



Parameterized Complexity of Path Set Packing

N. R. Aravind and Roopam Saxena^(✉)

Department of Computer Science and Engineering, IIT Hyderabad, Hyderabad, India
aravind@cse.iith.ac.in, cs18resch11004@iith.ac.in

Abstract. In PATH SET PACKING, the input is an undirected graph G , a collection \mathcal{P} of simple paths in G , and a positive integer k . The problem is to decide whether there exist k edge-disjoint paths in \mathcal{P} . We study the parameterized complexity of PATH SET PACKING with respect to both natural and structural parameters. We show that the problem is $W[1]$ -hard with respect to vertex cover plus the maximum length of a path in \mathcal{P} , and $W[1]$ -hard with respect to pathwidth plus maximum degree plus solution size. These results answer an open question raised in [17]. On the positive side, we present an FPT algorithm parameterized by feedback vertex set plus maximum degree, and also provide an FPT algorithm parameterized by treewidth plus maximum degree plus maximum length of a path in \mathcal{P} .

Keywords: Path set packing · Set packing · Parameterized complexity · Fixed parameter tractability · Graph algorithms

1 Introduction

Xu and Zhang [17] introduced the PATH SET PACKING (PSP) problem and discussed its various applications, such as in software defined networks. The problem asks if for a given graph $G = (V, E)$ and a collection \mathcal{P} of simple paths in G , there exists a set $S \subseteq \mathcal{P}$ of edge disjoint paths such that $|S| \geq k$. PSP is closely related to the well known SET PACKING problem. Xu and Zhang [17] showed that PSP is NP -complete even when the maximum length of the given paths is 3. Considering the optimization version of the problem, they showed that PSP is hard to approximate within $O(|E|^{\frac{1}{2}-\epsilon})$ unless $NP = ZPP$. They showed that PSP can be solved in polynomial time when the input graph is a tree and gave a parameterized algorithm with running time $O(|\mathcal{P}|^{tw(G)\Delta}|V|)$ where $tw(G)$ is treewidth of G and Δ is maximum degree. Further, they left open the question whether PSP is fixed parameter tractable with respect to treewidth of the input graph.

PATH SET PACKING:

Input: An instance $I = (G, \mathcal{P}, k)$, where $G = (V, E)$ is an undirected graph, \mathcal{P} is a collection of simple paths in G , and $k \in \mathbb{N}$.

Output: YES if there is a set of k pairwise edge-disjoint paths in \mathcal{P} , NO otherwise.

1.1 Related Work

In SET PACKING we are given a list S of subsets of a universe U and it is asked if S has k pairwise disjoint sets. The SET PACKING problem is W[1]-hard when parameterized by solution size k [6]. For the maximum size of a set d , FPT algorithms for the combined parameter of k and d have been obtained [14, 15]. Kernel of size $O(k^{d-1})$ [1] has also been obtained. Since PSP can be seen as a special case of SET PACKING, all the positive results obtained for SET PACKING are also applicable to PSP.

Path Set Packing (PSP) can also be seen as the problem of finding a Maximum Independent Set on the *conflict graph* obtained by considering each path as a vertex with two vertices being adjacent if the corresponding paths share an edge. When the input graph to a PSP instance is a grid graph, the corresponding conflict graphs are called EPG graphs [12]. It was shown in [12] that every graph is an EPG graph. This immediately implies the following, because of well-known results on MIS on general graphs.

Corollary 1. *PSP is W[1]-hard on Grid graphs when parameterized by solution size k .*

Corollary 2. *PSP doesn't admit a $2^{o(|\mathcal{P}|)}$ time exact algorithm, assuming ETH.*

Thus, it is natural to consider PSP with further or different restrictions on the input graph G . We mention some known results of this type.

1. When G is a tree, the conflict graph is called an EPT graph [11]. Recognizing EPT graphs is NP-Complete [10]; nevertheless MIS is solvable in polynomial time on the class of EPT graphs [16].
2. The class of B_k -EPG graphs was defined as graphs obtained as the edge intersection graph of paths on a grid, with the restriction that each path have at most k bends. MIS on B_1 -EPG graphs is NP-hard [7]. In [3], the authors showed that for the class of B_1 -EPG graphs, when the number of path shapes is restricted to three, MIS admits an FPT algorithm, while remaining W[1]-hard on B_2 -EPG graphs.

1.2 Our Results

We studied PSP with respect to combination of both natural and structural parameters of the input graph G and obtained following results for PSP.

Theorem 1. *PSP is $W[1]$ -hard when parameterized by vertex cover number of G + maximum length of a path in \mathcal{P} .*

Theorem 2. *PSP is $W[1]$ -hard when parameterized by pathwidth of G + maximum degree of G + solution size.*

Theorem 3. *PSP admits an FPT algorithm when parameterized by feedback vertex number of G + maximum degree of G .*

Theorem 4. *PSP admits an FPT algorithm when parameterized by treewidth of G + maximum degree of G + maximum length of a path in \mathcal{P} .*

We note that the above positive results complement the hardness of PSP with respect to any subset of the parameters used in the respective algorithms. The hardness with respect to treewidth plus maximum path length, and hardness with respect to treewidth plus maximum degree are implied by (Theorem 1 and 2). The hardness with respect to feedback vertex number is implied by (Theorem 1). And for maximum degree plus maximum path length, we note that the reduction from maximum independent set to PSP given in [17] to prove the inapproximability of PSP also works to prove NP-hardness of PSP for bounded maximum degree in G and bounded maximum length of a path in \mathcal{P} using the fact that independent set is NP-hard on bounded degree graphs [9].

2 Preliminaries

We use $[n]$ to denote the set $\{1, 2, \dots, n\}$. For a sequence ρ with n elements, $set(\rho)$ denotes the set of all the elements of ρ , and for $j \in [n]$, $\rho[j]$ is the element of ρ at position j . All the graphs considered in this paper are simple and finite. We use standard graph notations and terminologies and refer the reader to [5]. A path P is simple if no vertex occurs more than once in it. We denote the set of all the edges and all the vertices of a path P by $E(P)$ and $V(P)$ respectively.

For a connected graph $G = (V, E)$, we say that a subset $S \subseteq V$ is a vertex cover if $V \setminus S$ is an independent set in G , and we say that S is a feedback vertex set if $G[V \setminus S]$ induces a forest. The minimum size of a vertex cover of G is called its vertex cover number and the minimum size of a feedback vertex set of G is called its feedback vertex number. For two disjoint sets $A, B \subseteq V$, $E(A, B)$ is the set of all the edges with one endpoint in A and another in B . For details on pathwidth and treewidth, we refer to [4]. For details on parameterized complexity and fixed parameter tractability (FPT) we refer to [4, 6]. Informally, a $W[1]$ -hard problem is unlikely to be fixed parameter tractable.

3 Hardness with Respect to Vertex Cover + Maximum Path Length

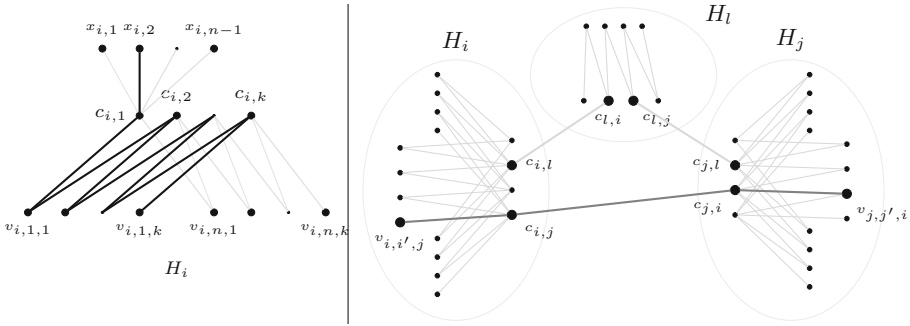


Fig. 1. Left: An example of vertex selection gadget H_i , the darkened edges forms a long path $P_{x_{i,2}, v_{i,1}}$. Right: An example of inter gadget edges and darkened edges forms a short path $P_{v_{i,i'}, v_{j,j'}, v_{j,j',i}}$ corresponding to an edge $v_{i,i'} v_{j,j'}$ in G .

In the k -MULTI COLORED CLIQUE (k -MCC) problem we are given a graph $G = (V, E)$, where V is partitioned into k disjoint sets V_1, \dots, V_k , each of size n , and the question is if G has a clique \mathcal{C} of size k such that $|\mathcal{C} \cap V_i| = 1$ for every $i \in [k]$. It is known that k -MCC is $W[1]$ -hard parameterized by k [8].

We will give a parameterized reduction from k -MCC to PSP. Let $G = (V, E)$ and $\{V_1, \dots, V_k\}$ be an input of k -MCC. Let the vertices of V_i be labeled $v_{i,1}$ to $v_{i,n}$. We will construct an equivalent instance $(G' = (V', E'), \mathcal{P})$ of PSP (see Fig. 1 for overview). For every set V_i , we construct a vertex selection gadget H_i (an induced subgraph of G') as follows .

- Create a set $C_i = \{c_{i,1}, \dots, c_{i,k}\}$ of k vertices, a set $X_i = \{x_{i,1}, x_{i,2}, \dots, x_{i,n-1}\}$ of $n - 1$ vertices, and connect every $x_{i,j}$ to $c_{i,1}$ we call these edges E_{C_i, X_i} .
- For every $v_{i,j} \in V_i$, create a vertex set $V_{i,j} = \{v_{i,j,1}, v_{i,j,2}, \dots, v_{i,j,k}\}$ of k vertices. Connect $v_{i,j,l}$ to $c_{i,l}$ and $c_{i,l+1}$ where $l \in [k - 1]$, and connect $v_{i,j,k}$ to $c_{i,k}$. We denote these edges by $E_{C_i, V_{i,j}}$.

Formally, $H_i = (\bigcup_{j=1}^n V_{i,j} \cup X_i \cup C_i, \bigcup_{j=1}^n E_{C_i, V_{i,j}} \cup E_{C_i, X_i})$. Further, let $C = \bigcup_{i=1}^k C_i$, we add the following edges in G' .

- For $1 \leq i < j \leq k$, we connect $c_{i,j} \in C_i$ to $c_{j,i} \in C_j$. We call these edges the inter gadget edges and denote them by E_C . Observe that there are $\binom{k}{2}$ inter gadget edges (Fig. 1).

The above completes the construction of $G' = (\bigcup_{i=1}^k V(H_i), \bigcup_{i=1}^k E(H_i) \cup E_C)$.

We now move on to the construction of the collection \mathcal{P} .

- Let $P_{x_{i,l}, V_{i,j}} = (x_{i,l}, c_{i,1}, v_{i,j,1}, c_{i,2}, v_{i,j,2}, \dots, c_{i,k}, v_{i,j,k})$, that is a path starting at $x_{i,l}$ and then alternates between a vertex in C_i and $V_{i,j}$ and ending at $v_{i,j,k}$. We call such a path a long path. For every $i \in [k]$, $l \in [n-1]$, and $j \in [n]$ we add $P_{x_{i,l}, V_{i,j}}$ in \mathcal{P} . Observe that there are $n(n-1)$ long paths added from every H_i .
- For every edge $e = v_{i,i'}v_{j,j'} \in E$ where $v_{i,i'} \in V_i$ and $v_{j,j'} \in V_j$, w.l.o.g. assuming $i < j$, we add a path $P_{v_{i,i'}, j, v_{j,j'}, i} = (v_{i,i'}, c_{i,j}, c_{j,i}, v_{j,j'}, i)$ in \mathcal{P} and call such a path, a short path. There are $|E|$ short paths added to \mathcal{P} .

The above completes the construction of instance $(G' = (V', E'), \mathcal{P})$ with $|\mathcal{P}| = |E| + kn(n-1)$.

The vertex set C forms a vertex cover for G' which is of size k^2 and the length of every long path is $2k + 1$. Further, the time taken for construction is $poly(|V|)$; the following concludes the correctness of the reduction and proof of Theorem 1.

Lemma 1 (\star^1). *$(G = (V, E), \{V_1, \dots, V_k\})$ is a yes instance of k -MCC if and only if G' has $k(n-1) + \binom{k}{2}$ edge disjoint paths in \mathcal{P} .*

4 Hardness with Respect to Pathwidth + Maximum Degree + Solution Size

We give a parameterized reduction from k -MCC to PSP. Let $G = (V, E)$ and $\{V_1, \dots, V_k\}$ be the input for k -MCC, and let the vertices of set V_i be labeled $v_{i,1}$ to $v_{i,n}$. We will construct an equivalent instance $(G' = (V', E'), \mathcal{P})$ of PSP (see Fig. 2 for overview), the construction of G' is as follows.

- For every V_i , we construct a gadget (subgraph of G') which includes a vertex selection path P_i , a vertex set W_i , and k edge verification paths $P_{i,l}^e$ as follows.
 - Corresponding to V_i , we start with creating $n + 1$ paths of $2k$ vertices each, one path for every vertex $v_{i,i'} \in V_i$ and an additional path. For every $i' \in [n+1]$, the i' path is $(v_{i,i',1}, u_{i,i',1}, v_{i,i',2}, u_{i,i',2}, \dots, v_{i,i',k}, u_{i,i',k})$. We now combine these $n + 1$ paths into one path P_i by adding an edge between $u_{i,i',k}$ and $v_{i,i'+1,1}$ for every $i' \in [n]$.
 - We create n vertices $w_{i,1}$ to $w_{i,n}$ and call the set of these vertices W_i . For every $i' \in [n]$, we connect $w_{i,i'}$ to $v_{i,i',1}$ and $v_{i,i'+1,1}$.
 - We create k edge verification paths $P_{i,1}^e$ to $P_{i,k}^e$ with $n + 1$ vertices each. The path $P_{i,j}^e$ is $(x_{i,1,j}, x_{i,2,j}, \dots, x_{i,n,j}, c_{i,j})$.
 - For every $j \in [k]$, $i' \in [n]$, we connect $u_{i,i',j}$ to $x_{i,i',j}$. These edges connects the vertices of vertex selection path P_i to edge verification paths $P_{i,j}^e$.

¹ The proofs of statements marked with a \star have been omitted.

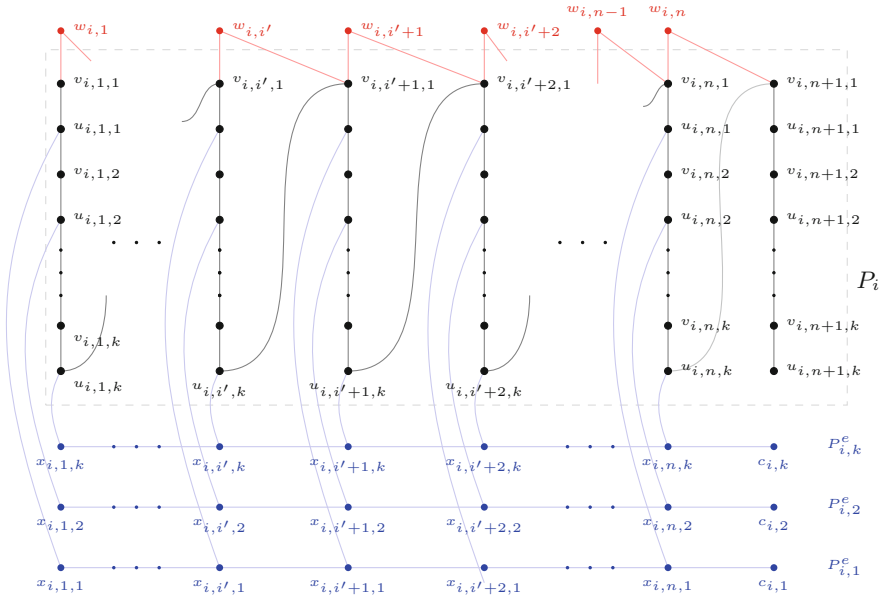


Fig. 2. An example of path P_i , edge verification paths $P_{i,1}^e, P_{i,2}^e$, and $P_{i,k}^e$, also the edges between vertices of vertex selection paths and edge verification paths.

- After constructing above mentioned gadgets for every vertex set in $\{V_1, \dots, V_k\}$, for $1 \leq i < j \leq k$, we connect $c_{i,j}$ to $c_{j,i}$. We call these edges the inter gadget edges. Observe that there are $\binom{k}{2}$ inter gadget edges.

The above completes the construction of G' . We now construct collection \mathcal{P} of size $nk + |E|$ as follows.

- For every $i \in [k]$, from the subgraph of G' induced by $V(P_i) \cup W_i$, we will add n paths in the collection \mathcal{P} as follows.
 - Add a path $l_{i,\vec{i}'} = (P_i(v_{i,1,1}, v_{i,i',1}), w_{i,i'}, P_i(v_{i,i'+1,1}, u_{i,n+1,k}))$ for every $i' \in [n]$, where $P_i(v_{i,1,1}, v_{i,i',1})$ is the path from vertex $v_{i,1,1}$ to $v_{i,i',1}$ in P_i (a unique path since P_i is a path). Intuitively, for every $i' \in [n]$ the $l_{i,\vec{i}'}$ contains all the edges of P_i except the edges which belong to subpath $P_i(v_{i,i',1}, v_{i,i'+1,1})$. We call these paths the long paths.
- For every edge $v_{i,i'}v_{j,j'} \in E$ where $i < j$, we add the path $s_{i,i',j,j'} = (v_{i,i',j}, u_{i,i',j}, P_{i,j}^e(x_{i,i',j}, c_{i,j}), P_{j,i}^e(c_{j,i}, x_{j,j',i}), u_{j,j',i}, v_{j,j',i})$ in \mathcal{P} , where $P_{i,j}^e(x_{i,i',j}, c_{i,j})$ is the path from $x_{i,i',j}$ to $c_{i,j}$ in $P_{i,j}^e$ (a unique path, since $P_{i,j}^e$ is a path). We note that every $s_{i,i',j,j'}$ contains exactly one inter gadget edge $c_{i,j}, c_{j,i}$. This finishes the construction of \mathcal{P} .

Observe that the construction of (G', \mathcal{P}) takes time polynomial in $|V|$. We now claim the bounds on pathwidth and maximum degree of G' .

Lemma 2 (\star). *Pathwidth of G' is $O(k^2)$ and maximum degree of G' is $O(k)$.*

The following concludes the correctness of reduction and proof of Theorem 2.

Lemma 3 (\star). *$G = (V, E)$ with partition V_1 to V_k is a yes instance of k -MCC if and only if \mathcal{P} has $k + \binom{k}{2}$ pairwise edge disjoint paths.*

5 FPT Parameterized by Feedback Vertex Number + Maximum Degree

In this section, we will show that PSP is FPT parameterized by Γ (feedback vertex number) plus maximum degree Δ and prove Theorem 3. For a connected graph $G = (V, E)$, its feedback edge number, denoted by λ , is the minimum number of edges whose removal results in a tree and equals $|V| - |E| + 1$. Since the set of all edges incident on a feedback vertex set forms a feedback edge set, we have: $\lambda \leq \Gamma \cdot \Delta$. We will provide an algorithm which solves PSP in time $(\lambda \cdot \Delta)^{O(\lambda \cdot \Delta)} \cdot \text{poly}(|V| + |\mathcal{P}|)$. The approach used here is a non trivial adaptation of the approach given in [13].

5.1 Preliminaries: Defining Structures and Nice Solutions

Let $G = (V, E)$ be the input graph. We create 3 vertices z_1, z_2, z_3 and arbitrarily choose a vertex $v \in V$ and connect z_1, z_2, z_3 , and v to each other forming a clique on 4 vertices in G . This modification increases the size of minimum feedback vertex set and maximum degree by only a constant, and is safe for our purposes. In this section we define structures (adapted from [13]) that we will need.

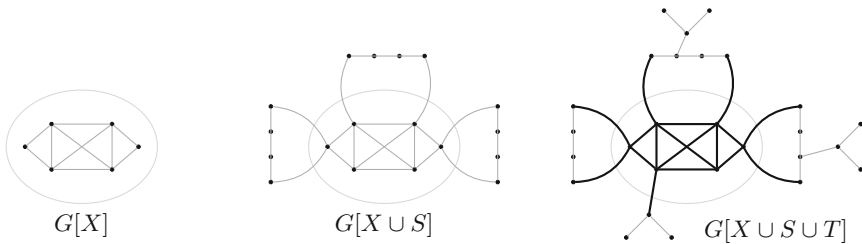


Fig. 3. An example induced graphs $G[X]$, $G[X \cup S]$, and $G[X \cup S \cup T]$. The darkened edges in rightmost graph represent core edges E_X .

Definition 1. *We define the vertex set T, S and X by the following process (refer Fig. 3).*

- Initialize T as an empty set and $G' = (V', E')$ as a copy of G .
- While there is a vertex v in G' with degree $d_{G'}(v) = 1$, we set $T = T \cup \{v\}$ and $G' = G' - \{v\}$.

- X is the set of all the vertices with degree at least 3 in $G[V \setminus T]$, and S is the set $V \setminus (T \cup X)$.

Observe that $G[T]$ is a forest, and every component of $G[T]$ is connected to $G[V \setminus T]$ by a single edge in G . $G[V \setminus T]$ has minimum degree 2. X is a non empty set as it contains at least z_1, z_2, z_3 , and v as they form a clique. Every vertex in S has degree exactly 2 in $G[V \setminus T]$ and $G[S]$ is a union of paths (Fig. 3).

Definition 2. We define $E_X = E(G[X]) \cup E(X, S \cup T)$, i.e. all the edges in G with at least one endpoint in X . We will call these edges the core edges of G .

Observation 1 (\star). Every component in $G[S \cup T]$ contains at most 1 component from $G[S]$.

Let \mathcal{D} be the set of all the components of $G[S \cup T]$ which contain a component of $G[S]$. Further, let \mathcal{T} be the set of all other components in $G[S \cup T]$, every component $C \in \mathcal{D} \cup \mathcal{T}$ is a tree, and the edges which connect C to $G[X]$, are called *external edges* of C , and these edges belong to E_X .

Observation 2 (\star). Every component in \mathcal{D} is connected to $G[X]$ by two edges in G (has 2 external edges), and every component in \mathcal{T} is connected to $G[X]$ by one edge in G (has one external edge).

We call \mathcal{D} the set of 2-external edge components, and \mathcal{T} the set of 1-external edge components. To bound the size of X and \mathcal{D} , we recall the following from [13].

Proposition 1 ([13]). Let G be a connected graph of minimum degree at least two with cyclomatic number (feedback edge number) λ . Let X be the set of all the vertices of degree at least three in G . Then $|X| \leq 2\lambda - 2$ and if $X \neq \emptyset$, then the number of connected components of $G[V \setminus X]$ is at most $\lambda + |X| - 1$.

We get the following corollary.

Corollary 3. The size of vertex set $|X| = O(\lambda) = O(\Gamma \cdot \Delta)$, and $|E_X| = O(\lambda \cdot \Delta) = O(\Gamma \cdot \Delta^2)$. The size of component set $|\mathcal{D}| = O(\lambda) = O(\Gamma \cdot \Delta)$.

Definition 3. For a subgraph $H \subseteq G$, a path p in G is an internal path of H if $E(p) \subseteq E(H)$. Further, given a set P of paths in G , we define $INT(H, P) = \{p_i \mid p_i \in P \wedge E(p_i) \subseteq E(H)\}$, that is all the paths in P which are internal to H .

Definition 4. Let (G, \mathcal{P}, k) be an instance of PSP, for a subgraph $H \subseteq G$, we define $OPT(H)$ as the maximum number of edge disjoint paths in $INT(H, \mathcal{P})$.

Lemma 4 (\star). Let (G, \mathcal{P}, k) be an instance of PSP. And let $M \subseteq \mathcal{P}$ be a path set packing of maximum size. Then there exists a path set packing $M' \subseteq \mathcal{P}$, such that $|M'| = |M|$ and the following holds

- for every component $D_i \in \mathcal{D}$, $OPT(D_i) \geq |M' \cap INT(D_i, \mathcal{P})| \geq OPT(D_i) - 1$, and
- for every component $T_i \in \mathcal{T}$, $|M' \cap INT(T_i, \mathcal{P})| = OPT(T_i)$.

We call such M' as a **nice solution**.

While searching for a solution, we will only search for a nice solution of maximum size.

Proposition 2 ([16, 17]). *PSP can be solved in time polynomial in $|V| + |\mathcal{P}|$ if the input graph is a tree.*

Corollary 4. *$OPT(D_i)$ and $OPT(T_i)$ can be computed in time polynomial in $|V| + |\mathcal{P}|$ for every $D_i \in \mathcal{D}$ and every $T_i \in \mathcal{T}$.*

5.2 Guessing and Extending the Solution

We first guess the number of internal paths that every $D_i \in \mathcal{D}$ will have in the solution. Formally, $f_d : \mathcal{D} \rightarrow \{0, 1\}$ is a guessing that $OPT(D_i) - f_d(D_i)$ internal paths of D_i will be in the solution. For every $T_i \in \mathcal{T}$ we know that $OPT(T_i)$ internal paths will be in the solution. Now, it is left to optimize the number of edge disjoint paths which are having one or more edges from E_X . Let $f_e : E_X \rightarrow [|E_X|] \cup \{0\}$. Let $E_i = \{e | e \in E_X \wedge f_e(e) = i\}$. Let there be l non-empty sets E_i except E_0 ; let them be E_1 to E_l . This is our second guess, where we are guessing the partition of E_X such that if a path p in the solution intersects with E_X , then $E(p) \cap E_X$ should be E_i for an $i \in [l]$. Further, all the edges of E_0 are guessed to be not part of any path in the solution. Thus, we are guessing that there will be l paths in the solution containing the edges from E_X .

Definition 5. *We say a path $p \in \mathcal{P}$ is of type E_i if $(E(p) \cap E_X) = E_i$.*

Definition 6. *We say that the pair (f_d, f_e) has a feasible solution, if there exists a set of edge disjoint paths $M \subseteq \mathcal{P}$ such that the following holds.*

1. *For every $i \in [l]$, there exists a path $p \in M$ such that p is of type E_i ;*
2. *For every $D_i \in \mathcal{D}$, $|M \cap INT(D_i, M)| = OPT(D_i) - f_d(D_i)$;*
3. *For every $T_i \in \mathcal{T}$, $|M \cap INT(T_i, M)| = OPT(T_i)$.*

If a pair (f_d, f_e) has a feasible solution M , then $|M|$ will be equal to the sum of $OPT(T_i)$ over every $T_i \in \mathcal{T}$, plus sum of $OPT(D_i) - f_d(D_i)$ over every $D_i \in \mathcal{D}$, plus l . Thus, taking maximum over the size of the feasible solution of every possible pair (f_d, f_e) which has a feasible solution will give us the maximum size of a path set packing as both f_d and f_e are exhaustive guesses, and we are searching for a nice solution (Lemma 4). There are at most $2^{O(\lambda)}$ distinct guesses f_d and at most $(\lambda \cdot \Delta)^{O(\lambda \cdot \Delta)}$ distinct guesses f_e . If we can verify whether a pair (f_d, f_e) has a feasible solution in time polynomial in $(|V| + |\mathcal{P}|)$, then we can bound the running time of the algorithm to $(\lambda \cdot \Delta)^{O(\lambda \cdot \Delta)} \cdot poly(|V| + |\mathcal{P}|)$. Given a pair (f_d, f_e) , we will now discuss how to verify if (f_d, f_e) has a feasible solution.

Definition 7. A set of paths P is (f_d, f_e) -compatible if the following holds.

1. For every pair $p, q \in P$, if $p \neq q$ then p and q are edge disjoint;
2. For every $p \in P$, there exists $i \in [l]$ such that p is of type E_i ;
3. For every $D_i \in \mathcal{D}$, $OPT(D_i - E(P)) \geq OPT(D_i) - f_d(D_i)$;
4. For every $T_i \in \mathcal{T}$, $OPT(T_i - E(P)) = OPT(T_i)$.

Observation 3 (\star). Given a path set $P \subseteq \mathcal{P}$, in time polynomial in $(|V| + |\mathcal{P}|)$ we can verify if P is (f_d, f_e) -compatible or not. Further, if P is (f_d, f_e) -compatible then every subset P' of P is also (f_d, f_e) -compatible.

Lemma 5 (\star). (f_d, f_e) has a feasible solution if and only if there exists a path set $P \subseteq \mathcal{P}$ such that $|P| = l$ and P is (f_d, f_e) -compatible.

Due to above lemma, to verify if (f_d, f_e) has a feasible solution, it will suffice to verify if there exist an (f_d, f_e) -compatible path set $P \subseteq \mathcal{P}$ of size l .

Observation 4 (\star). If there exist an E_i which contains exactly one external edge of three or more components in $\mathcal{D} \cup \mathcal{T}$, then no path can contain all the edges of E_i .

If for any $i \in [l]$, E_i contains exactly one external edge of three or more components in $\mathcal{D} \cup \mathcal{T}$, then using Observation 4, we can conclude that (f_d, f_e) has no feasible solution. Thus we shall henceforth assume that no E_i where $i \in [l]$, contains exactly one external edge of three or more components in $\mathcal{D} \cup \mathcal{T}$.

Consider an auxiliary graph H with vertex set $\{E_1, \dots, E_l\}$. In H , two vertices E_i and E_j are adjacent if and only if there exists a $D \in \mathcal{D}$ such that one of the external edges of D belongs to E_i and the other external edge belongs to E_j . Every D has two external edges, and thus, every D can cause at most one edge to be created in H . Combining this with Observation 4, we conclude that H has degree at most two. Thus, H is a union of paths and cycles. Let there be l' components in H which are labelled from 1 to l' , let π_i be the the path formed by the component i , that is each π_i is a sequence of vertices (arbitrarily chose the first vertex of the path in case the component i is a cycle). We call π_i a type sequence. Let $\Pi = \{\pi_i | i \in [l']\}$.

Definition 8. A sequence ρ of paths is a candidate for type sequence π_i , if $|\rho| = |\pi_i|$ and for every $j \in [|\rho|]$, $\rho[j]$ is of type $\pi_i[j]$.

Lemma 6 (\star). There exists $P \subseteq \mathcal{P}$ such that $|P| = l$ and P is (f_d, f_e) -compatible if and only if for every sequence $\pi_i \in \Pi$, there exists a candidate ρ such that $set(\rho)$ is (f_d, f_e) -compatible.

Since there are at most $l' \leq |V|^2$ type sequences, it suffices to prove that given a type sequence $\pi_i \in \Pi$, in time polynomial in $(|\mathcal{P}| + |V|)$ we can verify if π_i has a candidate ρ_i such that $set(\rho_i)$ is (f_d, f_e) -compatible. Consider the following.

Lemma 7 (\star). For every type sequence $\pi_i \in \Pi$, let ρ_i be a candidate of π_i . Then, $set(\rho_i)$ is (f_d, f_e) -compatible if and only if

- For every $j \in [|\pi_i|]$, $\{\rho_i[j], \rho_i[(j + 1) \bmod |\pi_i|]\}$ is (f_d, f_e) -compatible.

The above lemma helps us find a desired candidate for type sequence π_i in polynomial time, since we now only need to check if the paths of adjacent types in a sequence are (f_d, f_e) -compatible or not. If the size of π_i is ≤ 2 , then we can verify if π_i has a candidate ρ_i such that $set(\rho_i)$ is (f_d, f_e) -compatible by checking for every distinct $p, q \in \mathcal{P}$ if (p, q) is a candidate of π_i as well as (f_d, f_e) -compatible or not, this will take time polynomial in $(|V| + |\mathcal{P}|)$. We now move on to the case when $|\pi_i| \geq 3$.

Given a type sequence π_i we create an auxiliary directed graph H_i as follows.

- For every type $\pi_i[j]$, create a vertex set $V_j = \{v_p \mid p \in \mathcal{P} \wedge p \text{ is of type } \pi_i[j] \wedge \{p\} \text{ is } (f_d, f_e)\text{-compatible}\}$. That is V_j contain vertices corresponding to every path of type $\pi_i[j]$ which is (f_d, f_e) -compatible.
- For every $j \in [|\pi_i|]$, let $j' = (j + 1) \bmod |\pi_i|$, we add an arc (directed edge) from $v_p \in V_j$ to $v_q \in V_{j'}$ if and only if $\{p, q\}$ are (f_d, f_e) -compatible.

Lemma 8 (\star). *π_i has a candidate ρ_i such that $set(\rho_i)$ is (f_d, f_e) -compatible if and only if H_i has a cycle containing exactly one vertex from every V_j where $j \in [|\pi_i|]$.*

Lemma 9 (\star). *In time polynomial in $(|V| + |\mathcal{P}|)$, we can find a cycle containing exactly one vertex from every V_j in H_i or conclude that no such cycle exists.*

The above lemma finishes the proof of Theorem 3.

6 FPT When Combining Three Parameters

Given a pair (G, \mathcal{P}) , let H be the conflict graph with vertex set \mathcal{P} . As noted earlier, solving PSP on (G, \mathcal{P}) is equivalent to finding a MIS in H . We can deduce the following fact about the structure of H .

Lemma 10 (\star). *If G has treewidth k and maximum degree Δ , and each path in \mathcal{P} is of length at most r , then the treewidth of H is at most $(k + 1)\Delta^r$.*

The Maximum Independent Set problem admits a $O(2^\tau)$ algorithm on graphs of treewidth τ [2]. Hence we obtain Theorem 4 as a corollary.

Acknowledgements. We thank anonymous reviewers of this and an earlier version of this paper for useful suggestions.

References

1. Abu-Khazam, F.N.: An improved kernelization algorithm for r-set packing. Inf. Process. Lett. **110**(16), 621–624 (2010)
2. Arnborg, S., Proskurowski, A.: Linear time algorithms for np-hard problems restricted to partial k-trees. Discret. Appl. Math. **23**(1), 11–24 (1989)

3. Bessy, S., Bougeret, M., Chaplick, S., Gonçalves, D., Paul, C.: On independent set in B_1 -EPG graphs. *Discret. Appl. Math.* **278**, 62–72 (2020)
4. Cygan, M., et al.: *Parameterized Algorithms*. Springer, Heidelberg (2015). <https://doi.org/10.1007/978-3-319-21275-3>
5. Diestel, R.: *Graph Theory*. Graduate Texts in Mathematics, vol. 173, 4th edn. Springer, Heidelberg (2012)
6. Downey, R.G., Fellows, M.R.: *Parameterized Complexity*. Monographs in Computer Science, Springer, New York (1999). <https://doi.org/10.1007/978-1-4612-0515-9>
7. Epstein, D., Golumbic, M.C., Morgenstern, G.: Approximation algorithms for B_1 -EPG graphs. In: Dehne, F., Solis-Oba, R., Sack, J.-R. (eds.) *WADS 2013*. LNCS, vol. 8037, pp. 328–340. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-40104-6_29
8. Fellows, M.R., Hermelin, D., Rosamond, F.A., Viallette, S.: On the parameterized complexity of multiple-interval graph problems. *Theor. Comput. Sci.* **410**(1), 53–61 (2009)
9. Garey, M.R., Johnson, D.S.: The rectilinear steiner tree problem is $\$np\$$ -complete. *SIAM J. Appl. Math.* **32**(4), 826–834 (1977)
10. Golumbic, M.C., Jamison, R.E.: Edge and vertex intersection of paths in a tree. *Discret. Math.* **55**(2), 151–159 (1985)
11. Golumbic, M.C., Jamison, R.E.: The edge intersection graphs of paths in a tree. *J. Comb. Theory Ser. B* **38**(1), 8–22 (1985)
12. Golumbic, M.C., Lipshteyn, M., Stern, M.: Edge intersection graphs of single bend paths on a grid. *Networks* **54**(3), 130–138 (2009)
13. Jansen, B.M.P.: On structural parameterizations of hitting set: hitting paths in graphs using 2-SAT. In: Mayr, E.W. (ed.) *WG 2015*. LNCS, vol. 9224, pp. 472–486. Springer, Heidelberg (2016). https://doi.org/10.1007/978-3-662-53174-7_33
14. Jia, W., Zhang, C., Chen, J.: An efficient parameterized algorithm for m -set packing. *J. Algorithms* **50**(1), 106–117 (2004)
15. Koutis, I.: A faster parameterized algorithm for set packing. *Inf. Process. Lett.* **94**(1), 7–9 (2005)
16. Tarjan, R.E.: Decomposition by clique separators. *Discret. Math.* **55**(2), 221–232 (1985)
17. Xu, C., Zhang, G.: The path set packing problem. In: Wang, L., Zhu, D. (eds.) *COCOON 2018*. LNCS, vol. 10976, pp. 305–315. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-94776-1_26