harder case: when it is possible starting from I_s to move a token to some vertex of $\mathcal{C}_2 \setminus \mathcal{C}_2'$, and this is also possible starting from I_t.

▶ **Construction 13.** *To form G' from G, we do the following.*

(1) *If $|\mathcal{C}_2| \leq \chi_{up}(G)(N_2(G)(3r-2)+k)$, then do nothing; that is, $\mathcal{C}_2' := \mathcal{C}_2$.*

(2) *Otherwise, by Pigeonhole pick $I \subseteq \mathcal{C}_2$ such that I is an independent set and $|I| = N_2(G)(3r-2)+k$.*

(3) *A 2-class $\mathcal{C}_Y$ is* big *if $|\mathcal{C}_Y| \geq \chi_{up}(G)(2r-1)+1$; otherwise, $\mathcal{C}_Y$ is* small.

(4) *For each small 2-class, do nothing.*

(5) *For each big 2-class $\mathcal{C}_Y$, do the following.*

  (a) *If $\mathcal{C}_Y$ has at least $3r-1$ vertices of I, then delete all vertices of $\mathcal{C}_Y \setminus I$.*

  (b) *If $\mathcal{C}_Y$ has at most $3r-2$ vertices of I, then:*
  Keep in $\mathcal{C}_Y$ an arbitrary independent set of size $2r$ and remove all other vertices of $\mathcal{C}_Y$.
  Remove all the vertices of $I \cap \mathcal{C}_Y$ from I.

▶ **Proposition 14.** *We have $|\mathcal{C}_2'| \leq \chi_{up}(G)(N_2(G)(4r-1)+k)$.*

Proof. If $|\mathcal{C}_2| \leq \chi_{up}(G)(N_2(G)(4r-1)+k)$, then we are done, trivially. So assume not. Now we define I and delete vertices of big 2-classes as in Construction 13. If a 2-class $\mathcal{C}_Y$ is either small or intersects I in at most $3r-2$ vertices, then in $\mathcal{C}_2'$, we keep at most $\chi_{up}(G)(2r-1)$ vertices of $\mathcal{C}_Y$. Thus, the total number of vertices in these classes (restricted to G') is at most $N_2(G)\chi_{up}(G)(2r-1)$.

If a 2-class $\mathcal{C}_Y$ is big and intersects I in at least $3r-1$ vertices, then in G' we keep in $\mathcal{C}_Y$ only its vertices in I. Thus, the total number of vertices in these classes (restricted to G') is at most $|I| \leq N_2(G)(3r-2)+k$. So the total size of $\mathcal{C}_2'$ is at most $N_2(G)\chi_{up}(G)(2r-1) + N_2(G)(3r-2)+k \leq \chi_{up}(G)(N_2(G)(4r-1)+k)$. ◀

A *helpful independent set* is any subset of I of size k. Since I is independent, the following is clear.

▶ Remark 15. Any helpful independent set can be transformed into any other.

For a 2-class $\mathcal{C}_Y$, we call the 2 vertices in Y the *key vertices* of $\mathcal{C}_Y$. To *unlock* a big 2-class $\mathcal{C}_Y$ is to move tokens to reach an independent set I' such that $|I' \cap N(\mathcal{C}_Y)| \leq 1$. After we unlock a class $\mathcal{C}_Y$, we can move the single token in $N(\mathcal{C}_Y)$, if it exists, onto $\mathcal{C}_Y$ and then move additional tokens onto $\mathcal{C}_Y$ (provided that $\mathcal{C}_Y$ contains a large enough independent set).

▶ **Lemma 16.** *Fix $x, x' \in X$ and let C be the $\{x, x'\}$-class. Every component of $G[V \setminus (C \cup \{x, x'\})]$ is adjacent to at most $r-1$ vertices of C.*

Proof. If not, then G contains a $K_{3,r}$-minor, where one side consists of the vertices x, x', and the component A with r neigbhbors in C, and the other side consists of r vertices of $N(A) \cap C$. ◀

▶ Remark 17. Having defined G' and bounded its size, all that remains is to prove that this new instance ISR-TJ(G', I_s, I_t) is equivalent to the original ISR-TJ(G, I_s, I_t). This equivalence is precisely the assertion of Lemma 4, and it follows immediately from Lemma 18.

▶ **Lemma 18.** *If in G we can from I_s (resp. I_t) unlock a big class, then in G' we can from I_s (resp. I_t) reach a helpful independent set. That is, there exists a transformation from I_s (resp. I_t) into a helpful independent set in G that only uses vertices of G'.*

Proof. Assume that there exists an independent set J_0' that can be reached from I_s such that $|N(C) \cap J_0'| = 1$ for some big class C. Among all transformations from I_s to J_0', take a transformation $\mathcal{R}$ of minimum length. The case when $\mathcal{R}$ has length 0 (that is $J_0' = I_s$) is easier, so we handle it briefly at the end. For now we assume that $\mathcal{R}$ has positive length.

We claim that: (i) the last step of $\mathcal{R}$ consists in moving a token on a key vertex x of class C to some vertex z and, (ii) each jump of the transformation except the last one consists of moving a token (from its current vertex) to an adjacent vertex. Point (i) follows from the minimality of the transformation. At some step in $\mathcal{R}$, we move a token away from a key vertex of some big 2-class. If $\mathcal{R}$ continues with further steps, then we can omit these steps, contradicting the minimality of $\mathcal{R}$. Point (ii) holds because if, prior to the last step in $\mathcal{R}$, we moved a token from a vertex v to a vertex w, with w not adjacent to v, then we should have instead moved the token on the key vertex x to w; this gives a shorter transformation, again contradicting the minimality of $\mathcal{R}$.

We form G'' from G by deleting all vertices in big classes and all key vertices for big classes. Now (ii) above implies that all vertices that have gained or lost a token during $\mathcal{R}$ must be in the same component of G''; otherwise we can simply omit from $\mathcal{R}$ all moves in components of G'' other than the component where we move our token on our final move, which unlocks C.

We claim that each big class C' has at most $2(r-1)$ vertices with neighbors in $J_0' \setminus I_s$, as follows. Let z denote the vertex where we move a token on the final step of $\mathcal{R}$ (unlocking C). Note that all vertices of $J_0' \setminus (I_s \cup \{z\})$ belong to the same component of G''; this follows from the previous paragraph, since each of these vertices received a token during $\mathcal{R}$ (excluding its last step). Lemma 16 ensures that each big class C' has at most $r-1$ vertices with neighbors in $J_0' \setminus (I_s \cup \{z\})$. Lemma 16 also ensures that each big 2-class C' has at most $r-1$ neighbors of z. Thus, the claim holds.

By the definition of J_0', there is a big 2-class C that has been unlocked; that is J_0' only contains a token on at most one of the two key vertices of C. Regardless of whether or not C contained (in G) at least $3r-1$ vertices of I, we know that C contains in G' an independent set of size at least $2r$. From above, at most $2r-2$ of these vertices have neighbors in $J_0' \setminus I_s$. So at least $2r-(2r-2) = 2$ of them have no such neighbor. We denote by $\{a,b\}$ an independent set of size 2 in $C \setminus N(J_0' \setminus I_s)$.

By Lemma 16, each of a, b has at most $r-1$ neighbors in each other big class C'. Actually, let x, y denote the key vertices for the big 2-class C that contains a, b. For each big class C', with $C' \neq C$, one of x and y is not a key vertex for C'; by symmetry, we assume x is not. If $|N(\{a,b\}) \cap C'| \geq r$, then we get a $K_{3,r}$-minor by contracting $\{a,b\}$ onto x (with x and the two key vertices of C' as the part of size 3). Thus, $|N(\{a,b\}) \cap C'| \leq r-1$. So for each class C' containing at least $3r-1$ vertices of I, we know that $|(C' \cap I) \setminus N(\{a,b\} \cup (J_0' \setminus X))| \geq (3r-1)-(r-1)-(r-1)-(r-1) = 2$. We denote by I' the set $I \setminus N(\{a,b\})$. Note in particular that by construction I' contains at least 2 vertices in each big 2-class that contains (in G') vertices of I.

Let $C_1, \dots, C_\ell$ be the big classes intersecting I in G' (different from C if C also intersects I). Let b be a vetex of $I' \cap C_1$ and x_1 be a key vertex of C_1. Note that no vertex of J_0' is adjacent to b. So we can move the token[4] on x_1 to b to reach an independent set that we denote by J_0. Note that C_1 is unlocked in J_0 (since $|N(C_1) \cap X| = 1$.) We can now unlock all the big classes $C_1, \dots C_i, \dots, C_\ell$ intersecting I in G', by induction.

---

[4] If $x_1 = x$, then we have already moved the token that was on x_1, so we now do nothing.

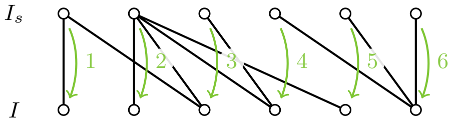

Figure 1 An I_s-greedy independent set, together with the transformation ↷ .

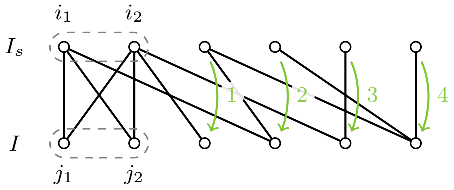

Figure 2 A weakly I_s-greedy independent set. ⬚ are the activation pairs and ↷ is the tranformation.

At the end of this transformation, we get an independent set J_ℓ where every big class containing vertices of I has been unlocked. Moreover, by construction, vertices of $J_\ell \setminus I$ have at most $3r-3$ neighors in each big class. When constructing G' from G, we deleted vertices of I only in step (5b). So the number of vertices of I we deleted is at most $N_2(G)(3r-2)$. Thus, the total number of vertices in I (in G) that are unavailable in G' to receive tokens from J_ℓ is at most $N_2(G)(3r-2)$. So the number of vertices available to receive tokens, from J_ℓ is at least $|I| - N_2(G)(3r-2) = k$. Hence, in G' all the tokens of J_ℓ can be moved to I, as desired.

Now we remark briefly on the case that $\mathcal{R}$ has length 0; that is, some big 2-class C is already unlocked. In this case, we just begin immediately moving tokens to a, b in C. Now each big 2-class C' has at most $r-1$ vertices in $N(\{a,b\})$, but we do not need to worry about neighbors of z in the component of G'' where other moves occurred. So the analysis above still holds. ◀

3 Improved Kernel for Planar Graphs

3.1 Proof Outline

Now we provide an algorithm that outputs a smaller kernel in the specific case of planar graphs. More precisely, the goal of Section 3 is to prove the following result.

▶ **Theorem 19.** *On planar graphs* ISR-TJ *admits a kernel of size* $42k$*.*

The general idea of the proof is similar to that of Lemma 18. We prove that if G is large enough, then G contains an independent set I of size at least k with the following property: if we can unlock one of the "big classes" from I_s (or I_t), then we can transform I_s (or I_t) into a size k subset of I. To obtain a smaller kernel, we need two main ingredients. First, we give a more subtle way to define this independent set I. This allows us to find such an independent set I in planar graphs much smaller than required by Lemma 18. Second, we prove that, as in Lemma 18, if we can unlock one class that is big enough, then we can transform I_s into I. The difference from Lemma 18 is that our new notion of "big enough" is actually much smaller; but this savings comes at the cost of slightly more involved analysis.

We now explain in more detail how we find this set I and describe some of its properties. Let $X := I_s \cup I_t$. Let Y be a subset of vertices. The *X-neighborhood of Y* is $\cup_{y\in Y} N(y) \cap X$. One of the key ideas in the proof is the concept of (weakly) greedy independent sets. Let I be an independent subset of $V \setminus X$ consisting of vertices in 2-classes.

We say that I is *I_s-greedy* (resp. I_t-greedy) if there is a greedy algorithm that moves tokens (one-by-one) from the vertices of I_s (resp. I_t) onto vertices of I, while keeping an independent set all throughout the transformation; see Figure 1. To rephrase, this means that, at each step of the transformation, we can find a vertex of $I_s \setminus I$ that can be replaced by a vertex of $I \setminus I_s$. Again equivalently, but in a more structural way, there is an ordering $i_1, \ldots, i_k$ of I_s and $j_1, \ldots, j_k$ of I such that, for every $t \leq k$, vertices $j_1, \ldots, j_t, i_{t+1}, \ldots, i_k$ form an independent set. Note, if I_1 and I_2 are both independent sets of size k, that I_1 is I_2-greedy if and only if I_2 is I_1-greedy. That is, being greedy is symmetric.

We will also need the following weakening of I_s-greedy independent sets. The set I is *weakly I_s-greedy* if there exist vertices $i_1, i_2 \in I_s$ and $j_1, j_2 \in I$ such that $(I_s \setminus \{i_1, i_2\}) \cup \{j_1, j_2\}$ is an independent set, call it I', and I is I'-greedy; see Figure 2. In other words, there are orderings $i_1, \ldots, i_k$ of I_s and $j_1, \ldots, j_k$ of I such that, for every $t \in \{2, \ldots, k\}$, the vertex subset $\{j_1, \ldots, j_t, i_{t+1}, \ldots, i_k\}$ is independent. Note that an I_s-greedy independent set is indeed weakly I_s-greedy. If I is weakly I_s-greedy, then the pair $\{i_1, i_2\}$ (resp. $\{j_1, j_2\}$) is called the *I_s-activation pair* (resp. I-activation pair). These pairs "activate" the transformation in the sense that, if $\{i_1, i_2\}$ has been replaced by $\{j_1, j_2\}$, then we can greedily finish the transformation from I_s into I. (In the rest of the proof, I might have size larger than k, since we simply want to transform I_s into a subset of I; but imagining that these set sizes are equal keeps all the hardness of the problem.)

Assume that G contains an independent set I of size at least k that is weakly I_s-greedy and weakly I_t-greedy. If I is I_s-greedy and I_t-greedy, then we can transform I_s into I_t, passing through I. But if I is only weakly I_s-greedy, then nothing ensures that we can transform I_s into I. Nevertheless, by definition, if we can replace the activation vertices of I_s with the activation vertices of I (and similarly for I_t), then we can transform I_s into I. But (i) there might not exist any transformation between I_s and I and, (ii) if there is a transformation, nothing guarantees that some such transformation satisfies this condition.

To overcome point (ii), we exhibit certain special weakly greedy independent sets (called weakly clean independent sets) in Section 3.3. And we also prove that if G is large enough, then either there is no transformation from I_s into I or there is a transformation that replaces the activation pair of I_s by the activation pair of I without moving the other tokens. Moreover, in the latter case, this transformation only uses a constant number of vertices in each 2-class, as well as (possibly) vertices in classes that are not 2-classes. (We optimize this constant knowing that the graph is planar.) Finally, we prove that every planar graph that is large enough has an independent set I that is both weakly I_s-greedy and weakly I_t-greedy; this completes the proof.

Organization

In Section 3.2, we start with a few observations. In Section 3.3, we define clean independent sets, and prove that if G is large enough, then it contains an independent set I that is clean both for I_s and for I_t. In the longer version [10] we combine all these arguments to get the desired smaller kernel, assuming the truth of a key lemma about transformations of I_s into I; finally, we conclude by proving this key lemma.

3.2 First Observations

When constructing a sequence to reconfigure I_s to I_t, we would prefer to be able to move tokens onto vertices of distinct 2-classes independently of each other. But this may be impossible, because of edges between some of these vertices. So this first subsection is about how we can allow ourselves this desired freedom. Throughout this section, we extensively use the following simple remark; it is due to the fact that all vertices belonging to a given 2-class are adjacent to the same two key vertices.

▶ Remark 20. If G is a planar graph, then the vertices of each 2-class C induce either a cycle or a disjoint union of paths.

Proof. If some vertex v in C has at least 3 neighbors in C, say w_1, w_2, w_3, then G contains as a subgraph $K_{3,3}$, with $\{w_1, w_2, w_3\}$ in one part and with v and the key vertices for C in the other. This $K_{3,3}$ contradicts that G is planar. And if C induces a cycle H, then H separates the key vertices, so no other vertex is adjacent to both key vertices. Thus, H spans C. ◀

In particular, every 2-class C contains an independent set of size $\lfloor |C|/2 \rfloor$. Moreover, if we consider a strict subset of C, that is a set $D \subsetneq C$, then D induces a disjoint union of paths; so D admits an independent set of size at least $\lceil |D|/2 \rceil$. The subset D is typically formed from C by deleting 2 vertices. We usually remove vertices to guarantee that 2-classes are anticomplete to each other (disjoint vertex subsets A and B are *anticomplete* to each other if no edge of G has one endpoint in A and the other in B). Namely, we often use the following remark.

▶ Remark 21. If C, C' are distinct 2-classes, then the following 2 statements hold.

(a) The set $N(C) \cap C'$ has size at most 2. Moreover, if $N(C) \cap C'$ has size 2 then its two elements must be either (i) consecutive vertices on a path of $G[C']$; or (ii) two endpoints of a single path of $G[C']$; or (iii) two endpoints of disjoint paths of $G[C']$.

(b) If x, y are the key vertices of C, then, for every $z \in X$ distinct from x, y, the union of the classes incident to z has at most two neighbors in C. (Otherwise, G has a $K_{3,3}$-minor.)

In the case of planar graphs, we can actually strengthen Remark 21(a) to apply to more than two classes, and we prove the following version.

▶ **Lemma 22.** *Let G be a planar graph. Let $C_1, \ldots, C_r$ be 2-classes. For every $i < r$, there exist $C'_i \subseteq C_i$ with $|C'_i| \geq |C_i| - 2$ such that $C'_1, \ldots, C'_{r-1}, C_r$ are pairwise anticomplete.*

Proof. Consider a plane drawing of the subgraph of G induced by $\cup_{i \leq r} C_i \cup X$ such that the outer face of the drawing contains either the 2 key vertices of C_r or 2 vertices contained in C_r, possibly both. For every class C_i, let G_i denote the subgraph induced by C_i and its 2 key vertices. A class C_i is *nested* in C_j if all the vertices of C_i lie within a face of G_j distinct from its outer face. Two classes C_i and C_j are *incomparable* if the vertices of C_a are on the outer face of G_b whenever $\{a, b\} = \{i, j\}$. For every 2-class C, the vertices of C plus its two key vertices form a $K_{2,|C|}$ (possibly with extra edges among C), so every pair of classes is either nested or incomparable.

Now, for every $i < r$, let B_i denote the at most two vertices of C_i on the boundary of G_i, and let $C'_i := C_i \setminus B_i$. We claim that, C'_i is anticomplete to C'_j for every $j \neq i$ (with $C'_r := C_r$). If the two classes are incomparable, then the only vertices of C_i that can be adjacent to C_j are the vertices of B_i and B_j, but the vertices of at least one of these sets are excluded from $C'_i \cup C'_j$ (and the vertices of both are if $r \notin \{i, j\}$). Otherwise, up to symmetry, C_i is nested in C_j. By the definition of the plane drawing, $i \neq r$ and the vertices of the outer face of C_i are excluded from $\mathcal{C}'_i$. ◀

Later we will need a slight variation of Lemma 22. (Its proof is nearly identical to that above.)

▶ **Lemma 23.** *Let G be a planar graph, and let C_1, C_2 be two 2-classes. If C is the vertex set of a connected subgraph of $G[V \setminus (C_1 \cup C_2 \cup X)]$, then the following 2 statements hold.*

(a) *There exist C'_1, C'_2 such that $|C'_i| \geq |C_i| - 2$ for every $i \in \{1, 2\}$ and C'_1, C'_2, C are pairwise anticomplete.*

(b) *If C_1 and C_2 are anticomplete, then there exists $C' \subseteq C$ such that $|C'| \geq |C| - 4$ and C', C_1, C_2 are pairwise anticomplete.*

3.3 Clean Independent Sets

We define two types of independent sets; these are similar to the helpful independent sets of Section 2. A weakly I_s-greedy independent set I is 3-*clean for* I_s if it satisfies both of the 2 conditions below; see Figure 3.

1. Some 2-class C contains at least three vertices of I and,
2. The vertices of the activation pair[5] of I are in C.

A weakly I_s-greedy independent set I is $(2,2)$-*clean for* I_s if it satisfies both of the 2 conditions below; see Figure 4.

1. Some vertex x is a key vertex for two 2-classes C, C' that both contain two vertices of I and,
2. The vertices of the activation pair of I are the vertices of $C \cap I$.

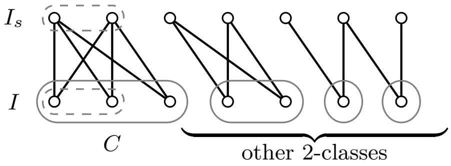

Figure 3 The independent set I is 3-clean for I_s. ⬚ are the activation pairs.

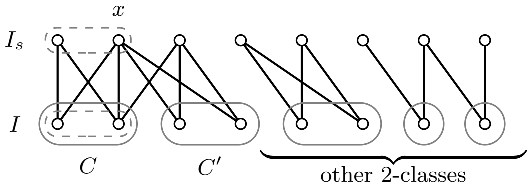

Figure 4 The independent set I is (2,2)-clean for I_s. ⬚ are the activation pairs.

The set I is *weakly clean for* I_s if it is either 3-clean or $(2,2)$-clean. Note that if I is weakly clean for I_s, then the I_s-activation pair is in $N(C) \cap X$. The key vertices of C (and C' for $(2,2)$-clean independent sets) might not be in I_s. But in that case, we can easily transform I_s into I, as we show in the next lemma.

▶ **Lemma 24.** *Let I be a 3-clean (resp. $(2,2)$-clean) independent set for I_s. If at least one of the two vertices of $N(C) \cap X$ (resp. $N(C \cup C') \cap X$) is not in I_s then we can transform I_s into I using only vertices of $I_s \cup I$.*

Proof. Since I is I_s-weakly greedy, we simply need to prove that we can move the tokens from the activation pair of I_s to the activation pair of I. Indeed, if we can do this, then afterward we can complete the transformation, from I_s into I, greedily.

First suppose that I is 3-clean. By definition, the activation pair $\{i_1, i_2\}$ of I is in C and the activation pair $\{j_1, j_2\}$ of I_s contains the vertex j_1 of $N(C) \cap I_s$ (if it exists). We can replace[6] j_1 with i_1 and replace j_2 with i_2 (since $N(C) \cap I_s$ has at most one vertex, namely j_1) and the conclusion follows.

Now instead assume that I is $(2,2)$-clean. If $|N(C) \cap X| \leq 1$, then we can perform the same moves and the conclusion follows similarly. So we assume that the activation pair of I_s is $\{x, j_2\}$, the two vertices of $N(C) \cap X$. By assumption not all the vertices of $N(C \cup C') \cap X$ are in I_s. So the second key vertex of C' is not in I_s. Thus, we can replace x with a vertex y' of $I \cap C'$, replace j_2 with i_1, and finally replace y with i_2. This completes the proof. ◀

---

[5] There might exist several orders, and several activation pairs. In this case, we select one pair (and corresponding order) that satisfies the condition and call this *the* activation pair.

[6] By *replace a with b* we mean to move the token from vertex a to vertex b; this replaces a with b in the independent set defined by the vertices currently with a token.

From now on, by Lemma 24, we assume that the classes C and C' are *locked*, that is $N(C) \cap I_s$ and $N(C') \cap I_s$ each have size 2. In particular, the activation pair of I is $N(C) \cap I_s$. If I is 3-clean (resp. $(2,2)$-clean), then the vertex of $I \cap C$ (resp. the two vertices of $I \cap C'$) that is not in the I-activation pair (if several such vertices exist, then we choose one of them arbitrarily) is called the *auxiliary activation vertex* (resp. vertices) of I. Moreover, if I is $(2,2)$-clean, then the key vertex of C' that is not a key vertex of C is called the *auxiliary activation vertex of* I_s.

An independent set is *clean for* I_s if it is I_s-greedy, 3-clean for I_s, or $(2,2)$-clean for I_s. As we already mentioned, it is *weakly clean* if it is clean but not I_s-greedy. And if I is I_s-greedy, then we can transform I into I_s. We now explain informally how we will make use of an independent set that is 3-clean or $(2,2)$-clean for I_s. We will argue that if we can unlock a large enough class D, then (if we consider a good transformation) we can replace an X-vertex adjacent to C (resp. replace x, when I is $(2,2)$-clean) with a vertex in the class D. This fact, together with the fact that $C \cap I$ (resp. $(C \cup C') \cap I$) is large enough will allow us to find a transformation from I_s into I. We formalize this intuition with the following simple example.

▶ **Lemma 25.** *Let G be a $K_{3,3}$-free graph, and let an independent set I be clean for I_s. If $|V(G) \setminus N(I_s)| \geq 3$, then we can transform I_s into I using at most one vertex that is not in $I_s \cup I_t$.*

Proof. The conclusion follows if the independent set I is I_s-greedy. So we instead assume that I is weakly I_s-greedy. Thus, if we can replace the activation pair of I_s by the activation pair of I, then we can complete the transformation greedily. The rest of the proof consists in showing that we can do this. Let $\{i_1, i_2\}$ be the I-activation pair and i_3 be an auxiliary I-activation vertex (and i_4 be the other if I is $(2,2)$-clean). We denote by C the class containing i_1, i_2 (and by C' the class containing i_3 if I is $(2,2)$-clean). Since G is $K_{3,3}$-free, there is a non-edge between some vertex $a \in V(G) \setminus N(I_s)$ and some vertex i_b of $\{i_1, i_2, i_3\}$.

If I is 3-clean, then we let $\{j_1, j_2\}$ be the activation pair of I_s (these are also the key vertices of C). We replace j_1 with a, replace j_2 with i_b, replace a with some vertex i_c in $\{i_1, i_2\} \setminus i_b$, and finally replace i_b with $\{i_1, i_2\} \setminus i_c$ if $i_b = i_3$.

If I is $(2,2)$-clean, then we let $\{x, j_1\}$ be the I_s-activation pair, and let j_2 be the vertex of $N(C') \cap I_s$ distinct from x. If i_b is in the I-activation pair, then we can conclude as above for 3-clean independent sets. So we assume that $i_b = i_3$. We replace x with a, replace j_2 with i_3, replace a with i_4, replace j_1 with i_1, replace i_3 with i_2, and replacer i_4 with j_2. ◀

To conclude this section, we prove that if G is large enough, then the graph contains an independent set that is clean for both I_s and I_t. Namely, the following holds.

▶ **Lemma 26.** *Let G be a planar graph and I_s, I_t be two independent sets of size k. If the number of vertices in 2-classes is at least $21k$, then there exists an independent set I of size at most $2k$ that is both clean for I_s and clean for I_t.*

Proof. Note that if an independent set I is weakly I_s-greedy (resp. I_t-greedy), then I remains so when we add vertices to I. So it suffices to find an independent set I' of size k that is clean for I_s, find another I'' for I_t, and take their union, as long as this union is also independent. To ensure this union is indeed independent, we first find a large independent set I_0 and choose the clean independent sets I' and I'' from within I_0.

Let $X := I_s \cup I_t$. By Lemma 8, the number of 2-classes in G is at most $3|X| \leq 6k$. For each 2-class C, we call C a *good class for* I_s (resp. *for* I_t) if C has at most one neighbor in I_s. A class that is not good is called *bad*.

By Lemma 22, after removing at most 2 vertices per class, we assume that all the 2-classes are anticomplete to each other. The total number of vertices that we remove is at most $2|N_2(X)| \leq 6|X| \leq 12k$. Moreover, by Remark 20, the remaining vertices of each 2-class induce a disjoint union of paths, so each contains an independent set with at least half its vertices. We denote by I_0 the union of these independent sets; note that I_0 is also independent and $|I_0| \geq (21k-12k)/2 = 4.5k$. As we mentioned above, we choose from among vertices of I_0 an independent set of size k that is clean for I_s, and we do the same for I_t; the union of these two sets is the desired independent set I, with size at most $2k$. So below it suffices only to construct the independent subset of I_0 that is clean for I_s.

If the number of vertices of I_0 appearing in good classes for I_s is at least k, then we take an arbitrary set of k of them; this set is I_s-greedy, so we are done. Thus, we assume instead that the number of vertices of I_0 in good classes is at most $k-1$. Hence, the number of vertices of I_0 that are in bad classes is at least $4.5k-(k-1) = 3.5k+1$.

If more than $0.5k$ bad classes each have at least 2 vertices in I_0, then by Pigeonhole 2 such bad 2-classes C, C' share a key vertex of I_s. To form an independent set that is $(2,2)$-clean for I_s, we take the union of the vertices of I_0 in C, C', and then continue picking a bad class (with at least two vertices of I_0) and adding all of its vertices, until we reach a set of size k. To see that this set is $(2,2)$-clean, we start the ordering with the vertices of $C \cup C'$.

So we assume instead that all but at most $0.5k$ bad 2-classes contain at most 1 vertex of I_0; we call them *small classes*. We denote by ℓ the number of non-small bad classes; thus, $\ell \leq 0.5k$. By the proof of Lemma 8, at most $3k$ classes are bad; so the number of small bad classes is at most $3k-\ell$. Thus, the number of vertices of I_0 in small bad classes is at most $3k-\ell$. So the number of vertices of I_0 in non-small bad classes is at least $3.5k+1-(3k-\ell) = 0.5k+\ell+1 \geq 2\ell+1$, since $\ell \leq 0.5k$. Thus, by Pigeonhole, some bad 2-class C has at least 3 vertices of I_0. To form our 3-clean independent set (for I_s), we take all of the vertices of I_0 in C and add vertices of I_0 from bad 2-classes that are not small, up to a set of size k (or when no remaining bad class has two vertices of I_0).

To order I, we begin with the vertices in a class of size at least 3, and continue to other bad 2-classes, always making all vertices in a bad 2-class successive in the order. If we reach a set of size k, then this independent set I is weakly I_s-greedy, and we are done; so we assume we do not reach a set of size k.

However, we do reach a set I of size at least $0.5k+\ell+1$. So we need to add to I at most $0.5k-\ell-1$ vertices. After moving in the tokens from all key vertices needed to unlock I, these $0.5k+\ell+1$ vertices of I have at least $0.5k+\ell+1-2\ell = 0.5k-\ell+1$ vertices with no token. Thus we can move to them one token from each of $0.5k-\ell-1$ small bad classes. Afterwards, all of these small bad classes are unlocked, so the remaining tokens can move to these vertices. Thus, the resulting set is weakly I_s-greedy. ◀

▶ **Corollary 27.** *In fact, if there exist q 2-classes that each have a key vertex outside of I_s, and the union of these classes has size at least $2q+2k$, then I_s is I-greedy.*

Proof. By Pigeonhole, one of these classes C has size at least $\lceil (2q+2k)/q \rceil \geq 3$. Recall that C is I_s-unlocked, since C has a key vertex outside of I_s. So we can start by replacing the first 3 vertices of I_s with these 3 vertices of C. The proof that we can finish the desired order of I is precisely the final 2 paragraphs of the previous proof. ◀

Due to space constraints, the proof of this statement is not included in the extended abstract but can be found in the long version [10]. The proof that the kernel has the desired size is centered around the following definition and technical lemma. A vertex is *important* if it is a key vertex for either (a) at least one 2-class of size at least 7 or (b) at least two 2-classes of size at least 5.

▶ **Lemma 28.** *Let G be a planar graph and I be an independent set that is weakly clean for I_s. We can transform I_s into an independent subset of I whenever we can unlock from I_s either (a) a 2-class of size at least 7 or (b) a 2-class of size at least 5 with a key vertex adjacent to a second 2-class of size at least 5. Moreover, such a transformation still exists in every subgraph G' formed from G by deleting vertices (outside of I) in 2-classes of size at least 5 such that every vertex important in G remains important in G'.*

References

1 Faisal N. Abu-Khzam, Michael R. Fellows, Michael A. Langston, and W. Henry Suters. Crown structures for vertex cover kernelization. *Theory Comput. Syst.*, 41(3):411–430, 2007. `doi:10.1007/s00224-007-1328-0`.

2 Jochen Alber, Michael R. Fellows, and Rolf Niedermeier. Polynomial-time data reduction for dominating set. *J. ACM*, 51(3):363–384, 2004. `doi:10.1145/990308.990309`.

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. `doi:10.1007/s00453-021-00848-1`.

4 Valentin Bartier, Nicolas Bousquet, Jihad Hanna, Amer E. Mouawad, and Sebastian Siebertz. Token sliding on graphs of girth five. *Algorithmica*, 86(2):638–655, 2024. `doi:10.1007/S00453-023-01181-5`.

5 Valentin Bartier, Nicolas Bousquet, and Amer E. Mouawad. Galactic token sliding. *J. Comput. Syst. Sci.*, 136:220–248, 2023. `doi:10.1016/j.jcss.2023.03.008`.

6 Valentin Bartier, Nicolas Bousquet, and Moritz Mühlenthaler. Independent set reconfiguration in h-free graphs. In Daniel Král and Martin Milanic, editors, *Graph-Theoretic Concepts in Computer Science - 50th International Workshop, WG 2024, Gozd Martuljek, Slovenia, June 19-21, 2024, Revised Selected Papers*, volume 14760 of *Lecture Notes in Computer Science*, pages 35–49. Springer, 2024. `doi:10.1007/978-3-031-75409-8_3`.

7 Laurent Beaudou, Jan Bok, Florent Foucaud, Daniel A. Quiroz, and Jean-Florent Raymond. Profile and neighbourhood complexity of graphs with excluded minors and tree-structured graphs, 2025. `doi:10.48550/arXiv.2501.08895`.

8 Rémy Belmonte, Eun Jung Kim, Michael Lampis, Valia Mitsou, Yota Otachi, and Florian Sikora. Token sliding on split graphs. *Theory Comput. Syst.*, 65(4):662–686, 2021. `doi:10.1007/S00224-020-09967-8`.

9 Paul S. Bonsma, Marcin Kaminski, and Marcin Wrochna. Reconfiguring independent sets in claw-free graphs. In R. Ravi and Inge Li Gørtz, editors, *Algorithm Theory - SWAT 2014 - 14th Scandinavian Symposium and Workshops, Copenhagen, Denmark, July 2-4, 2014. Proceedings*, volume 8503 of *Lecture Notes in Computer Science*, pages 86–97. Springer, 2014. `doi:10.1007/978-3-319-08404-6_8`.

10 Nicolas Bousquet and Daniel W. Cranston. A linear kernel for independent set reconfiguration in planar graphs, 2025. `doi:10.48550/arXiv.2506.03319`.

11 Nicolas Bousquet, Quentin Deschamps, Arnaud Mary, Amer E Mouawad, and Théo Pierron. The tape reconfiguration problem and its consequences for dominating set reconfiguration, 2025. `doi:10.48550/arXiv.2505.00988`.

12 Nicolas Bousquet, Arnaud Mary, and Aline Parreau. Token jumping in minor-closed classes. In *Fundamentals of computation theory*, volume 10472 of *Lecture Notes in Comput. Sci.*, pages 136–149. Springer, Berlin, 2017. `doi:10.1007/978-3-662-55751-8_12`.

13 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, April 2022. `doi:10.48550/arXiv.2204.10526`.

14 Daniel W. Cranston, Moritz Mühlenthaler, and Benjamin Peyrille. A simple quadratic kernel for token jumping on surfaces. In H. Fernau and P. Kindermann, editors, *Graph-Theoretic Concepts in Computer Science: 51st International Workshop, WG 2025, Otzenhausen, Germany, June 11–13, 2025, Revised Selected Papers*, volume 16124 of *Lecture Notes in Computer Science*, pages 149–161, Berlin, Heidelberg, 2026. Springer-Verlag. `doi:10.1007/978-3-032-11835-6_11`.

15 Rodney G. Downey and Michael R. Fellows. Fixed-parameter tractability and completeness I: basic results. *SIAM J. Comput.*, 24(4):873–921, 1995. `doi:10.1137/S0097539792228228`.

16 Grzegorz Fabianski, Michal Pilipczuk, Sebastian Siebertz, and Szymon Torunczyk. Progressive algorithms for domination and independence. In Rolf Niedermeier and Christophe Paul, editors, *36th International Symposium on Theoretical Aspects of Computer Science, STACS 2019, March 13-16, 2019, Berlin, Germany*, volume 126 of *LIPIcs*, pages 27:1–27:16. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2019. `doi:10.4230/LIPIcs.STACS.2019.27`.

17 Jiong Guo and Rolf Niedermeier. Linear problem kernels for NP-hard problems on planar graphs. In Lars Arge, Christian Cachin, Tomasz Jurdzinski, and Andrzej Tarlecki, editors, *Automata, Languages and Programming, 34th International Colloquium, ICALP 2007, Wroclaw, Poland, July 9-13, 2007, Proceedings*, volume 4596 of *Lecture Notes in Computer Science*, pages 375–386. Springer, 2007. `doi:10.1007/978-3-540-73420-8_34`.

18 Robert A. Hearn and Erik D. Demaine. PSPACE-completeness of sliding-block puzzles and other problems through the nondeterministic constraint logic model of computation. *Theoret. Comput. Sci.*, 343(1-2):72–96, 2005. `doi:10.1016/j.tcs.2005.05.008`.

19 J.E. Hopcroft, J.T. Schwartz, and M. Sharir. On the complexity of motion planning for multiple independent objects; PSPACE-hardness of the "warehouseman's problem". *The International Journal of Robotics Research*, 3(4):76–88, 1984. `doi:10.1177/027836498400300405`.

20 Takehiro Ito, Marcin Kamiński, and Hirotaka Ono. Fixed-parameter tractability of token jumping on planar graphs. In *Algorithms and computation*, volume 8889 of *Lecture Notes in Comput. Sci.*, pages 208–219. Springer, Cham, 2014. `doi:10.1007/978-3-319-13075-0_17`.

21 Takehiro Ito, Marcin Kaminski, Hirotaka Ono, Akira Suzuki, Ryuhei Uehara, and Katsuhisa Yamanaka. On the parameterized complexity for token jumping on graphs. In T. V. Gopal, Manindra Agrawal, Angsheng Li, and S. Barry Cooper, editors, *Theory and Applications of Models of Computation - 11th Annual Conference, TAMC 2014, Chennai, India, April 11-13, 2014. Proceedings*, volume 8402 of *Lecture Notes in Computer Science*, pages 341–351. Springer, 2014. `doi:10.1007/978-3-319-06089-7_24`.

22 Gwenaël Joret and Clément Rambaud. Neighborhood complexity of planar graphs. *Comb.*, 2023. `doi:10.1007/s00493-024-00110-6`.

23 Marcin Kaminski, Paul Medvedev, and Martin Milanic. Complexity of independent set reconfigurability problems. *Theor. Comput. Sci.*, 439:9–15, 2012. `doi:10.1016/j.tcs.2012.03.004`.

24 A. V. Kostochka and N. Prince. Dense graphs have $K_{3,t}$ minors. *Discrete Math.*, 310(20):2637–2654, 2010. `doi:10.1016/j.disc.2010.03.026`.

25 Daniel Lokshtanov and Amer E. Mouawad. The complexity of independent set reconfiguration on bipartite graphs. *ACM Trans. Algorithms*, 15(1):7:1–7:19, 2019. `doi:10.1145/3280825`.

26 Daniel Lokshtanov, Amer E. Mouawad, Fahad Panolan, M. S. Ramanujan, and Saket Saurabh. Reconfiguration on sparse graphs. *J. Comput. Syst. Sci.*, 95:122–131, 2018. `doi:10.1016/j.jcss.2018.02.004`.

27 Amer E. Mouawad, Naomi Nishimura, Venkatesh Raman, Narges Simjour, and Akira Suzuki. On the parameterized complexity of reconfiguration problems. *Algorithmica*, 78(1):274–297, 2017. `doi:10.1007/s00453-016-0159-2`.

28 Naomi Nishimura. Introduction to reconfiguration. *Algorithms (Basel)*, 11(4):Paper No. 52, 25, 2018. `doi:10.3390/a11040052`.

29 Jan van den Heuvel. The complexity of change. In *Surveys in combinatorics 2013*, volume 409 of *London Math. Soc. Lecture Note Ser.*, pages 127–160. Cambridge Univ. Press, Cambridge, 2013. `doi:10.1017/CBO9781139506748.005`.

30 Marcin Wrochna. Reconfiguration in bounded bandwidth and tree-depth. *J. Comput. Syst. Sci.*, 93:1–10, 2018. `doi:10.1016/j.jcss.2017.11.003`.

31 David Zuckerman. Linear degree extractors and the inapproximability of max clique and chromatic number. *Theory of Computing*, 3(6):103–128, 2007. `doi:10.4086/toc.2007.v003a006`.