

# On the complexity of list $\mathcal{H}$ -packing for sparse graph classes

Tatsuya Gima\*    Tesshu Hanaka†    Yasuaki Kobayashi‡    Yota Otachi§  
 Tomohito Shirai¶    Akira Suzuki¶    Yuma Tamura¶    Xiao Zhou¶

December 15, 2023

## Abstract

The problem of packing as many subgraphs isomorphic to  $H \in \mathcal{H}$  as possible in a graph for a class  $\mathcal{H}$  of graphs is well studied in the literature. Both vertex-disjoint and edge-disjoint versions are known to be NP-complete for  $H$  that contains at least three vertices and at least three edges, respectively. In this paper, we consider “list variants” of these problems: Given a graph  $G$ , an integer  $k$ , and a collection  $\mathcal{L}_{\mathcal{H}}$  of subgraphs of  $G$  isomorphic to some  $H \in \mathcal{H}$ , the goal is to compute  $k$  subgraphs in  $\mathcal{L}_{\mathcal{H}}$  that are pairwise vertex- or edge-disjoint. We show several positive and negative results, focusing on classes of sparse graphs, such as bounded-degree graphs, planar graphs, and bounded-treewidth graphs.

## 1 Introduction

Packing as many graphs as possible into another graph is a fundamental problem in the field of graph algorithms. To be precise, for a fixed graph  $H$ , given an undirected graph  $G$  and a non-negative integer  $k$ , the VERTEX DISJOINT  $H$ -PACKING problem (resp. the EDGE DISJOINT  $H$ -PACKING problem) asks for finding a collection  $\mathcal{S}$  of  $k$  vertex-disjoint (resp. edge-disjoint) subgraphs of  $G$  that are isomorphic to  $H$ . For a connected graph  $H$ , both problems are polynomially solvable if  $H$  has at most two vertices (resp. at most two edges) because they can be reduced to the MAXIMUM MATCHING problem, whereas the problems are shown to be NP-complete if  $H$  has at least three vertices (resp. at least three edges) [11, 24]. Furthermore, both problems are naturally extended to VERTEX DISJOINT  $\mathcal{H}$ -PACKING and EDGE DISJOINT  $\mathcal{H}$ -PACKING [10, 24], which respectively ask for finding a collection  $\mathcal{S}$  of  $k$  vertex-disjoint and edge-disjoint subgraphs of  $G$  that are isomorphic to some graph in a (possibly infinite) fixed collection  $\mathcal{H}$  of graphs. These problems are also well studied in specific cases of  $\mathcal{H}$ . In particular, when  $\mathcal{H}$  is paths or cycles, it has received much attention in the literature because of the variety of possible applications [4, 5, 8, 25, 27]. In both cases, VERTEX DISJOINT  $\mathcal{H}$ -PACKING and EDGE DISJOINT  $\mathcal{H}$ -PACKING remain NP-complete for planar graphs [5, 13].

Recently, Xu and Zhang proposed a new variant of EDGE DISJOINT  $\mathcal{H}$ -PACKING, which they call PATH SET PACKING, from the perspective of network design [38]. In the PATH SET PACKING problem, given an undirected graph  $G$ , a non-negative integer  $k$ , and a collection  $\mathcal{L}$  of simple paths in  $G$ , we are required to find a subcollection  $\mathcal{S} \subseteq \mathcal{L}$  of (at least)  $k$  paths that are mutually edge-disjoint. Notice that  $\mathcal{L}$  may not be exhaustive: Some paths in  $G$  may not appear in  $\mathcal{L}$ . If  $\mathcal{H}$  consists of a finite number of paths, EDGE DISJOINT  $\mathcal{H}$ -PACKING can be (polynomially) reduced to PATH SET PACKING because  $\mathcal{H}$  is fixed and hence all paths in  $G$  isomorphic to some graph in  $\mathcal{H}$  can be enumerated in polynomial time. Xu and Zhang showed that for a graph  $G$  with  $n$  vertices and  $m$  edges, the optimization variant of PATH SET PACKING is hard

\*Graduate School of Informatics, Nagoya University, Nagoya, Japan and JSPS Research Fellow. [gima@nagoya-u.jp](mailto:gima@nagoya-u.jp)

†Department of Informatics, Kyushu University, Fukuoka, Japan. [hanaka@inf.kyushu-u.ac.jp](mailto:hanaka@inf.kyushu-u.ac.jp)

‡Graduate School of Information Science and Technology, Hokkaido University, Hokkaido, Japan. [koba@ist.hokudai.ac.jp](mailto:koba@ist.hokudai.ac.jp)

§Graduate School of Informatics, Nagoya University, Nagoya, Japan. [otachi@nagoya-u.jp](mailto:otachi@nagoya-u.jp)

¶Graduate School of Information Sciences, Tohoku University, Sendai, Japan. [{akira,tamura,zhou}@tohoku.ac.jp](mailto:{akira,tamura,zhou}@tohoku.ac.jp)

Table 1: The complexity of VERTEX DISJOINT  $C_\ell$ -PACKING (left) and EDGE DISJOINT  $C_\ell$ -PACKING (right).

	$\Delta(G) = 3$	$\Delta(G) \geq 4$		$\Delta(G) = 3$	$\Delta(G) = 4$	$\Delta(G) \geq 5$
$\ell = 3$	P [9]	NPC [9]	$\ell = 3$	P [9]		NPC [9]
$\ell = 4$	P [Thm 4]	NPC [Thm 6]	$\ell = 4$	P [Thm 4]	NPC [Thm 7]	
$\ell = 5$			$\ell = 5$			
$\ell \geq 6$	NPC [Thm 5]		$\ell \geq 6$	NPC [Thm 5]		

to approximate within a factor  $O(m^{1/2-\epsilon})$  for any constant  $\epsilon > 0$  unless  $\text{NP} = \text{ZPP}$ , while the problem is solvable in  $O(|\mathcal{L}|n^2)$  time if  $G$  is a tree and in  $O(|\mathcal{L}|^{\text{tw}\Delta}n)$  time if  $G$  has treewidth  $\text{tw}$  and maximum degree  $\Delta$  [38]. Very recently, Aravind and Saxena investigated the parameterized complexity of PATH SET PACKING for various parameters. For instance, PATH SET PACKING is  $\text{W}[1]$ -hard even when parameterized by pathwidth plus maximum degree plus solution size [2]. To the best of our knowledge, except for PATH SET PACKING, such a variant has not been studied for EDGE DISJOINT  $\mathcal{H}$ -PACKING, nor VERTEX DISJOINT  $\mathcal{H}$ -PACKING.

**Our contributions.** In this paper, motivated by PATH SET PACKING, we introduce *list variants* of VERTEX DISJOINT  $\mathcal{H}$ -PACKING and EDGE DISJOINT  $\mathcal{H}$ -PACKING. In the VERTEX DISJOINT LIST  $\mathcal{H}$ -PACKING (resp. EDGE DISJOINT LIST  $\mathcal{H}$ -PACKING) problem, we are given a graph  $G$ , a non-negative integer  $k$ , and a collection (list)  $\mathcal{L}_{\mathcal{H}}$  of subgraphs of  $G$  such that each subgraph in  $\mathcal{L}_{\mathcal{H}}$  is isomorphic to some graph in  $\mathcal{H}$ . The problem asks whether there exists a subcollection  $\mathcal{S} \subseteq \mathcal{L}_{\mathcal{H}}$  such that  $|\mathcal{S}| \geq k$  and subgraphs of  $G$  in  $\mathcal{S}$  are vertex-disjoint (resp. edge-disjoint). If  $\mathcal{L}_{\mathcal{H}}$  contains all subgraphs of  $G$  isomorphic to some graph in  $\mathcal{H}$ , the problem is equivalent to VERTEX DISJOINT  $\mathcal{H}$ -PACKING (resp. EDGE DISJOINT  $\mathcal{H}$ -PACKING). Thus, the tractability of the list variants implies that of the original problems. If  $\mathcal{H} = \{H\}$  for a fixed graph  $H$ , then we call these problems simply VERTEX (EDGE) DISJOINT LIST  $H$ -PACKING and VERTEX (EDGE) DISJOINT  $H$ -PACKING, respectively. For a positive integer  $\ell$ , we denote by  $P_\ell$  and  $C_\ell$  the path and the cycle of  $\ell$  vertices, respectively. (We assume  $\ell \geq 3$  for  $C_\ell$ .) When  $\mathcal{P} = \{P_\ell : \ell \geq 1\}$ , EDGE DISJOINT LIST  $\mathcal{P}$ -PACKING is equivalent to PATH SET PACKING. Therefore, EDGE DISJOINT LIST  $\mathcal{H}$ -PACKING generalizes both EDGE DISJOINT  $H$ -PACKING and PATH SET PACKING.

We first give sufficient conditions to solve VERTEX (EDGE) DISJOINT LIST  $H$ -PACKING on bounded degree graphs in polynomial time. These conditions directly indicate the polynomial-time solvability of VERTEX (EDGE) DISJOINT LIST  $C_\ell$ -PACKING on graphs of maximum degree 3 for  $\ell \in \{3, 4, 5\}$ . It is worth noting that VERTEX DISJOINT  $P_3$ -PACKING remains NP-complete even for 2-connected bipartite planar cubic graphs [27]. In contrast, we show that VERTEX (EDGE) DISJOINT  $C_\ell$ -PACKING on planar graphs of maximum degree 3 is NP-complete for any  $\ell \geq 6$ . As VERTEX (EDGE) DISJOINT  $C_\ell$ -PACKING can be represented by its list variant, this result also indicates the hardness of VERTEX (EDGE) DISJOINT LIST  $C_\ell$ -PACKING. We also give the NP-completeness of VERTEX (EDGE) DISJOINT  $C_\ell$ -PACKING on planar graphs of maximum degree 4 for any  $\ell \geq 4$ . Therefore, we provide the complexity dichotomy of VERTEX (EDGE) DISJOINT  $C_\ell$ -PACKING with respect to the maximum degree of a given graph and  $\ell$ , as summarized in Table 1.

Second, we design a polynomial-time algorithm for VERTEX DISJOINT LIST  $\mathcal{H}$ -PACKING on bounded-treewidth graphs, provided that all graphs in  $\mathcal{H}$  are connected. This implies that VERTEX DISJOINT LIST  $\mathcal{H}$ -PACKING belongs to XP parameterized by treewidth. Note that the connectivity condition on  $\mathcal{H}$  is essential, because otherwise one can see that the problem is NP-complete even on forests (see Theorem 3). On the other hand, we show that VERTEX DISJOINT LIST  $\mathcal{P}$ -PACKING and VERTEX DISJOINT LIST  $\mathcal{C}$ -PACKING parameterized by pathwidth plus  $k$  are  $\text{W}[1]$ -hard, where  $\mathcal{P} = \{P_\ell : \ell \geq 1\}$  and  $\mathcal{C} = \{C_\ell : \ell \geq 3\}$ . This result implies that there is probably no FPT algorithm for the problems parameterized by treewidth. One might think that XP algorithms parameterized by treewidth could also be designed for the edge-disjoint versions. We give a negative answer. We show that EDGE DISJOINT LIST  $\mathcal{P}$ -PACKING and EDGE DISJOINT LIST  $\mathcal{C}$ -PACKING parameterized by bandwidth plus  $k$  are  $\text{W}[1]$ -hard even for outerplanar and two-terminal series-parallel graphs, which have treewidth at most 2. In particular, the  $\text{W}[1]$ -hardness for EDGE DISJOINT

LIST  $\mathcal{P}$ -PACKING, which is equivalent to PATH SET PACKING, strengthens the result of [2].

The above hardness results prompt us to further investigate the complexity of EDGE DISJOINT LIST  $P_\ell$ -PACKING and EDGE DISJOINT LIST  $C_\ell$ -PACKING on bounded-treewidth graphs. In this paper, we focus on series-parallel graphs, also known as graphs of treewidth at most 2. We show that EDGE DISJOINT LIST  $P_4$ -PACKING and EDGE DISJOINT LIST  $C_5$ -PACKING remain NP-complete even for series-parallel graphs. Since EDGE DISJOINT LIST  $P_3$ -PACKING is solvable in polynomial time for general graphs by reducing to MAXIMUM MATCHING, the former implies the complexity dichotomy of EDGE DISJOINT LIST  $P_\ell$ -PACKING on series-parallel graphs with respect to  $\ell$ . The remaining task is to settle the complexity of EDGE DISJOINT LIST  $C_\ell$ -PACKING on series-parallel graphs for  $\ell \leq 4$ . We finally provide an algorithm that, given an  $n$ -vertex series-parallel graph and a collection  $\mathcal{L}_\mathcal{H}$  of cycles with length  $\ell \leq 4$ , solves EDGE DISJOINT LIST  $C_\ell$ -PACKING in  $O(|\mathcal{L}_\mathcal{H}| + n^{2.5})$  time.

**Related work.** The packing problems have a long history. VERTEX DISJOINT  $\mathcal{C}$ -PACKING (also known simply as CYCLE PACKING) has been studied in numerous papers after the well-known Erdős-Pósa theorem was given [14]. In particular, CYCLE PACKING has received a lot of attention in the field of parameterized complexity. CYCLE PACKING is fixed-parameter tractable when parameterized by a solution size  $k$  [7, 26] or by treewidth [12]. The complexity of restricted variants of CYCLE PACKING has also been explored. For any  $\ell \geq 3$ , VERTEX DISJOINT  $C_\ell$ -PACKING on planar graphs is NP-complete [5]. In addition, VERTEX DISJOINT  $C_3$ -PACKING is NP-complete for planar graphs of maximum degree 4 [9], chordal graphs, line graphs, and total graphs [18], while the problem is solvable in polynomial time for graphs of maximum degree 3 [9], split graphs, and cographs [18].

In the edge-disjoint variant of CYCLE PACKING, for any  $\ell \geq 3$ , EDGE DISJOINT  $C_\ell$ -PACKING is NP-complete for planar graphs [19] and balanced 2-interval graphs [21]. Furthermore, for any even  $\ell \geq 4$ , EDGE DISJOINT  $C_\ell$ -PACKING on bipartite graphs and balanced 2-track interval graphs is NP-complete [21]. EDGE DISJOINT  $C_3$ -PACKING is solvable for graphs of maximum degree 4 [9] and outerplanar graphs [19], while the problem is NP-complete for planar graphs of maximum degree 5 [9].

As for problems of packing disjoint paths, VERTEX DISJOINT  $P_\ell$ -PACKING for any  $\ell \geq 3$  is NP-complete for graphs of maximum degree 3 and EDGE DISJOINT  $P_\ell$ -PACKING for any  $\ell \geq 4$  is NP-complete for graphs of maximum degree 4 [28]. Moreover, VERTEX DISJOINT  $P_3$ -PACKING remains NP-complete even for 2-connected bipartite planar cubic graphs [27]. On the other hand, for any positive integer  $\ell$ , VERTEX DISJOINT  $P_\ell$ -PACKING is solvable in polynomial time for trees [28]. VERTEX DISJOINT  $P_2$ -PACKING is equivalent to MAXIMUM MATCHING, which is solvable in polynomial time [20]. VERTEX DISJOINT  $P_3$ -PACKING is polynomially solvable for certain cases [1, 22], and the problem is in FPT and has a linear kernel when parameterized by a solution size [16, 33, 37].

Compared to the other packing problems, as far as we know, not much is known about tractable cases of EDGE DISJOINT  $P_\ell$ -PACKING. EDGE DISJOINT  $P_\ell$ -PACKING for any positive integer  $\ell$  is solvable in polynomial time for trees [28] and EDGE DISJOINT  $P_3$ -PACKING is solvable in linear time for general graphs [28].

VERTEX (EDGE) DISJOINT  $\mathcal{H}$ -PACKING is closely related to the SET PACKING problem. Given a universe  $U$ , a non-negative integer  $k$ , and a collection  $\mathcal{L}$  of subsets of  $U$ , SET PACKING asks whether there exists a subcollection  $\mathcal{S} \subseteq \mathcal{L}$  such that  $|\mathcal{S}| \geq k$  and  $\mathcal{S}$  are mutually disjoint. If  $\mathcal{H}$  contains all (possibly disconnected) graphs, then these problems are equivalent because it is possible to construct a graph whose vertices (edges) correspond to elements of  $U$ . If  $\mathcal{H}$  contains restricted graphs, then VERTEX (EDGE) DISJOINT  $\mathcal{H}$ -PACKING are special cases of SET PACKING. Therefore, all tractable results for SET PACKING are applicable to our problems. This paper further seeks tractable cases of SET PACKING from the viewpoint of graph structure.

## 2 Preliminaries

For a positive integer  $i$ , we denote  $[i] = \{1, 2, \dots, i\}$ .

Let  $G$  be a graph. Throughout this paper, we assume that  $G$  is simple, that is, it has neither self-loops nor parallel edges. The sets of vertices and edges of  $G$  are denoted by  $V(G)$  and  $E(G)$ , respectively. For  $v \in V(G)$ , we denote by  $N_G(v)$  the set of neighbor of  $v$ , by  $E_G(v)$  the set of incident edges of  $v$ , and by  $d_G(v)$

the degree of  $v$  in  $G$ . The maximum degree of a vertex in  $G$  is denoted by  $\Delta(G)$  and the minimum degree of a vertex in  $G$  is denoted by  $\delta(G)$ . We may simply write  $uv$  to denote an edge  $\{u, v\}$ . For a positive integer  $t$ , we denote by  $tG$  the disjoint union of  $t$  copies of  $G$ . For a graph  $H$ , the  $H$ -vertex-conflict graph of  $G$ , denoted  $I_H^V(G)$ , is defined as follows. Each vertex of  $I_H^V(G)$  corresponds to a subgraph isomorphic to  $H$  in  $G$ . Two vertices of  $I_H^V(G)$  are adjacent if and only if the corresponding subgraphs in  $G$  share a vertex. The  $H$ -edge-conflict graph of  $G$ , denoted  $I_H^E(G)$ , is defined by replacing the adjacency condition in the definition of  $I_H^V(G)$  as: two vertices of  $I_H^E(G)$  are adjacent if and only if they share an edge in  $G$ .

A *claw* is a star graph with three leaves. A graph is said to be *claw-free* if it has no claw as an induced subgraph. Minty [30] and Sbihi [34] showed that the maximum independent set problem can be solved in polynomial time on claw-free graphs. This immediately implies the following proposition, which is a key to our polynomial-time algorithms.

**Proposition 1.** *If  $I_H^V(G)$  is claw-free, then VERTEX DISJOINT LIST  $H$ -PACKING can be solved in  $n^{O(|V(H)|)}$  time. Moreover, if  $I_H^E(G)$  is claw-free, then EDGE DISJOINT LIST  $H$ -PACKING can be solved in  $n^{O(|V(H)|)}$  time as well.*

A *tree decomposition* of  $G$  is a pair  $\mathcal{T} = (T, \{X_i : i \in V(T)\})$  of a tree  $T$  and vertex sets  $\{X_i : i \in V(T)\}$ , called *bags*, that satisfies the following conditions:  $\bigcup_{i \in V(T)} X_i = V(G)$ ; for  $\{u, v\} \in E(G)$ , there is  $i \in V(T)$  such that  $\{u, v\} \subseteq X_i$ ; and for  $v \in V(T)$ , the bags containing  $v$  induces a subtree of  $T$ . The *width* of  $\mathcal{T}$  is defined as  $\max_{i \in V(T)} |X_i| - 1$ , and the *treewidth* of  $G$  is the minimum integer  $k$  such that  $G$  has a tree decomposition of width  $k$ . A *path decomposition* of  $G$  is a tree decomposition of  $G$ , where  $T$  forms a path. The *pathwidth* of  $G$  is defined analogously. Let  $\pi: V(G) \rightarrow [|V(G)|]$  be a bijection, which we call a *linear layout* of  $G$ . The width of a linear layout  $\pi$  of  $G$  is defined as  $\max_{\{u, v\} \in E(G)} |\pi(u) - \pi(v)|$ . The *bandwidth* of  $G$  is the minimum integer  $k$  such that  $G$  has a linear layout of width  $k$ .

For the treewidth, pathwidth, and bandwidth of  $G$ , we denote them by  $\text{tw}(G)$ ,  $\text{pw}(G)$ , and  $\text{bw}(G)$ . We may simply write them as  $\text{tw}$ ,  $\text{pw}$ , and  $\text{bw}$  without specific reference to  $G$ . It is well known that  $\text{tw}(G) \leq \text{pw}(G) \leq \text{bw}(G)$  for every graph  $G$  [6].

### 3 List $C_\ell$ -packing on bounded degree graphs

In this section, we focus on VERTEX DISJOINT LIST  $C_\ell$ -PACKING and EDGE DISJOINT LIST  $C_\ell$ -PACKING on bounded degree graphs. We give sufficient conditions to solve VERTEX (EDGE) DISJOINT  $H$ -PACKING in polynomial time, which extend polynomial-time algorithms for  $\ell = 3$  [9] to more general cases. We also show that these conditions are “tight” in some sense.

**Theorem 1.** *VERTEX DISJOINT LIST  $H$ -PACKING can be solved in polynomial time if the following inequality holds:*

$$\Delta(G) \leq 2\delta(H) - \left\lfloor \frac{|V(H)|}{3} \right\rfloor.$$

*Proof.* By Proposition 1, it suffices to show that if  $I_H^V(G)$  has a claw as an induced subgraph, then  $G$  has a vertex of degree more than  $2\delta(H) - \lfloor |V(H)|/3 \rfloor$ . Let  $H^*$ ,  $H_1$ ,  $H_2$ , and  $H_3$  be induced copies of  $H$  in  $G$  such that they correspond to an induced claw in  $I_H^V(G)$  whose center is  $H^*$ . This implies that  $V(H^*) \cap V(H_i) \neq \emptyset$  for  $1 \leq i \leq 3$  and  $V(H_i) \cap V(H_j) = \emptyset$  for  $1 \leq i < j \leq 3$ . This implies that some copy of  $H$ , say  $H_1$ , satisfies  $|V(H^*) \cap V(H_1)| \leq \lfloor |V(H)|/3 \rfloor$ . Let  $v \in V(H^*) \cap V(H_1)$ . The vertex  $v$  has at least  $\delta(H)$  neighbors in each of  $H^*$  and  $H_1$  and at most  $\lfloor |V(H)|/3 \rfloor - 1$  of them belong to  $V(H^*) \cap V(H_1)$ . Thus,

$$\begin{aligned} d_G(v) &\geq 2\delta(H) - (|V(H^*) \cap V(H_1)| - 1) \\ &\geq 2\delta(H) - \left( \left\lfloor \frac{|V(H)|}{3} \right\rfloor - 1 \right) \\ &> 2\delta(H) - \left\lfloor \frac{|V(H)|}{3} \right\rfloor, \end{aligned}$$

which proves the claim.  $\square$

**Theorem 2.** EDGE DISJOINT LIST  $H$ -PACKING can be solved in polynomial time if the following inequality holds:

$$\Delta(G) \leq 2\delta(H) - \left\lfloor \frac{|E(H)|}{3} \right\rfloor,$$

except that  $H = tK_2$  for  $3 \leq t \leq 5$ .

*Proof.* We first assume that  $|E(H)| \geq 6$ . The proof strategy is analogous to that in Theorem 1. Let  $H^*$ ,  $H_1$ ,  $H_2$ , and  $H_3$  be induced copies of  $H$  in  $G$  such that they correspond to an induced claw in  $I_H^V(G)$  whose center is  $H^*$ . We assume that  $|E(H^*) \cap E(H_1)| \leq \lfloor |E(H)|/3 \rfloor$ . As  $E(H^*) \cap E(H_1) \neq \emptyset$ , there are at least two vertices, say  $v$  and  $w$ , in  $V(H^*) \cap V(H_1)$ . Both  $v$  and  $w$  have at least  $\delta(H)$  incident edges in each of  $H^*$  and  $H_1$ . If  $v$  has at most  $\lfloor |E(H)|/3 \rfloor - 1$  incident edges that belong to  $E(H^*) \cap E(H_1)$ , then we have

$$d_G(v) \geq 2\delta(H) - \left( \left\lfloor \frac{|E(H)|}{3} \right\rfloor - 1 \right) > 2\delta(H) - \left\lfloor \frac{|E(H)|}{3} \right\rfloor.$$

Suppose otherwise. Then, all the edges in  $E(H^*) \cap E(H_1)$  are incident to  $v$ . This implies that, at most one edge (i.e.,  $\{v, w\}$  if it exists) in  $E(H^*) \cap E(H_1)$  can be incident to  $w$ . Thus, we have  $d_G(w) \geq 2\delta(H) - 1$ . By the assumption  $|E(H)| \geq 6$ , we have  $d_G(w) > 2\delta(H) - \lfloor |E(H)|/3 \rfloor$ .

We next consider the other case, that is,  $|E(H)| \leq 5$ , and show that EDGE DISJOINT LIST  $H$ -PACKING is solvable in polynomial time under the assumption that  $\Delta(G) \leq 2\delta(H) - \lfloor |E(H)|/3 \rfloor$ . We can assume that  $|E(H)| \geq 3$  as otherwise  $I_H^E(G)$  has no induced claws. If  $\delta(H) \geq 3$ , then it has at least four vertices, and thus we have  $|E(H)| \geq |V(H)| \cdot \delta(H)/2 \geq 4\delta(H)/2 \geq 6$ . Thus, we assume  $1 \leq \delta(H) \leq 2$ . If  $\delta(H) = 2$ , then  $\Delta(G) \leq 4 - \lfloor |E(H)|/3 \rfloor = 3$ . In this case, two copies of  $H$  appearing in  $G$  are edge-disjoint if and only if they are vertex-disjoint. Moreover, as  $|E(H)| \geq |V(H)|$ , we have  $\Delta(G) \leq 2\delta(H) - \lfloor |V(H)|/3 \rfloor$ , yields a polynomial-time algorithm by Theorem 1. Thus, the remaining case is  $\delta(H) = 1$ , that is,  $H = tK_2$  for  $3 \leq t \leq 5$ .  $\square$

Let us note that the exception in Theorem 2 is critical for the tractability of EDGE DISJOINT LIST  $H$ -PACKING. In fact, EDGE DISJOINT LIST  $H$ -PACKING is NP-complete even if  $G = nK_2$  and  $H = 3K_2$ , which satisfy that  $\Delta(G) = 1 \leq 2\delta(H) - \lfloor |E(H)|/3 \rfloor$ . This intractability is shown by reducing EXACT COVER BY 3-SETS, which is known to be NP-complete [23], to EDGE DISJOINT LIST  $H$ -PACKING. In EXACT COVER BY 3-SETS, we are given a universe  $U$  and a collection  $\mathcal{C}$  of subsets of  $U$ , each of which has exactly three elements and asked whether there is a pairwise disjoint subcollection  $\mathcal{C}'$  of  $\mathcal{C}$  that covers  $U$  (i.e.,  $U = \bigcup_{S \in \mathcal{C}'} S$ ). This problem is a special case of EDGE DISJOINT LIST  $H$ -PACKING, where  $G$  has a copy of  $K_2$  corresponding to each element in  $U$  and  $\mathcal{S}$  consists of all copies of  $3K_2$  corresponding to  $\mathcal{C}$ . This reduction also proves that VERTEX DISJOINT LIST  $H$ -PACKING is NP-complete even if  $G = nK_2$  and  $H = 3K_2$ .

**Theorem 3.** VERTEX DISJOINT LIST  $H$ -PACKING and EDGE DISJOINT LIST  $H$ -PACKING are NP-complete even if  $G = nK_2$  and  $H = 3K_2$ .

As consequences of Theorems 1 and 2, we have the following positive results.

**Theorem 4.** For  $\ell \in \{4, 5\}$ , there are polynomial-time algorithms for VERTEX DISJOINT LIST  $C_\ell$ -PACKING and EDGE DISJOINT LIST  $C_\ell$ -PACKING on graphs of maximum degree 3.

Contrary to this tractability, for any  $\ell \geq 6$ , VERTEX DISJOINT  $C_\ell$ -PACKING and EDGE DISJOINT  $C_\ell$ -PACKING are NP-complete even on planar graphs of maximum degree 3.

**Theorem 5.** For  $\ell \geq 6$ , VERTEX DISJOINT  $C_\ell$ -PACKING and EDGE DISJOINT  $C_\ell$ -PACKING are NP-complete even on planar graphs of maximum degree 3.

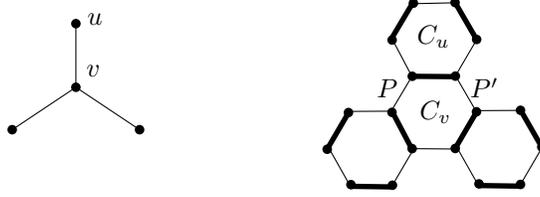


Figure 1: An example of the construction of  $G'$  for  $\ell = 6$ . Bold lines indicate a matching of each cycle.

*Proof.* Since VERTEX DISJOINT  $C_\ell$ -PACKING and EDGE DISJOINT  $C_\ell$ -PACKING are equivalent on graphs of maximum degree 3, we only consider VERTEX DISJOINT  $C_\ell$ -PACKING.

To show NP-hardness, we perform a polynomial-time reduction from INDEPENDENT SET, which is known to be NP-complete even on planar graphs with maximum degree 3 and girth at least  $p$  for any integer  $p$  [31]. Here, the *girth* of  $G$  is the length of a shortest cycle in  $G$ .

Let  $G$  be a planar graph with  $\Delta(G) \leq 3$  and girth at least  $\ell + 1$ . We construct a graph  $G'$  as follows. For each  $v \in V(G)$ ,  $G'$  contains a cycle  $C_v$  of length  $\ell$ . These cycles are called *primal cycles* in  $G'$ . Let  $M_v$  be an arbitrary matching in  $C_v$  with  $|M_v| = 3$ . For two adjacent vertices  $u, v$  in  $G$ , we identify one of the edges in  $M_u$  and in  $M_v$  as shown in Figure 1. The edges in  $M_v$  are called *shared edges*. By removing the shared edges from  $C_v$ , we obtain three paths, which we call *private paths* in  $C_v$ . Note that each private path belongs to exactly one primal cycle in  $G'$ . The construction of  $G'$  is done. It is easy to observe that  $G'$  is planar and has maximum degree 3. Also observe that the degree of each internal vertex of a private path is exactly 2.

We first claim that the length of cycles in  $G'$  except for primal cycles is greater than  $\ell$ . To see this, let  $C$  be an arbitrary cycle that is not a primal cycle in  $G'$ . As shared edges form a matching in  $G'$ ,  $C$  must contain at least one private path  $P$  in  $C_v$  for some  $v \in V(G)$ . If  $C$  contains all the private paths in  $C_v$ , the length of  $C$  is greater than  $\ell$ , except for the case  $C = C_v$ . Thus, we assume that  $C$  does not contain one of the private paths, say  $P'$ , in  $C_v$ . Let  $W = (v_0, v_1, \dots, v_t)$  be a sequence of vertices of  $G$  defined as follows. We first contract all shared edges in  $C$ . Then, the contracted cycle can be partitioned into maximal subpaths  $P_0, \dots, P_t$  such that each  $P_i$  consists of edges of (possibly more than one) private paths in  $C_{v_i}$  for some  $v_i \in V(G)$ . Due to the maximality of  $P_i$ , we have  $v_i \neq v_{i+1}$  for  $0 \leq i \leq t$ , where the addition in the subscript is taken modulo  $t + 1$ . Moreover, the sequence contains at least two vertices as  $C$  contains  $P$  and does not contain  $P'$ , meaning that it must have a private path in  $C_{v'}$  for some  $v' \neq v$ . For any pair of private paths in  $C_u$  and in  $C_w$ , they are adjacent with a shared edge if and only if  $u$  and  $w$  are adjacent in  $G$ . Thus,  $W$  is a closed walk in  $G$ .

Suppose that  $W$  contains a “turn”, that is,  $v_i = v_{i+2}$  for some  $i$ . This implies that  $C$  contains all private paths of  $C_{v_{i+1}}$ , which implies that the length of  $C$  is more than  $\ell$ . Otherwise,  $W$  contains a cycle in  $G$ . Since the girth of  $G$  is at least  $\ell + 1$ ,  $W$  has more than  $\ell$  edges. Hence,  $C$  contains more than  $\ell + 1$  private paths.

Now, we are ready to prove that  $G$  has an independent set of size at least  $k$  if and only if  $G'$  has a  $C_\ell$ -packing of size at least  $k$ . From an independent set of  $G$ , we can construct a vertex-disjoint  $C_\ell$ -packing by just taking primal cycles corresponding to vertices in the independent set. Since every cycle except for primal cycles has length more than  $\ell$ , this correspondence is reversible: From a vertex-disjoint  $C_\ell$ -packing of  $G'$  with size  $k$ , we can construct an independent set of  $G$  with size  $k$ .  $\square$

Using a similar strategy of Theorem 5, we prove the following theorems.

**Theorem 6.** For  $\ell \in \{4, 5\}$ , VERTEX DISJOINT  $C_\ell$ -PACKING is NP-complete even on planar graphs of maximum degree 4.

*Proof.* We again give a polynomial-time reduction from INDEPENDENT SET on planar graph with maximum degree at most 3 and girth at least  $\ell + 1$ . Let  $G$  be a planar graph with maximum degree 3 and girth at least  $\ell + 1$ . We construct a graph  $G'$  such that  $G$  has an independent set of size  $k$  if and only if  $G'$  has  $k$  vertex-disjoint  $C_\ell$ 's. The construction is almost analogous to one used in Theorem 5. For each  $v \in V(H)$ ,  $G'$

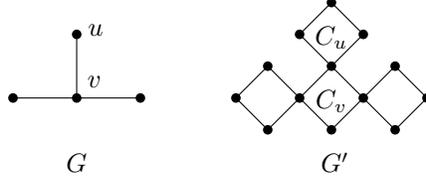


Figure 2: An example of the construction of  $G'$  for  $\ell = 4$ .

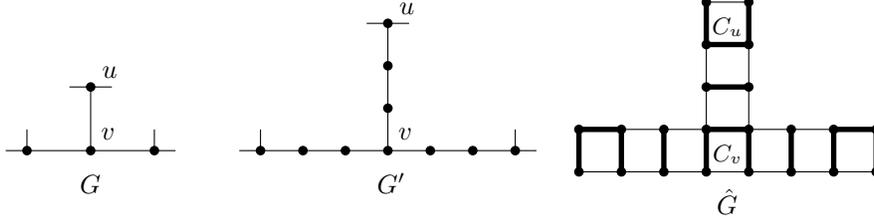


Figure 3: An example of the constructions of  $H'$  and  $G$  for  $\ell = 4$ . Bold lines depict shared edges.

contains a cycle  $C_v$  of length  $\ell$ . These cycles are called primal cycles in  $G'$ . We take arbitrary three vertices of  $C_v$ , which are called *shared vertices*. For two adjacent vertices  $u, v$  in  $G$ , we identify one of the shared vertices in  $C_u$  and in  $C_v$  as shown in Figure 2. It is easy to see that the constructed graph  $G'$  is planar and has degree at most 4.

Let  $C$  be a cycle of  $G'$  that is not a primal cycle. Since each primal cycle has only three shared vertices,  $C$  passes through a primal cycle at most once, where we say that  $C$  passes through a primal cycle  $C_v$  when  $C$  contains at least one edge of  $C_v$ . This fact implies that  $C$  induces a cycle of  $G$  by tracing  $C$  as in the proof of Theorem 5. Thus, we can conclude that  $C$  has more than  $\ell$  edges.

The remaining part of the proof is identical to that in Theorem 5. For every independent set  $S$  of  $G$ , the primal cycles of them are vertex-disjoint  $C_\ell$ 's in  $G$  and vice versa.  $\square$

**Theorem 7.** For  $\ell \in \{4, 5\}$ , EDGE DISJOINT  $C_\ell$ -PACKING is NP-complete even on planar graphs of maximum degree 4.

*Proof.* The proof is done by showing a polynomial-time reduction from INDEPENDENT SET on planar graphs with maximum degree 3. Let  $G$  be a planar graph with maximum degree 3. We construct a planar graph  $\hat{G}$  with maximum degree at most 4 such that  $G$  has an independent set of size  $k$  if and only if  $\hat{G}$  has  $k'$  edge-disjoint  $C_\ell$ 's for some  $k'$ . The construction is done in the same spirit of Theorems 5 and 6 but we need to make a minor modification to keep the upper bound on its degree. We first subdivide each edge of  $G$  twice. The subdivided graph is denoted by  $G'$ . It is well known that  $G$  has an independent set of size  $k$  if and only if  $G'$  has an independent set of size  $k + |E(G)|$  [32]. For each vertex  $v \in V(G')$ ,  $\hat{G}$  contains a cycle of length  $\ell$ , which we call a primal cycle of  $v$ . For such a cycle  $C_v$ , we take arbitrary three edges if the degree of  $v$  is 3 in  $G'$ , and take a matching with two edges otherwise. These edges are called shared edges. We then identify one of shared edges in  $C_u$  and that in  $C_v$  if they are adjacent in  $G'$ . See Figure 3 for an illustration. It is easy to see that  $\hat{G}$  is planar. As  $G'$  has no pair of adjacent vertices of degree 3, for each vertex in  $\hat{G}$ , there are at most two shared edges and at most two non-shared edges incident to it, which implies that the maximum degree of  $\hat{G}$  is at most 4.

Now, we show that  $G'$  has an independent set of size  $k'$  if and only if  $\hat{G}$  has  $k'$  edge-disjoint  $C_\ell$ 's. This can be shown by observing that every cycle of  $\hat{G}$  has more than  $\ell$  edges except for primal cycles. The remaining part of the proof is analogous to ones in Theorems 5 and 6.  $\square$

## 4 Vertex Disjoint List $\mathcal{H}$ -Packing on bounded-treewidth graphs

In Section 5, we will see that EDGE DISJOINT LIST  $\mathcal{H}$ -PACKING is intractable even if  $\mathcal{H}$  contains a single small connected graph and an input graph is series-parallel. In contrast to this intractability, VERTEX DISJOINT LIST  $\mathcal{H}$ -PACKING is polynomial-time solvable on series-parallel graphs and, more generally, bounded-treewidth graphs, if  $\mathcal{H}$  consists of a finite number of connected graphs. More precisely, we show that VERTEX DISJOINT LIST  $\mathcal{H}$ -PACKING is XP parameterized by treewidth, provided that  $\mathcal{H}$  consists of connected graphs. We also show that VERTEX DISJOINT LIST  $\mathcal{P}$ -PACKING and VERTEX DISJOINT LIST  $\mathcal{C}$ -PACKING are W[1]-hard parameterized by pathwidth. (Recall that  $\mathcal{P} = \{P_\ell : \ell \geq 1\}$  and  $\mathcal{C} = \{C_\ell : \ell \geq 3\}$ .)

**Theorem 8.** VERTEX DISJOINT LIST  $\mathcal{H}$ -PACKING is solvable in  $n^{O(\text{tw})}$  time, provided that all graphs in  $\mathcal{H}$  are connected, where  $n$  is the number of vertices in the input graph.

We only sketch the proof of Theorem 8 by giving a rough idea of a dynamic programming algorithm based on tree decompositions. (A similar idea is used in [35] for instance.) Let  $G$  be a graph with  $n$  vertices. We can find a tree decomposition  $\mathcal{T} = (T, \{X_i : i \in V(T)\})$  of  $G$  of width  $\text{tw}(G)$  in time  $n^{O(\text{tw})}$  using the algorithm of [3]. By taking an arbitrary node of  $T$ , we assume that  $T$  is rooted. For each bag  $X_i$  of  $\mathcal{T}$ , we denote by  $G_i$  the subgraph of  $G$  induced by the vertices contained in  $X_i$  or descendant bags of  $X_i$ .

Let  $\mathcal{L}_{\mathcal{H}}$  be a collection of connected subgraphs of  $G$ , each of which is isomorphic to some graph in  $\mathcal{H}$ . For  $i \in V(T)$ , a *partial  $\mathcal{H}$ -packing* of  $G_i$  is a vertex-disjoint subcollection  $\mathcal{S}$  of  $\mathcal{L}_{\mathcal{H}}$ , each of which contains at least one vertex of  $G_i$ . A subgraph in a partial  $\mathcal{H}$ -packing is said to be *active* if it has at least one vertex in  $V(G) \setminus V(G_i)$ . A key observation to our dynamic programming algorithm is that each active subgraph in any partial  $\mathcal{H}$ -packing contains at least one vertex of  $X_i$ . This follows from the fact that every graph in  $\mathcal{L}_{\mathcal{H}}$  is connected. This implies that there are at most  $|X_i|$  active subgraphs in any partial  $\mathcal{H}$ -packing of  $G_i$ .

To be slightly more precise, we define  $\text{opt}(i, \hat{\mathcal{S}})$  as the maximum cardinality of a partial  $\mathcal{H}$ -packing  $\mathcal{S}$  such that  $\hat{\mathcal{S}} \subseteq \mathcal{S}$  is the set of active subgraphs in it. By a standard argument in dynamic programming over tree decompositions, we can compute  $\text{opt}(i, \hat{\mathcal{S}})$  in a bottom-up manner. The total running time of computing all possible  $\text{opt}(i, \hat{\mathcal{S}})$  for  $i$  and  $\hat{\mathcal{S}}$  is upper bounded by  $|\mathcal{L}_{\mathcal{H}}|^{O(\text{tw})} n = n^{O(\text{tw})}$ .

We would like to note that the connectivity of  $\mathcal{H}$  is crucial as we have seen in Section 3 that VERTEX DISJOINT LIST  $\mathcal{H}$ -PACKING is NP-complete even if  $G = nK_2$  and  $H = 3K_2$ .

The following theorems complement the positive result of Theorem 8.

**Theorem 9.** VERTEX DISJOINT LIST  $\mathcal{P}$ -PACKING is W[1]-hard parameterized by  $\text{pw} + k$ .

*Proof.* The proof is done by showing a parameterized reduction from MULTICOLORED INDEPENDENT SET, which is known to be W[1]-complete [12, 15]. In MULTICOLORED INDEPENDENT SET, we are given a graph  $G$  with a partition of vertex set  $V(G) = V_1 \cup V_2 \cup \dots \cup V_t$  and asked whether  $G$  has an independent set  $S$  of size  $t$  that contains exactly one vertex from  $V_i$  for each  $1 \leq i \leq t$ .

From an instance of MULTICOLORED INDEPENDENT SET  $G$  with  $V(G) = V_1 \cup \dots \cup V_t$ , we construct a graph  $G'$  as follows. Let  $m = |E(G)|$ . The vertex set of  $G'$  consists of  $m + 1$  vertex sets  $V'_0 \cup V'_1 \cup \dots \cup V'_m$ , where each set  $V'_i$  with  $i \geq 1$  is associated with an edge  $e_i$  of  $G$ . The vertex set  $V'_0$  contains  $t$  vertices  $v_1, \dots, v_t$  and other sets  $V'_i$  for  $i \geq 1$  contain  $t + 1$  vertices  $v_{i,1}, \dots, v_{i,t}, w_i$ . For  $1 \leq i \leq m$ , we add an edge between every pair of vertices  $v \in V_{i-1}$  and  $v' \in V_i$ . The graph obtained in this way is denoted by  $G'$ . Observe that the pathwidth of  $G'$  is at most  $2t + 1$ . This can be seen by observing that  $G'$  has a path decomposition consisting of bags  $X_i := V'_{i-1} \cup V'_i$  for  $1 \leq i < m$ , which has width at most  $2t + 1$ . For each vertex  $v \in V_i$ , we define a path  $P_v$  in  $G'$  that consists of (1)  $v_i$  and (2) for all  $1 \leq j \leq m$ ,  $v_{i,j}$  if  $e_j$  is not incident to  $v$  and  $w_i$  otherwise. As  $P_v$  contains exactly one vertex from each  $V'_i$ ,  $P_v$  forms a path in  $G'$ . See Figure 4 for an illustration. Let  $\mathcal{L}_{\mathcal{H}} = \{P_v : v \in V(G)\}$ . Now, we claim that  $G$  has an independent set  $S$  that contains exactly one vertex from each  $V_i$  if and only if there is a  $P_{m+1}$ -packing  $\mathcal{S} \subseteq \mathcal{L}_{\mathcal{H}}$  with  $|\mathcal{S}| \geq t$  in  $G'$ .

Suppose that  $G$  has an independent set  $S$  with  $|S \cap V_i| = 1$  for  $1 \leq i \leq t$ . Let  $\mathcal{S} = \{P_v : v \in S\}$ . For distinct paths  $P_v, P_{v'} \in \mathcal{S}$  with  $v \in V_i$  and  $v' \in V_{i'}$ , suppose that they share a vertex  $u$ . As  $i \neq i'$ , the vertex  $u$  is contained in some  $V_j$  as  $u = w_j$ . This implies that  $v$  is adjacent to  $v'$ , contradicting to the fact that  $S$  is an independent set of  $G$ .

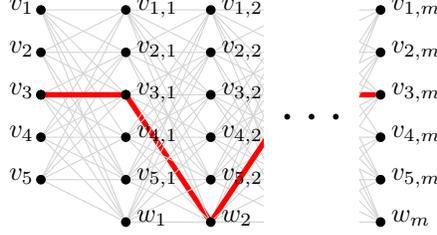


Figure 4: The figure illustrates the graph  $G'$ . The red thick lines indicate the edges of path  $P_v$  for  $v \in V_3$ . The edge  $e_1$  is not incident to  $v$  but  $e_2$  is incident to  $v$ .

For the converse direction, suppose that there is a  $P_{m+1}$ -packing  $\mathcal{S} \subseteq \mathcal{L}_{\mathcal{H}}$  with  $|\mathcal{S}| \geq t$  in  $G'$ . Let  $S = \{v \in V(G) : P_v \in \mathcal{S}\}$ . Since each subgraph contains exactly one vertex from  $V_0$ , we have  $|\mathcal{S}| = t$ . Moreover,  $S$  contains exactly one vertex from each  $V_i$ . If there are two vertices in  $S$  that are adjacent (by an edge  $e_j$ ) to each other, the corresponding paths in  $\mathcal{S}$  share the vertex  $w_j$  in  $V_j$ . Hence,  $\mathcal{S}$  is an independent set of  $G$ .  $\square$

**Theorem 10.** VERTEX DISJOINT LIST  $\mathcal{C}$ -PACKING is  $W[1]$ -hard parameterized by  $\text{pw} + k$ .

*Proof.* This proof is done by modifying the instance constructed in the proof of Theorem 9. For the graph  $G'$ , we additionally give an edge between every pair of vertices  $v \in V'_0$  and  $v' \in \cup V'_m$ . Observe that there exists a path decomposition consisting of bags  $X_i := V'_{i-1} \cup V'_i \cup V'_m$  for  $1 \leq i < m$ , which has width at most  $3t + 2$ . For each vertex  $v \in V_i$ , we define a cycle  $C_v$  in  $G'$  that consists of (1)  $v_i$  and (2) for all  $1 \leq j \leq m$ ,  $v_{i,j}$  if  $e_j$  is not incident to  $v$  and  $w_i$  otherwise. The remaining part of the proof is identical to that in Theorem 9.  $\square$

## 5 Edge Disjoint List $\mathcal{H}$ -Packing on series-parallel graphs

This section is devoted to showing several positive and negative results on *series-parallel graphs*. The class of series-parallel graphs is a well-studied class of graphs and is equivalent to the class of graphs of treewidth at most 2.

A *two-terminal labeled graph* is a graph  $G$  with distinguished two vertices called a *source*  $s$  and a *sink*  $t$ . Let  $G_1 = (V_1, E_1)$  (resp.  $G_2 = (V_2, E_2)$ ) be a two-terminal labeled graph with a source  $s_1$  and a sink  $t_1$  (resp. a source  $s_2$  and a sink  $t_2$ ). A *series composition* of  $G_1$  and  $G_2$  is an operation that produces the two-terminal labeled graph with a source  $s$  and a sink  $t$  obtained from  $G_1$  and  $G_2$  by identifying  $t_1$  and  $s_2$ , where  $s = s_1$  and  $t = t_2$ . A *parallel composition* of  $G_1$  and  $G_2$  is an operation that produces the two-terminal labeled graph with a source  $s$  and a sink  $t$  obtained from  $G_1$  and  $G_2$  by identifying  $s_1$  and  $s_2$ , and identifying  $t_1$  and  $t_2$ , where  $s = s_1 (= s_2)$  and  $t = t_1 (= t_2)$ . We denote  $G = G_1 \bullet G_2$  if  $G$  is created by a series composition of  $G_1$  and  $G_2$ , and denote  $G = G_1 \parallel G_2$  if  $G$  is created by a parallel composition of  $G_1$  and  $G_2$ . We say that a two-terminal labeled graph  $G$  is a *two-terminal series-parallel graph* if one of the following conditions is satisfied: (i)  $G = K_2$  with a source  $s$  and a sink  $t$ ; (ii)  $G = G_1 \bullet G_2$  for two-terminal series-parallel graphs  $G_1$  and  $G_2$ ; or (iii)  $G = G_1 \parallel G_2$  for two-terminal series-parallel graphs  $G_1$  and  $G_2$ .

We say that a graph  $G$  (without a source and a sink) is a *series-parallel graph* if each biconnected component is a two-terminal series-parallel graph by regarding some two vertices as a source and a sink<sup>1</sup>.

### 5.1 Hardness

A graph  $G$  is *outerplanar* if it has a planar embedding such that every vertex of  $G$  meets the unbounded face of the embedding. Every outerplanar graph is series-parallel but may not be two-terminal series-parallel.

<sup>1</sup>Some papers refer to a two-terminal series-parallel graph simply as a series-parallel graph. In this paper, we distinguish them explicitly to avoid confusion.

The following two theorems indicate that EDGE DISJOINT LIST  $\mathcal{H}$ -PACKING remains intractable even when a given graph is highly restricted.

**Theorem 11.** EDGE DISJOINT LIST  $\mathcal{P}$ -PACKING parameterized by  $\text{bw}(G) + k$  is  $W[1]$ -hard even for outerplanar and two-terminal series-parallel graphs, where  $k$  is a solution size.

We prove Theorem 11 by giving a parameterized reduction from MULTICOLORED INDEPENDENT SET to EDGE DISJOINT LIST  $\mathcal{P}$ -PACKING. Recall that, in MULTICOLORED INDEPENDENT SET, we are given a graph  $G'$  with a partition of its vertex set  $V(G') = V_1 \cup V_2 \cup \dots \cup V_t$ . We may assume that  $V_i$  forms a clique of  $G'$ .

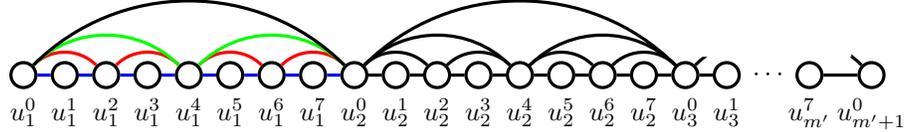


Figure 5: The figure partially illustrates the graph  $G$  for  $t = 3$ . The edges in  $E_1^1$ , in  $E_1^2$ , and in  $E_1^3$  are depicted as green segments, red segments, and blue segments, respectively.

From an instance of MULTICOLORED INDEPENDENT SET, we construct an instance  $(G, t, \mathcal{L}_{\mathcal{H}})$  of EDGE DISJOINT LIST  $\mathcal{P}$ -PACKING. To this end, we construct a gadget  $G_i$  for each  $i \in [m']$ , where  $m' = |E(G')|$ . (See also Figure 5.) The vertex set of  $G_i = (W_i, E_i)$  is defined as  $W_i = \{u_i^j : j \in [2^t] \cup \{0\}\}$ . For  $j \in [t] \cup \{0\}$  and  $p \in [2^j] \cup \{0\}$ , let denote  $\lambda(j, p) = 2^{t-j}p$ . Then we define  $E_i^j = \{u_i^{\lambda(j, p-1)}, u_i^{\lambda(j, p)} : p \in [2^j]\}$ . In other words,  $E_i^j$  consists of all edges in the path starting from  $u_i^0$  to  $u_i^q$  with  $q = 2^t$  that contains  $u_i^r$  for  $r = 2^{t-j}, 2 \cdot 2^{t-j}, 3 \cdot 2^{t-j}, \dots, (2^j - 1)2^{t-j}$  as internal vertices. We define the edge set of  $G_i$  as  $E_i = \bigcup_{j \in [t] \cup \{0\}} E_i^j$ . After constructing  $G_1, \dots, G_{m'}$ , we concatenate the graphs by identifying  $u_i^q$  of  $G_i$  and  $u_{i+1}^0$  of  $G_{i+1}$  for every  $i \in [m' - 1]$ . For notational convenience, we relabel a vertex  $u_i^q$  as  $u_{i+1}^0$  for every  $i \in [m']$ . We define the graph constructed above as  $G$ .

Next, we construct a collection  $\mathcal{L}_{\mathcal{H}}$  of paths in  $G$ . Suppose that  $E(G') = \{e_1, e_2, \dots, e_{m'}\}$ . For each  $v \in V(G')$ , a path  $P_v$  of  $G$  is defined as follows. Suppose that  $v \in V_j$ . The path  $P_v$  starts at  $u_1^0$  and ends at  $u_{m'+1}^0$ . For each  $i \in [m']$ ,  $P_v$  passes through  $u_i^0 u_{i+1}^0$  if  $e_i$  is incident to  $v$ ; otherwise,  $P_v$  passes through all edges in  $E_i^j$ . For example, consider  $t = 3$  and a vertex  $v \in V_2$  such that only edges  $e_1, e_4$  are incident to  $v$ . Then,

$$P_v = (u_1^0, u_2^0, u_2^2, u_2^4, u_2^6, u_3^0, u_3^2, u_3^4, u_3^6, u_4^0, u_5^0, u_5^2, \dots, u_{m'}^6, u_{m'+1}^0).$$

We define  $\mathcal{L}_{\mathcal{H}} = \{P_v : v \in V(G')\}$ . This completes the construction of the instance  $(G, t, \mathcal{L}_{\mathcal{H}})$ . Let denote  $n = |V(G)|$  and  $m = |E(G)|$ .

Observe that  $G$  is an outerplanar graph and a two-terminal series-parallel graph. For  $n' = |V(G')|$  and  $m' = |E(G')|$ , we have  $n = 2^t m' + 1$ ,  $m = (2^{t+1} - 1)m'$ , and  $|\mathcal{L}_{\mathcal{H}}| = n'$ . Thus, the construction of  $(G, k, \mathcal{L}_{\mathcal{H}})$  is completed in  $O(2^t n' m')$  time. Moreover,  $\text{bw}(G) \leq 2^t$  holds: consider the linear layout  $\pi(u_i^j) = 2^t(i-1) + j + 1$  for  $i \in [m']$  and  $j \in [2^t - 1] \cup \{0\}$ , which is the same as the ordering of the vertices depicted in Figure 5. The following lemma completes the proof of Theorem 11.

**Lemma 1.** The instance  $(G', t, \{V_1, V_2, \dots, V_t\})$  of MULTICOLORED INDEPENDENT SET is a yes-instance if and only if the instance  $(G, t, \mathcal{L}_{\mathcal{H}})$  of EDGE DISJOINT LIST  $\mathcal{P}$ -PACKING is a yes-instance.

*Proof.* We first prove the necessity. For a solution  $\mathcal{S}$  of a yes-instance  $(G, t, \mathcal{L}_{\mathcal{H}})$  of EDGE DISJOINT LIST  $\mathcal{P}$ -PACKING, let  $I = \{v \in V(G') : P_v \in \mathcal{S}\}$ . Since  $\mathcal{S}$  are mutually edge-disjoint on  $G$ , at most one path in  $\mathcal{S}$  passes through  $u_i^0 u_{i+1}^0$  for each  $i \in [m']$ . This implies that at most one of the endpoints of  $e_i \in E(G')$  is in  $I$ , that is,  $I$  is an independent set of  $G'$ . Moreover, we have  $|I| \geq t$ . Since  $V_j$  is a clique of  $G'$ ,  $|V_j \cap I| = 1$  holds for every  $j \in [t]$ . Thus,  $(G', t, \{V_1, V_2, \dots, V_t\})$  is a yes-instance of MULTICOLORED INDEPENDENT SET.

We next prove the sufficiency. For a solution  $I \subseteq V(G')$  of a yes-instance  $(G', t, \{V_1, V_2, \dots, V_t\})$  of MULTICOLORED INDEPENDENT SET, let  $\mathcal{S} = \{P_v : v \in I\}$ . Since  $|\mathcal{S}| = t$ , it suffices to show that  $\mathcal{S}$  is mutually edge-disjoint. Let  $u, v \in I$ . Suppose that  $u \in V_j$  and  $v \in V_{j'}$ . Since  $|V_j \cap I| = 1$  for every  $j \in [t]$ ,  $j \neq j'$  holds. For paths  $P_u, P_v \in \mathcal{S}$  corresponding to  $u, v \in I$  and  $i \in [m']$ , we denote by  $P_{u,i}$  and  $P_{v,i}$ , the subpaths of  $P_u$  and  $P_v$  that start at  $u_i^0$  and end at  $u_{i+1}^0$ , respectively. In the remainder of this proof, we show that  $P_{u,i}$  and  $P_{v,i}$  are edge-disjoint for every  $i \in [m']$ , meaning that  $P_u$  and  $P_v$  are also edge-disjoint. Suppose that  $e_i$  is incident to neither  $u$  nor  $v$ . Then, it holds that  $E(P_{u,i}) = E_i^j$  and  $E(P_{v,i}) = E_i^{j'}$ . Since  $j \neq j'$ , we have  $E_i^j \cap E_i^{j'} = \emptyset$ . Suppose otherwise that  $e_i$  is incident to  $u$  or  $v$ , say  $u$ . This implies that  $e_i$  is not incident to  $v$  as  $u, v \in I$ . Therefore, we have  $E(P_u) = E_i^0$  and  $E(P_{v,i}) = E_i^{j'}$ . Since  $j' \in [t]$ ,  $P_{u,i}$  and  $P_{v,i}$  are edge-disjoint.  $\square$

By a similar proof, we can show the W[1]-hardness of EDGE DISJOINT LIST  $\mathcal{C}$ -PACKING.

**Theorem 12.** EDGE DISJOINT LIST  $\mathcal{C}$ -PACKING parameterized by  $\text{bw}(G) + k$  is W[1]-hard even for outerplanar and two-terminal series-parallel graphs, where  $k$  is a solution size.

*Proof.* We reuse the instance  $(G, k, \mathcal{L}_{\mathcal{H}})$  of EDGE DISJOINT LIST  $\mathcal{P}$ -PACKING from the proof of Theorem 11, which is constructed from the instance  $(G', t, \{V_1, V_2, \dots, V_t\})$  of MULTICOLORED INDEPENDENT SET. Let  $G^c$  denote the graph obtained by identifying  $u_1^0$  and  $u_{m+1}^0$  of  $G$ . Observe that  $G^c$  is outerplanar and two-terminal series-parallel. Moreover, this identification produces a collection  $\mathcal{L}_{\mathcal{H}}^c$  of cycles of  $G^c$  from  $\mathcal{L}_{\mathcal{H}}$ , because every path in  $\mathcal{L}_{\mathcal{H}}$  starts at  $u_1^0$  and ends at  $u_{m+1}^0$ . As in the proof of Lemma 1, we can show that the instance  $(G', t, \{V_1, V_2, \dots, V_t\})$  of MULTICOLORED INDEPENDENT SET is a yes-instance if and only if the instance  $(G^c, k, \mathcal{L}_{\mathcal{H}}^c)$  of EDGE DISJOINT LIST  $\mathcal{C}$ -PACKING is a yes-instance.

In the remainder of this proof, we show that  $\text{bw}(G^c) \leq 3 \cdot 2^t$ . For each  $i \in [m']$ , let  $\pi_i = \langle u_i^0, u_i^1, \dots, u_i^{q-1} \rangle$ , where  $q = 2^t$ . In addition, we define a permutation  $\rho$  of  $[m']$  as follows:

$$\rho(i) = \begin{cases} \frac{i+1}{2} & \text{if } i \text{ is odd,} \\ m' + 1 - \frac{i}{2} & \text{otherwise.} \end{cases}$$

Consider an ordering  $\pi = \langle \pi_{\rho(1)}, \pi_{\rho(2)}, \dots, \pi_{\rho(m')} \rangle$ . Recall that  $W_i = \{u_i^j : j \in [2^t] \cup \{0\}\}$  and  $u_i^{2^t} = u_{i+1}^0$  for the gadget  $G_i$ . From the construction of  $G^c$ , every vertex in  $W_i$  for  $i \in [m']$  is adjacent to only vertices in  $W_{i-1} \cup W_i \cup W_{i+1}$ , where we consider  $W_{m'} = W_0$  and  $W_1 = W_{m'+1}$ . Thus, for every vertex  $u$  in  $\pi_{\rho(i)}$ , each adjacent vertex of  $u$  appears between  $\pi_{\rho(i-2)}$  and  $\pi_{\rho(i+2)}$ . Since  $\pi_{\rho(i)}$  has  $2^t$  vertices, the bandwidth of  $\pi$  is bounded by  $3 \cdot 2^t$ , that is,  $\text{bw}(G^c) \leq 3 \cdot 2^t$ . This completes the proof of Theorem 12.  $\square$

We next focus on the case where  $\mathcal{H}$  consists of a single graph and show that the problem remains hard. Let  $K_{2,n}$  denotes the complete bipartite graph such that one side consists of two vertices and the other side consists of  $n$  vertices.

**Theorem 13.** EDGE DISJOINT LIST  $P_4$ -PACKING remains NP-complete even for the class of  $K_{2,n}$ .

Observe that  $K_{2,n}$  is a two-terminal series-parallel graph. Since EDGE DISJOINT LIST  $P_3$ -PACKING is solvable for general graphs, Theorem 13 suggests that the complexity dichotomy with respect to path length still holds for very restricted graphs. Moreover, Theorem 13 immediately provides the following corollary, which strengthens the hardness result in [2] that PATH SET PACKING is W[1]-hard when parameterized by vertex cover number of  $G$  plus maximum length of paths in a given collection  $\mathcal{L}$ .

**Corollary 1.** PATH SET PACKING is NP-complete even when a given graph has vertex cover number 2 and every path in  $\mathcal{L}$  is of length 3.

To prove Theorem 13, we perform a polynomial-time reduction from INDEPENDENT SET on cubic graphs, that is, graphs such that every vertex has degree exactly 3. INDEPENDENT SET on cubic graphs is known to be NP-complete [17].

For a cubic graph  $G'$ , we construct vertex sets  $X = \{x_u, x'_u : u \in V(G')\}$ ,  $Y = \{y_{u,e}, y_{v,e} : e = uv \in E(G')\}$ , and  $Z = \{z_e : e \in E(G')\}$ . A graph  $G$  consists of a vertex set  $X \cup Y \cup Z \cup \{s, t\}$  such that  $X \cup Y \cup Z$  forms an independent of  $G$  and every vertex in  $X \cup Y \cup Z$  is adjacent to both  $s$  and  $t$ . Obviously,  $G = K_{2, 2n'+3m'}$ .

We next construct a collection  $\mathcal{L}_{\mathcal{H}}$  of paths in  $G$  as follows:

- for  $u \in V(G')$ , let  $P_u^X = \langle s, x_u, t, x'_u \rangle$ ;
- for  $u \in V(G')$  with three incident edges  $e, f, g \in E_{G'}(u)$ , let

$$\begin{aligned} P_{u,e}^Y &= \langle s, y_{u,e}, t, x'_u \rangle, \\ P_{u,f}^Y &= \langle s, y_{u,f}, t, x_u \rangle, \\ P_{u,g}^Y &= \langle x_u, s, y_{u,g}, t \rangle; \end{aligned}$$

- for  $e = uv \in E(G')$ , let  $P_{u,e}^Z = \langle y_{u,e}, s, z_e, t \rangle$  and  $P_{v,e}^Z = \langle y_{v,e}, s, z_e, t \rangle$ .

Then, we define

$$\mathcal{L}_{\mathcal{H}} = \{P_u^X : u \in V(G')\} \cup \{P_{u,e}^Y : u \in V(G'), e \in E_{G'}(u)\} \cup \{P_{u,e}^Z, P_{v,e}^Z : e = uv \in E(G')\}$$

and  $k = k' + 2m'$ . Clearly, the construction of the instance  $(G, k, \mathcal{L}_{\mathcal{H}})$  for EDGE DISJOINT LIST  $P_4$ -PACKING is completed in polynomial time. Our remaining task is to prove the following lemma.

**Lemma 2.** *An instance  $(G', k')$  of INDEPENDENT SET is a yes-instance if and only if an instance  $(G, k, \mathcal{L}_{\mathcal{H}})$  of EDGE DISJOINT LIST  $P_4$ -PACKING is a yes-instance.*

*Proof.* We first show the forward implication. For an independent set  $I$  with  $|I| \geq k'$ , we construct a subcollection  $\mathcal{S} \subseteq \mathcal{L}_{\mathcal{H}}$  as follows:

$$\mathcal{S} = \{P_u^X : u \in I\} \cup \{P_{u,e}^Z : u \in I, e \in E_{G'}(u)\} \cup \{P_{u,e}^Y : u \notin I, e \in E_{G'}(u)\}.$$

Observe that  $|\mathcal{S}| \geq k' + 3n'$ . Since  $G'$  is a cubic graph, we have  $3n' = 2m'$  by the handshaking lemma, and hence  $|\mathcal{S}| \geq k' + 2m' = k$ . We claim that  $\mathcal{S}$  is mutually edge-disjoint. Assume for a contradiction that there are two paths  $P$  and  $P'$  in  $\mathcal{S}$  that have a common edge. There are three cases to consider: (i)  $P = P_u^X$  for some  $u \in V(G')$ ; (ii)  $P = P_{u,e}^Y$  for some  $u \in V(G')$  and  $e \in E_{G'}(u)$ ; and (iii)  $P = P_{u,e}^Z$  and  $P' = P_{v,f}^Z$  for some  $u, v \in V(G')$ ,  $e \in E_{G'}(u)$ , and  $f \in E_{G'}(v)$ .

In the case (i), from the construction of  $\mathcal{L}_{\mathcal{H}}$ , we have  $P' = P_{u,e}^Y$  for some  $e \in E_{G'}(u)$ . As  $P_u^X \in \mathcal{S}$ , we have  $u \in I$  but as  $P_{u,e}^Y \in \mathcal{S}$ , we have  $u \notin I$ , a contradiction. Consider the case (ii). Only  $P_u^X$  and  $P_{u,e}^Z$  can share an edge with  $P_{u,e}^Y$ , and thus we consider the latter case:  $P' = P_{u,e}^Z$ . This implies that  $u \in I$ , while  $P_{u,e}^Y \in \mathcal{S}$  also implies  $u \notin I$ , a contradiction. In the case (iii), one can observe that  $e = f = uv$ . Then, we have  $u, v \in I$ , a contradiction. Therefore,  $\mathcal{S}$  is a solution for  $(G, k, \mathcal{L}_{\mathcal{H}})$ .

We next show the backward implication. Let  $\mathcal{S}$  be a subcollection of  $\mathcal{L}_{\mathcal{H}}$  such that  $\mathcal{S}$  is mutually edge-disjoint and  $|\mathcal{S}| \geq k = k' + 2m'$ . Suppose that  $\mathcal{S}$  contains both  $P_u^X$  and  $P_v^X$  with  $e = uv \in E(G')$ . As they share edges with  $P_{u,e}^Y$  and  $P_{v,e}^Y$ , we have  $P_{u,e}^Y, P_{v,e}^Y \notin \mathcal{S}$ . Furthermore, at least one of  $P_{u,e}^Z$  and  $P_{v,e}^Z$  is not contained in  $\mathcal{S}$ . We then remove  $P_u^X$  from  $\mathcal{S}$  and add  $P_{u,e}^Y$  into  $\mathcal{S}$  if  $P_{u,e}^Z \notin \mathcal{S}$ ; otherwise, we remove  $P_v^X$  from  $\mathcal{S}$  and add  $P_{v,e}^Y$  into  $\mathcal{S}$ . This exchange keeps  $\mathcal{S}$  mutually edge-disjoint because  $P_{u,e}^Y$  (resp.  $P_{v,e}^Y$ ) shares edges only with  $P_u^X$  (resp.  $P_v^X$ ) in  $\mathcal{S}$ . Hence, we may assume that  $\mathcal{S}$  contains at most one of  $P_u^X$  and  $P_v^X$  for every  $uv \in E(G')$ . Let denote  $\mathcal{P}_e = \{P_{u,e}^Y, P_{u,e}^Z, P_{v,e}^Y, P_{v,e}^Z\}$  for  $e = uv \in E(G')$ . Observe that  $|\mathcal{P}_e \cap \mathcal{S}| \leq 2$  due to the construction of  $\mathcal{L}_{\mathcal{H}}$ . Thus,  $|\bigcup_{e \in E(G')} \mathcal{P}_e \cap \mathcal{S}| \leq 2m'$ . This implies that  $|\{P_u^X : u \in V(G')\} \cap \mathcal{S}| \geq k - 2m' = k'$ . Let  $I = \{u \in V(G') : P_u^X \in \mathcal{S}\}$ . From the assumption of  $\mathcal{S}$ ,  $I$  is an independent set of  $G'$ . This completes the proof of the lemma.  $\square$

We also show the complexity of EDGE DISJOINT LIST  $C_5$ -PACKING, which highlights the positive result in Section 5.2.

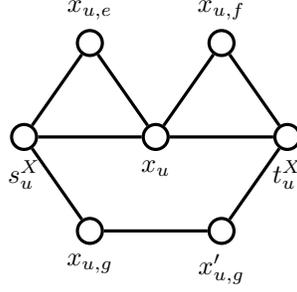


Figure 6: The gadget  $G_u^X$  for  $u \in V(G')$  with three incident edges  $e, f, g \in E_{G'}(u)$ .

**Theorem 14.** EDGE DISJOINT LIST  $C_5$ -PACKING remains NP-complete even for two-terminal series-parallel graphs.

We again perform a polynomial-time reduction from INDEPENDENT SET on cubic graphs. For a cubic graph  $G'$ , we use a gadget  $G_u^X$  depicted in Figure 6 for each  $u \in V(G')$ . The gadget  $G_u^X$  consists of seven vertices  $s_u^X, t_u^X, x_u, x_{u,e}, x_{u,f}, x_{u,g}$ , and  $x'_{u,g}$ , where  $e, f, g \in E_{G'}(u)$ . Observe that  $G_u^X$  is a two-terminal series-parallel graph with a source  $s_u^X$  and a sink  $t_u^X$ . For each pair  $u \in E(G')$  and  $e \in E_{G'}(u)$ , we prepare an additional two-terminal series-parallel graph  $G_{u,e}^Y$  consisting of a path  $\langle s_{u,e}^Y, y_{u,e}, t_{u,e}^Y \rangle$ , where  $s_{u,e}^Y$  and  $t_{u,e}^Y$  are a source and a sink of  $G_{u,e}^Y$ , respectively. Moreover, for each  $e \in E(G')$ , we construct a two-terminal series-parallel graph  $G_e^Z$  consisting of a path  $\langle s_e^Z, z_e, z'_e, t_e^Z \rangle$ , where  $s_e^Z$  and  $t_e^Z$  are a source and a sink of  $G_e^Z$ , respectively. We denote by  $G^X$  a graph obtained by parallel compositions of the all gadgets  $G_u^X$  for  $u \in V(G')$ . The graph  $G^Y$  and  $G^Z$  are similarly defined as parallel compositions of the gadgets. We then let  $G = G^X \parallel G^Y \parallel G^Z$ . Observe that  $G$  is also a two-terminal series-parallel graph. Let  $s$  and  $t$  denote the source and the sink of  $G$ , respectively.

We next construct a collection  $\mathcal{L}_{\mathcal{H}}$  of cycles in  $G$  as follows:

- for  $u \in V(G')$ , let  $C_u^X = \langle s, x_u, t, x'_{u,g}, x_{u,g}, s \rangle$ , where  $g \in E_{G'}(u)$ ;
- for  $u \in V(G')$  with three incident edges  $e, f, g \in E_{G'}(u)$ , let

$$\begin{aligned} C_{u,e}^Y &= \langle s, x_{u,e}, x_u, t, y_{u,e}, s \rangle, \\ C_{u,f}^Y &= \langle s, x_u, x_{u,f}, t, y_{u,f}, s \rangle, \\ C_{u,g}^Y &= \langle s, x_{u,g}, x'_{u,g}, t, y_{u,g}, s \rangle; \end{aligned}$$

- for  $e = uv \in E(G')$ , let  $C_{u,e}^Z = \langle s, y_{u,e}, t, z'_e, z_e, s \rangle$  and  $C_{v,e}^Z = \langle s, y_{v,e}, t, z'_e, z_e, s \rangle$ .

Then, we define

$$\mathcal{L}_{\mathcal{H}} = \{C_u^X : u \in V(G')\} \cup \{C_{u,e}^Y : u \in V(G'), e \in E_{G'}(u)\} \cup \{C_{u,e}^Z, C_{v,e}^Z : e = uv \in E(G')\}$$

and  $k = k' + 2m'$ , where  $m' = |E(G')|$ . The construction of the instance  $(G, k, \mathcal{L}_{\mathcal{H}})$  for EDGE DISJOINT LIST  $C_5$ -PACKING takes in polynomial time. From a vertex set  $I$  of  $G'$ , we can construct a set

$$\mathcal{S} = \{C_u^X : u \in I\} \cup \{C_{u,e}^Z : u \in I, e \in E_{G'}(u)\} \cup \{C_{u,e}^Y : u \notin I, e \in E_{G'}(u)\}$$

of cycles in  $\mathcal{L}_{\mathcal{H}}$ . Analogous to Lemma 2, we can prove that  $I$  is an independent set of  $G'$  if and only if cycles in  $\mathcal{S}$  are edge-disjoint, which is omitted because, in fact, it is essentially the same as the proof of Lemma 2.

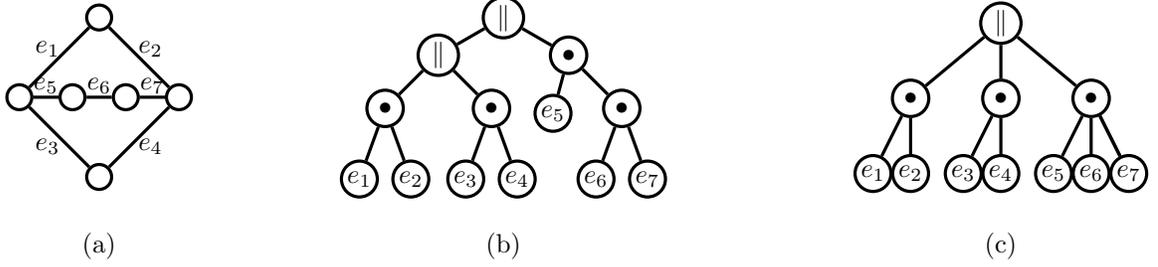


Figure 7: (a) The graph  $G$ , (b) the decomposition tree of  $G$ , and (c) the layered decomposition tree of  $G$ .

## 5.2 Polynomial-time algorithm of Edge Disjoint List $C_\ell$ -packing for $\ell \leq 4$

We design a polynomial-time algorithm for EDGE DISJOINT LIST  $C_\ell$ -PACKING for  $\ell \leq 4$  on two-terminal series-parallel graphs. Actually, we give a stronger theorem.

**Theorem 15.** *Let  $C_{\leq 4} = \{C_3, C_4\}$ . Given a series-parallel graph  $G$  with  $n$  vertices and a collection  $\mathcal{L}_{\mathcal{H}}$  of cycles in  $G$  of length at most 4, EDGE DISJOINT LIST  $C_{\leq 4}$ -PACKING is solvable in  $O(|\mathcal{L}_{\mathcal{H}}| + n^{2.5})$  time.*

We first note that we may assume that a given graph  $G$  is biconnected: the problem can be solved independently in each biconnected component. Moreover, from the definition of series-parallel graphs, every biconnected series-parallel graph can be regarded as a two-terminal series-parallel graph. We thus consider a polynomial-time algorithm that finds a largest solution of a given two-terminal series-parallel graph.

The recursive definition of a two-terminal series-parallel graph  $G$  naturally gives us a rooted full binary tree  $T$  representing  $G$ , called the *decomposition tree* of  $G$  (see Figure 7(a) and (b)). To avoid confusion, we refer to a vertex and an edge of  $T$  as a *node* and a *link*, respectively. For a node  $x$  of  $T$ , let  $T_x$  be a subtree of  $T$  rooted at  $x$ . Each leaf of  $T$  corresponds to an edge of  $G$  whose endpoints are labeled with a source  $s$  and a sink  $t$ . Each internal node  $x$  of  $T$  is labeled either  $\bullet$  or  $\parallel$ . Suppose that  $x$  has exactly two children  $x_1$  and  $x_2$ . The label  $\bullet$  indicates a series composition of two-terminal series-parallel graphs defined by  $T_{x_1}$  and  $T_{x_2}$ . The label  $\parallel$  indicates a parallel composition of two-terminal series-parallel graphs defined by  $T_{x_1}$  and  $T_{x_2}$ . We refer to nodes labeled  $\bullet$  as  $\bullet$ -nodes and to nodes labeled  $\parallel$  as  $\parallel$ -nodes. We denote by  $G_x$  the graph composed by  $T_x$ . Let  $r$  be the root of  $T$ . Then, we have  $G_r = G$ . Note that, since  $G_1 \bullet G_2$  and  $G_2 \bullet G_1$  produce different two-terminal graphs, we assume that children of a  $\bullet$ -node are ordered. In addition, since we have assumed  $G$  is 2-connected, the root  $r$  of  $T$  is labeled  $\parallel$  (assuming  $G$  has at least three vertices).

At the beginning of our algorithm, we construct a decomposition tree  $T'$  of a given graph  $G$  in linear time [36], and then transform it into a suitable form for our algorithm as follows (see also Figure 7(c)). If a  $\bullet$ -node  $x$  of  $T'$  has a child  $\bullet$ -node  $x'$ , then we contract a link  $xx'$  without changing the order of series compositions. For example, suppose that  $x$  has children  $x_1$  and  $x'$ ;  $x'$  has children  $x_2$  and  $x_3$ ; and  $G_x = G_{x_1} \bullet G_{x'}$ . Then, we contract the link  $xx'$  so that  $G_x = G_1 \bullet G_2 \bullet G_3$ . The contracted tree still tells how to construct  $G$ . Similarly, if a  $\parallel$ -node  $x$  of  $T'$  has a child  $\parallel$ -node  $x'$ , then we contract the link  $xx'$ . We iteratively contract such links until each  $\bullet$ -node has only leaves or  $\parallel$ -nodes as its children, and  $\parallel$ -node has only leaves or  $\bullet$ -nodes as its children. Note that each  $\parallel$ -node has at most one leaf of  $T'$  as its children because  $G$  has no multiple edges. The tree obtained in this way is called a *layered decomposition tree* of  $G$  and is denoted by  $T$ .

Let  $C$  be a cycle of a graph  $G = (V, E)$ . For a subgraph  $G' = (V', E')$  of  $G$ , we say that  $C$  *enters*  $G'$  if  $C$  has both an edge in  $E'$  and an edge in  $E \setminus E'$ . Suppose that there is a  $\bullet$ -node  $x$  of  $T$  such that  $x$  has  $c \geq 4$  children  $x_1, x_2, \dots, x_c$ . Then, no cycle of length at most 4 enters  $G_x$ ; since  $G_x$  is created by series compositions of at least four two-terminal labeled graphs, every cycle entering  $G_x$  has length at least 5. Thus, the problem can be solved independently in each of  $G_x$  and the remaining part.

Assume that each  $\bullet$ -node  $x$  of  $T$  has at most three children. Before explaining dynamic programming over  $T$ , we give the following key lemma.

**Lemma 3.** *Let  $x$  be any node of a layered decomposition tree  $T$  of a two-terminal series-parallel graph  $G$  and  $\mathcal{L}_{\mathcal{H}}$  be a collection of cycles in  $G$  of length at most 4. For any solution  $\mathcal{S} \subseteq \mathcal{L}_{\mathcal{H}}$  of  $G$ , there exists at most one cycle in  $\mathcal{S}$  that enters  $G_x$ .*

*Proof.* If  $x$  is a leaf or the root of  $T$ , the lemma trivially holds. Suppose otherwise, and assume for a contradiction that there exist two edge-disjoint cycles  $C, C' \in \mathcal{S}$  that enter  $G_x$ .

Suppose that  $x$  is a  $\bullet$ -node. Recall that  $x$  has two or three children. Suppose that  $x$  has exactly two children  $y$  and  $z$ , that is,  $G_x = G_y \bullet G_z$ . We denote by  $s_y$  and  $t_y$  the source and the sink of  $G_y$ , respectively, and by  $s_z$  and  $t_z$  analogously. Observe that each of  $C$  and  $C'$  contains at least one edge from  $G_y$  and at least one edge from  $G_z$ . Moreover, they have at least one edge outside of  $G_x$ , implying that they have at most three edges in  $G_x$ . As  $C$  contains at most three edges in  $G_x$ , without loss of generality, we assume that  $C$  contains exactly one edge (i.e.,  $s_y t_y$ ) of  $G_y$ . As  $G$  has no parallel edges and the cycles are edge-disjoint,  $C'$  contains exactly two edges of  $G_y$ . Similarly,  $C$  contains exactly two edges of  $G_z$  and  $C'$  contains exactly one edge of  $G_z$ . Thus, both cycles contain exactly three edges of  $G_x$ . This implies that  $C$  and  $C'$  contain the edge  $s_y t_z$  as they have at most four edges, contradicting the fact that  $G$  has no parallel edges. We can derive a similar contradiction for the case where  $x$  has three children.

Next, suppose that  $x$  is a  $\parallel$ -node. Let  $x'$  be the parent of  $x$ . From the definition of a layered decomposition tree,  $x'$  is a  $\bullet$ -node. As  $C$  enters  $G_x$ ,  $C$  passes through (1) an edge of  $G_y$ , (2) one of the source or sink of  $G_x$ , and (3) an edge of  $G_x$ , where  $y$  is a child node of  $x'$  with  $y \neq x$ . This implies that  $C$  also enters  $G_{x'}$  as otherwise  $C$  passes through the vertex of (2) twice, contradicting the fact that  $C$  is a cycle. Also,  $C'$  enters  $G_{x'}$ , which leads to a contradiction as mentioned above.  $\square$

Let  $\mathcal{S}$  be a largest solution of  $G$ . Suppose that  $x$  is a  $\bullet$ -node of  $T$  with  $c$  children  $x_1, x_2, \dots, x_c$ . For an integer  $i \in [c]$ , let  $s_i$  and  $t_i$  denote a source and a sink of  $G_{x_i}$ , respectively. We distinguish the following two cases to consider:

- ( $s_1$ ) at least one cycle  $C$  in  $\mathcal{S}$  enters  $G_x$  and  $s_i t_i \in E(C)$  for every integer  $i \in [c]$ ;
- ( $s_2$ ) at least one cycle  $C$  in  $\mathcal{S}$  enters  $G_x$  and  $s_i t_i \notin E(C)$  for some integer  $i \in [c]$ .

Similarly, for a  $\parallel$ -node  $x$  with a source  $s$  and a sink  $t$ , we also distinguish the following two cases to consider:

- ( $p_1$ ) at least one cycle  $C$  in  $\mathcal{S}$  enters  $G_x$  and  $st \in E(C)$ ;
- ( $p_2$ ) at least one cycle  $C$  in  $\mathcal{S}$  enters  $G_x$  and  $st \notin E(C)$ ;

We note that the above cases are not exhaustive: there may be no cycle in  $\mathcal{S}$  entering  $G_x$ . It is not necessary to consider such a case in the construction of our algorithm. We also note that by Lemma 3, there are no more than one (edge-disjoint) cycle satisfying these conditions.

Let  $\mathcal{L}_{\mathcal{H}}^x$  be a restriction of  $\mathcal{L}_{\mathcal{H}}$  to  $G_x$ , that is,  $\mathcal{L}_{\mathcal{H}}^x = \{H \in \mathcal{L}_{\mathcal{H}} : E(H) \subseteq E(G_x)\}$ . In our algorithm, for each node  $x$  of  $T$ , we compute the largest size of a subcollection  $\mathcal{S}_x$  with  $\mathcal{S}_x = \mathcal{S} \cap \mathcal{L}_{\mathcal{H}}^x$ . Let  $f^\bullet(x)$  be the largest size of  $\mathcal{S}_x$  for a  $\bullet$ -node  $x$ , and let  $f^\parallel(x)$  be the largest size of  $\mathcal{S}_x$  for a  $\parallel$ -node  $x$ . Notice that, originally, leaves of  $T$  are labeled neither  $\bullet$  nor  $\parallel$ , and hence  $f^\bullet(x)$  and  $f^\parallel(x)$  cannot be defined for the leaves. For algorithmic simplicity, we consider a leaf  $x$  as a  $\bullet$ -node if its parent is labeled  $\parallel$ , and as a  $\parallel$ -node if its parent is labeled  $\bullet$ . This simplification allows us to define  $f^\bullet(x)$  and  $f^\parallel(x)$  for a leaf  $x$  accordingly.

We also define the truth values  $b_j^\bullet(x)$  and  $b_j^\parallel(x)$  for each  $j \in \{1, 2\}$  and each node  $x$  of  $T$ . We set  $b_j^\bullet(x) = 1$  (resp.  $b_j^\parallel(x) = 1$ ) if and only if there exists a mutually edge-disjoint subcollection  $\mathcal{S}'_x$  of  $\mathcal{L}_{\mathcal{H}}^x$  that satisfies the following conditions:

- $|\mathcal{S}'_x| = f^\bullet(x)$  (resp.  $|\mathcal{S}'_x| = f^\parallel(x)$ );
- there exists a cycle  $C \in \mathcal{L}_{\mathcal{H}} \setminus \mathcal{L}_{\mathcal{H}}^x$  corresponding to the case ( $s_j$ ) (resp. ( $p_j$ ));
- all subgraphs in  $\mathcal{S}'_x$  and  $C$  are edge-disjoint.

Intuitively speaking,  $b_j^\bullet(x) = 1$  (and  $b_j^\parallel(x) = 1$ ) if and only if we can further add a cycle  $C$  entering  $G_x$  into a partial solution at an ancestor of  $x$ .

We are ready to explain how to compute  $f^\bullet(x)$ ,  $f^\parallel(x)$ ,  $b_j^\bullet(x)$ , and  $b_j^\parallel(x)$  for each node  $x$  of  $T$  and each  $j \in \{1, 2\}$ .

**Leaf node.** Suppose that  $x$  is a leaf of  $T$ . Let  $s$  and  $t$  be the source and the sink of  $G_x$ , respectively. One can verify that the following equalities hold:  $f^\bullet(x) = f^\parallel(x) = 0$ ;  $b_1^\bullet(x) = b_1^\parallel(x) = 1$  if and only if there exists a cycle  $C$  in  $\mathcal{L}_\mathcal{H}$  such that  $st \in E(C)$ ; and  $b_2^\bullet(x) = b_2^\parallel(x) = 0$ .

**Internal  $\bullet$ -node.** Suppose that  $x$  is a  $\bullet$ -node with  $c$  children  $x_1, \dots, x_c$ . Since  $G_x$  consists of series compositions of  $G_{x_1}, \dots, G_{x_c}$ , every cycle in  $G_x$  is contained in  $G_{x_i}$  for some  $i$ . We thus have

$$f^\bullet(x) = \sum_{i \in [c]} f^\parallel(x_i).$$

We next compute  $b_j^\bullet(x)$  for each  $j \in \{1, 2\}$ . Recall that  $x$  has at most three children.

Suppose that  $c = 3$ . If there exists a cycle  $C \in \mathcal{L}_\mathcal{H} \setminus \mathcal{L}_\mathcal{H}^x$  that enters  $G_x$ , then it passes through  $s_1$  and  $t_3$ , meaning that it enters  $G_{x_i}$  for all  $i \in [3]$ . Conversely, for every  $i \in [3]$ , if there is a cycle  $C_i \in \mathcal{L}_\mathcal{H} \setminus \mathcal{L}_\mathcal{H}^{x_i}$  such that  $C_i$  enters  $G_{x_i}$ , then it must have  $E(C_i) = \{s_1 t_1, s_2 t_2, s_3 t_3, t_3 s_1\}$ , that is, the cycle is uniquely determined  $C = C_i$  for  $i \in [3]$ . It is easy to observe that  $C$  is edge-disjoint from any cycles in  $\mathcal{S}_x$  if and only if it is edge-disjoint from any cycles in  $\mathcal{S}_{x_i}$  for all  $i \in [3]$ . Hence, we have  $b_1^\bullet(x) = b_1^\parallel(x_1) \wedge b_1^\parallel(x_2) \wedge b_1^\parallel(x_3)$ . This also implies that there is no cycle  $C \in \mathcal{S}$  that enters  $G_x$  and  $s_i t_i \notin E(C)$ , which yields that  $b_2^\bullet(x) = 0$ .

Suppose next that  $c = 2$ . By the similar argument to the case  $c = 3$ , we have  $b_1^\bullet(x) = 1$  if and only if  $b_1^\parallel(x_1) \wedge b_1^\parallel(x_2) = 1$  and there is a cycle  $C \in \mathcal{L}_\mathcal{H} \setminus \mathcal{L}_\mathcal{H}^x$  such that  $s_1 t_1, s_2 t_2 \in E(C)$ . We explain how to decide  $b_2^\bullet(x)$ . If there is a cycle  $C \in \mathcal{L}_\mathcal{H} \setminus \mathcal{L}_\mathcal{H}^x$  with  $s_1 t_1 \notin E(C)$  that enters  $G_x$ , then it enters both  $G_{x_1}$  and  $G_{x_2}$ , and it holds that  $s_2 t_2 \in E(C)$  because the length of  $C$  is at most 4. Conversely, for a cycle  $C_1 \in \mathcal{L}_\mathcal{H} \setminus \mathcal{L}_\mathcal{H}^{x_1}$  such that  $C_1$  enters  $G_{x_1}$  and  $s_1 t_1 \notin E(C_1)$ ,  $C_1$  also enters  $G_{x_2}$  and  $G_x$ , and  $s_2 t_2 \in E(C_1)$  holds. The same argument is applied to a cycle  $C \in \mathcal{L}_\mathcal{H} \setminus \mathcal{L}_\mathcal{H}^x$  with  $s_2 t_2 \notin E(C)$  that enters  $G_x$ . Thus, we have  $b_2^\bullet(x) = (b_2^\parallel(x_1) \wedge b_1^\parallel(x_2)) \vee (b_1^\parallel(x_1) \wedge b_2^\parallel(x_2))$ .

**Internal  $\parallel$ -node.** Suppose that  $x$  is a  $\parallel$ -node with  $c$  children  $x_1, \dots, x_c$ . Let  $s$  and  $t$  be the source and the sink of  $G_x$ , respectively.

To compute  $f^\parallel(x)$ , we construct an *auxiliary graph*  $A_x$  whose vertex set is  $\{a_1, a_2, \dots, a_c\}$ . We associate each child  $x_i$  of  $x$  with a vertex  $a_i$ . Let  $i, j \in [c]$  be distinct integers. Suppose that  $x_i$  and  $x_j$  are internal nodes of  $T$ . Then,  $A_x$  has an edge  $a_i a_j$  if  $b_1^\bullet(x_i) \wedge b_1^\bullet(x_j) = 1$  and there exists a cycle in  $\mathcal{L}_\mathcal{H}^x$  that enters both  $G_{x_i}$  and  $G_{x_j}$ . Note that such a cycle  $C$  satisfies  $|E(C) \cap E(G_{x_i})| = |E(C) \cap E(G_{x_j})| = 2$ , which means that  $C$  must satisfy the case  $(s_1)$  for  $\bullet$ -nodes  $x_i$  and  $x_j$ . Suppose next that  $x_i$  is an internal node and  $x_j$  is a leaf of  $T$ . In this case,  $st \in E(G_{x_j})$ . Then,  $A_x$  has an edge  $a_i a_j$  if at least one of the following conditions is satisfied:

1.  $x_i$  has exactly  $c$  children with  $c \in \{2, 3\}$ ,  $b_1^\bullet(x_i) = 1$ , and there exists a cycle  $C$  in  $\mathcal{L}_\mathcal{H}^x$  of length  $c + 1$  that enters both  $G_{x_i}$  and  $G_{x_j}$ ; or
2.  $b_2^\bullet(x_i) = 1$ .

Note that, in the second case  $b_2^\bullet(x_i) = 1$ , there is a cycle  $C \in \mathcal{L}_\mathcal{H} \setminus \mathcal{L}_\mathcal{H}^{x_i}$  entering  $G_{x_i}$  such that  $x_i$  has a child  $y$  with  $|E(C) \cap E(G_y)| \geq 2$ . This implies that  $C$  has exactly three edges in  $G_{x_i}$  and hence we have  $st \in E(C)$ . Also note that there is no case that both  $x_i$  and  $x_j$  are leaves because  $G$  has no parallel edges. We complete the construction of  $A_x$ .

The intuition of the auxiliary graph  $A_x$  is as follows. If there is an edge  $a_i a_j \in A_x$ , then we can further add a cycle  $C$  in  $G_{x_i} \parallel G_{x_j}$  that is edge-disjoint from any cycles in  $\bigcup_{h \in [c]} \mathcal{S}_{x_h}$ . We can simultaneously add such cycles for other edges in  $A_x$ . However, by Lemma 3, we cannot add more than one cycles entering  $G_{x_i}$ .

Thus, in order to add as many such cycles as possible, the corresponding edges must form a matching in  $A_x$ . In fact, the following equality holds.

$$f^{\parallel}(x) = \sum_{i \in [c]} f^{\bullet}(x_i) + |M_x^*|, \quad (1)$$

where  $M_x^*$  be a maximum matching of  $A_x$ . The correctness of Equation (1) will be given in Section 5.2.1.

To compute  $b_j^{\parallel}(x)$  for each  $j \in \{1, 2\}$ , we construct additional auxiliary graphs  $A_x^1$  and  $A_x^2$  from  $A_x$ . Let  $A_x^1$  be the graph obtained from  $A_x$  as follows. We first add a vertex  $a'$ . Then, we add an edge  $a'a_i$  if  $x$  has a leaf child  $x_i$  and there is a cycle  $C \in \mathcal{L}_{\mathcal{H}} \setminus \mathcal{L}_{\mathcal{H}}^x$  such that  $C$  enters  $G_x$  and  $st \in E(C)$ .

Similarly, let  $A_x^2$  be the graph obtained from  $A_x$  as follows. We first add a vertex  $a''$ . Then, for each  $i \in [c]$ , we add an edge  $a''a_i$  if  $x_i$  is an internal node of  $T$ ,  $b_1^{\bullet}(x_i) = 1$  and there is a cycle  $C \in \mathcal{L}_{\mathcal{H}} \setminus \mathcal{L}_{\mathcal{H}}^x$  that enters  $G_{x_i}$ .

Let  $M_x^1$  and  $M_x^2$  be maximum matchings of  $A_x^1$  and  $A_x^2$ , respectively. We let  $b_1^{\parallel}(x) = 1$  if and only if  $|M_x^1| > |M_x^*|$ ; and  $b_2^{\parallel}(x) = 1$  if and only if  $|M_x^2| > |M_x^*|$ . The correctness of the computation of these truth values will be given in Section 5.2.1.

Finally, we conclude that  $f^{\parallel}(r)$  is the size of a largest solution of  $G$ .

### 5.2.1 Correctness

**Lemma 4.** *Equation (1) is correct.*

*Proof.* We first show that  $f^{\parallel}(x) \geq \sum_{i \in [c]} f^{\bullet}(x_i) + |M_x^*|$ . For  $i \in [c]$ , let  $\mathcal{S}_{x_i}$  be a mutually edge-disjoint subcollection of  $\mathcal{L}_{\mathcal{H}}^{x_i}$  such that  $|\mathcal{S}_{x_i}| = f^{\bullet}(x_i)$ . Recall that each edge  $a_i a_j$  in  $M_x^*$  corresponds to a cycle in  $\mathcal{L}_{\mathcal{H}}^x$  that enters both  $G_{x_i}$  and  $G_{x_j}$  and edge-disjoint from any cycle in  $\mathcal{S}_{x_i} \cup \mathcal{S}_{x_j}$ . Let  $\mathcal{M}_x$  be a subcollection of cycles in  $\mathcal{L}_{\mathcal{H}}^x$  corresponding to  $M_x^*$ . Observe that cycles in  $\mathcal{M}_x$  are mutually edge-disjoint because  $M_x^*$  is a matching of  $A_x$ . Therefore,  $(\bigcup_{i \in [c]} \mathcal{S}_{x_i}) \cup \mathcal{M}_x$  is mutually edge-disjoint, and hence we have  $f^{\parallel}(x) \geq \sum_{i \in [c]} f^{\bullet}(x_i) + |M_x^*|$ .

We next show that  $f^{\parallel}(x) \leq \sum_{i \in [c]} f^{\bullet}(x_i) + |M_x^*|$ . Suppose that  $\mathcal{S}_x$  is a mutually edge-disjoint subcollection of  $\mathcal{L}_{\mathcal{H}}^x$  such that  $|\mathcal{S}_x| = f^{\parallel}(x)$ . Consider a cycle  $C$  in  $\mathcal{S}_x$  entering both  $G_{x_i}$  and  $G_{x_j}$  such that  $x_i$  and  $x_j$  are internal nodes of  $T$ . If  $a_i a_j \notin E(A_x)$ , then it holds that  $b_1^{\bullet}(x_i) \wedge b_1^{\bullet}(x_j) = 0$ . Without loss of generality, assume that  $b_1^{\bullet}(x_i) = 0$ . Let  $\mathcal{S}_{x_i} = \mathcal{S}_x \cap \mathcal{L}_{\mathcal{H}}^{x_i}$ . Observe that  $|\mathcal{S}_{x_i}| - 1 \leq f^{\bullet}(x_i)$  as otherwise, the fact that the cycle  $C$  is edge-disjoint from any cycles in  $\mathcal{S}_{x_i}$  implies that  $b_1^{\bullet}(x_i) = 1$ . Then, we can replace  $\mathcal{S}_{x_i} \cup \{C\}$  with an edge-disjoint subcollection  $\mathcal{S}'_{x_i}$  with  $|\mathcal{S}'_{x_i}| = f^{\bullet}(x_i)$  in  $\mathcal{S}_x$  (i.e.,  $\mathcal{S}_x := (\mathcal{S}_x \setminus (\mathcal{S}_{x_i} \cup \{C\})) \cup \mathcal{S}'_{x_i}$ ). The same argument is applicable to the case where one of  $x_i$  and  $x_j$  is a leaf of  $T$ . Applying the above replacement exhaustively, we eventually obtain a mutually edge-disjoint subcollection  $\mathcal{S}_x^* \subseteq \mathcal{L}_{\mathcal{H}}^x$  with  $|\mathcal{S}_x^*| = f^{\parallel}(x)$  that satisfied the following condition: For each cycle  $C$  in  $\mathcal{S}_x^*$  that enters both  $G_{x_i}$  and  $G_{x_j}$  for distinct  $i, j \in [c]$ ,  $A_x$  has an edge  $a_i a_j$ . We denote by  $\mathcal{C} \subseteq \mathcal{S}_x^*$  the set of cycles, each of which enters  $G_{x_i}$  and  $G_{x_j}$  for distinct  $i, j \in [c]$ . Let  $M$  be the subset of  $E(A_x)$  corresponding to  $\mathcal{C}$ . Then,  $M$  forms a matching of  $A_x$ ; otherwise, there are two edge-disjoint cycles that enter  $G_{x_i}$  for some  $i \in [c]$ , which contradicts Lemma 3. Denote  $\mathcal{S}_{x_i}^* = \mathcal{S}_x^* \cap \mathcal{L}_{\mathcal{H}}^{x_i}$  for each  $i \in [c]$ . Obviously, it holds that  $|\mathcal{S}_{x_i}^*| \leq f^{\bullet}(x_i)$ . Since  $\mathcal{S}_x^* = (\bigcup_{i \in [c]} \mathcal{S}_{x_i}^*) \cup \mathcal{C}$ , we thus have  $f^{\parallel}(x) \leq \sum_{i \in [c]} f^{\bullet}(x_i) + |M| \leq \sum_{i \in [c]} f^{\bullet}(x_i) + |M_x^*|$  as  $M_x^*$  is a maximum matching of  $A_x$ . This completes the proof of Lemma 4.  $\square$

**Lemma 5.** *Let  $M_x^1$  and  $M_x^2$  denote maximum matchings of  $A_x^1$  and  $A_x^2$ , respectively. Then,  $b_1^{\parallel}(x) = 1$  if and only if  $|M_x^1| > |M_x^*|$ ; and  $b_2^{\parallel}(x) = 1$  if and only if  $|M_x^2| > |M_x^*|$ .*

*Proof.* We prove the former claim of the lemma. Suppose that  $b_1^{\parallel}(x) = 1$ . From the definition of  $b_1^{\parallel}(x)$ , there exists a mutually edge-disjoint subcollection  $\mathcal{S}'_x$  of  $\mathcal{L}_{\mathcal{H}}^x$  with  $|\mathcal{S}'_x| = f^{\parallel}(x)$  and a cycle  $C \in \mathcal{L}_{\mathcal{H}} \setminus \mathcal{L}_{\mathcal{H}}^x$  with  $st \in E(C)$  that is edge-disjoint from any cycle in  $\mathcal{S}'_x$ . Then, we can construct a matching  $M$  of  $A_x$  corresponding to a maximum subcollection  $\mathcal{C} \subseteq \mathcal{S}'_x$  such that each cycle in  $\mathcal{C}$  enters  $G_{x_i}$  and  $G_{x_j}$  for some  $i, j \in [c]$ . By Lemma 3, such a subcollection  $\mathcal{C}$  is uniquely determined. As  $M_x^*$  is a maximum matching of

$A_x$ , we have  $|M| \leq |M_x^*|$ . We now claim that  $|M| = |M_x^*|$ . Suppose to the contrary that  $|M| < |M_x^*|$ . Denote  $\mathcal{S}'_{x_i} = \mathcal{S}'_x \cap \mathcal{L}_{\mathcal{H}}^{x_i}$  for each  $i \in [c]$ . Since  $\mathcal{S}'_x = (\bigcup_{i \in [c]} \mathcal{S}'_{x_i}) \cup \mathcal{C}$ , we have  $|\mathcal{S}'_x| = (\sum_{i \in [c]} |\mathcal{S}'_{x_i}|) + |M|$ . Combined with the facts  $f^{\parallel}(x) = \sum_{i \in [c]} f^{\bullet}(x_i) + |M_x^*|$ ,  $|M| < |M_x^*|$ , and  $|\mathcal{S}'_x| = f^{\parallel}(x)$ , we can observe that there exists  $p \in [c]$  such that  $|\mathcal{S}'_{x_p}| > f^{\bullet}(x_p)$ , which contradicts the definition of  $f^{\bullet}(x_p)$ . Thus, we have  $|M| = |M_x^*|$ .

Let  $x_i$  be the child of  $x$  corresponding to the edge  $st$ . Since  $st \in E(C)$  and  $C$  is edge-disjoint from any cycles in  $\mathcal{S}'_x$ , every edge in  $M$  is not incident to  $a_i$ . Thus,  $M \cup \{a'a_i\}$  forms a matching of  $A_x^1$ . As  $|M| = |M_x^*|$ , we have  $|M_x^1| \geq |M \cup \{a'a_p\}| > |M| = |M_x^*|$ .

Conversely, suppose that  $|M_x^1| > |M_x^*|$ . This implies that there is an edge  $e \in M_x^1$  incident to  $a'$ . Let  $C$  be the cycle in  $\mathcal{L}_{\mathcal{H}} \setminus \mathcal{L}_{\mathcal{H}}^x$  corresponding to  $e$ . As in the proof of Lemma 4, we can construct a mutually edge-disjoint subcollection  $\mathcal{S}'_x$  of  $\mathcal{L}_{\mathcal{H}}^x$  such that  $|\mathcal{S}'_x| = \sum_{i \in [c]} f^{\bullet}(x_i) + |M_x^1 \setminus \{e\}|$ . Since  $|M_x^1 \setminus \{e\}| = |M_x^1| - 1 \geq |M_x^*|$ , we have  $|\mathcal{S}'_x| \geq \sum_{i \in [c]} f^{\bullet}(x_i) + |M_x^*| = f^{\parallel}(x)$ . It clearly holds that  $|\mathcal{S}'_x| \leq f^{\parallel}(x)$ , and hence we have  $|\mathcal{S}'_x| = f^{\parallel}(x)$ .

Moreover, every subgraph in  $\mathcal{S}'_x$  and  $C$  are edge-disjoint. Therefore, we have  $b_1^{\parallel}(x) = 1$ .

The latter claim can be proved in the similar way, which completes the proof of Lemma 5.  $\square$

### 5.2.2 Running time

Let  $n$  and  $m$  be the number of vertices and edges of a given series-parallel graph  $G$ , respectively, and let  $\mathcal{L}_{\mathcal{H}}$  be a collection of cycles in  $G$  of length at most 4. As noted at the beginning of Section 5.2, our algorithm is applied to each of  $\beta^*$  biconnected components  $G_1, G_2, \dots, G_{\beta^*}$  of  $G$ . We can enumerate biconnected components of  $G$  in  $O(n+m)$  time [20] and partition  $\mathcal{L}_{\mathcal{H}}$  into  $\mathcal{L}_{\mathcal{H}}^1, \mathcal{L}_{\mathcal{H}}^2, \dots, \mathcal{L}_{\mathcal{H}}^{\beta^*}$  so that  $\mathcal{L}_{\mathcal{H}}^{\beta}$  is the subcollection of  $\mathcal{L}_{\mathcal{H}}$  consisting of all cycles in  $G_{\beta}$  for each  $\beta \in [\beta^*]$  in  $O(|\mathcal{L}_{\mathcal{H}}|)$  time.

For  $\beta \in [\beta^*]$ , we denote by  $n_{\beta}$  and  $m_{\beta}$  the number of vertices and edges of  $G_{\beta}$ , respectively. We may assume that edges of  $G_{\beta}$  are labeled distinct integers  $1, 2, \dots, m_{\beta}$ . Each cycle  $C$  of  $G_{\beta}$  with  $h$  edges produces a word  $x_1 x_2 \dots x_h$  such that  $x_i \in [m_{\beta}]$  for every  $i \in [h]$  and  $x_1 < x_2 < \dots < x_h$ . As a preprocessing, we convert every cycle in  $\mathcal{L}_{\mathcal{H}}^{\beta}$  to a corresponding word, and then lexicographically sort  $\mathcal{L}_{\mathcal{H}}^{\beta}$  with radix sort. Since every cycle in  $\mathcal{L}_{\mathcal{H}}^{\beta}$  is of length at most 4, this can be done in  $O(|\mathcal{L}_{\mathcal{H}}^{\beta}| + m_{\beta})$  time. Using this data structure, we can check whether given a cycle  $C$  is in  $\mathcal{L}_{\mathcal{H}}$  in time  $O(\log |\mathcal{L}_{\mathcal{H}}^{\beta}|)$  with binary search.

After the preprocessing, we construct an (original) decomposition tree of  $G_{\beta}$  in  $O(n_{\beta})$  time [36]. It is not hard to see that the decomposition tree can be modified into a layered decomposition tree  $T_{\beta}$  in  $O(n_{\beta})$  time with depth-first search. Moreover, we record the source and the sink of  $G_x$  in each node  $x$  of  $T_{\beta}$ .

We bound the running time of our dynamic programming. Obviously, for each leaf  $x$  of  $T_{\beta}$ , we can compute  $f^{\bullet}(x)$ ,  $f^{\parallel}(x)$ ,  $b_j^{\bullet}(x)$ , and  $b_j^{\parallel}(x)$  for each  $j \in \{1, 2\}$  in  $O(1)$  time. Moreover,  $f^{\bullet}(x)$ ,  $b_1^{\bullet}(x)$ , and  $b_2^{\bullet}(x)$  are computed in  $O(1)$  time for each  $\bullet$ -node  $x$  of  $T_{\beta}$ .

Consider the case where  $x$  is an internal  $\parallel$ -node of  $T_{\beta}$ . Let  $c_x$  denote the number of children of  $x$ . Our algorithm first constructs the auxiliary graph  $A_x$  with the vertex set  $\{a_1, \dots, a_{c_x}\}$ . Let  $i, j \in [c_x]$  be distinct integers. Suppose that  $x_i$  and  $x_j$  are internal nodes of  $T_{\beta}$ . We check whether  $b_1^{\bullet}(x_i) \wedge b_1^{\bullet}(x_j) = 1$  in  $O(1)$  time. Moreover, we check whether there exists a cycle  $C$  that enters both  $G_{x_i}$  and  $G_{x_j}$ . In fact, the cycle  $C$  is uniquely determined if it exists: each of  $G_{x_i}$  and  $G_{x_j}$  is obtained by a series composition of two graphs, meaning that  $|E(C) \cap E(G_{x_i})| = |E(C) \cap E(G_{x_j})| = 2$ . We can compute such a cycle  $C$  in  $O(1)$  time by recording the source and the sink of a graph corresponding to each node, while we can decide whether  $C \in \mathcal{L}_{\mathcal{H}}^{\beta}$  in  $O(\log |\mathcal{L}_{\mathcal{H}}^{\beta}|)$  time using the above data structure. As  $|\mathcal{L}_{\mathcal{H}}^{\beta}|$  is bounded by  $n_{\beta}^4$  above, this can be done in  $O(\log n_{\beta})$  time, which also decides whether  $a_i a_j \in E(A_x)$  or not. Similarly, for the case where  $x_i$  is an internal node and  $x_j$  is a leaf of  $T_{\beta}$ , we can determine in  $O(\log n_{\beta})$  time whether  $a_i a_j \in E(A_x)$  or not. Therefore, the construction of  $A_x$  takes  $O(c_x^2 \log n_{\beta})$  time.

We then construct graphs  $A_x^1$  and  $A_x^2$  from  $A_x$  to compute  $b_1^{\parallel}(x)$  and  $b_2^{\parallel}(x)$ . For the graph  $A_x^1$ , we need to decide whether  $a'a_i \in E(A_x^1)$ , where  $x_i$  for  $i \in [c_x]$  is the unique leaf child of  $x$  (if it exists). To this end, we construct the collection  $\mathcal{C}_x^1$  of all cycles that enter  $G_x$  and contain the edge of  $G_{x_i}$ . If  $x$  is the root of  $T$ , then clearly  $\mathcal{C}_x^1 = \emptyset$ . Otherwise  $x$  has the parent  $y$  labeled  $\bullet$  and  $y$  has the parent  $z$  labeled  $\parallel$  as the root of  $T_{\beta}$  has label  $\parallel$ . If  $y$  has three children,  $\mathcal{C}_x^1$  can be obtained in  $O(1)$  time because a possible cycle contained in  $\mathcal{C}_x^1$  is uniquely determined. Suppose that  $y$  has exactly two children and let  $x'$  be a child of  $y$

with  $x' \neq x$ . For any cycle  $C \in \mathcal{C}_x^1$ , the following two cases are considered: (i)  $C$  shares exactly one edge with  $G_{x'}$ ; and (ii)  $C$  shares exactly two edges with  $G_{x'}$ . In the case (i), for each child  $c_z$  of  $z$ , at most two edges in  $G_{c_z}$  that can be shared with  $C$  are uniquely determined because  $c_z$  is a leaf of  $T_\beta$  or labeled  $\bullet$ . In the case (ii), for each child  $c_y$  of  $y$ , exactly two edges in  $G_{c_y}$  that can be shared with  $C$  are uniquely determined and  $C$  contains an edge between the source and the sink of  $G_z$ . In both cases,  $\mathcal{C}_x^1$  of size  $O(n_\beta)$  can be constructed in  $O(n_\beta)$  time. We thus decide in  $O(n_\beta \log n_\beta)$  time whether there is a cycle  $C \in \mathcal{C}_x^1 \cap \mathcal{L}_\mathcal{H}^\beta$ , that is,  $a'a_i \in E(A_x^1)$ . For the graph  $A_x^2$ , we decide whether  $a''a_i \in E(A_x^1)$  for each  $i \in [c_x]$  such that  $x_i$  is an internal node of  $T_\beta$ . We check whether  $b_1^\bullet(x_i) = 1$ , and if so, there exists a cycle  $C \in \mathcal{L}_\mathcal{H} \setminus \mathcal{L}_\mathcal{H}^x$  that enters  $G_{x_i}$ . Since  $G_{x_i}$  shares exactly two edges with  $C$ , the cycle is uniquely determined for each  $i \in [c_x]$  as in the case (ii) above. Thus,  $A_x^2$  is constructed in  $O(c_x \log n_\beta)$  time. After the construction of  $A_x$ ,  $A_x^1$ , and  $A_x^2$ , we obtain maximum matchings  $M_x^*$ ,  $M_x^1$ , and  $M_x^2$  in  $O(c_x^{2.5})$  time, respectively [29]. Therefore,  $f^\parallel(x)$  is computed in  $O(c_x^2 \log n_\beta + n_\beta \log n_\beta + c_x^{2.5})$  time for each  $\parallel$ -node  $x$  of  $T$ .

In summary,  $f^\parallel(r)$  is obtained in time

$$O(m_\beta + |\mathcal{L}_\mathcal{H}^\beta| + \sum_{x \in V(T)} (c_x^2 \log n_\beta + n_\beta \log n_\beta + c_x^{2.5})).$$

Recall that  $\sum_{x \in V(T)} c_x = O(n_\beta)$  and  $m_\beta = O(n_\beta)$  hold. Therefore, our algorithm for  $G_\beta$  runs in  $O(|\mathcal{L}_\mathcal{H}^\beta| + n_\beta^{2.5})$  time. Since  $\sum_{\beta \in [\beta^*]} n_\beta = n + b^* \leq 2n$ , we conclude that the total running time for a given series-parallel graph  $G$  is bounded by  $O(|\mathcal{L}_\mathcal{H}| + n^{2.5})$ . This completes the proof of Theorem 15.

## Acknowledgements

We thank the referees for their valuable comments and suggestions which greatly helped to improve the presentation of this paper.

## References

- [1] Jin Akiyama and Vasek Chvátal. “Packing paths perfectly”. In: *Discret. Math.* 85.3 (1990), pp. 247–255. DOI: [10.1016/0012-365X\(90\)90382-R](https://doi.org/10.1016/0012-365X(90)90382-R).
- [2] N. R. Aravind and Roopam Saxena. “Parameterized Complexity of Path Set Packing”. In: *the 17th International Conference and Workshops (WALCOM 2023)*. Ed. by Chun-Cheng Lin, Bertrand M. T. Lin, and Giuseppe Liotta. Cham: Springer Nature Switzerland, 2023, pp. 291–302. DOI: [10.1007/978-3-031-27051-2\\_25](https://doi.org/10.1007/978-3-031-27051-2_25).
- [3] Stefan Arnborg, Derek G. Corneil, and Andrzej Proskurowski. “Complexity of Finding Embeddings in a  $k$ -Tree”. In: *SIAM Journal on Algebraic Discrete Methods* 8.2 (1987), pp. 277–284. DOI: [10.1137/0608024](https://doi.org/10.1137/0608024).
- [4] V. Bafna and P.A. Pevzner. “Genome rearrangements and sorting by reversals”. In: *Proceedings of 1993 IEEE 34th Annual Foundations of Computer Science*. 1993, pp. 148–157. DOI: [10.1109/SFCS.1993.366872](https://doi.org/10.1109/SFCS.1993.366872).
- [5] Fran Berman, David Johnson, Tom Leighton, Peter W. Shor, and Larry Snyder. “Generalized planar matching”. In: *Journal of Algorithms* 11.2 (1990), pp. 153–184. DOI: [10.1016/0196-6774\(90\)90001-U](https://doi.org/10.1016/0196-6774(90)90001-U).
- [6] Hans L. Bodlaender. “A partial  $k$ -arboretum of graphs with bounded treewidth”. In: *Theoretical Computer Science* 209.1 (1998), pp. 1–45. DOI: [10.1016/S0304-3975\(97\)00228-4](https://doi.org/10.1016/S0304-3975(97)00228-4).
- [7] Hans L. Bodlaender. “On Disjoint Cycles”. In: *Int. J. Found. Comput. Sci.* 5.1 (1994), pp. 59–68. DOI: [10.1142/S0129054194000049](https://doi.org/10.1142/S0129054194000049).
- [8] Koen M. J. De Bontridder, Bjarni V. Halldórsson, Magnús M. Halldórsson, Cor A. J. Hurkens, Jan Karel Lenstra, R. Ravi, and Leen Stougie. “Approximation algorithms for the test cover problem”. In: *Math. Program.* 98.1-3 (2003), pp. 477–491. DOI: [10.1007/s10107-003-0414-6](https://doi.org/10.1007/s10107-003-0414-6).

- [9] Alberto Caprara and Romeo Rizzi. “Packing triangles in bounded degree graphs”. In: *Inf. Process. Lett.* 84.4 (2002), pp. 175–180. DOI: [10.1016/S0020-0190\(02\)00274-0](https://doi.org/10.1016/S0020-0190(02)00274-0).
- [10] F.R.K Chung and R.L. Graham. “Recent results in graph decompositions”. In: *London Mathematical Society, Lecture Note Series* 52 (1981), pp. 103–123.
- [11] Derek G. Corneil, Shigeru Masuyama, and S. Louis Hakimi. “Edge-disjoint packings of graphs”. In: *Discret. Appl. Math.* 50.2 (1994), pp. 135–148. DOI: [10.1016/0166-218X\(92\)00153-D](https://doi.org/10.1016/0166-218X(92)00153-D).
- [12] Marek Cygan, Fedor V. Fomin, Lukasz Kowalik, Daniel Lokshtanov, Dániel Marx, Marcin Pilipczuk, Michał Pilipczuk, and Saket Saurabh. *Parameterized Algorithms*. Springer, 2015. DOI: [10.1007/978-3-319-21275-3](https://doi.org/10.1007/978-3-319-21275-3).
- [13] M.E. Dyer and A.M. Frieze. “On the complexity of partitioning graphs into connected subgraphs”. In: *Discrete Applied Mathematics* 10.2 (1985), pp. 139–153. DOI: [10.1016/0166-218X\(85\)90008-3](https://doi.org/10.1016/0166-218X(85)90008-3).
- [14] P. Erdős and L. Pósa. “On Independent Circuits Contained in a Graph”. In: *Canadian Journal of Mathematics* 17 (1965), pp. 347–352. DOI: [10.4153/CJM-1965-035-8](https://doi.org/10.4153/CJM-1965-035-8).
- [15] Michael R. Fellows, Danny Hermelin, Frances A. Rosamond, and Stéphane Vialette. “On the parameterized complexity of multiple-interval graph problems”. In: *Theor. Comput. Sci.* 410.1 (2009), pp. 53–61. DOI: [10.1016/j.tcs.2008.09.065](https://doi.org/10.1016/j.tcs.2008.09.065).
- [16] Henning Fernau and Daniel Raible. “A parameterized perspective on packing paths of length two”. In: *J. Comb. Optim.* 18.4 (2009), pp. 319–341. DOI: [10.1007/s10878-009-9230-0](https://doi.org/10.1007/s10878-009-9230-0).
- [17] M.R. Garey, D.S. Johnson, and L. Stockmeyer. “Some simplified NP-complete graph problems”. In: *Theoretical Computer Science* 1.3 (1976), pp. 237–267. DOI: [10.1016/0304-3975\(76\)90059-1](https://doi.org/10.1016/0304-3975(76)90059-1).
- [18] Venkatesan Guruswami, C. Pandu Rangan, Maw-Shang Chang, Gerard J. Chang, and C. K. Wong. “The Vertex-Disjoint Triangles Problem”. In: *Graph-Theoretic Concepts in Computer Science, 24th International Workshop, WG '98*. Ed. by Juraj Hromkovic and Ondrej Sýkora. Vol. 1517. Lecture Notes in Computer Science. Springer, 1998, pp. 26–37. DOI: [10.1007/10692760\\_3](https://doi.org/10.1007/10692760_3).
- [19] Lenwood S. Heath and John Paul C. Vergara. “Edge-Packing in Planar Graphs”. In: *Theory Comput. Syst.* 31.6 (1998), pp. 629–662. DOI: [10.1007/s002240000107](https://doi.org/10.1007/s002240000107).
- [20] John Hopcroft and Robert Tarjan. “Algorithm 447: Efficient Algorithms for Graph Manipulation”. In: *Communications of the ACM* 16.6 (1973), pp. 372–378. DOI: [10.1145/362248.362272](https://doi.org/10.1145/362248.362272).
- [21] Minghui Jiang, Ge Xia, and Yong Zhang. “Edge-disjoint packing of stars and cycles”. In: *Theoretical Computer Science* 640 (2016), pp. 61–69. DOI: <https://doi.org/10.1016/j.tcs.2016.06.001>.
- [22] Atsushi Kaneko, Alexander Kelmans, and Tsuyoshi Nishimura. “On packing 3-vertex paths in a graph”. In: *Journal of Graph Theory* 36.4 (2001), pp. 175–197. DOI: [https://doi.org/10.1002/1097-0118\(200104\)36:4<175::AID-JGT1005>3.0.CO;2-T](https://doi.org/10.1002/1097-0118(200104)36:4<175::AID-JGT1005>3.0.CO;2-T).
- [23] Richard M. Karp. “Reducibility Among Combinatorial Problems”. In: *Proceedings of a symposium on the Complexity of Computer Computations, held March 20-22, 1972, at the IBM Thomas J. Watson Research Center, Yorktown Heights, New York, USA*. Ed. by Raymond E. Miller and James W. Thatcher. The IBM Research Symposia Series. Plenum Press, New York, 1972, pp. 85–103. DOI: [10.1007/978-1-4684-2001-2\\_9](https://doi.org/10.1007/978-1-4684-2001-2_9).
- [24] David G. Kirkpatrick and Pavol Hell. “On the Complexity of General Graph Factor Problems”. In: *SIAM J. Comput.* 12.3 (1983), pp. 601–609. DOI: [10.1137/0212040](https://doi.org/10.1137/0212040).
- [25] Adrian Kosowski, Michał Małafiejski, and Paweł Żyliński. “Parallel Processing Subsystems with Redundancy in a Distributed Environment”. In: *Parallel Processing and Applied Mathematics, 6th International Conference, PPAM 2005*. Ed. by Roman Wyrzykowski, Jack J. Dongarra, Norbert Meyer, and Jerzy Wasniewski. Vol. 3911. Lecture Notes in Computer Science. Springer, 2005, pp. 1002–1009. DOI: [10.1007/11752578\\_121](https://doi.org/10.1007/11752578_121).
- [26] Daniel Lokshtanov, Amer E. Mouawad, Saket Saurabh, and Meirav Zehavi. “Packing Cycles Faster Than Erdos-Posa”. In: *SIAM J. Discret. Math.* 33.3 (2019), pp. 1194–1215. DOI: [10.1137/17M1150037](https://doi.org/10.1137/17M1150037).

- [27] Michał Małafiejski and Paweł Żyliński. “Weakly Cooperative Guards in Grids”. In: *Computational Science and Its Applications – ICCSA 2005*. Ed. by Osvaldo Gervasi, Marina L. Gavrilova, Vipin Kumar, Antonio Laganà, Heow Pueh Lee, Youngsong Mun, David Taniar, and Chih Jeng Kenneth Tan. Berlin, Heidelberg: Springer Berlin Heidelberg, 2005, pp. 647–656. DOI: [10.1007/11424758\\_68](https://doi.org/10.1007/11424758_68).
- [28] Shigeru Masuyama and Toshihide Ibaraki. “Chain Packing in Graphs”. In: *Algorithmica* 6.6 (1991), pp. 826–839. DOI: [10.1007/BF01759074](https://doi.org/10.1007/BF01759074).
- [29] Silvio Micali and Vijay V. Vazirani. “An  $O(\sqrt{|V|}|E|)$  algorithm for finding maximum matching in general graphs”. In: *21st Annual Symposium on Foundations of Computer Science (sfcs 1980)*. 1980, pp. 17–27. DOI: [10.1109/SFCS.1980.12](https://doi.org/10.1109/SFCS.1980.12).
- [30] George J. Minty. “On maximal independent sets of vertices in claw-free graphs”. In: *J. Comb. Theory, Ser. B* 28.3 (1980), pp. 284–304. DOI: [10.1016/0095-8956\(80\)90074-X](https://doi.org/10.1016/0095-8956(80)90074-X).
- [31] Owen J. Murphy. “Computing independent sets in graphs with large girth”. In: *Discret. Appl. Math.* 35.2 (1992), pp. 167–170. DOI: [10.1016/0166-218X\(92\)90041-8](https://doi.org/10.1016/0166-218X(92)90041-8).
- [32] Svatopluk Poljak. “A note on stable sets and colorings of graphs”. eng. In: *Commentationes Mathematicae Universitatis Carolinae* 015.2 (1974), pp. 307–309.
- [33] Elena Prieto-Rodriguez and Christian Sloper. “Looking at the stars”. In: *Theor. Comput. Sci.* 351.3 (2006), pp. 437–445. DOI: [10.1016/j.tcs.2005.10.009](https://doi.org/10.1016/j.tcs.2005.10.009).
- [34] Najiba Sbihi. “Algorithme de recherche d’un stable de cardinalité maximum dans un graphe sans étoile”. In: *Discret. Math.* 29.1 (1980), pp. 53–76. DOI: [10.1016/0012-365X\(90\)90287-R](https://doi.org/10.1016/0012-365X(90)90287-R).
- [35] Petra Scheffler. *Practical linear time algorithm for disjoint paths in graphs with bounded tree-width*. Tech. rep. FU Berlin, Fachbereich 3 Mathematik, 1994.
- [36] Jacobo Valdes, Robert E. Tarjan, and Eugene L. Lawler. “The Recognition of Series Parallel Digraphs”. In: *SIAM Journal on Computing* 11.2 (1982), pp. 298–313. DOI: [10.1137/0211023](https://doi.org/10.1137/0211023).
- [37] Jianxin Wang, Dan Ning, Qilong Feng, and Jianer Chen. “An Improved Parameterized Algorithm for a Generalized Matching Problem”. In: *Theory and Applications of Models of Computation, 5th International Conference*. Ed. by Manindra Agrawal, Ding-Zhu Du, Zhenhua Duan, and Angsheng Li. Vol. 4978. Lecture Notes in Computer Science. Springer, 2008, pp. 212–222. DOI: [10.1007/978-3-540-79228-4\\_19](https://doi.org/10.1007/978-3-540-79228-4_19).
- [38] Chenyang Xu and Guochuan Zhang. “The Path Set Packing Problem”. In: *the 24th International Conference (COCOON 2018)*. Ed. by Lusheng Wang and Daming Zhu. Cham: Springer International Publishing, 2018, pp. 305–315. DOI: [10.1007/978-3-319-94776-1\\_26](https://doi.org/10.1007/978-3-319-94776-1_26).