# Minimum Separator Reconfiguration


**Guilherme C. M. Gomes**
Department of Computer Science, Federal University of Minas Gerais, Belo Horizonte, Brazil

**Clément Legrand-Duchesne**[1]
LaBRI, CNRS, Université de Bordeaux, France

**Reem Mahmoud**
Virginia Commonwealth University, Richmond, VA, USA

**Amer E. Mouawad**
Department of Computer Science, American University of Beirut, Beirut, Lebanon

**Yoshio Okamoto**
Graduate School of Informatics and Engineering, The University of Electro-Communications, Chofu, Japan

**Vinicius F. dos Santos**
Department of Computer Science, Federal University of Minas Gerais, Belo Horizonte, Brazil

**Tom C. van der Zanden**
Department of Data Analytics and Digitalisation, Maastricht University, The Netherlands



## Abstract

We study the problem of reconfiguring one minimum $s$-$t$-separator $A$ into another minimum $s$-$t$-separator $B$ in some $n$-vertex graph $G$ containing two non-adjacent vertices $s$ and $t$. We consider several variants of the problem as we focus on both the token sliding and token jumping models. Our first contribution is a polynomial-time algorithm that computes (if one exists) a minimum-length sequence of slides transforming $A$ into $B$. We additionally establish that the existence of a sequence of jumps (which need not be of minimum length) can be decided in polynomial time (by an algorithm that also outputs a witnessing sequence when one exists). In contrast, and somewhat surprisingly, we show that deciding if a sequence of at most $\ell$ jumps can transform $A$ into $B$ is an NP-complete problem. To complement this negative result, we investigate the parameterized complexity of what we believe to be the two most natural parameterized counterparts of the latter problem; in particular, we study the problem of computing a minimum-length sequence of jumps when parameterized by the size $k$ of the minimum $s$-$t$-separators and when parameterized by the number $\ell$ of jumps. For the first parameterization, we show that the problem is fixed-parameter tractable, but does not admit a polynomial kernel unless $\mathsf{NP} \subseteq \mathsf{coNP/poly}$. We complete the picture by designing a kernel with $\mathcal{O}(\ell^2)$ vertices and edges for the length $\ell$ of the sequence as a parameter.



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

**Keywords and phrases** minimum separators, combinatorial reconfiguration, parameterized complexity, kernelization

**Digital Object Identifier** 10.4230/LIPIcs.IPEC.2023.9

**Related Version** *Full Version*: `https://arxiv.org/abs/2307.07782`

**Funding** *Yoshio Okamoto*: JSPS KAKENHI Grant Numbers JP20K11670, JP20H05795, JP23K10982.
*Vinicius F. dos Santos*: FAPEMIG grant APQ-01707-21 and CNPq grants 406036/2021-7 and 312069/2021-9.

**Acknowledgements** This work started during the Combinatorial Reconfiguration Workshop (CoRe 2022) which was hosted at the Banff International Research Station for Mathematical Innovation and Discovery (BIRS), Alberta, Canada, from May 8–13, 2022. We would like to thank everyone who made this collaboration possible and, in particular, the organizers Daniel Cranston, Marthe Bonamy, Moritz Mühlenthaler, Naomi Nishimura, Nicolas Bousquet, Ryuhei Uehara, and Takehiro Ito for their continuous support of the combinatorial reconfiguration community in general.


[1] Corresponding author




18th International Symposium on Parameterized and Exact Computation (IPEC 2023).
Editors: Neeldhara Misra and Magnus Wahlström; Article No. 9; pp. 9:1–9:12
Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

# 1 Introduction

We study the problem of computing reconfiguration sequences between minimum $s$-$t$-separators[2]. A set $S$ of vertices in a graph $G$ is an *$s$-$t$-separator* if vertices $s$ and $t$ are separated in $G - S$, i.e, $s$ and $t$ belong to different components of $G - S$. A *minimum $s$-$t$-separator* is an $s$-$t$-separator of minimum size. We always let $k$ denote the size of a minimum $s$-$t$-separator in $G$. The token[3] jumping (TJ-) (resp. token sliding (TS-)) Minimum Separator Reconfiguration (MSR) problem is defined as follows. Given a graph $G$ and minimum $s$-$t$-separators $A$ and $B$, the goal is to determine if there exists a sequence of sets $A = S_1, S_2, \ldots, S_r = B$, such that $S_i$ is a minimum $s$-$t$-separator, $S_i = (S_{i-1} \setminus \{v\}) \cup \{u\}$ for some $v \in S_{i-1}$, and $u \in V(G) \setminus S_{i-1}$ (resp. $u \in N_{G-S_{i-1}}(v)$) for every $i \in [r] \setminus \{1\}$.

**Motivation.** Reconfiguration problems arise in various applications and, as a result, have gained considerable attention in recent literature [1, 2, 9, 18]. They appear in power supply problems, such as operating switches in a network to transform between different arrangements of power supply from stations to homes without causing a blackout [17]. They also show up in evolutionary biology, such as in the transformation of genomes via mutations [22]. Moreover, reconfiguration problems contribute to numerous fields of study, such as computational geometry with polygon reconfiguration [5], or statistical physics with the transformation of a particle's spin system [6]. At the same time, vertex separators are useful in the factorization of sparse matrices [23], as well as, partitioning hypergraphs [19]. They also lend themselves to problems in cyber security and telecommunication [20], bioinformatics and computational biology [15], and many divide-and-conquer graph algorithms [12]. Given the importance of vertex separators, we believe that it is a natural question to study the problem of reconfiguration between different vertex separators.

**Related work.** Gomes, Nogueira, and dos Santos [16] initiated the study of the problem of computing reconfiguration sequences between $s$-$t$-separators, $A$ and $B$, without restricting the size of the separators (to minimum). We call the corresponding problem Vertex Separator Reconfiguration (VSR). They show that for token sliding, checking if $A$ can be transformed to $B$, i.e., Vertex Separator Reconfiguration, is a PSPACE-complete problem even on bipartite graphs. In contrast, under the token jumping model the problem becomes NP-complete for bipartite graphs.

**Our results.** Unlike the VSR problem, the requirement in the MSR problem that the separators in the reconfiguration sequence must be minimum introduces a lot of structure. In particular, we can rely on the duality between minimum separators and disjoint paths, observing that tokens are always constrained to move on a set of disjoint $s$-$t$-paths, which we call *canonical paths*. Using this property, we prove that, in an (optimal) solution, tokens always move "forward" towards their target locations and we never need to take a step back. This immediately prevents the problems from being PSPACE-complete since this gives a (polynomial) bound on the length of a solution. In fact, the "always-forward" property immediately implies a greedy algorithm that decides whether we can reconfigure one $s$-$t$-separator into another or not for both the token sliding and token jumping models. We then

[2] It is important to note that graphs may have an exponential number of minimum $s$-$t$-separators as otherwise the problem is trivial.

[3] The notion of tokens is intentionally kept abstract as tokens can represent any type of agents.

turn our attention to finding shortest reconfiguration sequences. While TS-MSR is still solvable in polynomial-time, finding an optimal solution for the TJ-MSR problem is shown to be NP-complete by a reduction from Vertex Cover; finding the largest set of vertices that can be "skipped" by jumping over them is "similar" to finding a minimum vertex cover.

We give a complete characterization of the (parameterized) complexity of the TJ-MSR problem for its natural parameterizations. In particular, we complement our NP-hardness result by showing that the problem of finding a shortest sequence of token jumps is fixed-parameter tractable when parameterized by $k$, the size of a minimum separator; this is accomplished by further exploiting the structure imposed by the separators' minimality as yes-instances have pathwidth bounded by $\mathcal{O}(k)$. Unfortunately, unless $\mathsf{NP} \subseteq \mathsf{coNP/poly}$, the problem admits no polynomial kernel under this parameterization. Finally, we show that if we parameterize the problem by the length of the reconfiguration sequence, $\ell$, then we obtain a kernel with $\mathcal{O}(\ell^2)$ vertices and edges.

## 2 Preliminaries

We denote the set of natural numbers by $\mathbb{N}$ and, for $n \in \mathbb{N}$, we let $[n] = \{1, 2, \dots, n\}$. We only deal with finite simple undirected graphs. We let $V(G)$ and $E(G)$ denote the vertex set and edge set of graph $G$, respectively. The *open neighborhood* of a vertex $v$ is denoted by $N_G(v) = \{u \in V(G) \mid \{u, v\} \in E(G)\}$ and the *closed neighborhood* by $N_G[v] = N_G(v) \cup \{v\}$. For a set $S \subseteq V(G)$ of vertices, we define $N_G(S) = \bigcup_{v \in S} N_G(v) \setminus S$ and $N_G[S] = N_G(S) \cup S$; if the context is clear, we omit the subscript $G$. The subgraph of $G$ *induced* by $S$ is denoted by $G[S]$, where $G[S]$ has vertex set $S$ and edge set $\{\{u, v\} \in E(G) \mid u, v \in S\}$; we also define $G - S = G[V(G) \setminus S]$. A *walk* of length $q$ from $v_0$ to $v_q$ in $G$ is a vertex sequence $v_0, \dots, v_q$ such that $\{v_i, v_{i+1}\} \in E(G)$ for all $i \in \{0, \dots, q-1\}$. It is a *path* if all vertices are distinct. An *$s$-$t$-path* is one with endpoints $s$ and $t$. Two $s$-$t$-paths $P_1$ and $P_2$ are *(internally) disjoint* if $V(P_1) \cap V(P_2) = \{s, t\}$. The following is a celebrated theorem attributed to Menger [21] and later generalized and made algorithmic by Ford and Fulkerson [13].

▶ **Theorem 1.** *The size of a minimum $s$-$t$-separator is equal to the maximum number of pairwise internally disjoint $s$-$t$-paths, which can be computed in polynomial time.*

**Parameterized complexity.** A *parameterized problem* $Q$ is a subset of $\Sigma^* \times \mathbb{N}$, where the second component denotes the *parameter*. A parameterized problem is *fixed-parameter tractable* with respect to a parameter $\kappa$, FPT for short, if there exists an algorithm to decide whether $(x, \kappa) \in Q$ in time $f(\kappa) \cdot |x|^{\mathcal{O}(1)}$, where $f$ is a computable function. We say that two instances are *equivalent* if they are both yes-instances or both no-instances. A *kernelization algorithm*, or a *kernelization* for short, is a polynomial-time algorithm that reduces an input instance $(x, \kappa)$ into an equivalent instance $(x', \kappa')$ such that $|x'|, \kappa' \leq f(\kappa)$, for some computable function $f$. Such $x'$ is called a *kernel*. Every fixed-parameter tractable problem admits a kernel, however, possibly of exponential or worse size. For efficient algorithms it is therefore most desirable to obtain kernels of polynomial, or even linear size. We refer to the textbooks [7, 10] for extensive background on parameterized complexity.

## 3 Preprocessing and general observations

Let $(G, s, t, A, B)$ denote an instance of Minimum Separator Reconfiguration, where $A$ and $B$ are minimum $s$-$t$-separators of size $k$. The reconfiguration model, i.e., jumping vs. sliding, will be clear from context. We begin by making some general observations (that hold for both models) about the structure of sequences of minimum $s$-$t$-separators, which we make extensive use of. We also describe some preprocessing operations: we assume they have been applied on every instance in the rest of the paper. We begin by introducing the notion of canonical paths, which describe the possible locations for each token.

▶ **Definition 2** (Canonical paths). *Let $A$ and $B$ denote the starting and target separators, respectively. By Theorem 1, we begin by fixing a maximum set of pairwise internally disjoint $s$-$t$-paths, which has size $k = |A| = |B|$ (since $A, B$ are minimum separators). We may assume that all paths are* chordless*, i.e., if two vertices of the same path are adjacent the edge connecting them is part of the path; otherwise, we could decrease the length of the path by shortcutting along the chord. We repeat this procedure until all paths are chordless; it terminates since in each iteration the total number of vertices involved in the paths decreases. We call these $k$ chordless pairwise internally disjoint $s$-$t$-paths $P_1, \ldots, P_k$ the* canonical paths *of our instance.*

Note that canonical paths may not be uniquely defined; however, we can fix any set of $k$ internally vertex disjoint paths as canonical.

▶ **Lemma 3.** *Let $S$ be a minimum $s$-$t$-separator. Then, $S$ contains exactly one vertex of each canonical path.*

**Proof.** The set $S$ has to contain at least one vertex of each path since otherwise there would be an $s$-$t$-path in $G - S$. Since the number of paths is equal to the size of a minimum separator and the paths are disjoint, it has to be exactly one in each (Theorem 1). ◀

The next observation follows immediately from Lemma 3.

▶ **Observation 4.** *For both token sliding and token jumping, each token is confined to its respective canonical path and in the case of sliding, a token can only slide to either one of its two neighbours along its canonical path.*

Thus, our view of the problem is that we are sliding (resp. jumping) tokens along a set of $k$ paths and that each token is confined to its respective path. We now show that we can always slide (resp. jump) a token in the direction of the target separator $B$ and never have to do a "backward" move.

Let $L(i)$ denote the number of vertices on the canonical path $P_i$, including $s$ and $t$. Let $u_{i,1}, \ldots u_{i,L(i)}$ denote the vertices on the canonical path $P_i$ in the order in which they appear on it, with $u_{i,1} = s$ and $u_{i,L(i)} = t$. Let $a_i$ and $b_i$ denote the indices such that $V(P_i) \cap A = \{u_{i,a_i}\}$ and $V(P_i) \cap B = \{u_{i,b_i}\}$, i.e., $a_i$ is the index of the starting vertex of the token on $P_i$ and $b_i$ the index of the goal vertex for this token. Let $l_i = \min(a_i, b_i)$ and $r_i = \max(a_i, b_i)$. We first show that, in any (shortest) reconfiguration sequence, we only need to consider configurations of tokens in which, for all $i$, the token on the path $P_i$ remains between (or on) $u_{i,l_i}$ and $u_{i,r_i}$.

▶ **Lemma 5.** *For all $i \in [k]$, let $\phi_i$ be the function such that for all $1 < a < L(i)$,*

$$\phi_i(a) := \begin{cases} l_i & \text{if } a < l_i, \\ r_i & \text{if } a > r_i, \\ a & \text{otherwise.} \end{cases}$$

*Let $f(u_{i,a}) := u_{i,\phi_i(a)}$. If $X$ is a minimum $s$-$t$-separator, then so is $f(X)$.*

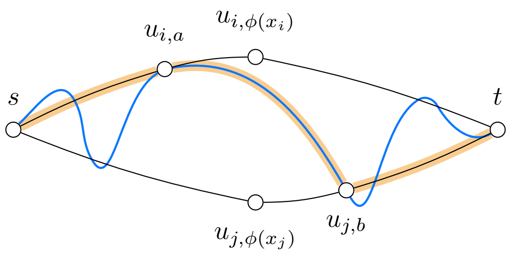


**Figure 1** $P$ is in blue and $P'$ is highlighted in orange.

**Proof.** Given a set $X = \{u_{i,x_i} \mid i \in [k]\}$, let $\mathrm{Left}(X) = \bigcup_i \{u_{i,a} \mid a < x_i\}$.

Let $X = \{u_{i,x_i} \mid i \in [k]\}$ be an $s$-$t$-separator. Assume that $Y = f(X)$ is not an $s$-$t$-separator. Let $P$ be an $s$-$t$-path in $G - Y$. Let $u_{i,a}$ be the last vertex of $P$ belonging to $\mathrm{Left}(Y)$ (possibly $u_{i,a} = s$). Let $u_{j,b}$ be the first vertex of $P$ lying on a canonical path after $u_{i,a}$ (possibly $u_{j,b} = t$). We have $u_{j,b} \notin \mathrm{Left}(Y)$ and thus $b > \phi_j(x_j)$. Note that $a \neq \phi_i(x_i)$ and $b \neq \phi_j(x_j)$ by the definition of $P$. Let $P'$ be the path in $G$ going from $s$ to $u_{i,a}$ via $P_i$, then to $u_{j,b}$ via $P$ and finally to $t$ via $P_j$; see Figure 1. We have $i \neq j$ since otherwise $P'$ would be an $s$-$t$-path in $G - A$ (respectively $G - B$ or $G - X$) if $\phi_i(x_i) = a_i$ (respectively $\phi_i(x_i) = b_i$ or $\phi_i(x_i) = x_i$).

We cannot have $a < x_i$ and $x_j < b$ at the same time since otherwise $P'$ would be an $s$-$t$-path in $G - X$. Without loss of generality, assume that $x_i \leq a$. As a result, $x_i \leq a < \phi_i(x_i) = l_i$. We must have $b < l_j$, or otherwise $P'$ would be an $s$-$t$-path in $G - A$ or $G - B$. Thus, $b < l_j \leq \phi_j(x_j)$ which contradicts the aforementioned property that $b > \phi_j(x_j)$ and proves that $f(X)$ is an $s$-$t$-separator. ◀

▶ **Corollary 6.** *In both the token jumping and token sliding models, if there exists a reconfiguration sequence from $A$ to $B$, then there exists a shortest sequence such that, for any $i$, the $i^{th}$ token remains between $l_i$ and $r_i$ at all times. As a result, deleting all vertices on canonical paths that are not beteween $l_i$ and $r_i$ and replacing them by the edges $\{\{s, l_i\} \mid 1 \leq i \leq k\} \cup \{\{r_i, t\} \mid 1 \leq i \leq k\}$ yields an equivalent instance.*

**Proof.** Two $s$-$t$-separators differing only by a token jump (resp. slide) are mapped by $f$ to two $s$-$t$-separators that are either equal or differing by only a token jump (resp. slide). Thus, applying $f$ to all separators in the reconfiguration sequence, we get a reconfiguration sequence with the claimed property. ◀

Given a vertex $u_{i,x}$ on a canonical path $P_i$, we define the set $F(u_{i,x})$ as the set of vertices of $P_i$ between $u_{i,x}$ and $u_{i,b_i}$ (see Figure 2). We say that a jump (resp. slide) from $u_{i,a}$ to $u_{i,b}$ is *forward* if $u_{i,b} \in F(u_{i,a})$. Intuitively, this means that a jump (resp. slide) is forward if it moves the token closer to its target location (along the canonical path) without going past it.

▶ **Lemma 7** (Forward-moving lemma)**.** *If there exists a reconfiguration sequence from $A$ to $B$, then there exists a shortest sequence $\mathcal{S}$ of jumps (resp. slides) going from $A$ to $B$ containing only forward jumps (resp. slides).*

**Proof.** We proceed by induction on the length of a shortest sequence. If $A = B$, there is nothing to prove. Otherwise, let $\mathcal{S}$ be a shortest sequence of moves going from $A$ to $B$. By Corollary 6, we can assume that the first move in $\mathcal{S}$ is a forward move. Let $A'$ be the separator obtained after this move. The tail of $\mathcal{S}$ is a shortest sequence of move between $A'$ and $B$ and by induction, it may be replaced with a shortest sequence that only contains forward jumps (resp. slides). ◀

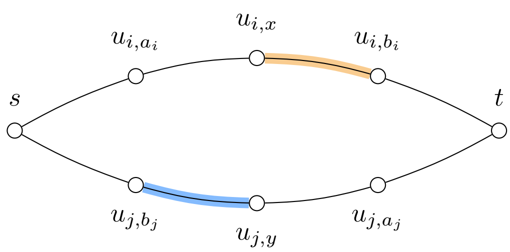


**Figure 2** $F(u_{i,x})$ is highlighted in orange and $F(u_{j,y})$ in blue.

## 4 Polynomial-time algorithms

The forward-moving lemma immediately implies that several problems can be solved in polynomial time by a greedy algorithm.

▶ **Theorem 8.** *A minimum-length sequence of token slides reconfiguring one minimum $s$-$t$-separator to another can be computed in polynomial time.*

**Proof.** Since slides are reversible, doing any slide can never turn a yes-instance into a no-instance. Thus, we can greedily apply moves that slide a token forward. That is, we iteratively find any token that can slide forward (as long as possible) and execute the slide. Since we never need to do a backward slide (Lemma 7), this always finds a solution if one exists. Moreover, this is optimal; the paths are chordless, so any slide can only advance a token one step closer towards its target position, and since there are no backward slides, the solution is optimal. Clearly, checking if a token can slide forward requires polynomial time and given that an optimal solution has polynomial length we get the claimed running time. ◀

▶ **Theorem 9.** *A (feasible, but not necessarily minimum-length) sequence of token jumps reconfiguring one minimum $s$-$t$-separator to another can be computed in polynomial time.*

**Proof.** Jumps are also reversible, so doing a jump can never turn a yes-instance into a no-instance. Again, we can greedily apply forward jumps. Since a solution never needs to contain a backward jump, this yields a feasible solution if one exists. ◀

In the case of token jumping, the solution produced by the greedy algorithm is not necessarily optimal (not guaranteed to be a shortest sequence of jumps); by choosing a different order for the jumps, it might be possible to make "longer" jumps, i.e., jumping over more vertices. In fact, we show that deciding whether a sequence of at most $\ell$ jumps can tranform one minimum $s$-$t$-separator into another is an NP-complete problem.

## 5 Hardness of finding short sequences of jumps

First, we note that the problem of deciding whether a sequence of at most $\ell$ jumps between two minimum $s$-$t$-separators exists is in NP. Indeed, Lemma 7 implies that the length of a reconfiguration sequence cannot exceed $|V(G)|$; we can therefore directly use a reconfiguration sequence as a certificate.

We show NP-hardness by reducing the Vertex Cover problem. Given a graph $G = (V, E)$ and the size $\kappa$ of a desired vertex cover, we construct our instance $(G', s, t, A, B)$ as follows. We first create a copy of the graph $G$, and for every $v \in V(G)$ we add two additional vertices $s_v, t_v$ and edges $\{s_v, v\}, \{v, t_v\}$. We further add vertices $s, t$ and for all $v \in V(G)$, edges $\{s, s_v\}, \{t_v, t\}$; see Figure 3. We ask whether we can reconfigure the $s$-$t$-separator $A = \{s_v \mid v \in V(G)\}$ to the separator $B = \{t_v \mid v \in V(G)\}$ using at most $|V(G)| + \kappa$ token jumps. Note that the canonical paths are of the form $\{s, s_v, v, s_t, t \mid v \in V(G)\}$.

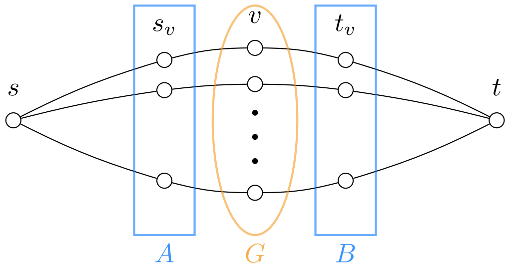


**Figure 3** The graph $G'$ formed from $G$ (highlighted in orange), along with the initial and target separators $A$ and $B$ (highlighted in blue), respectively.

▶ **Theorem 10.** *Deciding whether a sequence of at most $\ell$ token jumps can tranform one minimum $s$-$t$-separator into another is an NP-complete problem.*

## 6 Preprocessing for token jumping

In this section, we describe some preprocessing rules that can be applied to the token jumping variant of Minimum Separator Reconfiguration. In the remainder of this paper, we assume that all instances are preprocessed according to these rules. We first show that we can reduce the graph so that it contains no vertices that are not on the canonical $s$-$t$-paths.

▶ **Lemma 11.** *Given an instance $(G, s, t, A, B)$ of* Minimum Separator Reconfiguration *in the token jumping model, it is possible to compute in polynomial time an equivalent instance $(G', s, t, A, B)$ in which all vertices are on the (chordless) canonical paths and both the minimum $s$-$t$-separator size and the length of minimum reconfiguration sequences are preserved. Moreover, all vertices in $A \cup B$ are adjacent to either $s$ or $t$.*

Going forward, we assume that all graphs are preprocessed according to these rules. We next observe that to ensure that a configuration of tokens forms a valid $s$-$t$-separator it suffices to check the existence of relatively simple $s$-$t$-paths.

▶ **Lemma 12.** *To check whether a configuration of tokens that assigns exactly one token to each canonical path forms an $s$-$t$-separator, it suffices to check whether there exists an $s$-$t$-path that from $s$, follows one of the canonical paths, then follows one edge (a* crossing edge*) from that canonical path to another, and then follows that canonical path to $t$.*

We now show that it suffices to consider instances in which all vertices have degree at least 3 and at most $2k$, where $k \geq 3$ is the size of the minimum separator (for $k \leq 2$, the problem can be solved in polynomial time by a simple algorithm that computes a shortest path in an auxiliary graph $H$ having one vertex for each of the at most $n^2$ $s$-$t$-separators and where two vertices of $H$ share an edge whenever the corresponding $s$-$t$-separators are one reconfiguration step away from each other).

▶ **Lemma 13.** *Given an instance $(G, s, t, A, B)$ of* Minimum Separator Reconfiguration *in the token jumping model, it is possible to compute in polynomial time an equivalent instance $(G', s, t, A, B)$ in which the length of minimum reconfiguration sequences is preserved and all vertices have degree at least 3 and at most $2k$, where $k = |A| = |B|$ (assuming $k \geq 3$). In particular, every vertex can have at most two neighbors on each canonical path.*

We proceed by showing that we can always assume that the source and target minimum $s$-$t$-separators, i.e., $A$ and $B$, are disjoint.

▶ **Lemma 14.** *Given an instance $(G, s, t, A, B)$ of* Minimum Separator Reconfiguration *in the token jumping model, it is possible to compute in polynomial time an equivalent instance $(G', s, t, A', B')$ in which $A' \cap B' = \emptyset$ and such that the length of minimum reconfiguration sequences is preserved.*

We conclude this section by formalizing the notion of unskippable vertices; a notion that will be useful in many of our subsequent results. Given an instance $(G, s, t, A, B)$, we say that $v \in V(G) \setminus (A \cup B \cup \{s, t\})$ is *unskippable* if for every sequence (if any exist) of minimum $s$-$t$-separators that transforms $A$ to $B$ there exists at least one $s$-$t$-separator $S$ in the sequence such that $v \in S$. In other words, there is no transformation from $A$ to $B$ that can skip over $v$ and not jump a token onto $v$ at some point. Similarly, we say that a set $U \subseteq V(G) \setminus (A \cup B \cup \{s, t\})$ of vertices is *unskippable* whenever there exists at least one $s$-$t$-separator $S$ in every reconfiguration sequence (from $A$ to $B$) such that $|U \cap S| \geq 1$.

▶ **Lemma 15.** *Let $(G, s, t, A, B)$ be an instance of* Minimum Separator Reconfiguration *in the token jumping model. A vertex $v \in V(G) \setminus (A \cup B \cup \{s, t\})$ having two neighbors on a canonical path other than its own is unskippable. If $u, v \in V(G) \setminus (A \cup B \cup \{s, t\})$, $u, v$ belong to two different canonical paths, and $\{u, v\} \in E(G)$, then $\{u, v\}$ is unskippable.*

## 7 Parameterizing by the size of the separators

In this section, we study the natural parameterization by $k$: the number of tokens, or the size of minimum separators. We first prove that there exists an FPT algorithm for this parameterization, then show that no polynomial kernel exists unless NP $\subseteq$ coNP/poly.

### 7.1 FPT algorithm

In this section, we show that finding a shortest sequence for the token jumping variant of Minimum Separator Reconfiguration is fixed-parameter tractable with respect to $k$, the size of the minimum $s$-$t$-separators. We do so by constructing a path decomposition of the graph from an arbitrary sequence of token jumps, and then proceed by designing a dynamic programming algorithm that operates on that path decomposition. We shall work towards proving the following.

▶ **Lemma 16.** *Given a graph $G$ preprocessed by our reduction rules and two minimum $s$-$t$-separators $A$ and $B$ of $G$ with $|A| = k$, if $A$ and $B$ can be reconfigured into each other, then $G \setminus \{s, t\}$ has pathwidth at most $k$.*

Let $\mathcal{S} = A, S_1, \ldots, S_\ell, B$ be a reconfiguration sequene of minimum separators. Note that the symmetric difference between two elements of $\mathcal{S}$ are two vertices $u_{j,p} \in S_i \setminus S_{i+1}$ and $u_{j,r} \in S_{i+1} \setminus S_i$ belonging to the *same* canonical path. As such, we can construct a width-$k$ path decomposition where (i) each $S_i \in \mathcal{S}$ is placed in the order they appear in $\mathcal{S}$ and (ii) between $S_i$ and $S_{i+1}$ we have the bags $X \cup \{u_{j,q}\}$, where $X = S_i \cap S_{i+1}$ and $p < q < r$. Using the previous statement and dynamic programming, we prove our main result.

▶ **Theorem 17.** *The optimization version of* Minimum Separator Reconfiguration *under token jumping parameterized by the size of a minimum separator is in* FPT.

## 7.2 No polynomial kernel for parameter $k$

We use the cross-composition framework developed by the work of Drucker [11], Bodlaender et al. [3, 4], Dell and van Melkebeek [8], and Fortnow and Santhanam [14]. Roughly speaking, in this framework, a problem $\Pi$ *and-cross-composes* into a parameterized problem $\Gamma$ if, given instances $\{I_1, \dots, I_r\}$ of $\Pi$, we can construct an instance $(O, k)$ of $\Gamma$ such that: $O$ is a yes-instance if and only if $I_i$ is a yes-instance for all $i \in [r]$, $|O|$ is polynomial on $\sum_i |I_i| + r$, and $k \leq \mathsf{poly}(\max_i |I_i| + \log r)$.

For Minimum Separator Reconfiguration under token jumping, we and-cross-compose Vertex Cover, employing a construction very similar to the one used in Theorem 10. In fact, for each of the $r$ vertex cover instances $(G_i, \kappa)$, we repeat the construction of Theorem 10 and serialize all the graphs in a "linear fashion" using synchronization gadgets, as depicted in Figure 4. We ask for a reconfiguration sequence between the leftmost and rightmost sets (each of size $k = \mu + 1$) of length $\mathcal{O}(rk + \sum(|V(G_i)|))$. As the name suggests, the purpose of the synchronization gadgets is to guarantee that all tokens jump over each graph $G_i$ before moving on to the next graph (allowing us to accurately calculate the total number of required jumps). Formally, our result can be stated as follows:

▶ **Theorem 18.** *There exists an and-cross-composition from* Vertex Cover *into TJ* Minimum Separator Reconfiguration*, parameterized by the minimum size $k$ of an $s$-$t$-separator. Consequently, when parameterized by $k$, TJ* Minimum Separator Reconfiguration *does not admit a polynomial kernel unless* NP $\subseteq$ coNP/poly.

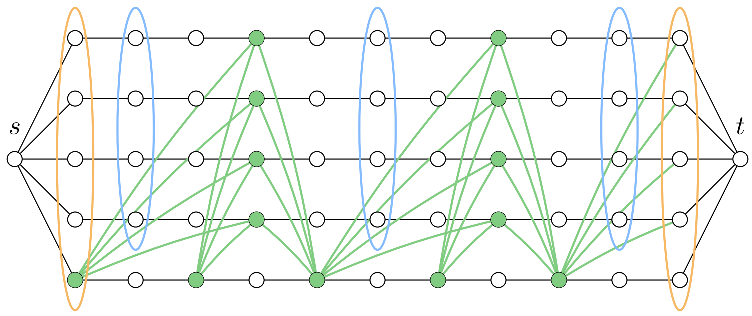


**Figure 4** An overview of the and-cross-composition with $r = 3$ and $k = 5$. The orange blobs are $A$ and $B$, the blue blobs represent the graphs $G_i$, and the green vertices and edges represent the synchronization gadgets.

## 8 Polynomial kernel for parameter $\ell$

Observe that the construction employed in Theorem 18 heavily relies on the fact that the length $\ell$ of the reconfiguration sequence is linearly proportional to the number of instances. We show that this dependence cannot be broken. That is, when parameterizing by $\ell$, we prove that Minimum Separator Reconfiguration admits a quadratic kernel.

Recall that, by Lemma 11, we can preprocess the graph so that all vertices are on the canonical paths and the vertices in $A$ and $B$ are adjacent to (at least one of) $s$ or $t$. Similarly, by Lemma 13, each vertex in $G$ will have degree at least 3 and at most $2k$ (with at most two neighbors on each canonical path) and, by Lemma 14, we know that $A \cap B = \emptyset$. We refer to an instance satisfying all of the above as a reduced instance.

▶ **Lemma 19.** *In a reduced yes-instance where $\ell$ is the parameter, all the following properties must be satisfied: (i) $A \cap B = \emptyset$ and $|A| = |B| \leq \ell$; (ii) all vertices are on the (at most $\ell$) canonical paths; (iii) vertices in $A$ and $B$ are adjacent to (at least one of) $s$ or $t$; and (iv) each vertex in $G$ has degree at least 3, degree at most $2\ell$, and at most two neighbors on each canonical path.*

**Proof.** The lemma follows immediately from the fact that if after obtaining a reduced instance we have $|A| = |B| > \ell$, then we have a no-instance; as at least $\ell + 1$ jumps are needed. Consequently, the minimum separator size will be at most $\ell$ and all the remaining properties follow from Lemma 11, Lemma 13, and Lemma 14. ◀

▶ **Lemma 20.** *Assume that in a reduced instance one of the canonical paths contains more than $4(\ell + 1)^2 + 4$ vertices. Then, the instance is a no-instance.*

**Proof.** Let $P$ denote such a canonical path. We claim that at least $\ell + 1$ jumps are required for the token on $P$. We assume otherwise, i.e, that $\ell$ jumps or fewer are enough, and work towards a contradiction.

First, recall that if a vertex $v$ on a canonical path is adjacent to two distinct vertices on another canonical path, then $v$ can never be jumped over, i.e., $v$ is unskippable (Lemma 15). We decompose $P$ into $\ell + 1$ subpaths each consisting of at least $4\ell + 4$ vertices (excluding $s$, $t$, and the initial and target vertices of $A$ and $B$). For the token on $P$ to reach its final position in at most $\ell$ jumps, it must (at least once) jump over $2\ell + 1$ (consecutive) vertices of $P$ or more (landing on the vertex $2\ell + 2$ away or more). Let us denote those vertices that are jumped over by $Q$. Moreover, let $S_i$ denote the $s$-$t$-separator preceeding the jump and let $S_{i+1}$ denote the resulting $s$-$t$-separator after the jump.

If $Q$ contains a vertex having two distinct neighbors on another canonical path, then the vertex is unskippable and we get a contradiction. Hence, every vertex $v$ of $Q$ (which has degree 3 or more) can have at most one neighbor on every canonical path $P' \neq P$ (and we know that $v$ must have at least one neighbor not in $P$). Since $S_i$ and $S_{i+1}$ are $s$-$t$-separators, every vertex $v \in Q$ can only be adjacent to vertices in $S_i \setminus V(P) = S_{i+1} \setminus V(P)$; otherwise an $s$-$t$-path can be easily constructed, contradicting the fact that $S_i$ and $S_{i+1}$ are $s$-$t$-separators. Now, given that $|Q| \geq 2\ell + 1$, we know that there exists at least 3 distinct vertices in $Q$ all having the same neighbor in $S_{i+1} \setminus V(P)$. This contradicts the fact that after our reductions each vertex in $G$ can have at most two distinct neighbors on any canonical path. ◀

Lemmas 19 and 20 immediately imply a kernel with $\mathcal{O}(\ell^3)$ vertices: a yes-instance consists of at most $\ell$ canonical paths each having $\mathcal{O}(\ell^2)$ vertices. We obtain an improved bound by refining our analysis slightly and strengthening the result of Lemma 20 in Theorem 21.

▶ **Theorem 21.** *The optimization version of* TJ Minimum Separator Reconfiguration *admits a kernel with $\mathcal{O}(\ell^2)$ vertices and edges when parameterized by the length $\ell$ of a reconfiguration sequence.*

## 9 Concluding Remarks

We studied the minimum $s$-$t$-separator reconfiguration problem through several lenses. First, we considered the token sliding and token jumping reconfiguration models, showing that the reachability question is answerable in polynomial time in both cases. Afterwards, we considered the task of finding a shortest reconfiguration sequence; we proved that it is easy under the first model but NP-complete under the second. To tackle this hardness, we studied the parameterized complexity of the token jumping version for the natural parameterizations

$k$ (the number of tokens) and $\ell$ (the length of the sequence). In this context, we designed an FPT algorithm for parameter $k$, a quadratic kernel when parameterized by $\ell$, and showed that no polynomial kernel exists for $k$ unless $\mathsf{NP} \subseteq \mathsf{coNP/poly}$.

In terms of future work on minimum $s$-$t$-separator reconfiguration itself, we are interested in understanding shortest token jumping sequences for some graph classes, in particular for planar graphs. Other possibilities include the study of structural parameterizations, such as treewidth, feedback edge set, and vertex deletion distance metrics (e.g., distance to cluster, clique, etc.). In a different spirit, studying the connectivity of the minimum $s$-$t$-separator reconfiguration graph might yield different insights than the ones presented in this paper. Beyond minimum separators, the work of Gomes, Nogueira, and dos Santos [16] investigated arbitrary $s$-$t$-separator reconfiguration, but no research has yet been done on bounded size separators. As such, we believe work on the complexity of reconfiguration of minimum $+ r$ $s$-$t$-separators might be of independent interest.

## References


1. Kira Adaricheva, Chassidy Bozeman, Nancy E. Clarke, Ruth Haas, Margaret-Ellen Messinger, Karen Seyffarth, and Heather C. Smith. Reconfiguration graphs for dominating sets. In *Research Trends in Graph Theory and Applications*, volume 25 of *Assoc. Women Math. Ser.*, pages 119–135. Springer, Cham, 2021.
2. Akanksha Agrawal, Soumita Hait, and Amer E. Mouawad. On finding short reconfiguration sequences between independent sets. In *33rd International Symposium on Algorithms and Computation*, volume 248 of *LIPIcs. Leibniz Int. Proc. Inform.*, pages Paper No. 39, 14. Schloss Dagstuhl. Leibniz-Zent. Inform., Wadern, 2022.
3. Hans L. Bodlaender, Rodney G. Downey, Michael R. Fellows, and Danny Hermelin. On problems without polynomial kernels. *J. Comput. Syst. Sci.*, 75(8):423–434, 2009. `doi:10.1016/j.jcss.2009.04.001`.
4. Hans L. Bodlaender, Bart M. P. Jansen, and Stefan Kratsch. Kernelization lower bounds by cross-composition. *SIAM J. Discret. Math.*, 28(1):277–305, 2014. `doi:10.1137/120880240`.
5. Robert Connelly, Erik D. Demaine, and Günter Rote. Blowing up polygonal linkages. *Discrete Comput. Geom.*, 30:205–239, 2003.
6. Daniel W. Cranston and Reem Mahmoud. In most 6-regular toroidal graphs all 5-colorings are Kempe equivalent. *European J. Combin.*, 104:Paper No. 103532, 21, 2022.
7. Marek Cygan, Fedor V. Fomin, Lukasz Kowalik, Daniel Lokshtanov, Dániel Marx, Marcin Pilipczuk, Michal Pilipczuk, and Saket Saurabh. *Parameterized Algorithms*. Springer, 2015. `doi:10.1007/978-3-319-21275-3`.
8. Holger Dell and Dieter van Melkebeek. Satisfiability allows no nontrivial sparsification unless the polynomial-time hierarchy collapses. *J. ACM*, 61(4):23:1–23:27, 2014. `doi:10.1145/2629620`.
9. Quentin Deschamps, Carl Feghali, František Kardoš, Clément Legrand-Duchesne, and Théo Pierron. Strengthening a theorem of Meyniel, 2022.
10. Rodney G. Downey and Michael R. Fellows. *Parameterized Complexity*. Monographs in Computer Science. Springer, 1999. `doi:10.1007/978-1-4612-0515-9`.
11. Andrew Drucker. New limits to classical and quantum instance compression. *SIAM J. Comput.*, 44(5):1443–1479, 2015. `doi:10.1137/130927115`.
12. Cem Evrendilek. Vertex separators for partitioning a graph. *Sensors*, 8(2):635–657, 2008.
13. Lester R. Ford and Delbert R. Fulkerson. *Flows in Networks*. Princeton University Press, USA, 2010.
14. Lance Fortnow and Rahul Santhanam. Infeasibility of instance compression and succinct PCPs for NP. *J. Comput. Syst. Sci.*, 77(1):91–106, 2011. `doi:10.1016/j.jcss.2010.06.007`.
15. Bin Fu and Zhixiang Chen. Sublinear time width-bounded separators and their application to the protein side-chain packing problem. *J. Comb. Optim.*, 15(4):387–407, 2008.

16 Guilherme C. M. Gomes, Sérgio H. Nogueira, and Vinícius F. dos Santos. Some results on vertex separator reconfiguration, 2020.

17 Takehiro Ito, Erik D. Demaine, Xiao Zhou, and Takao Nishizeki. Approximability of partitioning graphs with supply and demand. *J. Discrete Algorithms*, 6(4):627–650, 2008.

18 Takehiro Ito, Naonori Kakimura, Naoyuki Kamiyama, Yusuke Kobayashi, and Yoshio Okamoto. Shortest reconfiguration of perfect matchings via alternating cycles. *SIAM J. Discrete Math.*, 36(2):1102–1123, 2022.

19 Enver Kayaaslan, Ali Pinar, Ümit Çatalyürek, and Cevdet Aykanat. Partitioning hypergraphs in scientific computing applications through vertex separators on graphs. *SIAM J. Sci. Comput.*, 34(2):A970–A992, 2012.

20 Charles E. Leiserson. Area-efficient graph layouts. In *21st Annual Symposium on Foundations of Computer Science*, pages 270–281, 1980.

21 Karl Menger. Zur allgemeinen Kurventheorie. *Fundamenta Mathematicae*, 10:96–115, 1927. URL: `http://eudml.org/doc/211191`.

22 István Miklós and Heather Smith. Sampling and counting genome rearrangement scenarios. *BMC Bioinformatics*, 16, 2015.

23 Xie Xian-fen, Gu Wan-rong, He Yi-chen, and Mao Yi-jun. Matrix transformation and factorization based on graph partitioning by vertex separator for recommendation. *Computer Science*, 49(6):272–279, 2022.