Theory of Computing Systems manuscript No.
(will be inserted by the editor)

Finding Disjoint Paths in Split Graphs

Pinar Heggernes[1] · Pim van 't Hof[1] · Erik Jan van Leeuwen[2] · Reza Saei[1]

Received: date / Accepted: date

Abstract The well-known Disjoint Paths problem takes as input a graph G and a set of k pairs of terminals in G, and the task is to decide whether there exists a collection of k pairwise vertex-disjoint paths in G such that the vertices in each terminal pair are connected to each other by one of the paths. This problem is known to be NP-complete, even when restricted to planar graphs or interval graphs. Moreover, although the problem is fixed-parameter tractable when parameterized by k due to a celebrated result by Robertson and Seymour, it is known not to admit a polynomial kernel unless NP $\subseteq$ coNP/poly. We prove that Disjoint Paths remains NP-complete on split graphs, and show that the problem admits a kernel with $O(k^2)$ vertices when restricted to this graph class. We furthermore prove that, on split graphs, the edge-disjoint variant of the problem is also NP-complete and admits a kernel with $O(k^3)$ vertices. To the best of our knowledge, our kernelization results are the first non-trivial kernelization results for these problems on graph classes.

Keywords Disjoint paths · Computational complexity · Parameterized complexity · Polynomial kernel · Split graphs

1 Introduction

Finding vertex-disjoint or edge-disjoint paths with specified endpoints is a classical and fundamental problem in algorithmic graph theory and combinatorial optimization, with applications in such areas as VLSI layout, transportation networks, and network reliability; see, for example, the surveys by Frank [9]

This research is supported by the Research Council of Norway. A preliminary version of this work was presented at SOFSEM 2014 [13].

[1] Department of Informatics, University of Bergen, Postboks 7803, 5020 Bergen, Norway, {pinar.heggernes,pim.vanthof,reza.saeidinvar}@ii.uib.no
[2] MPI für Informatik, Campus E1 4, 66123 Saarbrücken, Germany, erikjan@mpi-inf.mpg.de

and by Vygen [26]. The Vertex-Disjoint Paths problem takes as input a graph G and a set of k pairs of terminals in G, and the task is to decide whether there exists a collection of k pairwise vertex-disjoint paths in G such that the vertices in each terminal pair are connected to each other by one of the paths. The Edge-Disjoint Paths problem is defined analagously, but here the task is to decide whether there exist k pairwise edge-disjoint paths instead of vertex-disjoint paths.

The Vertex-Disjoint Paths problem was shown to be NP-complete by Karp [15], one year before Even et al. [8] proved that the same holds for Edge-Disjoint Paths. A celebrated result by Robertson and Seymour [24], obtained as part of their groundbreaking graph minors theory, states that the Vertex-Disjoint Paths problem can be solved in $O(n^3)$ time for every fixed k. This implies that Edge-Disjoint Paths can be solved in $O(m^3)$ time for every fixed k. As a recent development, an $O(n^2)$-time algorithm for each of the problems, for every fixed k, was obtained by Kawarabayashi, Kobayashi and Reed [16]. The above results show that both problems are fixed-parameter tractable when parameterized by the number of terminal pairs. On the negative side, Bodlaender, Thomassé and Yeo [3] showed that, under the same parameterization, the Vertex-Disjoint Paths problem does not admit a polynomial kernel, unless NP $\subseteq$ coNP/poly.

Both problems have been intensively studied on graph classes. A trivial reduction from Edge-Disjoint Paths to Vertex-Disjoint Paths implies that the latter is NP-complete on line graphs. By a slightly more complicated argument, Edge-Disjoint Paths can also be shown to be NP-complete on line graphs[1]. It is known that both problems remain NP-complete when restricted to planar graphs [18,19]. On the positive side, Vertex-Disjoint Paths can be solved in linear time for every fixed k on planar graphs [23], or more generally, on graphs of bounded genus [7,17]. Interestingly, Vertex-Disjoint Paths can be solved in polynomial-time on graphs of bounded treewidth [22], while Edge-Disjoint Paths is NP-complete on series-parallel graphs [21], and thus on graphs of treewidth at most 2. Gurski and Wanke [11] proved that the line graph of a graph of treewidth at most k has clique-width at most $2k + 2$, and therefore Vertex-Disjoint Paths (following [11]) and Edge-Disjoint Paths (using the reduction of footnote 1) are NP-complete on graphs of clique-width at most 6. On the other hand, Vertex-Disjoint Paths can be solved in linear time on graphs of clique-width at most 2 [11]. Natarajan and Sprague [20] proved the NP-completeness of Vertex-Disjoint Paths on interval graphs, and thus also on all superclasses of interval graphs such as circular-arc graphs and chordal graphs. On chordal graphs, Vertex-Disjoint Paths is linear-time solvable for each fixed k [14].

[1] We briefly sketch a reduction from Edge-Disjoint Paths on general graphs to Edge-Disjoint Paths on line graphs. Replace every edge uv of the original graph G by a three-vertex path u, x_{uv}, v (where x_{uv} is a new vertex), and take the line graph of the resulting graph. Then each edge uv of G corresponds to an edge of the line graph (namely, the edge between the vertices corresponding to edges ux_{uv} and $x_{uv}v$). This creates a correspondence between paths in G and paths in the line graph, and completes the proof.

Given the fact that the Vertex-Disjoint Paths problem is unlikely to admit a polynomial kernel on general graphs, and the amount of known results for both problems on graph classes, it is surprising that no kernelization result has been known on either problem when restricted to graph classes. Interestingly, even the classical complexity status of both problems has been open on split graphs, i.e., graphs whose vertex set can be partitioned into a clique and an independent set, which form a well-studied graph class and a famous subclass of chordal graphs [4,10].

We present the first non-trivial kernelization algorithms for the Vertex-Disjoint Paths and Edge-Disjoint Paths problems on graph classes by showing that the problems admit kernels with $O(k^2)$ and $O(k^3)$ vertices, respectively, on split graphs. To complement these results, we prove that both problems remain NP-complete on this graph class.

2 Preliminaries

All the graphs considered in this paper are finite, simple, and undirected. We refer to the monograph by Diestel [5] for graph terminology and notation not defined below. A *split graph* is a graph whose vertex set can be partitioned into a clique C and an independent set I, either of which may be empty; such a partition (C, I) is called a *split partition*. Note that, in general, a split graph can have more than one split partition. However, split graphs can be recognized in linear time, and a split partition can be found in linear time if one exists [12]. A *chordal graph* is a graph for which no set of four or more vertices induces a cycle, or more formally, a graph G is chordal if for any $X \subseteq V(G)$ with $|X| \geq 4$, the graph $(X, E(G) \cap (X \times X))$ is not a cycle. Observe that any split graph is chordal.

Let G be a graph. For any vertex v in G, we write $N_G(v)$ to denote the neighborhood of v, and $d_G(v) = |N_G(v)|$ to denote the degree of v. Given a path P in G and a vertex $v \in V(G)$, we say that P *visits* v if $v \in V(P)$.

The two problems we consider in this paper are formally defined as follows:

Vertex-Disjoint Paths
Instance: A graph G, and a set $\mathcal{X} = \{(s_1, t_1), \ldots, (s_k, t_k)\}$ of k pairs of vertices in G, called *terminals*, where $s_i \neq t_i$ for each $i \in \{1, \ldots, k\}$.
Question: Do there exist k pairwise vertex-disjoint paths $P_1, \ldots, P_k$ in G such that P_i connects s_i to t_i for each $i \in \{1, \ldots, k\}$?

Edge-Disjoint Paths
Instance: A graph G, and a set $\mathcal{X} = \{(s_1, t_1), \ldots, (s_k, t_k)\}$ of k pairs of vertices in G, called *terminals*, where $s_i \neq t_i$ for each $i \in \{1, \ldots, k\}$.
Question: Do there exist k pairwise edge-disjoint paths $P_1, \ldots, P_k$ in G such that P_i connects s_i to t_i for each $i \in \{1, \ldots, k\}$?

Throughout the paper, we write n and m to denote the number of vertices and edges, respectively, of the input graph G in each of the two problems. Note that in both problems, we allow different terminals to coincide. This is motivated by the close relationship between the problem of finding vertex-disjoint and edge-disjoint paths and the problem of finding topological minors and immersions in graphs, respectively [24]. For this reason, we define two paths to be *vertex-disjoint* if they are distinct, and if none of the paths contains an inner vertex of the other. Observe that this definition ensures that no terminal appears as an inner vertex of any path in a solution for VERTEX-DISJOINT PATHS. For EDGE-DISJOINT PATHS, it seems unnatural to impose such a restriction[2]. Therefore, the definition of edge-disjointness is as expected: we define two paths to be *edge-disjoint* if they do not share an edge.

Suppose $(G, \mathcal{X}, k)$ is a yes-instance of the VERTEX-DISJOINT PATHS problem. A solution $\mathcal{P} = \{P_1, \ldots, P_k\}$ for the instance $(G, \mathcal{X}, k)$ is *minimum* if there is no solution $\mathcal{Q} = \{Q_1, \ldots, Q_k\}$ for $(G, \mathcal{X})$ such that $\sum_{i=1}^{k} |E(Q_i)| < \sum_{i=1}^{k} |E(P_i)|$.

For any problem Π, two instances I_1, I_2 of Π are *equivalent* if I_1 is a yes-instance of Π if and only if I_2 is a yes-instance of Π. A *parameterized problem* is a subset $Q \subseteq \Sigma^* \times \mathbb{N}$ for some finite alphabet Σ, where the second part of the input is called the *parameter*. A parameterized problem $Q \subseteq \Sigma^* \times \mathbb{N}$ is said to be *fixed-parameter tractable* if for each pair $(x, k) \in \Sigma^* \times \mathbb{N}$ it can be decided in time $f(k)\,|x|^{O(1)}$ whether $(x, k) \in Q$, for some function f that only depends on k; here, $|x|$ denotes the length of input x. We say that a parameterized problem Q has a *kernel* if there is an algorithm that transforms each instance (x, k) in time $(|x| + k)^{O(1)}$ into an instance (x', k'), such that $(x, k) \in Q$ if and only if $(x', k') \in Q$ and $|x'| + k' \leq g(k)$ for some function g. Here, g is typically an exponential function of k. If g is a polynomial or a linear function of k, then we say that the problem has a *polynomial* kernel or a *linear* kernel, respectively. We refer the interested reader to the monograph by Downey and Fellows [6] for more background on parameterized complexity. It is known that a parameterized problem is fixed-parameter tractable if and only if it is decidable and has a kernel, and several fixed-parameter tractable problems are known to have polynomial or even linear kernels. Recently, methods have been developed for proving the non-existence of polynomial kernels, under some complexity theoretical assumptions [1–3].

In the NP-completeness proofs in Section 3, we will reduce from a restricted variant of the SATISFIABILITY (SAT) problem. In order to define this variant, we need to introduce some terminology. Let x be a variable and c a clause of a Boolean formula φ in conjunctive normal form (CNF). We say that x *appears* in c if either x or $\neg x$ is a literal of c. If x is a literal of clause c, then we say that x *appears positively* in c. Similarly, if $\neg x$ is a literal of c, then x *appears*

[2] We mention, however, that our polynomial kernel for EDGE-DISJOINT PATHS, presented in Section 4.2, can be easily modified to take this restriction into account. It suffices to insist in Lemma 8 that, instead of the degree, the number of non-terminal neighbors is at least the number of terminals on it. The rest of the proof goes through (mutatis mutandis). Also, our NP-completeness result is not influenced by this issue.

negatively in c. Given a Boolean formula φ, we say that a variable x appears positively (respectively negatively) if there is a clause c in φ in which x appears positively (respectively negatively). The following result, which we will use to prove that Vertex-Disjoint Paths is NP-complete on split graphs, is due to Tovey [25].

Theorem 1 ([25]) *The* SAT *problem is NP-complete when restricted to CNF formulas satisfying the following three conditions:*

- *every clause contains two or three literals;*
- *every variable appears in two or three clauses;*
- *every variable appears at least once positively and at least once negatively.*

3 Finding Disjoint Paths in Split Graphs is NP-Hard

Lynch [19] gave a polynomial-time reduction from SAT to Vertex-Disjoint Paths, thereby proving the latter problem to be NP-complete in general. By modifying his reduction, he then strengthened his result and proved that Vertex-Disjoint Paths remains NP-complete when restricted to planar graphs. In this section, we first show that the reduction of Lynch can also be modified to prove that Vertex-Disjoint Paths is NP-complete on split graphs. We then show that Edge-Disjoint Paths is NP-complete on split graphs as well, using a reduction from the Edge-Disjoint Paths problem on general graphs.

We first describe the reduction from SAT to Vertex-Disjoint Paths due to Lynch [19]. Let $\varphi = c_1 \wedge c_2 \wedge \ldots \wedge c_m$ be a CNF formula, and let $v_1, \ldots, v_n$ be the variables that appear in φ. We assume that every variable appears at least once positively and at least once negatively; if this is not the case, then we can trivially reduce the instance to an equivalent instance that satisfies this property. Given the formula φ, we create an instance $(G_\varphi, \mathcal{X}_\varphi)$ of Vertex-Disjoint Paths as follows (see Figure 1 for an illustrative example).

The vertex set of the graph G_φ consists of three types of vertices: variable vertices, clause vertices, and literal vertices. For each variable v_i in φ, we create two *variable vertices* v_i and w_i; we call (v_i, w_i) a *variable pair*. For each clause c_j, we add two *clause vertices* c_j and d_j and call (c_j, d_j) a *clause pair*. For each clause c_j, we also add a *literal vertex* for each literal that appears in c_j as follows. If c_j contains a literal v_i, that is, if variable v_i appears positively in clause c_j, then we add a vertex $\ell^+_{i,j}$ to the graph, and we make this vertex adjacent to vertices c_j and d_j. Similarly, if c_j contains a literal $\neg v_i$, then we add a vertex $\ell^-_{i,j}$ and make it adjacent to both c_j and d_j. This way, we create $|c_j|$ paths of length exactly 2 between each clause pair (c_j, d_j), where $|c_j|$ is the number of literals in clause c_j.

For each $i \in \{1, \ldots, n\}$, we add edges to the graph in order to create exactly two vertex-disjoint paths between the variable pair (v_i, w_i) as follows. Let $c_{j_1}, c_{j_2}, \ldots, c_{j_p}$ be the clauses in which v_i appears positively, where $j_1 < j_2 < \cdots < j_p$. Similarly, let $c_{k_1}, c_{k_2}, \ldots, c_{k_q}$ be the clauses in which v_i appears

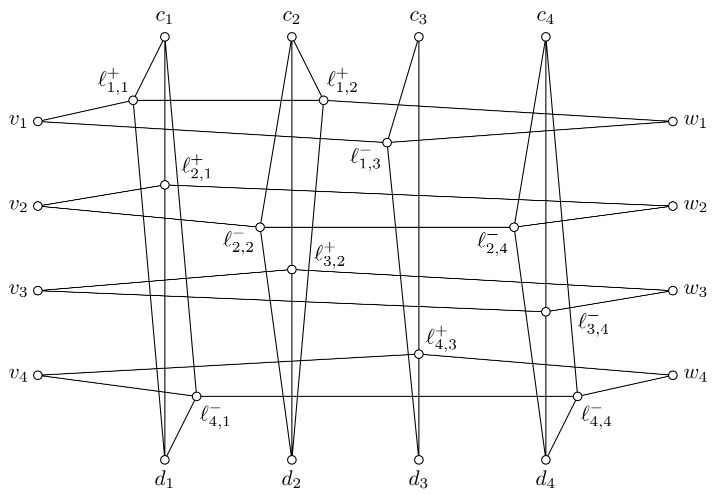

Fig. 1 The graph G_φ constructed in the reduction of Lynch from the CNF formula $\varphi = (v_1 \vee v_2 \vee \neg v_4) \wedge (\neg v_2 \vee v_3 \vee v_1) \wedge (\neg v_1 \vee v_4) \wedge (\neg v_2 \vee \neg v_3 \vee \neg v_4)$.

negatively, where $k_1 < k_2 < \cdots < k_p$. Note that $p \geq 1$ and $q \geq 1$ due to the assumption that every variable appears at least once positively and at least once negatively. We now add the edges $v_i\ell^+_{i,j_1}$ and $\ell^+_{i,j_p}w_i$, as well as the edges $\ell^+_{i,j_1}\ell^+_{i,j_2}$, $\ell^+_{i,j_2}\ell^+_{i,j_3}, \ldots,$ $\ell^+_{i,j_{p-1}}\ell^+_{i,j_p}$. Let $L^+_i = v_i\ell^+_{i,j_1}\ell^+_{i,j_2}\cdots\ell^+_{i,j_{p-1}}\ell^+_{i,j_p}w_i$ denote the path between v_i and w_i that is created this way. Similarly, we add exactly those edges needed to create the path $L^-_i = v_i\ell^-_{i,k_1}\ell^-_{i,k_2}\cdots\ell^-_{i,j_{q-1}}\ell^-_{i,j_q}w_i$. This completes the construction of the graph G_φ.

Let $\mathcal{X}_\varphi$ be the set consisting of all the variable pairs and all the clause pairs in G_φ, i.e., $\mathcal{X}_\varphi = \{(v_i, w_i) \mid 1 \leq i \leq n\} \cup \{(c_j, d_j) \mid 1 \leq j \leq m\}$. The pair $(G_\varphi, \mathcal{X}_\varphi)$ is the instance of Vertex-Disjoint Paths corresponding to the instance φ of SAT.

Theorem 2 ([19]) *Let φ be a CNF formula. Then φ is satisfiable if and only if $(G_\varphi, \mathcal{X}_\varphi)$ is a yes-instance of the* Vertex-Disjoint Paths *problem.*

Proof For the sake of completeness, we sketch the proof of Lynch [19] of this theorem. We only prove one direction; the other direction is similar.

Suppose that φ is satisfiable, and consider any truth assignment to the variables such that φ is satisfied. Construct a solution to $(G_\varphi, \mathcal{X}_\varphi)$ as follows. If variable v_i is set to true by the truth assignment, then use L^-_i to connect v_i and w_i; otherwise, use L^+_i. Since φ is satisfied, each clause c_j has a literal that is true. Say this literal is v_i. Therefore, v_i is set to true by the truth assignment, and thus vertex $\ell^+_{i,j}$ is not used to connect v_i and w_i. Hence, it can be used to connect c_j and d_j. It can now be readily verified that $(G_\varphi, \mathcal{X}_\varphi)$ is a yes-instance. □

We are now ready to prove our first result.

Theorem 3 *The* VERTEX-DISJOINT PATHS *problem is NP-complete on split graphs.*

Proof We reduce from the NP-complete variant of SAT defined in Theorem 1. Let $\varphi = c_1 \wedge c_2 \wedge \ldots \wedge c_m$ be a CNF formula that satisfies the three conditions mentioned in Theorem 1, and let $v_1, \ldots, v_n$ be the variables that appear in φ. Let $(G_\varphi, \mathcal{X}_\varphi)$ be the instance of VERTEX-DISJOINT PATHS constructed from φ in the way described at the beginning of this section. Now let G be the graph obtained from G_φ by adding an edge between each pair of distinct literal vertices, i.e., by adding all the edges needed to make the literal vertices form a clique. The graph G clearly is a split graph.

We will show that $(G, \mathcal{X}_\varphi)$ is a yes-instance of VERTEX-DISJOINT PATHS if and only if $(G_\varphi, \mathcal{X}_\varphi)$ is a yes-instance of VERTEX-DISJOINT PATHS. Since $(G_\varphi, \mathcal{X}_\varphi)$ is a yes-instance of VERTEX-DISJOINT PATHS if and only if the formula φ is satisfiable due to Theorem 2, this suffices to prove the theorem.

If $(G_\varphi, \mathcal{X}_\varphi)$ is a yes-instance of VERTEX-DISJOINT PATHS, then so is $(G, \mathcal{X}_\varphi)$ due to the fact that G is a supergraph of G_φ. Hence, it remains to prove the reverse direction. Suppose $(G, \mathcal{X}_\varphi)$ is a yes-instance of VERTEX-DISJOINT PATHS. Let $\mathcal{P} = \{P_1, \ldots, P_n, Q_1, \ldots, Q_m\}$ be a minimum solution, where each path P_i connects the two terminals in the variable pair (v_i, w_i), and each path Q_j connects the terminals in the clause pair (c_j, d_j). We will show that all the paths in $\mathcal{P}$ exist also in the graph G_φ, implying that $\mathcal{P}$ is a solution for the instance $(G_\varphi, \mathcal{X}_\varphi)$.

Since $\mathcal{P}$ is a minimum solution and no two terminals coincide, it holds that every path in $\mathcal{P}$ is an induced path in G. By the construction of G, this implies that all the inner vertices of every path in $\mathcal{P}$ are literal vertices. Moreover, since the literal vertices form a clique in G, every path in $\mathcal{P}$ has at most two inner vertices.

Let $j \in \{1, \ldots, m\}$. Since $N_G(c_j) = N_G(d_j)$, the vertices c_j and d_j are non-adjacent, and Q_j is an induced path between c_j and d_j, the path Q_j must have length 2, and its only inner vertex is a literal vertex. Recall that we only added edges between distinct literal vertices when constructing the graph G from G_φ. Hence, the path Q_j exists in G_φ.

Now let $i \in \{1, \ldots, n\}$. We consider the path P_i between v_i and w_i. As we observed earlier, the path P_i contains at most two inner vertices, and all inner vertices of P_i are literal vertices. If P_i has exactly one inner vertex, then P_i exists in G_φ for the same reason as why the path Q_j from the previous paragraph exists in G_φ. Suppose P_i has two inner vertices. Recall the two vertex-disjoint paths L_i^+ and L_i^- between v_i and w_i, respectively, that were defined just above Theorem 2. Since v_i appears in at most three different clauses, at least once positively and at least once negatively, one of these paths has length 2, while the other path has length 2 or 3. Without loss of generality, suppose L_i^+ has length 2, and let ℓ denote the only inner vertex of L_i^+. Note that both v_i and w_i are adjacent to ℓ. Since P_i is an induced path from v_i to w_i with exactly two inner vertices, P_i cannot contain the vertex ℓ. From the construction of G, it is then clear that both inner vertices of P_i

must lie on the path L_i^-. This implies that L_i^- must have length 3, and that $P_i = L_i^-$. We conclude that the path P_i exists in G_φ. □

We now turn our attention to Edge-Disjoint Paths on split graphs. The following lemma will be used in the proof of Theorem 4 below.

Lemma 1 *Let $(G, \mathcal{X})$ be an instance of* Edge-Disjoint Paths. *Let $\mathcal{X}'$ be a subset of $\mathcal{X}$ such that for every pair $(s,t) \in \mathcal{X}'$, it holds that $d_G(s) = d_G(t) = 1$ and there is a (unique) path P_{st} of length 3 between s and t. If the paths in $\mathcal{P}' = \{P_{st} \mid (s,t) \in \mathcal{X}'\}$ are pairwise edge-disjoint and $(G, \mathcal{X})$ is a yes-instance, then there is a solution $\mathcal{P}$ for the instance $(G, \mathcal{X})$ such that $\mathcal{P}' \subseteq \mathcal{P}$.*

Proof Suppose the paths in $\mathcal{P}' = \{P_{st} \mid (s,t) \in \mathcal{X}'\}$ are pairwise edge-disjoint and $(G, \mathcal{X})$ is a yes-instance. Let $\mathcal{P}$ be a solution for $(G, \mathcal{X})$ that contains as many paths from $\mathcal{P}'$ as possible. We claim that $\mathcal{P}' \subseteq \mathcal{P}$. Suppose, for contradiction, that there exists a pair of terminals $(s,t) \in \mathcal{X}'$ such that $P_{st} \notin \mathcal{P}$. Let Q denote the path in $\mathcal{P}$ connecting s and t. Recall that both s and t have degree 1 in G, and let u and v be the unique neighbors of s and t, respectively. If none of the paths in $\mathcal{P}$ uses the edge uv, then the set $(\mathcal{P} \setminus Q) \cup P_{st}$ is a solution for $(G, \mathcal{X})$ containing more paths from $\mathcal{P}'$ than $\mathcal{P}$ does, contradicting the choice of $\mathcal{P}$. Hence, there must be a path $P^* \in \mathcal{P}$ that uses the edge uv. Let s^* and t^* be the two terminals that are connected by the path P^*. Let Q^* denote the path between s^* and t^* obtained from P^* by replacing the edge uv by the subpath of Q between u and v. Note that $P^* \notin \mathcal{P}'$, as otherwise the paths $P^* = P_{s^*t^*}$ and P_{st} would share the edge uv, contradicting the assumption that the paths in $\mathcal{P}'$ are pairwise edge-disjoint. Hence, the set obtained from $\mathcal{P}$ by replacing Q by the path P_{st} and replacing P^* by the path Q^* yields a solution for $(G, \mathcal{X})$ that contains one more path from $\mathcal{P}'$. This contradicts the choice of $\mathcal{P}$ and finishes the proof. □

We now prove the analogue of Theorem 3 for Edge-Disjoint Paths.

Theorem 4 *The* Edge-Disjoint Paths *problem is NP-complete on split graphs.*

Proof We reduce from Edge-Disjoint Paths on general graphs, which is well-known to be NP-complete [18]. Let $(G, \mathcal{X})$ be an instance of Edge-Disjoint Paths, where $\mathcal{X} = \{(s_1, t_1), \ldots, (s_k, t_k)\}$. Let G' be the graph obtained from G by adding, for every $i \in \{1, \ldots, k\}$, two new vertices s_i' and t_i' as well as two edges $s_i's_i$ and $t_i't_i$. Let $\mathcal{X}' = \{(s_1', t_1'), \ldots, (s_k', t_k')\}$. Clearly, $(G, \mathcal{X})$ is a yes-instance of Edge-Disjoint Paths if and only if $(G', \mathcal{X}')$ is a yes-instance of Edge-Disjoint Paths.

From G', we create a split graph G'' as follows. For every pair of vertices $u, v \in V(G)$ such that $uv \notin E(G)$, we add to G' the edge uv as well as two new terminals s_{uv} and t_{uv} that we connect to u and v respectively. Let G'' be the resulting graph, let $Q = \{(s_{uv}, t_{uv}) \mid u, v \in V(G), uv \notin E(G)\}$ be the terminal pairs that were added to G' to create G'', and let $\mathcal{X}'' = \mathcal{X}' \cup \mathcal{Q}$. We claim that $(G'', \mathcal{X}'')$ and $(G', \mathcal{X}')$ are equivalent instances of Edge-Disjoint Paths. Since G'' is a split graph, this suffices to prove the theorem.

Since G'' is a supergraph of G', it is clear that if $(G', \mathcal{X}')$ is a yes-instance of Edge-Disjoint Paths, then so is $(G'', \mathcal{X}'')$. For the reverse direction, suppose that $(G'', \mathcal{X}'')$ is a yes-instance. For every pair $(s_{uv}, t_{uv}) \in \mathcal{Q}$, let P_{uv} be unique path of length 3 in G'' between s_{uv} and t_{uv}, and let $\mathcal{P}'$ be the set consisting of these paths. By Lemma 1, there is a solution $\mathcal{P}$ for $(G'', \mathcal{X}'')$ such that $\mathcal{P}' \subseteq \mathcal{P}$. Note that the paths in $\mathcal{P}'$ contain all the edges that were added between non-adjacent vertices in G' in the construction of G''. This implies that for every $(s, t) \in \mathcal{X}'$, the path in $\mathcal{P}$ connecting s to t contains only edges that already existed in G'. Hence, $\mathcal{P} \setminus \mathcal{P}'$ is a solution for the instance $(G', \mathcal{X}')$. □

4 Two Polynomial Kernels

In this section, we present polynomial kernels for Vertex-Disjoint Paths and Edge-Disjoint Paths on split graphs, parameterized by the number of terminal pairs. We will denote instances of the parameterized version of the problems by $(G, \mathcal{X}, k)$, where $k = |\mathcal{X}|$ is the parameter.

Before we present the kernels, we introduce some additional terminology. Let $(G, \mathcal{X}, k)$ be an instance of either the Vertex-Disjoint Paths problem or the Edge-Disjoint Paths problem, where $\mathcal{X} = \{(s_1, t_1), \ldots, (s_k, t_k)\}$. Every vertex in $\{s_1, \ldots, s_k, t_1, \ldots, t_k\}$ is called a *terminal*. If $s_i = v$ (resp. $t_i = v$) for some $v \in V(G)$, then we say that s_i (resp. t_i) is a *terminal on* v; note that, in general, there can be more than one terminal on v. A vertex $v \in V(G)$ is a *terminal vertex* if there is at least one terminal on v, and v is a *non-terminal vertex* otherwise. Let $uv \in E(G)$. If there exists a subset $S \subseteq \{1, \ldots, k\}$ of size at least 2 such that $\{u, v\} = \{s_i, t_i\}$ for every $i \in S$, then we call uv a *heavy edge*, and $|S|$ is the *weight* of the edge uv. In other words, a heavy edge of weight s is an edge whose endpoints coincide with $s \geq 2$ pairs of terminals.

Recall that a solution $\mathcal{P}$ for an instance $(G, \mathcal{X}, k)$ of Vertex-Disjoint Paths is *minimum* if there is no solution $\mathcal{Q}$ for $(G, \mathcal{X})$ that uses strictly fewer edges of G. It is easy to see that if all the terminals in $\mathcal{X}$ are distinct, then every path in a minimum solution is an induced path. In general, this is not true if we allow (pairs of) terminals to coincide. The following lemma shows that if the input graph G is chordal, then the non-induced paths in a minimum solution are of a very restricted type. This lemma will be used in the correctness proof of two reduction rules in our kernelization algorithm for Vertex-Disjoint Paths on split graphs.

Lemma 2 *Let $(G, \mathcal{X}, k)$ be an instance of* Vertex-Disjoint Paths *such that G is a chordal graph. If $(G, \mathcal{X}, k)$ is a yes-instance and $\mathcal{P} = \{P_1, \ldots, P_k\}$ a minimum solution, then every path $P_i \in \mathcal{P}$ satisfies exactly one of the following two statements:*

- *P_i is an induced path;*

– *P_i is a path of length 2, and there exists a path $P_j \in \mathcal{P}$ of length 1 whose endpoints coincide with the endpoints of P_i.*

Proof Suppose $(G, \mathcal{X}, k)$ is a yes-instance, and let $\mathcal{P}$ be a minimum solution for this instance. If all the paths in $\mathcal{P}$ are induced, then the statement of the lemma holds. Suppose there is a path $P_i \in \mathcal{P}$ that is not induced. Then there exists an edge $xy \in E(G)$ such that $x, y \in V(P_i)$ and $xy \notin E(P_i)$. Since $\mathcal{P}$ is a minimum solution, there must exist a path $P_j \in \mathcal{P}$ such that $xy \in E(P_j)$. The paths P_i and P_j are vertex-disjoint, so by definition, x and y are the endpoints of both P_i and P_j. It remains to argue that P_i has length 2. For contradiction, suppose P_i has length at least 3. The path P_i together with the edge xy forms a cycle of length at least 4 in G. Since G is chordal, this cycle has a chord e such that at least one endpoint of e is an internal vertex of P_i. Since the paths in $\mathcal{P}$ are pairwise vertex-disjoint, the edge e is not used in any path in $\mathcal{P}$. But then we could have replaced P_i by a shorter path P_i', containing the edge e, to obtain another solution. This contradicts the minimality of $\mathcal{P}$. □

4.1 Polynomial Kernel for Vertex-Disjoint Paths on Split Graphs

Our kernelization algorithm for Vertex-Disjoint Paths on split graphs consists of four reduction rules. We fix a split partition (C^*, I^*) of the split graph G^* that forms part of the instance on which the kernelization algorithm is applied. In each of the rules below, we write $(G, \mathcal{X}, k)$ to denote the instance of Vertex-Disjoint Paths on which the rule is applied, and (C, I) denotes the unique split partition of G satisfying $I = V(G) \cap I^*$; from the description of the rules it will be clear that G indeed has such a split partition. The instance that is obtained after the application of the rule on $(G, \mathcal{X}, k)$ is denoted by $(G', \mathcal{X}', k')$. We say that a reduction rule is *safe* if $(G, \mathcal{X}, k)$ and $(G', \mathcal{X}', k')$ are equivalent instances of Vertex-Disjoint Paths.

Given an instance $(G, \mathcal{X}, k)$ of Vertex-Disjoint Paths and a split partition (C, I) of G, we can construct an auxiliary bipartite graph H as follows. Let T be the set of all terminal vertices in G. The vertex set of H consists of the independent set $I \setminus T$ and an independent set A that is constructed as follows. For every pair of vertices $v, w \in C$ such that vw is a heavy edge of weight $s \geq 2$, we add $s - 1$ vertices $a_1^{vw}, \ldots, a_{s-1}^{vw}$. The edge set of H is constructed by adding, for each $x \in I \setminus T$, an edge from x to vertices $a_1^{vw}, \ldots, a_{s-1}^{vw}$ if and only if x is adjacent to both v and w in G. An example is given in Figure 2.

Using graph H, we can now define our first reduction rule. Here, given a matching M of H, we say that $x \in I$ is *matched* by M if x is an endpoint of an edge in M.

Rule 1 *If there exists a non-terminal vertex in I, then we construct the bipartite graph H as described above, and find a maximum matching M in H. Let R be the set of vertices in $I \setminus T$ that are not matched by M. We set $G' = G - R$, $\mathcal{X}' = \mathcal{X}$, and $k' = k$.*

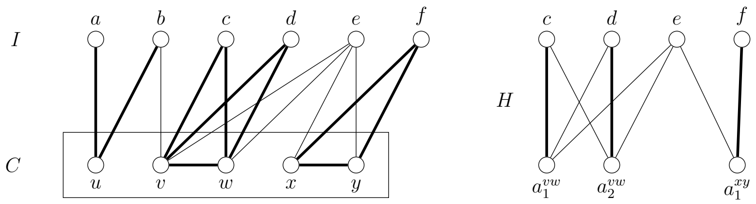

Fig. 2 The left picture shows a split graph G with split partition (C, I). To keep the picture clean, not all edges of C are drawn. Let $\mathcal{X} = \{(a, b), (v, w), (v, w), (v, w), (x, y), (x, y)\}$ and consider the instance $(G, \mathcal{X}, 6)$. Observe that edges vw and xy are heavy. The thick edges indicate a solution $\mathcal{P}$ for this instance. The right figure shows the auxiliary bipartite graph H for $(G, \mathcal{X}, 6)$. The thick edges indicate a matching M' that is induced by $\mathcal{P}$.

Lemma 3 *Rule 1 is safe.*

Proof It is clear that if $(G', \mathcal{X}', k')$ is a yes-instance of Vertex-Disjoint Paths, then $(G, \mathcal{X}, k)$ is also a yes-instance of Vertex-Disjoint Paths, as G is a supergraph of G'. For the reverse direction, suppose $(G, \mathcal{X}, k)$ is a yes-instance of Vertex-Disjoint Paths. Among all minimum solutions for this instance, let $\mathcal{P}$ be one for which the total number of vertices in R visited by the paths in $\mathcal{P}$ is as small as possible. We will show that $\mathcal{P}$ is a solution for $(G', \mathcal{X}', k')$, which suffices to prove the lemma.

The idea of the proof will be to show that $\mathcal{P}$ induces a matching M' in H. If a path in $\mathcal{P}$ uses a vertex in R, then by inspecting the symmetric difference of M and M', we can reroute some paths in $\mathcal{P}$ to avoid this vertex; in particular, we can reduce the total number of vertices of R that are visited by the paths in $\mathcal{P}$. Therefore, no path of $\mathcal{P}$ visits a vertex of R, proving that $\mathcal{P}$ is a solution for $(G', \mathcal{X}', k')$.

We first define a matching M' in H as follows. Consider any path $P_i \in \mathcal{P}$ that visits a vertex $x \in I \setminus T$. Since $x \notin T$, there are two vertices $v, w \in C$ such that the edges vx and xw appear consecutively on the path P_i. Hence, P_i is not induced. By Lemma 2, P_i has length 2, and its endpoints are v and w, which are also the endpoints of some path $P_j \in \mathcal{P}$ of length 1. In particular, this implies that vw is a heavy edge. Let $s \geq 2$ be the weight of vw, and consider the vertices $a_1^{vw}, \ldots, a_{s-1}^{vw}$ in the graph H. By construction, x is adjacent to each of these vertices. We arbitrarily choose an edge xa_j^{vw}, where $j \in \{1, \ldots, s-1\}$. We do the same for every other non-induced path in $\mathcal{P}$ whose middle vertex belongs to $I \setminus T$. By the construction of H, we can do this in such a way that the chosen edges form a matching. Let M' be such a matching. See Figure 2 for an illustration of such a matching M'.

Now suppose, for contradiction, that there exists a path $P_i \in \mathcal{P}$ that visits a vertex $x \in R$. Let v and w be the endpoints of P_i. As argued before, it follows from Lemma 2 that $P_i = vxw$. Hence, there exists an edge $xa_j^{vw} \in M'$ with $j \in \{1, \ldots, s-1\}$, where s is the weight of the edge vw. Let Q be a maximal path in H starting in x whose edges alternatingly belong to M' and M. Since no edge in M is incident with a vertex in R, we know that x is the only vertex of Q that belongs to R. Let $y \neq x$ be the other endpoint of Q. The last edge

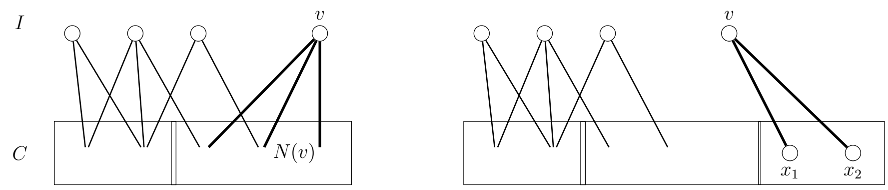

Fig. 3 Illustration of Rule 3. The left picture shows a split graph G with a split partition (C, I). Consider an instance on G with $k = 2$ such that v is a terminal of both pairs. Then $p = 2$. Since $d_G(v) = 3 \geq 2 = 2k - p$, we can indeed apply Rule 3 to v. The right picture shows the graph after the application of Rule 3.

of Q, i.e., the edge of Q incident with y, belongs to M, as otherwise Q would be an M-augmenting path in H, contradicting the fact that M is a maximum matching in H. In particular, this implies that $y \in I \setminus T$ and y is not incident with an edge in M'. By the definition of M', we know that y is not visited by any non-induced path of $\mathcal{P}$, and thus y is not visited by any path in $\mathcal{P}$.

Let $z_1, \ldots, z_\ell$ be the consecutive vertices of Q that belong to $I \setminus T$, such that $z_1 = x$ and $z_\ell = y$. For every $j \in \{1, \ldots, \ell - 1\}$, we replace the path $P \in \mathcal{P}$ whose middle vertex is z_j by the path obtained from P by replacing z_j by z_{j+1}. Since y is not visited by any path in $\mathcal{P}$, this yields a new solution $\mathcal{P}'$. In fact, $\mathcal{P}'$ is a minimum solution, as the total number of edges in all the paths did not change. Since the paths in $\mathcal{P}'$ visit one fewer vertex of R than the paths in $\mathcal{P}$, we arrive at the desired contradiction. We conclude that no path in $\mathcal{P}$ visits a vertex in R. Therefore, $\mathcal{P}$ is a solution for $(G', \mathcal{X}', k')$. □

Rule 2 *If there exists a non-terminal vertex $v \in I$ that is adjacent to a non-terminal vertex in C, then we set G' to be the graph obtained from G by deleting edge vw for every non-terminal vertex $w \in C$. We also set $\mathcal{X}' = \mathcal{X}$ and $k' = k$.*

Lemma 4 *Rule 2 is safe.*

Proof Suppose $v \in I$ is a non-terminal vertex that is adjacent to a non-terminal vertex $w \in C$. If $(G, \mathcal{X}, k)$ is a yes-instance and $\mathcal{P}$ is a minimum solution for this instance, then no path in $\mathcal{P}$ uses the edge vw as a result of Lemma 2. Hence, it is safe to delete any such edge vw from the graph. □

Rule 3 *If there exists a terminal vertex $v \in I$ with $d_G(v) \geq 2k-p$, where $p \geq 1$ is the number of terminals on v, then we set G' to be the graph obtained from G by deleting all edges incident with v, adding p new vertices $\{x_1, \ldots, x_p\}$, and making these new vertices adjacent to v, to each other, and to all the other vertices in C. We also set $\mathcal{X}' = \mathcal{X}$ and $k' = k$.*

See Figure 3 for an illustration of this rule.

Lemma 5 *Rule 3 is safe.*

Proof Suppose there exists a terminal vertex $v \in I$ with $d_G(v) \geq 2k-p$, where $p \geq 1$ is the number of terminals on v. Let $X = \{x_1, \ldots, x_p\}$ be the set of vertices that were added during the execution of the rule. Let $Y = \{y_1, \ldots, y_p\}$ be the set of terminals on v.

First suppose $(G, \mathcal{X}, k)$ is a yes-instance of Vertex-Disjoint Paths, and let $\mathcal{P} = \{P_1, \ldots, P_k\}$ be an arbitrary solution for this instance. We construct a solution $\mathcal{P}' = \{P_1', \ldots, P_k'\}$ for $(G', \mathcal{X}', k')$ as follows. Let $i \in \{1, \ldots, k\}$. First suppose that neither of the terminals in the pair (s_i, t_i) belongs to the set Y. Since the paths in $\mathcal{P}$ are pairwise vertex-disjoint and v is a terminal vertex, the path P_i does not contain an edge incident with v. Hence, P_i exists in G', and we set $P_i' = P_i$. Now suppose $v \in \{s_i, t_i\}$, and recall that $s_i \neq t_i$ by definition. Suppose, without loss of generality, that $v = s_i$. Then $s_i \in Y$, so $s_i = y_r$ for some $r \in \{1, \ldots, p\}$. Let vw be the first edge of the path P_i in G. We define P_i' to be the path in G' obtained from P_i by deleting the edge vw and adding the vertex x_r as well as the edges vx_r and $x_r w$. Let $\mathcal{P}' = \{P_1', \ldots, P_k'\}$ denote the collection of paths in G' obtained this way. Since the paths in $\mathcal{P}$ are pairwise vertex-disjoint in G, and every vertex in $\{x_1, \ldots, x_p\}$ is visited by exactly one path in $\mathcal{P}'$, it holds that the paths in $\mathcal{P}'$ are pairwise vertex-disjoint in G'. Hence, $\mathcal{P}'$ is a solution for the instance $(G', \mathcal{X}', k')$.

For the reverse direction, suppose $(G', \mathcal{X}', k')$ is a yes-instance of Vertex-Disjoint Paths, and let $\mathcal{Q} = \{Q_1, \ldots, Q_k\}$ be a minimum solution. Let $\mathcal{Q}^* \subseteq \mathcal{Q}$ be the set of paths in $\mathcal{Q}$ that visit a vertex in the set $X = \{x_1, \ldots, x_p\}$. Since there are p terminals on v, and v has exactly p neighbors in G' (namely, the vertices of X), every path in $\mathcal{Q}^*$ has v as one of its endpoints and $|\mathcal{Q}^*| = p$. Now consider the paths in $\mathcal{Q} \setminus \mathcal{Q}^*$. Due to Lemma 2, each such path visits at most three vertices of $N_G(v)$. If a path in $\mathcal{Q} \setminus \mathcal{Q}^*$ is induced, it visits at most two vertices of $N_G(v)$. If a path $Q_i \in \mathcal{Q} \setminus \mathcal{Q}^*$ visits three vertices in $N_G(v)$, then Lemma 2 ensures that there is another path $Q_j \in \mathcal{Q} \setminus \mathcal{Q}^*$ such that Q_j has length 1 and the endpoints of Q_j coincide with the endpoints of Q_i, implying that Q_i and Q_j together visit at most three vertices of $N_G(v)$. We deduce that at most $2(k-p)$ vertices of $N_G(v)$ are visited by the $k-p$ paths in $\mathcal{Q} \setminus \mathcal{Q}^*$. Recall that $d_G(v) \geq 2k - p$. Therefore, at least p vertices of $N_G(v)$, say $z_1, \ldots, z_p$, are not visited by any path in $\mathcal{Q} \setminus \mathcal{Q}^*$.

Armed with the above observations, we construct solution $\mathcal{P} = (P_1, \ldots, P_k)$ for $(G, \mathcal{X}, k)$ as follows. For every path $Q_i \in \mathcal{Q} \setminus \mathcal{Q}^*$, we define $P_i = Q_i$. Now let $Q_i \in \mathcal{Q}^*$. The path Q_i visits v, one vertex $x_\ell \in X$, one vertex $z \in C$, and possibly one (terminal) vertex in $I \setminus \{v\}$. If $z \in N_G(v)$, then we define P_i to be the path in G obtained from Q_i by deleting x_ℓ and its two incident edges and adding the edge vz. If $z \notin N_G(v)$, then we define P_i to be the path obtained from Q_i by replacing the vertex x_ℓ by z_ℓ. It is easy to verify that $\mathcal{P}$ is a solution for the instance $(G, \mathcal{X}, k)$. □

Rule 4 *If there exist more than $k-1$ non-terminal vertices in C that have no neighbors in I, then we set G' to be the graph obtained from G by deleting all but $k-1$ of those vertices. We also set $\mathcal{X}' = \mathcal{X}$ and $k' = k$.*

Lemma 6 *Rule 4 is safe.*

Proof As in the proof of Lemma 3, the fact that G is a supergraph of G' implies that $(G, \mathcal{X}, k)$ is a yes-instance of Vertex-Disjoint Paths if $(G', \mathcal{X}', k')$ is.

For the reverse direction, suppose $(G, \mathcal{X}, k)$ is a yes-instance, and let $\mathcal{P}$ be a minimum solution. Let v be a non-terminal vertex in C that has no neighbors in I. Due to Lemma 2, the only kind of path in $\mathcal{P}$ that can visit v is a path of length 2 whose endpoints are the endpoints of a heavy edge. Observe that there are at most $k-1$ such paths, as for every heavy edge pq in G, there is exactly one path in $\mathcal{P}$ of length 1 whose endpoints are p and q. Since all non-terminals in C that have no neighbors in I are true twins, we can safely delete all but $k-1$ of them from the graph. □

We now prove that the above four reduction rules yield a quadratic vertex kernel for Vertex-Disjoint Paths on split graphs.

Theorem 5 *The* Vertex-Disjoint Paths *problem on split graphs has a kernel with at most* $4k^2$ *vertices and (thus) at most* $8k^4$ *edges, where* k *is the number of terminal pairs.*

Proof We describe a kernelization algorithm for Vertex-Disjoint Paths on split graphs. Let $(G^*, \mathcal{X}, k)$ be an instance of Vertex-Disjoint Paths, where G^* is a split graph. We fix a split partition (C^*, I^*) of G^*. We start by applying Rule 1 once. We then exhaustively apply Rules 2 and 3. Finally, we apply Rule 4 once. Let $(G', \mathcal{X}', k')$ be the obtained instance. From the description of the reduction rules it is clear that G' is a split graph, and that $\mathcal{X}' = \mathcal{X}$ and $k' = k$. By Lemmas 3–6, $(G', \mathcal{X}', k')$ is a yes-instance of Vertex-Disjoint Paths if and only if $(G^*, \mathcal{X}, k)$ is a yes-instance. Hence, the algorithm indeed reduces any instance of Vertex-Disjoint Paths to an equivalent instance.

We now show that $|V(G')| \leq 4k^2$. Let (C', I') be the unique split partition of G' such that $I' = V(G') \cap I^*$, i.e., the independent set I' contains exactly those vertices of I^* that were not deleted during any application of the reduction rules. Note that this split partition indeed exists by the description of the reduction rules. We distinguish two cases.

First suppose that every vertex in I' is a terminal vertex. Then $|I'| \leq 2k$, and since Rule 3 cannot be applied, every vertex in I' has degree at most $2k-2$. This implies that C' contains at most $2k(2k-2)$ vertices that have at least one neighbor in I'. Since Rule 4 cannot be applied, there are at most $k-1$ vertices in C' that have no neighbor in I', and thus $|C'| \leq 4k^2 - 3k - 1$. This implies that $|V(G')| = |I'| + |C'| \leq 4k^2$, as desired.

Now suppose that I' contains at least one non-terminal vertex. Recall the graph $H = (I^* \setminus T, A, F)$ and the maximum matching M in H that are constructed during the execution of Rule 1. The assumption that I' contains a non-terminal vertex implies that $|M| \geq 1$. Each edge in M corresponds to a non-terminal vertex in I' and a terminal pair in C', and there is at least one heavy edge whose endpoints both belong to C'. This implies that I' contains $|M|$ non-terminal vertices, and there are at least $|M|+1$ terminal pairs that belong to C'. Hence, there are at most $2(k - (|M|+1))$ terminal vertices in I', and $|I'| \leq |M| + 2(k - (|M|+1)) \leq 2k - |M| - 2 \leq 2k - 3$, where the last inequality follows from the assumption that $|M| \geq 1$. Since Rules 2 and 3 cannot be applied, every vertex in I' has degree at most $2k$. This implies that

C' contains at most $2k(2k-3)$ vertices that have at least one neighbor in I', and C' contains at most $k-1$ other vertices due to Rule 4. We conclude that $|C'| \leq 4k^2 - 5k - 1$, and thus $|V(G')| \leq 4k^2$.

It remains to argue that the above algorithm runs in polynomial time. Recall that a split partition of G^* can be found in linear time [12]. Rules 1 and 4 are applied only once. Rules 2 and 3 together are applied at most $|I^*|$ times in total, as each vertex in I^* that is not deleted during the execution of Rule 1 is considered in at most one of these rules. Rules 2, 3, and 4 can trivially be executed in polynomial time. The same holds for Rule 1, since it only takes polynomial time to construct the auxiliary bipartite graph H and find a maximum matching M in H. We conclude that the overall running time of the kernelization algorithm is polynomial. □

Note that because the number of vertices of the clique in the reduced graph G' contributes the most to the total number of vertices in G', we cannot prove a significantly better bound on the number of edges of G' than the trivial bound: $8k^4$. This means that the actual size of the kernel will be $O(k^4)$, even though the number of vertices is significantly smaller.

4.2 Polynomial Kernel for Edge-Disjoint Paths on Split Graphs

In this section, we present a cubic vertex kernel for the Edge-Disjoint Paths problem on split graphs. We need the following two structural lemmas.

Lemma 7 *Let $(G, \mathcal{X}, k)$ be an instance of* Edge-Disjoint Paths *such that G is a complete graph. If $|V(G)| \geq 2k$, then $(G, \mathcal{X}, k)$ is a yes-instance.*

Proof An edge $vw \in E(G)$ is called *occupied by the terminal pair* (s_i, t_i) if there is a terminal pair (s_i, t_i) such that $\{v, w\} = \{s_i, t_i\}$; if vw is occupied by some terminal pair, we simply call it *occupied*. Recall that an edge vw is *heavy* if it is occupied by more than one terminal pair. If vw is occupied by exactly one terminal pair, then we call vw a *light* edge. Denote the number of light edges by ℓ_1 and the number of heavy edges by ℓ_2. Observe that $\ell_1 + 2\ell_2 \leq k$.

We claim that there exists a solution $\mathcal{P} = \{P_1, \ldots, P_k\}$ for the instance $(G, \mathcal{X}, k)$ such that every path in $\mathcal{P}$ contains at most three vertices, and we will construct such a solution below.

For every terminal pair $(s_i, t_i) \in \mathcal{X}$ that occupies a light edge, we define P_i to be the path whose only edge is $s_i t_i$. For every heavy edge vw, we arbitrarily choose a terminal pair (s_i, t_i) that occupies it, and again define P_i to be the path whose only edge is $vw = s_i t_i$. Let $\mathcal{P}'$ be the set of paths P_i that we have defined so far, and let $\mathcal{X}'$ consist of the corresponding terminal pairs in $\mathcal{X}$. Note that every path in $\mathcal{P}'$ contains exactly two vertices. Every path in $\mathcal{P} \setminus \mathcal{P}'$ will contain exactly three vertices. For every terminal pair $(s_i, t_i) \in \mathcal{X} \setminus \mathcal{X}'$, the middle vertex of the path $P_i \in \mathcal{P} \setminus \mathcal{P}'$ is called the *bouncer* for (s_i, t_i). We now describe how we can construct the set of all bouncers.

For each terminal pair $(s_i, t_i) \in \mathcal{X} \setminus \mathcal{X}'$, we choose an arbitrary vertex of $V(G) \setminus \{s_i, t_i\}$ that is not incident with any occupied edge as the bouncer for (s_i, t_i), and we do this in such a way that the bouncers for any two pairs in $\mathcal{X} \setminus \mathcal{X}'$ are distinct. To see why this is possible, we first observe that we need to choose exactly $|\mathcal{X} \setminus \mathcal{X}'| = k - \ell_1 - \ell_2$ bouncers. Since there are $\ell_1 + \ell_2$ occupied edges in G, there are at least $2k - 2(\ell_1 + \ell_2) = 2(k - \ell_1 - \ell_2)$ vertices of G that are not incident with any occupied edge. Recall that $\ell_1 + 2\ell_2 \leq k$, which implies that $k - \ell_1 - \ell_2 \geq \ell_2 \geq 0$ and thus $2(k - \ell_1 - \ell_2) \geq k - \ell_1 - \ell_2$. This means that we can indeed choose a unique bouncer for each terminal pair in $\mathcal{X} \setminus \mathcal{X}'$ in the way described above.

Now, for each terminal pair $(s_i, t_i) \in \mathcal{X} \setminus \mathcal{X}'$, let P_i be the path from s_i to its selected bouncer to t_i. Note that the paths in $\mathcal{P} = (P_1, \ldots, P_k)$ are pairwise edge-disjoint due the way we chose the bouncers, implying that $(G, \mathcal{X}, k)$ is a yes-instance. □

Lemma 8 *Let $(G, \mathcal{X}, k)$ be an instance of* EDGE-DISJOINT PATHS *such that G is a split graph with split partition (C, I), and the degree of each terminal vertex is at least the number of terminals on it. If $|C| \geq 2k$, then $(G, \mathcal{X}, k)$ is a yes-instance.*

Proof The proof of this lemma consists of two steps: project to C, and route within C. In the first step, we project the terminals to C. Consider any terminal vertex $x \in I$. For each terminal on x, we project it to a neighbor of x in such a way that no two terminals on x are projected to the same vertex; if the terminal is s_i, denote this neighbor by s'_i, and if the terminal is t_i, denote this neighbor by t'_i. Since the degree of every terminal vertex is at least the number of terminals on it, this is indeed possible. For any terminal s_i that is on a terminal vertex in C, let $s'_i = s_i$, and for any terminal t_i that is on a terminal vertex in C, let $t'_i = t_i$. Let $\mathcal{X}' = \{(s'_i, t'_i) \mid i = 1, \ldots, k\}$, and let $G' = G - I$.

Since G' is a complete graph and $|V(G')| = |C| \geq 2k$, there exists a solution $\mathcal{P}''$ for the instance $(G', \mathcal{X}'', k'')$ due to Lemma 7, where $\mathcal{X}'' = \{(s'_i, t'_i) \in \mathcal{X}' \mid s'_i \neq t'_i\}$ and $k'' = |\mathcal{X}''|$. Let $\mathcal{P}' = \{P'_1, \ldots, P'_k\}$ be the collection of paths obtained from $\mathcal{P}''$ as follows. For every pair $(s'_i, t'_i) \in \mathcal{X}''$, we define P'_i to be the path in $\mathcal{P}''$ connecting s'_i and t'_i. For every pair (s'_i, t'_i) with $s'_i = t'_i$, we define P'_i to be the trivial path consisting of the single vertex s'_i. We now show that we can extend the paths in $\mathcal{P}'$ to obtain a solution $\mathcal{P}$ for the instance $(G, \mathcal{X}, k)$. For every $i \in \{1, \ldots, k\}$, we extend the path P'_i using the edges $s_i s'_i$ (if $s_i \neq s'_i$) and $t_i t'_i$ (if $t_i \neq t'_i$); let the resulting path be P_i. Since for every terminal vertex $x \in I$, no two terminals on x were projected to the same neighbor of x, the paths in $\mathcal{P}$ are pairwise edge-disjoint. We conclude that $(G, \mathcal{X}, k)$ is a yes-instance. □

Apart from the above two lemmas, our kernelization algorithm for EDGE-DISJOINT PATHS on split graphs will use one reduction rule. Before formulating this rule, we first define an auxiliary bipartite graph H' that is similar to but different from the graph H used in Rule 1 in the previous subsection. Given

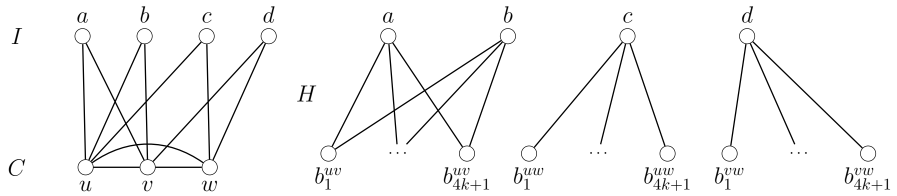

Fig. 4 The left picture shows a split graph G with split partition (C, I). Consider any instance on G such that no vertex of I is a terminal vertex. The right figure then shows the corresponding auxiliary bipartite graph H.

an instance $(G, \mathcal{X}, k)$ of Edge-Disjoint Paths and a split partition (C, I) of G, we define H' to be the bipartite graph whose vertex set consists of the independent set $I \setminus T$ and an independent set B that contains $4k+1$ vertices $b_1^{vw}, \dots, b_{4k+1}^{vw}$ for each pair v, w of vertices of C. For each $x \in I \setminus T$, we add edges from x to all of the vertices $b_1^{vw}, \dots, b_{4k+1}^{vw}$ if and only if x is adjacent to both v and w in G. An example is given in Figure 4.

We are now ready to formulate the reduction rule. The rule takes as input an instance $(G, \mathcal{X}, k)$ of Edge-Disjoint Paths and a split partition (C, I) of G, and it returns an instance $(G', \mathcal{X}', k')$ of Edge-Disjoint Paths.

Rule A *If there exists a non-terminal vertex in I, then we construct the bipartite graph H' as described above, and find a maximal matching M in H'. Let R be the set of vertices in $I \setminus T$ that are not matched by M. We set $G' = G - R$, $\mathcal{X}' = \mathcal{X}$, and $k' = k$.*

Lemma 9 *Rule A is safe.*

Proof It is clear that if $(G', \mathcal{X}', k')$ is a yes-instance of Edge-Disjoint Paths, then $(G, \mathcal{X}, k)$ is also a yes-instance of Edge-Disjoint Paths, as G is a supergraph of G'. For the reverse direction, suppose that $(G, \mathcal{X}, k)$ is a yes-instance of Edge-Disjoint Paths. Note that there exists a solution for $(G, \mathcal{X}, k)$ such that no path in the solution visits a vertex more than once. Among all such solutions, let $\mathcal{P} = (P_1, \dots, P_k)$ be one for which the total number of visits by all paths combined to vertices from R is minimized. We claim that no path in $\mathcal{P}$ visits a vertex in R.

For contradiction, suppose that some path $P_j \in \mathcal{P}$ visits some vertex $r \in R$. Since $r \notin T$, there are two vertices $v, w \in C$ such that the edges vr and wr appear consecutively on the path P_j. As $r \in R$, it is not matched by the maximal matching M used in Rule A. Since r is adjacent to all the vertices in $\{b_1^{vw}, \dots, b_{4k+1}^{vw}\}$ and M is a maximal matching, all the vertices in $\{b_1^{vw}, \dots, b_{4k+1}^{vw}\}$ are matched by M, and consequently at least $4k+1$ vertices of $I \setminus T$ that are adjacent to both v and w are matched by M. Let Z denote this set of vertices. By the choice of $\mathcal{P}$, no path of $\mathcal{P}$ visits a vertex twice. Hence, there are at most $4k$ edges of $\bigcup_{i=1}^{k} E(P_i)$ incident with v or w in G. Therefore, there exists a vertex $z \in Z$ such that $\bigcup_{i=1}^{k} E(P_i)$ contains neither the edge vz nor the edge wz. Let P'_j be the path obtained from P_j by replacing r with z and shortcutting it if necessary (i.e., if $z \in V(P_j)$). Then,

$\mathcal{P}' = (P_1, \ldots, P_{j-1}, P'_j, P_{j+1}, \ldots, P_k)$ is a solution for $(G, \mathcal{X}, k)$ where each path visits each vertex at most once, and where the total number of visits by all paths combined to vertices from R is at least one smaller than $\mathcal{P}$, contradicting the choice of $\mathcal{P}$. Therefore, no path of $\mathcal{P}$ visits a vertex of R. Hence, $\mathcal{P}$ is also a solution for $(G', \mathcal{X}', k')$, and thus it is a yes-instance. □

We now present our second kernelization result.

Theorem 6 *The* Edge-Disjoint Paths *problem on split graphs has a kernel with at most* $8k^3$ *vertices and at most* $16k^4$ *edges, where* k *is the number of terminal pairs.*

Proof We describe a kernelization algorithm for Edge-Disjoint Paths on split graphs. Let $(G, \mathcal{X}, k)$ be an instance of Edge-Disjoint Paths, where G is a split graph on n vertices and m edges. We assume that $k \geq 1$, and we fix a split partition (C, I) of G. If the degree of any terminal vertex is less than the number of terminals on it, then $(G, \mathcal{X}, k)$ is clearly a no-instance, and the kernelization algorithm outputs a trivial no-instance. Hence, we may assume that the degree of any terminal vertex is at least the number of terminals on it. If $|C| \geq 2k$, then the algorithm returns a trivial yes-instance, which is safe due to Lemma 8. Suppose $|C| \leq 2k - 1$. If every vertex in I is a terminal vertex, then $|I| \leq 2k$ and thus $|V(G)| \leq 4k - 1$ and $|E(G)| \leq 8k^2$, so we simply return the current instance as the desired kernel.

Suppose I contains at least one non-terminal vertex. We apply Rule A. Let $(G', \mathcal{X}', k')$ denote the resulting instance. By Lemma 9, $(G', \mathcal{X}', k')$ is a yes-instance of Edge-Disjoint Paths if and only if $(G, \mathcal{X}, k)$ is a yes-instance. Let (C', I') be the split partition of $V(G')$ such that $C' \subseteq C$ and $I' \subseteq I$. Since Rule A does not change the clique part of the instance, $C = C'$, and in particular, $|C'| = |C| \leq 2k - 1$ by assumption. By the construction of H', any maximal matching of H' has size at most $|B| = (4k+1) \cdot \binom{|C|}{2} \leq 8k^3 - 10k^2 + k + 1 \leq 8k^3 - 4k$. Hence, I' contains at most $8k^3 - 4k$ non-terminal vertices, and consequently $|I'| \leq 8k^3 - 2k$. We conclude that $|V(G')| = |C'| + |I'| \leq 8k^3$. To bound $|E(G')|$, we recall that C' is a clique and I' is an independent set. Therefore, $|E(G')| \leq \binom{|C'|}{2} + |C'| \cdot |I'| \leq 16k^4$.

It remains to argue that the above algorithm runs in polynomial time. Recall that a split partition of G can be found in linear time [12]. Checking the degrees of terminal vertices can be done in linear time. Therefore, it remains to analyze the time needed to apply Rule A. Note that H' has $O(n + k \cdot \binom{|C|}{2}) = O(n + k^3)$ vertices, and we can add all edges in $O(k \cdot \binom{|C|}{2} n) = O(k^3 n)$ time. Since finding a maximal matching takes linear time, the overall running time is indeed polynomial. □

Note that, as the number of edges of the kernel is at most $16k^4$, the actual size of the kernel will be $O(k^4)$, even though the number of vertices is significantly smaller.

5 Conclusion

We proved that Vertex-Disjoint Paths and Edge-Disjoint Paths admit kernels with $O(k^2)$ and $O(k^3)$ vertices, respectively, when restricted to split graphs. It would be interesting to investigate whether or not the problems admit *linear* vertex kernels on split graphs. Another interesting open question is whether either problem admits a polynomial kernel on chordal graphs, a well-known superclass of split graphs.

Recall that Bodlaender et al. [3] proved that Vertex-Disjoint Paths does not admit a polynomial kernel on general graphs, unless NP $\subseteq$ coNP/poly. They asked whether or not the problem admits a polynomial kernel when restricted to planar graphs. One could ask the same question about the Edge-Disjoint Paths problem. As far as we know, even the kernelization complexity of Edge-Disjoint Paths on general graphs is still open.

References

1. Bodlaender, H.L., Downey, R.G., Fellows, M.R., Hermelin, D.: On problems without polynomial kernels. J. Comp. Syst. Sci. 75(8), 423–434 (2009)
2. Bodlaender, H.L., Fomin, F.V., Lokshtanov, D., Penninkx, E., Saurabh, S., Thilikos, D.M.: (Meta) kernelization. In: 50th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2009, pp. 629–638. IEEE Computer Society (2009)
3. Bodlaender, H.L., Thomasse, S., Yeo, A.: Kernel bounds for disjoint cycles and disjoint paths. Theor. Comp. Sci. 412(35), 4570–4578 (2011)
4. Brandstädt, A., Le, V. B., Spinrad, J. Graph Classes: A Survey. SIAM Monographs on Discrete Mathematics and Applications (1999)
5. Diestel, R.: Graph Theory. Springer-Verlag, Electronic Edition (2005)
6. Downey, R.G., Fellows, M.R.: Parameterized Complexity. Monographs in Computer Science, Springer (1999)
7. Dvorak, Z., Král', D., Thomas, R.: Three-coloring triangle-free planar graphs in linear time. In: Mathieu, C. (ed.), SODA 2009, pp. 1176–1182. ACM-SIAM (2009)
8. Even, S., Itai, A., Shamir, A.: On the complexity of timetable and multicommodity flow problems. SIAM J. Comp. 5, 691–703 (1976)
9. Frank, A.: Packing paths, circuits, and cuts – a survey. In: Korte, B., Lovász, L., Prömel, H.J., Schrijver, A. (eds.), Paths, Flows, and VLSI-Layout, pp. 47–100. Springer-Verlag, Berlin (1990)
10. Golumbic, M.C.: Algorithmic Graph Theory and Perfect Graphs. Annals of Disc. Math. 57, Elsevier (2004).
11. Gurski, F., Wanke, E.: Vertex disjoint paths on clique-width bounded graphs. Theor. Comput. Sci. 359, 188–199 (2006)
12. Hammer, P.L., Simeone, B.: The splittance of a graph. Combinatorica 1, 275–284 (1981)
13. Heggernes, P., van 't Hof, P., van Leeuwen, E.J., Saei, R.: Finding disjoint paths in split graphs. In: Geffert, V., Preneel, B., Rovan, B., Stuller, J., Tjoa, A.M. (eds.), SOFSEM 2014. LNCS, vol. 8327, pp. 315–326 (2014)
14. Kammer, F., Tholey, T.: The k-disjoint paths problem on chordal graphs. In: Paul, C., Habib, M. (eds.), WG 2009. LNCS, vol. 5911, pp. 190–201 (2010)
15. Karp, R.M.: On the complexity of combinatorial problems. Networks 5, 45–68 (1975)
16. Kawarabayashi, K., Kobayashi, Y., Reed, B.A.: The disjoint paths problem in quadratic time. J. Comb. Theory B 102, 424–435 (2012)
17. Kobayashi, Y., Kawarabayashi, K.: Algorithms for finding an induced cycle in planar graphs and bounded genus graphs. In: Mathieu, C. (ed.), SODA 2009, pp. 1146–1155. ACM-SIAM (2009)

18. Kramer, M., van Leeuwen, J.: The complexity of wirerouting and finding minimum area layouts for arbitrary VLSI circuits. Adv. Comput. Res. 2, 129–146 (1984)
19. Lynch, J.F.: The equivalence of theorem proving and the interconnection problem ACM SIGDA Newsletter 5(3), 31–36 (1975)
20. Natarajan, S., Sprague, A.P.: Disjoint paths in circular arc graphs. Nordic J. Comp. 3, 256–270 (1996)
21. Nishizeki, T., Vygen, J., Zhou, X.: The edge-disjoint paths problem is NP-complete for series-parallel graphs. Discrete Applied Math. 115, 177–186 (2001)
22. Reed, B.A.: Tree width and tangles: A new connectivity measure and some applications. In: R.A. Bailey (ed.), Surveys in Combinatorics, pp. 87–162. Cambridge University Press (1997)
23. Reed, B.A., Robertson, N., Schrijver, A., Seymour, P.D.: Finding disjoint trees in planar graphs in linear time. In: Contemp. Math., vol. 147, pp. 295–301. Amer. Math. Soc., Providence, RI (1993)
24. Robertson, N., Seymour, P.D.: Graph minors XIII. The disjoint paths problem. J. Comb. Theory B 63(1), 65–110 (1995)
25. Tovey, C.A.: A simplified NP-complete satisfiability problem. Discrete Applied Math. 8, 85–89 (1984)
26. Vygen, J.: Disjoint paths. Technical report 94816, Research Institute for Discrete Mathematics, University of Bonn (1998)