Edge-disjoint Odd Cycles in 4-edge-connected Graphs

Ken-ichi Kawarabayashi*[1] and Yusuke Kobayashi†[2]

1 **National Institute of Informatics**
2-1-2, Hitotsubashi, Chiyoda-ku, Tokyo, Japan
`k_keniti@nii.ac.jp`

2 **University of Tokyo**
7-3-1, Hongo, Bunkyo-ku, Tokyo, Japan
`kobayashi@mist.i.u-tokyo.ac.jp`

Abstract

Finding edge-disjoint odd cycles is one of the most important problems in graph theory, graph algorithm and combinatorial optimization. In fact, it is closely related to the well-known max-cut problem. One of the difficulties of this problem is that the Erdős-Pósa property does not hold for odd cycles in general. Motivated by this fact, we prove that for any positive integer k, there exists an integer $f(k)$ satisfying the following: For any 4-edge-connected graph $G = (V, E)$, either G has edge-disjoint k odd cycles or there exists an edge set $F \subseteq E$ with $|F| \leq f(k)$ such that $G - F$ is bipartite. We note that the 4-edge-connectivity is best possible in this statement. Similar approach can be applied to an algorithmic question. Suppose that the input graph G is a 4-edge-connected graph with n vertices. We show that, for any $\varepsilon > 0$, if $k = O((\log\log\log n)^{1/2-\varepsilon})$, then the edge-disjoint k odd cycle packing in G can be solved in polynomial time of n.

1998 ACM Subject Classification G.2.2 Graph Theory

Keywords and phrases odd-cycles, disjoint paths problem, Erdős-Pósa property, packing algorithm, 4-edge-connectivity

Digital Object Identifier 10.4230/LIPIcs.STACS.2012.206

1 Introduction

Finding edge-disjoint odd cycles is one of the most important problems in combinatorial optimization, graph theory, and graph algorithm. Let us formulate our problem.

The edge-disjoint odd cycle packing
Input. A graph G with n vertices, and an integer k.
Problem. Does G have edge-disjoint k odd cycles?

Let us look at each importance of this problem.

* Research partly supported by Japan Society for the Promotion of Science, Grant-in-Aid for Scientific Research, by C & C Foundation, by Kayamori Foundation and by Inoue Research Award for Young Scientists.

† Supported by Grant-in-Aid for Scientific Research and by the Global COE Program "The research and training center for new development in mathematics", MEXT, Japan.

1.1 Importance in Combinatorial Optimization

In order to consider the edge-disjoint odd cycle packing, it is natural to consider the "fractional" version of the problem. Given a graph G, a *fractional edge-disjoint odd cycle packing* is a function f from $\mathcal{C}$ of odd cycles in G to $[0,1]$ satisfying $\sum_{C:e\in C} f(C) \leq 1$ for each edge e in G. The fractional version of the edge-disjoint odd cycle packing is defined to be maximizing $\sum_{C\in\mathcal{C}} f(C)$ over the fractional odd cycle packings f in G. This allows us to consider the integer programs whose linear program relaxations are duals. One can see that the edge-disjoint odd cycle packing is a "dual" problem of finding a minimum edge cover for the set of all odd cycles, which is one of the most important NP-complete problem, called the maximum cut problem. Fiorini et al. [7] proved that the integrality gap of the edge-disjoint odd cycle packing LP is bounded by a constant for planar graphs. But for general graphs, this is not true. Goemans and Williamson [10] proved that the integrality gap of the dual problem (the odd cycle covering LP) is at most 9/4 for planar graphs.

The edge-disjoint odd cycle packing is known to be NP-hard, even for planar graphs, if k is a part of input, see [7]. We remark that packing *disjoint cycles*, i.e., no parity requirement, has been also studied extensively. It is one of the most fundamental problems in graph theory with applications to several areas (see [2, 18]). For more details in this context, we refer the reader to the book by Schrijver [26].

1.2 Importance in Graph Theory

A family $\mathcal{F}$ of graphs is said to have the *Erdős-Pósa property*, if for every integer k there is an integer $f(k,\mathcal{F})$ such that every graph G contains k edge-disjoint subgraphs each isomorphic to a graph in $\mathcal{F}$ or a set F of at most $f(k,\mathcal{F})$ edges such that $G - F$ has no subgraph isomorphic to a graph in $\mathcal{F}$. The term *Erdős-Pósa property* arose because in [5], Erdős and Pósa proved that the family of cycles (without any parity condition) has this property.

On the other hand, for cycles with odd length, the situation is different. The Erdős-Pósa property does not hold for odd cycles in general. Let us give an example. For a graph G, an *odd cycle cover* is a set of edges $F \subseteq E(G)$ such that $G - F$ is bipartite. An *Escher wall of height* h consists of an elementary wall W of height h and h vertex disjoint paths $P_1, \ldots, P_h$ of length two such that:

(i) Each P_i has both endpoints on W but is otherwise disjoint from W.

(ii) One endpoint of P_i is in the ith brick of the top row of bricks of W, the other is in the $(h+1-i)$th brick of the bottom row of W. Furthermore, both of these vertices are in only one brick of W.

We remark that, as pointed out by Lovász and Schrijver (see [29]), an Escher wall of height h contains neither two edge-disjoint odd cycles nor an odd cycle cover with fewer than h edges. This shows that the Erdős-Pósa property does not hold for odd cycles. However, Reed [21] proved that the Erdős-Pósa property holds for the half integral version of the edge-disjoint odd cycle packing.

1.3 Importance in graph algorithm

The importance of finding edge-disjoint odd cycles comes also from the relation to the edge-disjoint paths problem. In the edge-disjoint paths problem, we are given a graph G and a set of k pairs of vertices (called *terminals*) in G, and we have to decide whether or not G has k edge-disjoint paths connecting given pairs of terminals. This is certainly a central problem in algorithmic graph theory and combinatorial optimization. See surveys [8, 23]. It has attracted attention in the contexts of transportation networks, VLSI layout and virtual circuit routing in high-speed networks or Internet.

We can see that the k edge-disjoint paths problem can be reduced to finding k edge-disjoint odd cycles as follows. Suppose we have an instance of the edge-disjoint paths problem with a graph $G = (V, E)$ and terminal pairs $(s_1, t_1), \ldots, (s_k, t_k)$. Let G' be the graph obtained from G by subdividing every edge into two edges and by adding an edge connecting s_i and t_i for $i = 1, \ldots, k$. Then finding edge-disjoint k odd cycles in G' is equivalent to finding k edge-disjoint paths in G.

Let us give previous known results on the edge-disjoint paths problem. If k is a part of the input of the problem, then this is known to be NP-complete [6] and it remains NP-complete even if G is constrained to be planar [17]. In fact, even for series-parallel graphs (allowing multiple edges), it remains NP-complete [19]. This is one of the few problems that are known to be NP-complete for series parallel graphs or bounded tree-width graphs. Let us observe that the vertex-disjoint paths problem is solvable for bounded tree-width graphs (and hence for series parallel graphs), see [20].

On the positive side, the seminal work of Robertson and Seymour [24] says that there is a polynomial time algorithm (actually $O(m^3)$ time algorithm, where m is the number of edges of an input graph G) for the edge-disjoint paths problem when the number k of terminals is fixed (the time complexity is improved to $O(n^2)$ in [13] where n is the number of vertices and a shorter correctness proof is given in [16]). Actually, this algorithm is one of the spin-offs of their groundbreaking work on Graph Minor project, spanning 23 papers, and giving several deep and profound results and techniques in discrete mathematics.

Recently, a faster algorithm and a much simpler correctness proof for the edge-disjoint paths problem in 4-edge-connected graphs are given in [12].

▶ **Theorem 1.** *Suppose that the input graph G is 4-edge-connected, which has n vertices. For any $\varepsilon > 0$, if $k = O((\log\log\log n)^{\frac{1}{2}-\varepsilon})$, then the k-edge-disjoint paths problem in G is solvable in polynomial time of n.*

1.4 Main Contributions

Lovász and Schrijver (see [29]) characterized graphs without two edge-disjoint odd cycles. However, their proof heavily depends on the seminal result by Seymour [28] for decomposing regular matroids. No such characterization has been known for k edge-disjoint odd cycles for any fixed k, even $k = 3$. In fact, Lovász and Schrijver considered the problem of finding a structure without many edge-disjoint odd cycles in early 1980's (actually, it seems that Gerards, Seymour, and Thomassen also considered this problem in early 1980's).

As we pointed out, one of the main difficulties is because the Erdős-Pósa property does not hold. The situation is not improved even if we assume a given graph to be 3-edge-connected, as we can easily make the above example 3-edge-connected by adding some parallel edges. On the other hand, if we assume some moderate edge-connectivity, i.e., if we assume 4-edge-connectivity, then the situation dramatically changes. Actually, our result holds also for graphs with no edge-cut of size exactly three, which we call 3-*edge-cut-free graphs*. The following is our main result.

▶ **Theorem 2.** *For any positive integer k, there exists an integer $f(k) = 2^{2^{O(k^2 \log k)}}$ satisfying the following. For any 4-edge-connected graph (or any 3-edge-cut-free graph) $G = (V, E)$, either G has edge-disjoint k odd cycles or there exists an edge set $F \subseteq E$ with $|F| \leq f(k)$ such that $G - F$ is bipartite.*

If we consider "vertex-disjoint" instead of "edge-disjoint", then we need vertex-connectivity $\Theta(k)$ as in [14]. So in the edge-disjoint case, we get a much better result. As we mentioned, the 4-edge-connectivity is best possible.

Similar proof technique for Theorem 2 can be applied to the edge-disjoint odd cycle packing. As we have already seen before, the edge-disjoint k odd cycle packing is a generalization of the k edge-disjoint paths problem. Since the edge-disjoint paths problem in 4-edge-connected graphs is much easier than the problem in general graphs [12], we expect that we can design a simpler algorithm for the edge-disjoint k odd cycle packing under the assumption that the input graph is 4-edge-connected. Here is our second contribution.

▶ **Theorem 3.** *Suppose that the input graph G is a 4-edge-connected graph (or a 3-edge-cut-free graph) with n vertices. For any $\varepsilon > 0$, if $k = O((\log\log\log n)^{1/2-\varepsilon})$, then the edge-disjoint k odd cycle packing in G is solvable in polynomial time of n.*

We have seen that the k edge-disjoint paths problem can be reduced to the edge-disjoint k odd cycle packing by subdividing every edge into two edges and by adding an edge connecting s_i and t_i for $i = 1, \dots, k$. If the original graph is 4-edge-connected, then the obtained graph is not 4-edge-connected but 3-edge-cut-free. Therefore, Theorem 3 implies Theorem 1 as a corollary.

The characterization by Lovász and Schrijver results in a polynomial time algorithm for testing whether or not a given graph contains two edge-disjoint odd cycles. In general, the following theorem is recently proved.

▶ **Theorem 4** (Kawarabayashi–Reed [15])**.** *For any fixed k, there is a polynomial time algorithm for the edge-disjoint odd cycle packing.*

However, the correctness proof of the algorithm needs the whole graph minor papers, and moreover, a hidden constant is huge.[1] On the other hand, our proof for Theorem 3 is within 5 pages, and the full proof is presented in this paper. In addition, our hidden constant concerning k is not so big and therefore we can handle superconstant concerning k.

It is natural to ask why we do not consider the weaker condition that the minimum degree being at least four, but in fact this weaker restriction would not gain us anything. Consider an instance of the edge-disjoint k odd cycle packing on an arbitrary graph G that may have degree three vertices. Then attach by two edges to each node in G a constant-sized bipartite graph of high minimum degree. This new graph G' has minimum degree high, but the resulting instance of the edge-disjoint k odd cycle packing is clearly equivalent to the original one in G. This example shows that 4-edge-connectivity is necessary. Thus we really need to stick the 4-edge-connectivity in our proof.

2 Preliminaries

In this paper, n and m always mean the numbers of vertices and edges of a given graph, respectively. A pair of subgraphs (A, B) is a *separation* if $G = A \cup B$ and there are no edges in $E(A) \cap E(B)$. The *order* of the separation (A, B) is $|V(A) \cap V(B)|$. We denote a clique (or a complete graph) with t vertices by K_t. A clique minor of order t, denoted by a *K_t-minor*,

[1] To quote David Johnson [11], "for any instance $G = (V, E)$ that one could fit into the known universe, one would easily prefer $|V|^{70}$ to even constant time, if that constant had to be one of Robertson and Seymour's." He estimates one constant in an algorithm for testing for a fixed minor H to be roughly $2 \uparrow 2^{2^{2^{2\uparrow(2\uparrow\Theta(|V(H)|))}}}$, where $2 \uparrow n$ denotes a tower $2^{2^{2^{\cdot^{\cdot^{\cdot}}}}}$ involving n 2's.

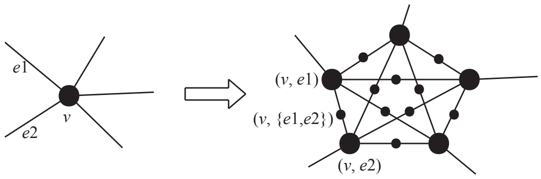

Figure 1 Construction of $L(G)$

can be thought of as t disjoint trees $T_1, \dots, T_t$ such that there is an edge between T_i and T_j for any i, j with $i \neq j$. Sometimes, one tree T_i is called a *node* of the clique minor. We say that a clique minor K consisting of disjoint trees $T_1, \dots, T_t$ is *odd*, if for every cycle C in K, $|E(C) \cap (\bigcup_i E(T_i))|$ is even. A *block* of a graph G is a maximal subgraph that is 2-connected (or a single vertex or a K_2).

It is well-known that the edge-disjoint paths problem can be reduced to the vertex-disjoint paths problem by considering the line graph. Similarly, edge-disjoint cycles in a graph correspond to vertex-disjoint cycles in its line graph. However, taking the line graph does not keep the information of parity. Therefore, instead of the line graph, we introduce the *extended line graph*, which is obtained from G by replacing every vertex by a clique whose each edge is subdivided into two edges. More precisely, for a graph $G = (V, E)$, the extended line graph $L(G) = (V^*, E^*)$ of G is defined by

$$\begin{aligned}
V_1^* &= \{(v, e) \mid v \in V,\ e \in E,\ e \text{ is incident to } v\},\\
V_2^* &= \{(v, \{e_1, e_2\}) \mid v \in V,\ e_1, e_2 \in E,\ e_1 \text{ and } e_2 \text{ are incident to } v\},\\
V^* &= V_1^* \cup V_2^*,\\
E^* &= \{(v, e_1)(v, \{e_1, e_2\}) \mid (v, e_1), (v, \{e_1, e_2\}) \in V^*\}\\
&\qquad \cup \{(v_1, e)(v_2, e) \mid e \text{ is an edge connecting } v_1 \text{ and } v_2 \text{ in } G\}.
\end{aligned}$$

See Figure 1 for the construction of $L(G)$. One can see that G contains edge-disjoint k odd cycles if and only if $L(G)$ contains vertex-disjoint k odd cycles.

We now look at definitions of the tree-width and wall. Let G be a graph, T a tree and let $\mathcal{V} = \{V_t \subseteq V(G) \mid t \in V(T)\}$ be a family of vertex sets $V_t \subseteq V(G)$ indexed by the vertices t of T. The pair $(T, \mathcal{V})$ is called a *tree-decomposition* of G if it satisfies the following three conditions:

- $V(G) = \bigcup_{t \in T} V_t$,
- for every edge $e \in E(G)$ there exists a $t \in T$ such that both ends of e lie in V_t,
- if $t, t', t'' \in V(T)$ and t' lies on the path of T between t and t'', then $V_t \cap V_{t''} \subseteq V_{t'}$.

The *width* of $(T, \mathcal{V})$ is the number $\max\{|V_t| - 1 \mid t \in T\}$ and the *tree-width* $\mathrm{tw}(G)$ of G is the minimum width of any tree-decomposition of G.

We can apply dynamic programming to solve problems on graphs of bounded tree-width, in the same way that we apply it to trees (see e.g. [1]), provided that we are given a bounded width tree decomposition. Bodlaender [3] developed a linear time algorithm.

▶ **Theorem 5.** *For an integer w, there exists a $(w^{O(w)})n^{O(1)}$ time algorithm that, given a graph G, either finds a tree decomposition of G of width w or concludes that the tree-width of G is more than w. Furthermore, if w is fixed, there exists an $O(n)$ time algorithm.*

If the tree-width and k are small, by a standard dynamic programming technique, the edge-disjoint k odd cycle packing can be solved efficiently (see e.g. [1]).

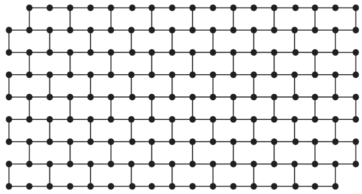

Figure 2 An elementary wall of height 8

▶ **Theorem 6.** *For integers w and k, there exists a $(w^{O(kw)})n^{O(1)}$ time algorithm for the edge-disjoint k odd cycle packing in graphs of tree-width w.*

An elementary wall of height eight is depicted in Figure 1. An *elementary wall* of height h for $h \geq 2$ is similar. It consists of h levels each containing h bricks, where a *brick* is a cycle of length six. A *wall* of height h is obtained from an elementary wall of height h by subdividing some of the edges, i.e. replacing the edges with internally vertex disjoint paths with the same endpoints. The *nails* of a wall are the vertices of degree three within it. Any wall has a unique planar embedding. We define a distance function d_W on the vertices of W so that $d_W(x, y)$ is the minimum number of regions of this embedding that an arc in the plane with endpoints x and y intersects. We define the distance between two subgraphs W_1, W_2 of W by

$$d_W(W_1, W_2) = \min\{d_W(x, y) \mid x \in V(W_1),\ y \in V(W_2)\}.$$

The *perimeter* of a wall W, denoted $\mathrm{per}(W)$, is the boundary of the unique face in this embedding which contains more than six vertices of the original elementary wall. For any wall W in a given graph G, there is a unique component U of $G - \mathrm{per}(W)$ containing $W - \mathrm{per}(W)$. The *compass* of W, denoted $\mathrm{comp}(W)$, is the subgraph of G induced by $V(U) \cup V(\mathrm{per}(W))$. A *subwall* of a wall W is a wall which is a subgraph of W. A subwall of W of height h is *proper* if it consists of h consecutive bricks from each of h consecutive rows of W. For a subgraph H, we say a proper subwall W' is *dividing* in H if H contains W' and the compass of W' in H is disjoint from $(W - W') \cap H$. A wall is *flat* if its compass does not contain two vertex-disjoint paths connecting the diagonally opposite corners. Note that if the compass of W has a planar embedding whose infinite face is bounded by the perimeter of W then W is clearly flat. Seymour [27], Thomassen [30], and others have characterized precisely which walls are flat.

One of the most important results concerning the tree-width is the main result of Graph Minors. V [22] which says the following.

▶ **Theorem 7.** *For any t, there exists a constant $f_1(t)$ such that if G has tree-width at least $f_1(t)$, then G contains a wall W of height t.*

The best known upper bound for $f_1(t)$ is 20^{2t^5}, see [4, 20, 25]. The best known lower bound is $\Theta(t^2 \log t)$, see [25]. Furthermore, such a wall can be found efficiently.

▶ **Theorem 8** ([24, 25])**.** *In a graph G with tree-width at least $f_1(t)$, we can find a wall W of height t in $(f_1(t)^{O(f_1(t))})n^{O(1)}$ time.*

3 Finding a Large Clique Minor

In our algorithm for the edge-disjoint k odd cycle packing, we divide the problem into two cases depending on whether the tree-width of the input graph is large or not. In order to deal with the case when the tree-width is large, we show the following theorem, which says that we can find a large clique minor in $L(G)$ if the tree-width of G is large.

▶ **Theorem 9.** *For any 4-edge-connected graph (or any 3-edge-cut-free graph) G and for any integer $t \geq 2$, there exists an integer $g(t) = 2^{(2^{O(t^2)})}$ such that one of the following holds:*
1. *G has tree-width at most $g(t)$.*
2. *The extended line graph $L(G)$ contains a clique minor of order t.*

Furthermore, either the tree decomposition of G of width at most $g(t)$ or the K_t-minor in $L(G)$ can be computed in $(g(t)^{O(g(t))})n^{O(1)}$ time.

The objective of this section is to give a proof of this theorem. Since the proof is almost the same as the proof of Theorem 4.1 in [12], we omit it. We note that [12] deals with the line graph instead of the extended line graph.

We now give a remark that a large clique minor plays an important role in the disjoint paths problem. By using the following theorem, we can reduce the disjoint paths problem to an equivalent smaller problem if the input graph has a large clique minor. We will use this theorem also in our proofs of Theorems 2 and 3.

▶ **Theorem 10** (Robertson and Seymour [24, Theorem (5.4)]). *Let $s_1, \ldots, s_k, t_1, \ldots, t_k$ be the terminals in a given G. If there is a clique minor of order at least $3k$ in G, and there is no separation (A, B) of order at most $2k-1$ in G such that A contains all the terminals and $B - A$ contains at least one node of the clique minor, then there are vertex-disjoint paths P_i with two ends in s_i, t_i for $i = 1, \ldots, k$.*

4 Erdős-Pósa Property (Proof of Theorem 2)

In this section, we give a proof of Theorem 2. An outline of the proof is described as follows. In Section 4.1, we show that if $L(G)$ has a large clique minor, then G is not a minimum counterexample of Theorem 2. In Section 4.2, we show that if $L(G)$ contains no large clique minor, then G cannot be a counterexample.

4.1 Property of a minimum counterexample

In this subsection, we show that if $L(G)$ has a large clique minor, then G is not a minimum counterexample of Theorem 2. To show this, we use the following theorem given in [9].

▶ **Theorem 11** (Geelen et al. [9, Theorem 13]). *There is a constant c such that if G contains a K_t-minor K, where $t \geq \lceil cl\sqrt{\log 12l} \rceil$ for a positive integer l, then one of the following holds.*
1. *G contains an odd K_l-minor.*
2. *There exists a vertex set X with $|X| < 8l$ such that the unique block (i.e., maximal 2-connected subgraph) U of $G - X$ that intersects all the nodes of K disjoint from X is bipartite.*

Furthermore, such an odd K_l-minor or a vertex set X can be found in $O(nm)$ time.

With the aid of this theorem, we show the following property of a minimum counterexample.

▶ **Proposition 12.** *Let k and l be positive integers. Suppose that $G = (V, E)$ is a 4-edge-connected graph (or a 3-edge-cut-free graph) with minimum number of edges such that it does not contain edge-disjoint k odd cycles and $G - F$ is not bipartite for any $F \subseteq E$ with $|F| \leq l$.*

Then, $L(G)$ has no clique minor of order $\max\{\lceil 3ck\sqrt{\log 36k}\rceil, 50k\}$*, where c is the constant given in Theorem 11.*

Proof. Assume that $L(G)$ has a clique minor K of order $\max\{\lceil 3ck\sqrt{\log 36k}\rceil, 50k\}$. By Theorem 11, we have one of the following.

1. $L(G)$ contains an odd K_{3k}-minor.
2. There exists a vertex set X with $|X| < 24k$ such that the unique block U of $L(G) - X$ that intersects all the nodes of K disjoint from X is bipartite.

When $L(G)$ contains an odd K_{3k}-minor, we can take vertex-disjoint k cycles each passing through three nodes of the clique minor in $L(G)$. Since these cycles are of odd length by the definition of the odd clique minor, we can find vertex-disjoint k odd cycles in $L(G)$. Hence, the corresponding cycles in G are edge-disjoint odd cycles, which contradicts the assumption.

Suppose that there exists a vertex set X with $|X| < 24k$ such that the unique block U of $L(G)-X$ that intersects all the nodes of K disjoint from X is bipartite. Let $U_1, \dots, U_q$ be the connected component of $L(G) - X - U$ that are not bipartite. Since $L(G)$ does not contain vertex-disjoint k odd cycles, we have $q < k$. By the definition of U, each U_i is adjacent to at most one vertex, say u_i, of U. Therefore, we have a separation (A', B') of $L(G)$ such that $V(A') \cap V(B') = X \cup \{u_1, \dots, u_q\}$ and $B' - A'$ is a bipartite graph containing $U - \{u_1, \dots, u_q\}$. We note that $B' - A'$ contains a clique minor of order $50k - |X \cup \{u_1, \dots, u_q\}| > 25k$. The following claim shows that we can find a separation of small order with some additional conditions.

▶ **Claim 13.** *There exists a separation (A, B) of $L(G)$ with $Y := V(A) \cap V(B)$ such that $|Y| < 25k$, $B - A$ is bipartite, $B - A$ contains a clique minor of order $25k$, we can link up Y by vertex-disjoint paths in any desired way in B, there is no edge with both end vertices in Y, and every vertex in Y is contained in V_1^*, where V_1^* is the vertex set as in the definition of $L(G)$.*

Proof of the claim. Let (A, B) be a separation of $L(G)$ of minimum order such that $V(B) \subseteq V(U) \cup X$, $B - A$ is bipartite, and $B - A$ contains at least one node of the clique minor. Furthermore, we assume that $|B|$ is minimum among such separations. We note that such a separation exists, because (A', B') satisfies these conditions. We show that this separation (A, B) satisfies the conditions in the above claim.

Define $Y = V(A) \cap V(B)$. By the definition of (A, B), it is obvious that $|Y| \le |X \cup \{u_1, \dots, u_l\}| < 25k$. Since $B - A$ contains at least one node of the clique minor, at least $50k - |Y| > 25k$ nodes are contained in $B - A$, that is, $B - A$ contains a clique minor K_B of order at least $25k$. By applying Theorem 10 with K_B and the terminal set Y, we can link up Y by vertex-disjoint paths in any desired way in B.

Assume that Y contains a vertex $v^* = (v, \{e_1, e_2\}) \in V_2^*$. Since v^* is adjacent to two vertices, say $v_1^*, v_2^* \in V_1^*$, by removing v^* from Y and adding v_1^* or v_2^*, we can obtain a separation with smaller B, which contradicts the definition of (A, B). Thus, we have $Y \subseteq V_1^*$. Furthermore, there exists no edge with both end vertices in Y, because we can remove one end vertex from Y if such an edge exists. ◀

Let (A, B) and Y be as in this claim, and we denote $Y = \{(v_i, e_i) \mid i = 1, 2, \dots, |Y|\}$ because $Y \subseteq V_1^*$. Let $F_Y = \{e_1, \dots, e_{|Y|}\}$ be the edge set in G that corresponds to Y. Then, one component H of $G - F_Y$ corresponds to $B - A$. More precisely, if $(v, e) \in V_1^*$ is contained in $B - A$, then the corresponding edge e is contained in H. Now we observe the following by the properties of $B - A$ and the definition of $L(G)$.

1. Since $B - A$ is bipartite, H is also bipartite.

2. Since we can link up Y by vertex-disjoint paths in any desired way in B, we can connect F_Y in H by edge-disjoint paths in any desired way. We note that every path in B connecting a fixed pair of vertices in Y has the same parity, and the same thing holds for paths connecting F_Y in H.
3. Since $B - A$ contains at least $25k + 1$ nodes of the clique minor, H contains at least $25k + 1$ edges.

Since H is bipartite by the first property, $V(H)$ is partitioned into two color classes $V_1(H)$ and $V_2(H)$. Now we contract $V_i(H)$ to a single vertex v_i for $i = 1, 2$ in G, and we remove some edges between v_1 and v_2 so that $25k$ edges of $E(H)$ remain between them. Let $\hat{G} = (\hat{V}, \hat{E})$ be the obtained graph. We note that F_Y might contain an edge between two vertices in $V_1(H)$ (or $V_2(H)$), because B is not necessarily bipartite. In such a case, $\hat{G}$ contains a self-loop. We can see that this reduction does not affect the existence of edge-disjoint k odd cycles by the second property. On the other hand, if $\hat{G} - \hat{F}$ is bipartite for some $\hat{F} \subseteq \hat{E}$ with $|\hat{F}| \leq l$, then we can make G bipartite by removing the edge set that corresponds to $\hat{F}$, which contradicts the assumption. Note that $\hat{F}$ does not contain an edge connecting v_1 and v_2, because the number of edges between v_1 and v_2 is bigger than $|F_Y|$. Thus, $\hat{G} - \hat{F}$ is not bipartite for any $\hat{F} \subseteq \hat{E}$ with $|\hat{F}| \leq l$. Since the number of edges decreases by the third property, this contradicts the minimality of G. ◀

4.2 When $L(G)$ has no large clique minor

In this subsection, we consider the case when $L(G)$ has no clique minor of order $t = \max\{\lceil 3ck\sqrt{\log 36k}\rceil, 50k\}$. By Theorem 9, the tree-width of G is bounded by some constant $g(t)$. Since each vertex of G has degree at most t, one vertex in G is replaced by at most $t + \binom{t}{2} < t^2$ vertices in $L(G)$. Thus, the tree-width of $L(G)$ is smaller than a constant $t^2 g(t)$. Now we show the following proposition.

▶ **Proposition 14.** *Let k be a positive integer. Let t and $g(t)$ be positive integers as above, and suppose that there exists an integer $f(i)$ satisfying the condition in Theorem 2 for $i = 1, 2, \ldots, k-1$. Suppose that G is a 4-edge-connected graph (or a 3-edge-cut-free graph) not containing edge-disjoint k odd cycles such that $L(G)$ has no clique minor of order t, and $F \subseteq E$ is a minimum edge set such that $G - F$ is bipartite. Then, we have $|F| \leq \max\{4f(k-1), 3t^2 g(t)\}$.*

Proof. Assume that $|F| > \max\{4f(k-1), 3t^2 g(t)\}$. First, we show that F is "highly-connected" in some sense. For two edges e_1, e_2, we say that a path P *connects* e_1 and e_2 if the first and last edges of P are e_1 and e_2. A path connects two edge sets F_1 and F_2 if it connects edges in F_1 and F_2. We show the following claim.

▶ **Claim 15.** *For any sets $F', F'' \subseteq F$ with $|F'| = |F''| \leq |F|/2$, there exist $|F'|$ edge-disjoint paths each connecting F' and F''.*

Proof of the claim. Let $F', F'' \subseteq F$ be sets with $|F'| = |F''| \leq |F|/2$. To derive a contradiction, assume that G does not contain $|F'|$ edge-disjoint paths connecting them. By Menger's theorem, there exists an edge set $C \subseteq E$ such that $|C| \leq |F'| - 1$ and $G - C$ contains no path connecting F' and F''. That is, there exists a partition (G_1, G_2) of $G - C$ such that $V(G_1 \cap G_2) = \emptyset$, $F' \subseteq E(G_1) \cup C$, and $F'' \subseteq E(G_2) \cup C$.

If both G_1 and G_2 contain an odd cycle, then each has at most $k-2$ edge-disjoint odd cycles. By induction hypothesis, for $i = 1, 2$, G_i has an edge set F_i with $|F_i| \leq f(k-1)$ such that

$G_i - F_i$ is bipartite. Then, $G-(F_1 \cup F_2 \cup C)$ is bipartite and $|F_1 \cup F_2 \cup C| < 2f(k-1)+\frac{|F|}{2} \leq |F|$, which contradicts the minimality of F.

Suppose that G_1 contains no odd cycle. Then, $G - ((F \setminus F') \cup C)$ is bipartite and $|(F \setminus F') \cup C| < |F|$, which contradicts the minimality of F. The same argument can be applied when G_2 contains no odd cycle. ◀

We choose one end vertex v_e arbitrary for each $e \in F$, and define a vertex set $V_F = \{(v_e, e) \mid e \in F\}$ of $L(G)$. Then, V_F is $|F|/2$-connected in $L(G)$ by the above claim, where we say that a vertex set X is *κ-connected* if $|X| \geq \kappa$ and for all subsets $X_1, X_2 \subseteq X$ with $|X_1| = |X_2| \leq \kappa$ there are $|X_1|$ vertex-disjoint paths connecting X_1 and X_2. In particular, since $|F| > \max\{4f(k-1), 3t^2 g(t)\}$, V_F is a $\frac{3}{2}t^2 g(t)$-connected set of size $3t^2 g(t)$. Now we use the following lemma.

▶ **Lemma 16** (Diestel et al. [4, Proposition 3]). *Let G be a graph and κ be a positive integer. If G contains a $(\kappa+1)$-connected set of size at least 3κ, then G has tree-width at least κ.*

Since $L(G)$ has a $\frac{3}{2}t^2 g(t)$-connected set V_F of size $3t^2 g(t)$, $L(G)$ has tree-width at least $t^2 g(t)$ by Lemma 16, which is a contradiction. ◀

By Propositions 12 and 14, $f(k)$ is bounded by $3t^2 g(t) = 2^{2^{O(k^2 \log k)}}$, which completes the proof of Theorem 2.

5 Packing Algorithm (Proof of Theorem 3)

In this section, we give an algorithm for the edge-disjoint k odd cycle packing in 4-edge-connected graphs (or 3-edge-cut-free graphs), and prove Theorem 3.

Since the case with small tree-width is easy by Theorem 6, it suffices to deal with the case when the extended line graph $L(G)$ has a large clique minor by Theorem 9. For this case, we give a procedure that reduces the original instance to a smaller instance.

▶ **Proposition 17.** *Let G be a 4-edge-connected graph (or a 3-edge-cut-free graph) and k be a positive integer. If a clique minor of order at least $\max\{\lceil 3ck\sqrt{\log 36k}\rceil, 50k\}$ is given in $L(G)$, where c is the constant given in Theorem 11, then we can reduce an instance of the edge-disjoint k odd cycle packing in G to an equivalent smaller instance in polynomial time.*

Proof. We use a similar argument to the proof of Proposition 12. Suppose that the extended line graph of G contains a clique minor K of order at least $\max\{\lceil 3ck\sqrt{\log 36k}\rceil, 50k\}$. Then, we have one of the following by Theorem 11.

1. $L(G)$ contains an odd K_{3k}-minor.
2. There exists a vertex set X with $|X| < 24k$ such that the unique block U of $L(G) - X$ that intersects all the nodes of K disjoint from X is bipartite.

When $L(G)$ contains an odd K_{3k}-minor, G contains edge-disjoint k odd cycles, which means that we can easily find edge-disjoint k odd cycles in G. (In other words, we can reduce the original instance to a trivial "YES" instance.)

Suppose that there exists a vertex set X with the above conditions. Let $A, B, Y, F_Y, H, V_1(H)$, and $V_2(H)$ be as in the proof of Proposition 12. Construct a smaller graph by contracting $V_i(H)$ to a single vertex v_i for $i = 1, 2$ and by removing some edges between v_1 and v_2, and let $\hat{G} = (\hat{V}, \hat{E})$ be the obtained graph. We have already seen in the proof of Proposition 12 that this reduction does not affect the existence of edge-disjoint k odd cycles. Since the obtained graph is smaller than the original one, the obtained instance is a desired one. ◀

Now we give a proof of Theorem 3.

Proof of Theorem 3. First, we apply Theorem 9 where $t = \max\{\lceil 3ck\sqrt{\log 36k}\rceil, 50k\}$ and c is the constant given in Theorem 11. Then, either the input graph G has tree-width at most $g(t)$, or $L(G)$ contains a clique minor of order t by Theorem 9. In the first case, we can solve the edge-disjoint k odd cycle packing in G in $(g(t)^{O(kg(t))})n^{O(1)}$ time by Theorem 6. In the second case, we apply Proposition 17 to obtain a smaller instance, and recurse the algorithm. We note that the running time $(g(t)^{O(kg(t))})n^{O(1)}$ is bounded by a polynomial of n if $k = O((\log\log\log n)^{1/2-\varepsilon})$. This shows that we can solve the problem in polynomial time. ◀

Finally, we give a remark on the time complexity for the case when k is a fixed constant. In this case, the most time consuming part is to execute Theorem 11 repeatedly. Since $L(G)$ might have $\Omega(n^3)$ vertices and edges when G has vertices of high degree, if we apply a naive reduction algorithm, then the time complexity of the reduction step becomes $O(n^6)$, and so the total running time is $O(n^6 m)$. If we find an edge set F_Y of G directly (i.e., without constructing $L(G)$), then the running time will be greatly improved, but we will not discuss this issue in this paper.

References

1 S. Arnborg and A. Proskurowski, Linear time algorithms for NP-hard problems restricted to partial k-trees, *Discrete Appl. Math.*, **23** (1989), 11–24.

2 P. Balister, Packing digraphs with directed closed trails, *Combin. Probab. Comput.*, **12** (2003), 1–15.

3 H.L. Bodlaender, A linear-time algorithm for finding tree-decomposition of small treewidth, *SIAM J. Comput.*, **25** (1996), 1305–1317.

4 R. Diestel, K.Y. Gorbunov, T.R. Jensen and C. Thomassen, Highly connected sets and the excluded grid theorem, *J. Combin. Theory Ser. B*, **75** (1999), 61–73.

5 P. Erdős and L. Pósa, On the independent circuits contained in a graph, *Canadian Journal of Mathematics*, **17** (1965), 347–352.

6 S. Even, A. Itai and A. Shamir, On the complexity of timetable and multicommodity flow problems, *SIAM J. Comput.*, **5** (1976), 691–703.

7 S. Fiorini, N. Hardy, B. Reed, and A. Vetta, Approximate min-max relations for odd cycles in planar graphs, *Mathematical Programming*, **110** (2007), 71–91.

8 A. Frank, Packing paths, cuts and circuits – a survey, in *Paths, Flows and VLSI-Layout*, B. Korte, L. Lovász, H.J. Promel and A. Schrijver (Eds.), Springer-Verlag, Berlin, 1990, 49–100.

9 J. Geelen, B. Gerards, B. Reed, P. Seymour, and A. Vetta, On the odd-minor variant of Hadwiger's conjecture, *J. Combin. Theory Ser. B*, **99** (2009), 20–29.

10 M.X. Goemans and D.P. Williamson, Primal-dual approximation algorithms for feedback problems in planar graphs, *Combinatorica*, **18** (1998), 37–59.

11 D. Johnson, The many faces of polynomial time, in The NP-completeness column: An ongoing guide, *J. Algorithms*, **8** (1987), 285–303.

12 K. Kawarabayashi and Y. Kobayashi, The edge disjoint paths problem in Eulerian graphs and 4-edge-connected graphs, *Proc. 21st Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, 2010, 45–353.

13 K. Kawarabayashi, Y. Kobayashi, and B. Reed, The disjoint paths problem in quadratic time, *J. Combin. Theory Ser. B*, to appear.

14 K. Kawarabayashi and B. Reed, Highly parity linked graphs, *Combinatorica*, **29** (2009), 215–225.

15 K. Kawarabayashi and B. Reed, Odd cycle packing, *Proc. 42nd ACM Symposium on Theory of Computing (STOC)*, 2010, 695–704.

16 K. Kawarabayashi and P. Wollan, A shorter proof of the graph minors algorithm – the unique linkage theorem, *Proc. 42nd ACM Symposium on Theory of Computing (STOC)*, 2010, 687–694. A full version of this paper is available at `http://research.nii.ac.jp/~k_keniti/uniquelink.pdf`.

17 M.R. Kramer and J. van Leeuwen, The complexity of wire-routing and finding minimum area layouts for arbitrary VLSI circuits, *Adv. Comput. Res.*, **2** (1984), 129–146.

18 M. Krivekevich, Z. Nutov, M.R. Salavatipour, J. Verstaete, and R. Yuster, Approximating algorithms and hardness results for cycle packing problems, *ACM transaction on Algorithms*, **3** (2007), article 48.

19 T. Nishizeki, J. Vygen, and X. Zhou, The edge-disjoint paths problem is NP-complete for series-parallel graphs, *Discrete Appl. Math.*, **115** (2001), 177–186.

20 B. Reed, Tree width and tangles: a new connectivity measure and some applications, in *Surveys in Combinatorics*, London Math. Soc. Lecture Note Ser. **241**, Cambridge Univ. Press, Cambridge, 1997, 87–162.

21 B. Reed, Mangoes and blueberries, *Combinatorica*, **19** (1999), 267–296.

22 N. Robertson and P.D. Seymour, Graph minors. V. Excluding a planar graph, *J. Combin. Theory Ser. B*, **41** (1986), 92–114.

23 N. Robertson and P.D. Seymour, An outline of a disjoint paths algorithm, in *Paths, Flows, and VLSI-Layout*, B. Korte, L. Lovász, H.J. Prömel, and A. Schrijver (Eds.), Springer-Verlag, Berlin, 1990, 267–292.

24 N. Robertson and P.D. Seymour, Graph minors. XIII. The disjoint paths problem, *J. Combin. Theory Ser. B*, **63** (1995), 65–110.

25 N. Robertson, P.D. Seymour, and R. Thomas, Quickly excluding a planar graph, *J. Combin. Theory Ser. B*, **62** (1994), 323–348.

26 A. Schrijver, *Combinatorial Optimization: Polyhedra and Efficiency*, Springer-Verlag, 2003.

27 P.D. Seymour, Disjoint paths in graphs, *Discrete Math.*, **29** (1980), 293–309.

28 P.D. Seymour, Decomposition of regular matroids, *J. Combin. Theory, Ser. B*, **28** (1980), 305–359.

29 P. Seymour, Matroid minors, in *Handbook of Combinatorics*, R.L. Graham, M. Grötschel, and L. Lóvasz (Eds.), North-Holland, Amsterdam, 1985, 419–431.

30 C. Thomassen, 2-linked graph, *European Journal of Combinatorics*, **1** (1980), 371–378.