

An Improved Approximation Algorithm for the Edge-Disjoint Paths Problem with Congestion Two*

Ken-ichi Kawarabayashi[†]

Yusuke Kobayashi[‡]

Abstract

In the maximum edge-disjoint paths problem, we are given a graph and a collection of pairs of vertices, and the objective is to find the maximum number of pairs that can be routed by edge-disjoint paths. An r -approximation algorithm for this problem is a polynomial-time algorithm that finds at least OPT/r edge-disjoint paths, where OPT denotes the maximum possible number of pairs that can be routed in a given instance. For a long time, an $O(n^{\frac{1}{2}})$ -approximation algorithm has been best known for this problem even if a congestion of two is allowed, i.e., each edge is allowed to be used in at most two of the paths¹. In this paper, we give a randomized $O(n^{\frac{3}{7}} \cdot \text{poly}(\log n))$ -approximation algorithm with congestion two. This is the first result that breaks the $O(n^{\frac{1}{2}})$ -approximation algorithm. In particular, we prove the following.

1. If we have a (randomized) polynomial-time algorithm for finding $\Omega(\text{OPT}^{\frac{1}{p}}/\text{polylog}(n))$ edge-disjoint paths for some $p > 1$, then we can give a randomized $O(n^{\frac{1}{2}-\alpha})$ -approximation algorithm for the edge-disjoint paths problem by using Rao-Zhou's algorithm for some $\alpha > 0$.
2. Based on the Chekuri-Khanna-Shepherd well-linked decomposition, we show that there is a randomized algorithm for finding $\Omega(\text{OPT}^{\frac{1}{4}}/\sqrt{\log n} \log n)$ edge-disjoint paths connecting given terminal pairs with congestion two.

Our framework for this algorithm is more general in the following sense. Indeed, the above two ingredients also work for the maximum edge-disjoint paths problem (with congestion one) if there is a (randomized) polynomial-time algorithm for finding $\Omega(\text{OPT}^{\frac{1}{p}})$ edge-disjoint paths connecting given terminal pairs for some $p > 1$.

Key words: Disjoint paths problem, Rao-Zhou algorithm, Chekuri-Khanna-Shepherd well-linked decomposition.

*An extended abstract of this paper appears in STOC 2011 [17].

[†]National Institute of Informatics, 2-1-2, Hitotsubashi, Chiyoda-ku, Tokyo, Japan. JST, ERATO, Kawarabayashi Large Graph Project. Supported by KAKENHI Grant Number 24106003. E-mail: k_keniti@nii.ac.jp

[‡]University of Tsukuba, Tsukuba, 305-8573, Japan. Supported by JST, ERATO, Kawarabayashi Large Graph Project, and by KAKENHI Grant Number 24106002, 24700004. E-mail: kobayashi@sk.tsukuba.ac.jp

¹After the extended abstract [17] of this paper appeared, a poly-logarithmic approximation algorithm was proposed in [9]. However, we believe that it is worth publishing the full version of [17].

1 Introduction

1.1 Background of the disjoint paths problem

In the edge- (vertex-) disjoint paths problem, we are given a graph G and a set of k pairs of vertices in G , and we have to decide whether or not G has k edge- (vertex-) disjoint paths connecting given pairs of terminals. This is certainly a central problem in algorithmic graph theory and combinatorial optimization. See [13, 34] for previous works on the disjoint paths. It has attracted attention in the contexts of transportation networks, VLSI layout and virtual circuit routing in high-speed networks or Internet. A basic technical problem is to interconnect certain prescribed “channels” on the chip such that wires belonging to different pins do not touch each other. In this simplest form, the problem mathematically amounts to finding vertex-disjoint trees or vertex-disjoint paths in a graph, each connecting a given set of vertices.

Let us give previous known results on the edge-disjoint paths problem. Early work on this problem focused on characterizing classes of instances for which the edge-disjoint paths problem can be solved in polynomial time. For example, the seminal work of Robertson and Seymour says that there is a polynomial-time algorithm for the edge-disjoint paths problem when the number of terminals, k , is fixed. Note that the running time of this algorithm is $O(n^3)$, where n is the number of vertices of an input graph G , and it is improved to $O(n^2)$ in [20]. 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. Note that the running time of their algorithm contains a huge hidden constant depending on k . If k is a part of the input of the problem, the edge-disjoint paths problem is known to be NP-complete [12], and it remains NP-complete even if an input graph G is constrained to be planar [26].

The focus has recently shifted to find approximation algorithms for the problem of finding maximum number of disjoint paths, which we call the *maximum edge- (vertex-) disjoint paths problem*. The maximum edge- (vertex-) disjoint paths problem receives considerable attention in view of approximation algorithms and hardness results. An r -approximation algorithm for this problem is a polynomial-time algorithm that on every given instance connects at least OPT/r terminal pairs using edge-disjoint paths, where OPT denotes the maximum possible number of pairs that can be routed in a given instance.

Despite a significant research in the recent years, there are wide gaps in understanding of the approximability of the maximum edge-disjoint paths problem. For *directed* graphs, no polynomial-time algorithm can achieve an approximation guarantee of $O(m^{\frac{1}{2}-\epsilon})$ for any $\epsilon > 0$, where m is the number of edges of an input graph, unless $P = NP$ [14]. But this result is based on the fact that the directed two disjoint paths problem is NP-hard, which does not hold for the undirected case. Currently, the strongest hardness result is due to [2], which gives a lower bound of $\Omega((\log m)^{\frac{1}{2}-\epsilon})$ for any $\epsilon > 0$. Constant factor and poly-logarithmic factor approximation algorithms for the maximum edge-disjoint paths problem are known for restricted classes of graphs such as trees, meshes, and highly connected graphs, Eulerian planar graphs [16, 23, 24, 22]. Here, we say that a function $f(n)$ is *poly-logarithmic* if $f(n) = O((\log n)^c)$ for some constant c , and such a function is denoted by $\text{poly}(\log n)$. Currently, the best known approximation is due to Chekuri, Khanna and Shepherd [7] who gave an $O(\sqrt{n})$ -approximation algorithm.

There are many studies on the maximum edge-disjoint paths problem in which we allow routable set to use each edge up to γ times for some positive integer γ . This setting appears when we would like to find a unit unsplittable flow between each terminal pair and each edge is endowed with some capacity γ , which is called *congestion*. Concerning approximation algorithms for the maximum edge-disjoint paths problem with congestion γ , there is still a hardness result

[2]. It says that no polynomial-time algorithm can achieve an approximation guarantee of $O(\log^{\frac{1}{\gamma+1}-\epsilon} n)$ for any $\epsilon > 0$ with congestion γ (up to $o(\log \log n / \log \log \log n)$), unless $P = NP$. On the positive side, there exists an $O(n^{\frac{1}{\gamma}})$ approximation algorithm for congestion γ [35], and when the congestion is allowed to be $O(\log n / \log \log n)$, there is an $O(1)$ -approximation algorithm via randomized rounding [28]. However, the best known algorithm for the congestion two case has been an $O(n^{\frac{1}{2}})$ -approximation algorithm [35] for a long time².

1.2 Our main results

As we see here, an $O(\sqrt{n})$ -approximation has been best known for the maximum edge-disjoint paths problem even for the congestion two case. Our main contribution is to give the first approximation algorithm that breaks an $O(\sqrt{n})$ -approximation algorithm for the congestion two case (Theorem 1.3). Actually, the approximation ratio of our algorithm is $\tilde{O}(n^{\frac{3}{7}})$, where we denote $f = \tilde{O}(g)$ if $f = O(g \log^t(g))$ for some t . Let us observe that the congestion two case has been well-studied by many researchers. Indeed there are some important results in this context, see [15, 19, 21, 25]. We have two ingredients for the proof of this result.

First, we use the algorithm of a very important result by Rao and Zhou [29] to show the following.

Theorem 1.1. *If we have a (randomized) polynomial-time algorithm that on every instance can route $\Omega(\text{OPT}^{\frac{1}{p}}/\beta(n))$ pairs with congestion γ for some $p > 1$, where OPT is the optimal value of the maximum edge-disjoint paths problem and β is some function, then there is a randomized $\tilde{O}(n^{\frac{p-1}{2p-1}}\beta(n))$ -approximation algorithm for the maximum edge-disjoint paths problem with congestion γ that runs in polynomial time. In particular, if $\beta(n)$ is a poly-logarithmic function, then we obtain a randomized $\tilde{O}(n^{\frac{p-1}{2p-1}})$ -approximation algorithm.*

The second ingredient is that using Chekuri–Khanna–Shepherd well-linked decomposition [8, 6], we can find $\Omega(\text{OPT}^{\frac{1}{4}}/\sqrt{\log n \log n})$ paths when congestion is two.

Theorem 1.2. *In any instance of the maximum edge-disjoint paths problem, we can find $\Omega(\text{OPT}^{\frac{1}{4}}/\sqrt{\log n \log n})$ edge-disjoint paths with congestion two.*

By combining Theorems 1.1 and 1.2, we have the main result as a corollary.

Theorem 1.3. *There is a randomized $\tilde{O}(n^{\frac{3}{7}})$ -approximation algorithm for the maximum edge-disjoint paths problem with congestion two.*

Our framework for the algorithm in Theorem 1.3 is more general in the following sense. Indeed, since Theorem 1.1 holds for the maximum edge-disjoint paths problem (with congestion one), we can break an $O(\sqrt{n})$ -approximation algorithm for the maximum edge-disjoint paths problem if the following conjecture is true.

Conjecture 1.4. *There is a polynomial-time algorithm for finding $\Omega(\text{OPT}^{\frac{1}{p}}/\beta(n))$ edge-disjoint paths connecting given terminal pairs for some $p > 1$, where OPT is the optimal value of the maximum edge-disjoint paths problem and β is a poly-logarithmic function.*

Let us observe that, unfortunately, the method we have developed for the proof of Theorem 1.2 would NOT work for the congestion one case. Indeed, the Chekuri–Khanna–Shepherd well-linked decomposition [8, 6] depends on the LP based algorithm, while the integrality gap of the LP-relaxation is $\Omega(\sqrt{n})$. For more details, see the next subsection.

²After the extended abstract of the present paper appeared, a poly-logarithmic approximation algorithm was proposed in [9].

Understanding the tractability of the disjoint paths problem is a fundamental problem in graph algorithms and combinatorial optimization, and progress in seeking for the boundary between approximability and inapproximability is certainly a crucial issue in this light. We believe that Theorem 1.3 gains us knowledge concerning these issues.

1.3 Overview of our algorithms

As we mentioned above, our technical contribution consists of two ingredients.

1. Using Rao–Zhou’s result [29] to obtain large number of edge-disjoint paths.
2. Using Chekuri–Khanna–Shepherd well-linked decomposition to prove Theorem 1.2.

Let us look at the first ingredient. Rao–Zhou’s result says that if the minimum cut in a given graph is $\Omega(\log^5 n)$, then there is a $\text{poly}(\log n)$ -approximation algorithm for the maximum edge-disjoint paths problem. This result was also used by Andrews [1] to give a $\text{poly}(\log n)$ -approximation algorithm with congestion $\text{poly}(\log \log n)$. We note that the algorithm that can route $\tilde{O}(\text{OPT}^{\frac{1}{p}}/\beta(n))$ pairs already gives a good solution when OPT is not so large. When OPT is large, we expect the graph to be well connected, and so Rao–Zhou’s algorithm comes as a natural alternative.

Of course, we cannot directly use Rao–Zhou’s algorithm because there may be a small cut in a given graph. However, we shall show the following.

1. If there is a cut of small size, which separates a given graph G into two parts A, B such that both A and B contain a terminal pair, then we can recursively apply our algorithm to A and B , respectively. Then the solutions of A and B give rise to a solution in the original graph G .
2. Alternatively, there is no such cut. In this case, we can select a set of edges that are incident on “small degree vertices” such that if we add $\Omega(\log^5 n)$ parallel “dummy” edges to each of the edge set then the resulting graph G' is $\Omega(\log^5 n)$ -edge-connected.

In the second case, we shall apply Rao–Zhou’s algorithm [29] to the resulting graph G' . Thus we get edge-disjoint paths \mathcal{P} of size $\text{OPT}/\text{poly}(\log n)$. We then do the following procedure:

Step 1. We pick up a path P from \mathcal{P} that goes through the minimum number of vertices with “small-degree” in the original graph.

Step 2. Remove all paths sharing a “small-degree” vertex with P from \mathcal{P} , and go to Step 1.

When we have an algorithm that can route $\tilde{O}(\text{OPT}^{\frac{1}{p}}/\beta(n))$ pairs, at the end, we shall show that this procedure would give a randomized $\tilde{O}(n^{\frac{p-1}{2p-1}}/\beta(n))$ -approximation algorithm for the edge-disjoint paths problem that runs in polynomial time, in the original graph G .

Let us now look at the second ingredient. We first adapt very interesting work by Chekuri, Khanna and Shepherd [8, 6]. The crucial ingredient of their work is the following. For a vertex set Z , we say that Z is *edge-well-linked* if for every vertex set S of G containing at most half of the vertices of Z , $|\delta(S)| \geq |S \cap Z|$, where $\delta(S)$ is the set of edges connecting S and $V(G) \setminus S$. Chekuri et al. showed that for any input graph G with the set of terminal pairs \mathcal{T} , one can compute vertex disjoint subgraphs G_1, \dots, G_r of G and their corresponding disjoint sets of vertex pairs $\mathcal{T}_1, \dots, \mathcal{T}_r$ of \mathcal{T} such that (a) each \mathcal{T}_i consists of the pairs of terminals and \mathcal{T}_i belongs to G_i ; (b) the members of the terminal pairs in \mathcal{T}_i are edge-well-linked in G_i ; (c) the total size of the sets \mathcal{T}_i is at least $\text{OPT}/\beta(n)$, where $\beta(n)$ is bounded by $O(\sqrt{\log n} \log n)$. Then,

we reduce each instance (G_i, \mathcal{T}_i) to the vertex-disjoint case by considering the line graph. We will refer to each obtained instance as a *well-linked instance*, and denote it by (G_i, \mathcal{T}_i) by abuse of notation. This well-linked instance allows us to find a “crossbar” to route the paths with terminals of \mathcal{T}_i in G_i with large congestion [8]. However, so far, no construction for a large size crossbar that allows us to route the paths of size $O(|\mathcal{T}_i|^{\frac{1}{p}})$ for $p > 1$ with constant congestion, has been known until this paper. In [8, 6], Chekuri et al. used a grid minor as a “crossbar”. They used the fact that if the graph has a large grid minor and the terminals are well-connected to it, then we can route many paths connecting terminal pairs by using the grid minor. This method works well if we consider a planar graph [6]. However, the best known result for a general graph is that if G has well-linked set of size $2^{O(t^2 \log t)}$ (equivalently, tree-width at least $2^{O(t^2 \log t)}$), then G has a $t \times t$ grid minor [18] (see also [11, 30, 33]), which only allows us to route at most $t = O(\log(\text{OPT}))$ terminals. So we cannot use this result.³

In order to find a larger crossbar structure, we shall use a “grid-like minor” as a crossbar instead, which is introduced by Reed and Wood [31], and investigated further by Kreutzer and Tazari [27]. Roughly, we say that G has a grid-like minor of order r if the graph $G \square K_2$, i.e., Cartesian product of G and K_2 (consisting of two vertices and one edge between them), has a clique minor of order r . It is shown in [27] that graphs with tree-width at least $\Omega(\text{poly}(t))$ has a grid-like minor of order t .

What we shall do is the following:

- (1) Starting with each well-linked instance (G_i, \mathcal{T}_i) , construct a grid-like minor M of order $\Omega(|\mathcal{T}_i|^{\frac{1}{4}})$, which is “attached” to the terminal \mathcal{T}_i in polynomial time.
- (2) Then construct edge-disjoint paths with the pairs of terminals in \mathcal{T}_i via M in G_i , with congestion two. To do so, we first take Cartesian product $G'_i = G_i \square K_2$. By the definition of the grid-like minor, G'_i has a clique minor M' of order $\Omega(|\mathcal{T}_i|^{\frac{1}{4}})$, because $M \square K_2$ is a subgraph of G'_i . By using the result of Robertson and Seymour [32] (and its generalization by Bollobás and Thomason [3, 4]), we can use M' as a “crossbar” to find $\Omega(|\mathcal{T}_i|^{\frac{1}{4}})$ vertex-disjoint paths with terminals in \mathcal{T}_i in G'_i . These paths clearly correspond to the disjoint paths in G_i with congestion two.

For both steps above, we shall provide details and polynomial-time algorithms in this paper.

Since the total size of the sets \mathcal{T}_i is at least $\Omega(\text{OPT}/\sqrt{\log n \log n})$, thus we can route $\Omega(\text{OPT}^{\frac{1}{4}}/\sqrt{\log n \log n})$ pairs of terminals. This allows us to prove Theorem 1.2, and consequently, we can prove Theorem 1.3 by combining Theorems 1.1 and 1.2.

Let us observe that, unfortunately, the method we just presented to use the well-linked instance would not work if we consider the congestion one case. To see that, the integrality gap for the LP-relaxation used in the above arguments is $O(\sqrt{n})$ [7]. This means that if the number of terminals is $O(\sqrt{n})$ (which is indeed the case for the known example, see [7]), then as long as we use the same LP-relaxation (which is indeed the case for the well-linked instance), there would be no way to prove Conjecture 1.4.

The paper is organized as follows: In Section 2, we give some notation. In Section 3, we shall give a proof of Theorem 1.1. In section 4, we introduce Chekuri, Khanna and Shepherd’s algorithm and the well-linked decomposition [8, 6]. In Section 5, we construct a grid-like minor that is attached to the well-linked set. In Section 6, we shall complete the proof of Theorem 1.2.

³After the submission of this paper, it was shown in [5] that every graph of tree-width $\Omega(t^{\frac{1}{98}-o(1)})$ has a $t \times t$ grid minor. This means that the grid minor approach has a potential to route $t = \Omega(\text{OPT}^{\frac{1}{p}})$ pairs with congestion two for some $p > 98$, but we will show a better result by using a “grid-like minor”.

2 Preliminaries

In this paper, n and m always mean the number of vertices of a given graph and the number of edges of a given graph, respectively. For a graph H , the vertex set and the edge set of H are denoted by $V(H)$ and $E(H)$, respectively. A *separation* (A, B) of a graph G is a pair of disjoint induced subgraphs of G such that $V(G) = V(A) \cup V(B)$, and there are no edges between $V(A - B)$ and $V(B - A)$. The order of the separation (A, B) is $|V(A) \cap V(B)|$. For a vertex set X in a graph $G = (V, E)$, let $\partial_G(X)$ be the set of edges between X and $V \setminus X$, and for $v \in V$, $\partial_G(\{v\})$ is denoted by $\partial_G(v)$. If no confusion may arise, we simply denoted by $\partial(X)$ or $\partial(v)$. For $X \subseteq V$, let $G[X]$ denote the subgraph of G induced by X , and let $N(X)$ denote the set of vertices in $V \setminus X$ that are adjacent to X . For a subgraph H of G , $N(V(H))$ is simply denoted by $N(H)$.

A complete graph with p vertices is denoted by K_p . For a graph H with $V(H) = \{v_1, \dots, v_h\}$, we say that H is a *minor* of a graph G or G contains a H -*minor* if G contains disjoint induced subgraphs G_1, \dots, G_h such that G_i is connected for every i and G contains an edge connecting G_i and G_j whenever $v_i v_j \in E(H)$. In this case, each G_i is called a *node* of the H -minor. For two graphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$, their *Cartesian product* $G_1 \square G_2 = (V^*, E^*)$ is the graph defined as follows:

- the vertex set V^* is $\{(v_1, v_2) \mid v_1 \in V_1, v_2 \in V_2\}$, and
- there exists an edge between $(v_1, v_2) \in V^*$ and $(v'_1, v'_2) \in V^*$ if and only if either $v_1 = v'_1$ and $v_2 v'_2 \in E_2$, or $v_2 = v'_2$ and $v_1 v'_1 \in E_1$.

In particular, for a graph $G = (V, E)$, $G \square K_2$ is the graph consisting of G , its copy $G' = (V', E')$, and $|V|$ edges each connecting one vertex in V and its corresponding vertex in V' .

3 Proof of Theorem 1.1

In this section, we give a proof of Theorem 1.1. As we mentioned above, we use the following result.

Theorem 3.1 (Rao–Zhou [29]). *There is a constant c such that a randomized $O(\log^{10} n)$ -approximation algorithm exists for the maximum edge-disjoint paths problem in a graph with edge-connectivity at least $c \log^5 n$.*

Suppose that $\text{OPT} = \tilde{O}(n^{\frac{p}{2p-1}})$. Since we can find $\Omega(\text{OPT}^{\frac{1}{p}}/\beta(n))$ edge-disjoint paths in the assumption, the approximation ratio is $O(\text{OPT}^{\frac{p-1}{p}}\beta(n))$ which is also represented as $\tilde{O}(n^{\frac{p-1}{2p-1}}\beta(n))$. Thus, in what follows in this section, we only consider the case of $\text{OPT} = \Omega(n^{\frac{p}{2p-1}} \log^t n)$ for any t . In particular, we may assume that $\text{OPT} = \Omega(n^{\frac{p}{2p-1}} \log^{10} n)$.

3.1 Reduction to graphs without small cuts

Let (X_1, X_2) be a partition of $V(G)$ such that $|\partial(X_1)| \leq c \log^5 n$, where c is the constant as in Theorem 3.1. Let OPT_1 (resp. OPT_2) be the optimal value of the maximum edge-disjoint paths problem when we restrict the problem to $G[X_1]$ (resp. $G[X_2]$). Since $|\partial(X_1)| \leq c \log^5 n$, it is easy to see that

$$\text{OPT} \leq \text{OPT}_1 + \text{OPT}_2 + c \log^5 n.$$

Let $n_1 = |X_1|$ and $n_2 = |X_2|$. We reduce the problem into two small subproblems using the following lemma.

Lemma 3.2. *Let α be a constant with $0 < \alpha < 1$, and let $n, n_1, n_2, \text{OPT}, \text{OPT}_1$, and OPT_2 be defined as above. If $\text{OPT}_1 \geq 1$, $\text{OPT}_2 \geq 1$, and $\text{OPT} \geq n^\alpha$, then*

$$\frac{\text{OPT}_1}{n_1^\alpha} + \frac{\text{OPT}_2}{n_2^\alpha} \geq \frac{\text{OPT}}{n^\alpha}$$

holds for sufficiently large n .

Proof. We may assume that $n_2 \geq n_1$. Let γ be a constant with $0 < \gamma < \alpha$ (e.g. $\gamma = \alpha/2$). We consider the following two cases separately.

Case 1: ($n_2/n_1 \geq n^\gamma$) In this case, we have

$$\begin{aligned} \frac{\text{OPT}_1}{n_1^\alpha} + \frac{\text{OPT}_2}{n_2^\alpha} &= \frac{\text{OPT}_1 + \text{OPT}_2 + (n_2^\alpha/n_1^\alpha - 1)\text{OPT}_1}{n_2^\alpha} \\ &\geq \frac{\text{OPT}}{n^\alpha} + \frac{-c \log^5 n + (n^{\alpha\gamma} - 1)\text{OPT}_1}{n_2^\alpha} \geq \frac{\text{OPT}}{n^\alpha} \end{aligned}$$

for sufficiently large n .

Case 2: ($n_2/n_1 \leq n^\gamma$) In this case, it holds that $n_2 \leq \frac{n^{1+\gamma}}{1+n^\gamma}$. Hence, we have

$$\begin{aligned} \frac{\text{OPT}_1}{n_1^\alpha} + \frac{\text{OPT}_2}{n_2^\alpha} &\geq \frac{\text{OPT}_1 + \text{OPT}_2}{n_2^\alpha} \\ &\geq \frac{\text{OPT}}{n^\alpha} + \frac{(1 - (\frac{n^\gamma}{1+n^\gamma})^\alpha)\text{OPT} - c \log^5 n}{n_2^\alpha} \\ &\geq \frac{\text{OPT}}{n^\alpha} + \frac{\frac{\alpha}{2}(1 - (\frac{n^\gamma}{1+n^\gamma}))\text{OPT} - c \log^5 n}{n_2^\alpha} \geq \frac{\text{OPT}}{n^\alpha} \end{aligned}$$

for sufficiently large n . Here we use the fact that for a positive integer t with $\frac{1}{2^{t-1}} > \alpha \geq \frac{1}{2^t}$ and for $x := \frac{n^\gamma}{1+n^\gamma}$, it holds that

$$1 - x^\alpha \geq 1 - x^{\frac{1}{2^t}} = \frac{1 - x}{(1 + x^{\frac{1}{2}})(1 + x^{\frac{1}{4}}) \cdots (1 + x^{\frac{1}{2^t}})} \geq \frac{1 - x}{2^t} > \frac{\alpha(1 - x)}{2}.$$

□

This lemma shows that by combining an $O(n^\alpha)$ -approximation solution in G_1 and one in G_2 , we obtain an $O(n^\alpha)$ -approximation solution in G . Thus, if there exists a partition (X_1, X_2) of $V(G)$ such that $|\partial(X_1)| \leq c \log^5 n$, $\text{OPT}_1 \geq 1$, and $\text{OPT}_2 \geq 1$, then we can reduce the problem into two subproblems.

Lemma 3.3. *Let α be a constant with $0 < \alpha < 1$, and let $n, n_1, n_2, \text{OPT}, \text{OPT}_1$, and OPT_2 be defined as above. Suppose that $\text{OPT}_i \geq 1$ and we have an $\tilde{O}(n_i^\alpha)$ -approximation solution of the subproblem in $G[X_i]$ for $i = 1, 2$. Then, we have an $\tilde{O}(n^\alpha)$ -approximation solution of the original problem in G .*

Proof. If n is bounded by a constant number, then we can solve the maximum edge-disjoint paths problem in G directly. Thus, we may assume that n is sufficiently large. When $\text{OPT} = \Omega(n^\alpha)$, by Lemma 3.3, by combining an $\tilde{O}(n^\alpha)$ -approximation solution in G_1 and one in G_2 , we obtain an $\tilde{O}(n^\alpha)$ -approximation solution in G . When $\text{OPT}/\tilde{O}(n^\alpha) \leq 1$, we can easily take one path in G to obtain an $\tilde{O}(n^\alpha)$ -approximation solution in G . □

3.2 Algorithm for highly connected graphs

By repeating the reduction as in Lemma 3.3, we may assume that the graph has no partition (X_1, X_2) of $V(G)$ such that $|\partial(X_1)| \leq c \log^5 n$, $\text{OPT}_1 \geq 1$, and $\text{OPT}_2 \geq 1$. Note that if $G[X_i]$ is connected, $\text{OPT}_i \geq 1$ is equivalent to that X_i contains at least one terminal pair. We also note that the edge-connectivity of the graph in this instance is not necessarily at least $c \log^5 n$. In order to apply Theorem 3.1, we now construct a new instance, in which the edge-connectivity of the graph is at least $c \log^5 n$. The whole algorithm will be described in Section 3.3.

First, while there exists a partition (X_1, X_2) of $V(G)$ such that $|\partial(X_1)| \leq c \log^5 n$, $\text{OPT}_1 = 0$, $|X_1| \geq 2$, and $G[X_1]$ is connected, we contract X_1 to a single vertex. We repeat the reduction and let G' be the obtained graph. We say that a vertex in G' corresponding to more than one vertex in G is a *contracted vertex*, and a vertex in G' with degree at most $c \log^5 n$ is a *small-degree vertex*. Then, G' satisfies that

- if a partition (X_1, X_2) of $V(G)$ satisfies that $|\partial(X_1)| \leq c \log^5 n$ and both $G[X_1]$ and $G[X_2]$ are connected, then one of X_1 or X_2 consists of a single vertex, and
- each contracted vertex is a small-degree vertex.

Let V_{small} be the set of small-degree vertices in $V(G')$. For each $v \in V_{\text{small}}$, we choose one edge $e \in \partial(v)$ arbitrarily and add $c \log^5 n - |\partial(v)|$ parallel edges of e . By the properties of G' , the obtained graph G'' is $c \log^5 n$ edge-connected and each vertex in V_{small} has degree at most $c^2 \log^{10} n$ in G'' .

Since contracting edges and adding edges do not decrease the number of edge-disjoint paths, G'' has at least OPT edge-disjoint paths. Hence, by Theorem 3.1, we can find $\text{OPT}/O(\log^{10} n)$ edge-disjoint paths \mathcal{P}_0 in G'' with high probability.

Now we find $\text{OPT}/\tilde{O}(n^{\frac{p-1}{2p-1}})$ edge-disjoint paths $\mathcal{P}' \subseteq \mathcal{P}_0$, in which no two paths share a vertex in V_{small} in common. Our algorithm for finding \mathcal{P}' is as follows. Here, recall that we assume that $\text{OPT} = \Omega(n^{\frac{p}{2p-1}} \log^{10} n)$.

Procedure A

Input. $\text{OPT}/O(\log^{10} n)$ edge-disjoint paths \mathcal{P}_0 in G'' , where $\text{OPT} = \Omega(n^{\frac{p}{2p-1}} \log^{10} n)$.

Output. $\text{OPT}/\tilde{O}(n^{\frac{p-1}{2p-1}})$ edge-disjoint paths $\mathcal{P}' \subseteq \mathcal{P}_0$, in which no two paths share a vertex in V_{small} in common.

Step 0. Set $\mathcal{P} := \mathcal{P}_0$ and $\mathcal{P}' = \emptyset$.

Step 1. If $\mathcal{P} = \emptyset$, then output \mathcal{P}' and stop the algorithm. Otherwise, take a path $P \in \mathcal{P}$ that goes through the minimum number of vertices in V_{small} , and add P to \mathcal{P}' .

Step 2. Remove from \mathcal{P} all paths sharing a vertex in V_{small} with P , and go to Step 1.

We show that this algorithm finds many edge-disjoint paths.

Claim 3.4. *There exists a constant $c_1 > 0$ such that the output \mathcal{P}' of Procedure A satisfies that $|\mathcal{P}'| \geq c_1 \cdot \text{OPT}/(n^{\frac{p-1}{2p-1}} \log^{31} n)$ for sufficiently large n .*

Proof. In this proof, we suppose that n is sufficiently large. While $|\mathcal{P}| \geq |\mathcal{P}_0|/2$, we can find a path $P \in \mathcal{P}$ that goes through at most $(n \cdot c^2 \log^{10} n)/|\mathcal{P}| \leq c_2 n \log^{20} n / \text{OPT}$ vertices in V_{small} for some $c_2 > 0$. Hence, P shares a vertex in V_{small} with at most $c_2 c^2 n \log^{30} n / \text{OPT}$ paths in \mathcal{P} . Since $\text{OPT} = \Omega(n^{\frac{p}{2p-1}} \log^{10} n)$ implies that $|\mathcal{P}_0| = \Omega(n^{\frac{p}{2p-1}})$, there exists a constant $c_1 > 0$ such that

$$(c_1 \cdot \text{OPT}/(n^{\frac{p-1}{2p-1}} \log^{31} n))(c_2 c^2 n \log^{30} n / \text{OPT}) = c_1 c_2 c^2 n^{\frac{p}{2p-1}} \log^{-1} n < |\mathcal{P}_0|/2.$$

This inequality means that the algorithm adds $c_1 \cdot \text{OPT} / (n^{\frac{p-1}{2p-1}} \log^{31} n)$ paths to \mathcal{P}' while $|\mathcal{P}| \geq |\mathcal{P}_0|/2$. \square

3.3 Algorithm description

In this section, we summarize the results in Sections 3.1 and 3.2, and describe the whole algorithm for the maximum edge-disjoint paths problem. In our algorithm, we assume that we have an algorithm for finding $\Omega(\text{OPT}^{\frac{1}{p}}/\beta(n))$ edge-disjoint paths, where β is some function and OPT is the optimal value of the maximum edge-disjoint paths problem. Recall that OPT_1 (resp. OPT_2) is the optimal value when we restrict the problem to the graph induced by X_1 (resp. X_2), and c is the constant as in Theorem 3.1.

Algorithm for the maximum edge-disjoint paths problem

Input. A graph G with terminal pairs.

Output. $\text{OPT}/\tilde{O}(n^{\frac{p-1}{2p-1}}\beta(n))$ edge-disjoint paths connecting the terminal pairs, where OPT is the maximum number of such edge-disjoint paths.

Step 0. If the number of vertices of G is small compared with constants in Lemma 3.2 and Claim 3.4, then we solve the problem in G directly.

Step 1. Find $\Omega(\text{OPT}^{\frac{1}{p}}/\beta(n))$ edge-disjoint paths \mathcal{P}_1 connecting the terminal pairs. Set $G' := G$.

Step 2. Find a partition (X_1, X_2) of $V(G')$ such that $|X_1| \geq 2$, $|X_2| \geq 2$, and $|\partial_{G'}(X_1)| \leq c \log^5 n$. If such a partition exists, then go to Step 3. Otherwise, go to Step 4.

Step 3. If both $\text{OPT}_1 \geq 1$ and $\text{OPT}_2 \geq 1$ hold, then divide the problem in the original graph G into two subproblems according to the partition, and solve the subproblems recursively. Then, obtain a solution of the original instance by putting two solutions together.

Otherwise, we may assume that $\text{OPT}_1 = 0$. Contract each connected component of $G'[X_1]$ (the subgraph of G' induced by X_1) to a single vertex. Then, go to Step 2.

Step 4. For each vertex $v \in V(G')$ with degree at most $c \log^5 n$, i.e., for each small-degree vertex v , choose one edge $e \in \partial_{G'}(v)$ arbitrarily and add $c \log^5 n - |\partial_{G'}(v)|$ parallel edges of e . Apply Theorem 3.1 and Procedure A to the obtained graph, and obtain edge-disjoint paths \mathcal{P}' , in which no two paths share a small-degree vertex in common. Let \mathcal{P}_2 be the paths in the original graph G corresponding to \mathcal{P}' , and go to Step 5.

Step 5. If $|\mathcal{P}_1| \geq |\mathcal{P}_2|$, then output \mathcal{P}_1 . Otherwise, output \mathcal{P}_2 .

It is obvious that this algorithm runs in polynomial time if we have an algorithm for finding $\Omega(\text{OPT}^{\frac{1}{p}}/\beta(n))$ edge-disjoint paths. The correctness of this algorithm follows from the arguments in Sections 3.1 and 3.2, which completes the proof of Theorem 1.1.

4 Well-linked instances – CKS Algorithm

The rest of this paper is devoted to an algorithm for finding $\Omega(\text{OPT}^{\frac{1}{4}})$ edge-disjoint paths with congestion two.

In our algorithm for the maximum disjoint paths problem with congestion two, we first decompose the given instance into well-linked instances with the aid of very useful work of Chekuri, Khanna and Shepherd (CKS) [6, 8]. Recall that a set of vertices Z in G is *edge-well-linked* if for every vertex set S containing at most half of Z , $|N(S)| \geq |S \cap Z|$. By Menger's theorem, if Z is edge-well-linked, then for any integer $q \leq \lfloor \frac{|Z|}{2} \rfloor$ and for any disjoint subsets

$Z_1, Z_2 \subseteq Z$ with $|Z_1| = |Z_2| = q$, there exist q edge-disjoint paths connecting Z_1 and Z_2 . Here is the key CKS theorem [6, 8]:

Theorem 4.1 (Chekuri–Khanna–Shepherd [6, 8]). *For an input graph G with the set of terminal pairs \mathcal{T} , one can compute vertex disjoint subgraphs G_1, \dots, G_r of G and their corresponding disjoint sets of vertex pairs $\mathcal{T}_1, \dots, \mathcal{T}_r$ of \mathcal{T} such that the following holds:*

- (1) *each \mathcal{T}_i consists of the pairs of terminals and \mathcal{T}_i belongs to G_i ;*
- (2) *the members of the terminal pairs in \mathcal{T}_i are edge-well-linked in G_i ;*
- (3) *the total size of the sets \mathcal{T}_i is at least $\text{OPT}/\beta(n)$, where $\beta(n)$ is bounded by $O(\sqrt{\log n} \log n)$.*

We now reduce each edge-well-linked instance of the maximum edge-disjoint paths problem to an instance of the maximum vertex-disjoint paths problem by considering the line graph. By the definition of edge-well-linked sets, the terminal set Z' of the obtained instance satisfies the following property: for any integer $q \leq \frac{|Z'|}{2}$ and for any disjoint subsets $Z_1, Z_2 \subseteq Z'$ with $|Z_1| = |Z_2| = q$, there exist q vertex-disjoint paths connecting Z_1 and Z_2 . Such a set Z' is called a *well-linked set*, and an instance is called *well-linked* if its terminal set is well-linked.

In what follows, we give an algorithm for finding $\Omega(\text{OPT}^{\frac{1}{4}})$ vertex-disjoint paths in a well-linked instance with congestion two. We note that in the maximum vertex-disjoint paths problem, “congestion two” means that each vertex is allowed to be used in at most two of the paths. Since we consider vertex-disjoint paths, in what follows, we assume that every graph has no loops and no multiple edges. In the rest of this paper, we denote “vertex-disjoint” simply by “disjoint”.

5 Algorithm for the disjoint paths problem with congestion two

In this section, we give a polynomial-time algorithm that finds $O(\text{OPT}^{\frac{1}{5}})$ disjoint paths connecting terminal pairs with congestion two.

Proposition 5.1. *In a well-linked instance of the disjoint paths problem, we can find $\Omega(\text{OPT}^{\frac{1}{5}})$ disjoint paths with congestion two.*

Later, in Section 6, we improve the number of disjoint paths to $\Omega(\text{OPT}^{\frac{1}{4}})$, using the similar line of the proof in this section.

5.1 Constructing webs from the well-linked set

In this subsection, we shall construct, what we call, a k -web from a well-linked set. To do so, we need some definition. A k -web of order h in a given graph G is a set of h disjoint trees T_1, \dots, T_h such that for any distinct i, j , there is a set of k disjoint paths connecting T_i and T_j . A tree T is called *subcubic* if its maximum degree is at most three. We now prove the following lemma.

Lemma 5.2. *Given a well-linked set Z in a given graph G , there exists a subcubic tree T with a vertex set $Z' \subseteq Z$ such that $Z' \subseteq V(T)$, $|Z'| \geq \frac{|Z|}{2}$, and there are at most $|Z'| - 1$ vertices of degree three in T . Moreover, given a well-linked set Z , there is an $O(nm)$ time algorithm to construct such a subcubic tree T and a vertex set $Z' \subseteq Z$.*

Proof. We inductively construct a vertex set $Z_i \subseteq Z$ and a subcubic tree T_i such that $Z_1 \subseteq Z_2 \subseteq \dots, Z_i \subseteq V(T_i)$, and there are at most $|Z_i| - 1$ vertices of degree 3 in T_i . When $|Z_i| \geq \frac{|Z|}{2}$, we set $Z' = Z_i$, and output the subcubic tree $T = T_i$.

Initially, we choose two vertices v, u in Z , and find a path between v and u in G . Since Z is a well-linked set (and hence G is connected), clearly such a path exists. Set this path T_1 with $Z_1 = T_1 \cap Z$ that contains both u and v . Clearly this satisfies the above conditions. Thus the base step of the induction is done.

Suppose we have constructed T_i and Z_i so far with $|Z_i| < \frac{|Z|}{2}$. Let C_i be the vertices of degree 3 in T_i , then $|C_i| \leq |Z_i| - 1$. By the definition of well-linked sets, there exist $|Z_i|$ disjoint paths between Z_i and $Z - Z_i$. This implies that there is a path P between $w' \in Z - Z_i$ and Z_i which avoids any vertex in C_i in G . We now follow the path P from w' , and let q be the first vertex in $V(T_i \cap P)$, and let P' be the subpath of P between w' and q . Then we set $T_{i+1} = T_i \cup P'$ and $Z_{i+1} = Z_i \cup (P' \cap Z)$. Clearly T_{i+1} is a tree, and there is no vertex of degree four or more in T_{i+1} since the path P' does not hit any vertex in C_i . It is also clear that q is the only vertex in T_{i+1} which could have degree three in $T_{i+1} - V(C_i)$. Thus T_{i+1} and Z_{i+1} satisfy our assumption. If $|Z_{i+1}| \geq \frac{|Z|}{2}$, we stop, otherwise, we go to the next iteration.

This proof can be clearly converted into a polynomial-time algorithm (actually $O(nm)$ time algorithm) since in each step, we just need to find a path in $G - C_i$, and there are at most $\frac{|Z|}{2}$ steps. This completes the proof. \square

This lemma implies the following corollary, which is of independent interest.

Corollary 5.3. *For a well-linked set Z in a given graph G , there exists a tree T such that $Z \subseteq V(T)$ and the maximum degree of T is at most four.*

Proof. By Lemma 5.2, there exists a subcubic tree T' with a vertex set $Z' \subseteq Z$ such that $Z' \subseteq V(T')$ and $|Z'| \geq \frac{|Z|}{2}$. By the definition of well-linked sets, there exist $|Z - Z'|$ disjoint paths between Z' and $Z - Z'$. By combining these paths with T' , we obtain a desired tree. \square

We are now ready to state our main result in this subsection.

Lemma 5.4. *Let h and k be positive integers. Given a well-linked set Z with $|Z| \geq 4hk$, there is a polynomial time algorithm to construct a k -web of order h . Moreover, the k -web of order h consists of h disjoint trees T_1, \dots, T_h such that each T_i contains at least k vertices in Z , and there are k disjoint paths connecting $T_i \cap Z$ and $T_j \cap Z$ in G for any i, j .*

Proof. We adapt Lemma 5.2 to obtain a subcubic tree T with a vertex set $Z' \subseteq Z$ such that $Z' \subseteq V(T)$ and $|Z'| \geq 2hk$. In [10, Lemma 12.4.6], the following was shown:

Let $k \geq 1$ be an integer. Let T be a tree of maximum degree at most three and $X \subseteq V(T)$. Then T has a set F of edges such that every component of $T - F$ has between k and $2k - 1$ vertices in X , except that one such component may have fewer vertices in X .

For the algorithmic purpose, we just give a sketch of proof, which is algorithmic. If $|X| \leq 2k - 1$, there is nothing to do. So we may assume $|X| \geq 2k$. Let e be an edge of T such that some component T' of $T - e$ contains at least k vertices of X and $|T'|$ is as small as possible (such a choice can be clearly found in polynomial time). Since the maximum degree of T is at most three, the end of e in T' has degree at most two in T' . Thus the minimality of T' implies $|X \cap V(T')| \leq 2k - 1$. We just recursively apply this argument to $T - T'$. Thus we can find the above edge set F in polynomial time.

Therefore, in polynomial time, we can find a set F of edges in T such that: (1) there are h subtrees T_1, \dots, T_h of $T - F$, and (2) $k \leq |V(T_i) \cap Z| \leq 2k - 1$ for each subtree T_i . Furthermore, for any $1 \leq i < j \leq h$, there are k disjoint paths between $T_i \cap Z'$ and $T_j \cap Z'$ the definition of well-linked sets. This completes the proof, because such paths can be found in polynomial time. \square

5.2 Using a grid-like minor connecting terminal pairs

In this subsection, we construct a large *grid-like minor* from a k -web, and use it in order to connect terminal pairs.

We now give some definitions. Let \mathcal{P}_1 and \mathcal{P}_2 be a set of disjoint connected subgraphs in a given graph G . We denote by $I(\mathcal{P}_1, \mathcal{P}_2)$ the *intersection graph* of \mathcal{P}_1 and \mathcal{P}_2 defined as follows: $I(\mathcal{P}_1, \mathcal{P}_2)$ is the bipartite graph with partite sets \mathcal{P}_1 and \mathcal{P}_2 , which has one vertex for each element of \mathcal{P}_1 and \mathcal{P}_2 , and an edge between two vertices exists if the corresponding subgraphs in \mathcal{P}_1 and in \mathcal{P}_2 , respectively, intersect. Thus there are $|\mathcal{P}_1|$ vertices in one partite set of the bipartite graph, and $|\mathcal{P}_2|$ vertices in the other partite set. For sets \mathcal{P}_1 and \mathcal{P}_2 of disjoint paths in G , we say that a pair $(\mathcal{P}_1, \mathcal{P}_2)$ is a *half-integral H -minor* if $I(\mathcal{P}_1, \mathcal{P}_2)$ contains the graph H as a minor. If G contains such a pair $(\mathcal{P}_1, \mathcal{P}_2)$, we say that G has a half-integral H -minor. In the special case when H is the complete graph K_ℓ , we call it a *grid-like minor* of order ℓ . Note that if a graph $G = (V, E)$ has a half-integral H -minor, then $G \square K_2$ contains H as a minor (see [31, Lemma 3.4]).

We construct a large grid-like minor with the aid of the following result of [27].

Theorem 5.5 (Kreutzer–Tazari [27, Lemma 4.2]). *Let G be a graph and let T_1, \dots, T_h be given to be the disjoint trees of a k -web of order h in G with $k \geq ch^2p^2$ for some constant c . Then there is a randomized polynomial-time algorithm to find either a grid-like minor of order p (thus there is a K_p -minor in $G \square K_2$) or a set of $\binom{h}{2}$ disjoint paths $Q_{i,j}$ in the k -web, $1 \leq i < j \leq h$, such that $Q_{i,j}$ connects T_i and T_j (thus there is a K_h -minor). Furthermore, if each of the k disjoint paths between T_i and T_j contains a terminal for any i, j , then every node of the obtained minor (the K_p -minor in $G \square K_2$ or the K_h -minor in G) contains a terminal or its copy.*

Although the latter half of Theorem 5.5 is not stated in [27], it is immediately derived from the construction (see the proof of Theorem 6.2). We also note that a k -web in this paper means a weak k -web in [27], but as mentioned in [27, Lemma 4.7], it still works.

Now we are ready to prove Proposition 5.1.

Proof of Proposition 5.1. Let Z be the terminal set. We apply Lemma 5.4 and Theorem 5.5 in which $p = h = \Theta(|Z|^{\frac{1}{5}})$ and $k = \Theta(|Z|^{\frac{4}{5}})$. Then, we have a K_p -minor in $G \square K_2$ whose each node contains a terminal or its copy. Note that a K_h -minor of G is also a K_h -minor of $G \square K_2$. Suppose that $G \square K_2$ consists of G , its copy $G' = (V', E')$, and edges each connecting one vertex in V and its corresponding vertex in V' . Now we use the following theorem.

Theorem 5.6 (Robertson–Seymour [32, Theorem (5.4)]). *Let $s_1, \dots, s_q, t_1, \dots, t_q$ be terminals in a given graph G . If there is a clique minor of order at least $3q$ in G , and there is no separation (A, B) of order at most $2q - 1$ in G such that A contains all the terminals and $B - A$ contains at least one node of the clique minor, then there are disjoint paths P_i with two ends in s_i, t_i for $i = 1, \dots, q$. Furthermore, given the above clique minor, desired disjoint paths can be found in $O(qm)$ time.*

We take $q = \max\{\lfloor p/6 \rfloor, 1\}$ terminal pairs in Z and let Z_0 be the set of such terminals. Note that these terminals are contained in the original graph G . In what follows, we show that the terminals can be connected by disjoint paths in $G \square K_2$ by applying Theorem 5.6. Since the case of $q = 1$ is obvious, we suppose that $q = \lfloor p/6 \rfloor$.

Assume that there is a separation (A, B) of order at most $2q - 1$ in $G \square K_2$ such that A contains Z_0 and $B - A$ contains at least one node of the clique minor. By connectivity of the clique minor, at least $p - (2q - 1) \geq 4q$ nodes are contained in $B - A$. Then, we have $|(Z \cup Z') \cap V(B - A)| \geq 4q$, where Z' is the copy of Z in G' . Since $v' \in Z' \cap V(B - A)$ implies that the corresponding vertex v is in $Z \cap V(B)$, B contains at least $2q$ terminals in Z . By

setting $A_1 = A \cap G$ and $B_1 = B \cap G$, we have a separation (A_1, B_1) of G such that its order is at most $2q - 1$, A_1 contains Z_0 , and B_1 contains $2q$ terminals, which contradicts the definition of well-linked sets.

Therefore, there is no such a separation (A, B) , and hence we can connect Z_0 by q disjoint paths in $G \square K_2$ by Theorem 5.6. These paths correspond to the disjoint paths in G with congestion two, which completes the proof. \square

We note that we improve the number of disjoint paths to $\Omega(\text{OPT}^{\frac{1}{4}})$ in the next section.

6 Improving the algorithm

In this section, we improve the number of disjoint paths in Proposition 5.1 to $\Omega(\text{OPT}^{\frac{1}{4}})$ and prove Theorem 1.2. For the improvement, we use a graph with large minimum degree instead of a clique minor. We denote the minimum degree of a graph H by $\delta(H)$.

Theorem 6.1. *In a well-linked instance of the disjoint paths problem, we can find $\Omega(\text{OPT}^{\frac{1}{4}})$ disjoint paths with congestion two.*

To prove Theorem 6.1, we begin with the modification of Theorem 5.5. Our proof essentially follows the proof given in [31] (see also [27, Lemma 4.7]).

Theorem 6.2. *Let G be a graph and let T_1, \dots, T_h be given to be the disjoint trees of a k -web of order h in G with $k \geq ch^2p$ for some constant c . Then there is a randomized polynomial-time algorithm to find either a half-integral H -minor, where H is some graph satisfying $2\delta(H) \geq |H| + 4p - 2$ or a set of $\binom{h}{2}$ disjoint paths $Q_{i,j}$ in the k -web, $1 \leq i < j \leq h$, such that $Q_{i,j}$ connects T_i and T_j (thus there is a K_h -minor).*

Furthermore, if each of the k disjoint paths between T_i and T_j contains a terminal, then every node of the obtained minor (a H -minor in $G \square K_2$ or a K_h -minor in G) contains a terminal.

Proof. Starting with a k -web of order h with $k \geq ch^2p$ for some constant c , we consider the disjoint paths $\mathcal{P}_{i,j}$ between the pairs of trees T_i and T_j from the k -web. Note that these paths can be found by a max-flow computation in polynomial time. For each pair of these paths $\mathcal{P}_{i,j}$, i.e., consider $\mathcal{P}_{i,j}$ and $\mathcal{P}_{i',j'}$ with $i \neq j$, $i' \neq j'$ and $(i, j) \neq (i', j')$, we construct the intersection graph $I(\mathcal{P}_{i,j}, \mathcal{P}_{i',j'})$. For each (i, j, i', j') , we check if the graph $I(\mathcal{P}_{i,j}, \mathcal{P}_{i',j'})$ is $22p$ -degenerate, where we say that a graph G is d -degenerate if each induced subgraph of G has a vertex of degree at most d . It can be checked in polynomial time by removing a vertex of degree at most $22p$, repeatedly. We now need to consider the two cases:

Case 1. $I(\mathcal{P}_{i,j}, \mathcal{P}_{i',j'})$ is not $22p$ -degenerate for some (i, j, i', j') .

In the case, we have a subgraph H_0 of $I(\mathcal{P}_{i,j}, \mathcal{P}_{i',j'})$ such that $|E(H_0)| > 11p|V(H_0)|$. Then, the following well-known result of Bollobás and Thomason [4] implies that the intersection graph $I(\mathcal{P}_{i,j}, \mathcal{P}_{i',j'})$ contains some graph H as a minor such that $2\delta(H) \geq |V(H)| + 4p - 2$.

Theorem 6.3 (Bollobás–Thomason [4, Lemma 3], see also [3, 36]). *There exists a constant $\alpha > 0.37$ satisfying the following. Let $\ell \geq 3$ be an integer and let G be a simple graph with $|E(G)| > \ell|V(G)|$. Then G contains some simple graph H as a minor such that $|H| \leq \ell + 2$ and $2\delta(H) \geq |V(H)| + \lfloor \alpha \ell \rfloor - 1$.*

We note that, since the proof of [4, Lemma 3] is constructive, such a H -minor can be found in polynomial time. Thus, by applying Theorem 6.3 in which $\ell = 11p$, we can find a H -minor in $G \square K_2$ such that $2\delta(H) \geq |V(H)| + 4p - 2$.

Case 2. $I(\mathcal{P}_{i,j}, \mathcal{P}_{i',j'})$ is $22p$ -degenerate for any (i, j, i', j') .

In this case, we consider the intersection graph I of all the $r = \binom{h}{2}$ sets of the paths, i.e., I is an r -partite graph having a vertex for each path out of $\mathcal{P}_{i,j}$ for $1 \leq i < j \leq h$, and an edge between two vertices if the corresponding paths intersect. Now we can use the following lemma in [27].

For $r \geq 2$, let V_1, \dots, V_r be the color classes in an r -partite graph G_0 . Suppose that $|V_i| \geq 2e(2r-3)d+1$ (where $e = 2.718\dots$) for all $1 \leq i \leq r$, and assume that any subgraph of G_0 induced by $V_i \cup V_j$ is d -degenerate for $1 \leq i < j \leq r$. Then there exists a randomized polynomial-time algorithm to find an independent set $\{x_1, \dots, x_r\}$ of G_0 such that $x_i \in V_i$ for $i = 1, \dots, r$.

By the above lemma in which $d = 22p$, $G_0 = I$, and c is an appropriate constant, we get a clique minor whose nodes correspond to $\{x_1, \dots, x_r\}$.

Furthermore, in either case, if each of the k disjoint paths between T_i and T_j contains a terminal, then every node of the obtained minor contains a terminal by the construction of the minor. \square

We use the H -minor or the K_h -minor obtained in Theorem 6.2 to construct our routing, and give a proof of Theorem 6.1. Our proof of Theorem 6.1 basically follows the same line with that of Proposition 5.1, and we use the following theorem instead of Theorem 5.6.

Theorem 6.4 (Bollobás–Thomason [3, Theorem 3]). *Let $s_1, \dots, s_q, t_1, \dots, t_q$ be the terminals in a given G . If G contains H as a minor, where H is some graph satisfying $2\delta(H) \geq |H| + 4q - 2$, and there is no separation (A, B) of order at most $2q - 1$ in G such that A contains all the terminals and $B - A$ contains at least one node of the H -minor, then there are disjoint paths P_i with two ends in s_i, t_i for $i = 1, \dots, q$.*

We note that the proof of [3, Theorem 3] implies that given the above H -minor, desired disjoint paths can be found in polynomial time. We are now ready to give our proof for Theorem 6.1.

Proof of Theorem 6.1. Let Z be the terminal set. We apply Lemma 5.4 and Theorem 6.2 in which $p = h = \Theta(|Z|^{\frac{1}{4}})$ and $k = \Theta(|Z|^{\frac{3}{4}})$. Then, we have a H -minor with $2\delta(H) \geq |V(H)| + 4p - 2$ in $G \square K_2$ whose each node contains a terminal or a K_h -minor in G whose each node contains a terminal. If we have a K_h -minor, then we can connect h terminal pairs by the same arguments as the proof of Proposition 5.1. Hence, we assume that we have a H -minor with $2\delta(H) \geq |V(H)| + 4p - 2$ in $G \square K_2$.

We take $q = \max\{\lfloor 2p/3 \rfloor, 1\}$ terminal pairs in Z and let Z_0 be the set of such terminals. Note that these terminals are contained in the original graph G . We also note that $2\delta(H) \geq |V(H)| + 4p - 2 \geq |V(H)| + 4q - 2$. In what follows, we show that the terminals can be connected by disjoint paths in $G \square K_2$ by applying Theorem 6.4. Since the case of $q = 1$ is obvious, we suppose that $q = \lfloor 2p/3 \rfloor$.

Assume that there is a separation (A, B) of order at most $2q - 1$ in $G \square K_2$ such that A contains Z_0 and $B - A$ contains at least one node of the H -minor. Since $\delta(H) \geq 4p - 1 + (|V(H)| - \delta(H) - 1) \geq 4p - 1$, at least $\delta(H) - (2q - 1) \geq 4q$ nodes are contained in $B - A$. Then, we have $|(Z \cup Z') \cap V(B - A)| \geq 4q$, where Z' is the copy of Z in G' . Since $v' \in Z' \cap V(B - A)$ implies that the corresponding vertex v is in $Z \cap V(B)$, B contains at least $2q$ terminals in Z . By setting $A_1 = A \cap G$ and $B_1 = B \cap G$, we have a separation (A_1, B_1) of G such that its order is at most $2q - 1$, A_1 contains Z_0 , and B_1 contains $2q$ terminals, which contradicts the definition of well-linked sets.

Therefore, there is no such a separation (A, B) , and hence we can connect Z_0 by q disjoint paths in $G \square K_2$ by Theorem 6.4. These paths correspond to the disjoint paths in G with congestion two, which completes the proof. \square

Finally, we give a proof for Theorem 1.2.

Proof of Theorem 1.2. By Theorem 4.1, we obtain instances $(G_1, \mathcal{T}_1), \dots, (G_r, \mathcal{T}_r)$ with edge-well-linked terminal sets. Let OPT_i be the optimal value of the instance (G_i, \mathcal{T}_i) , then we have $\sum_i \text{OPT}_i \geq \text{OPT}/\beta(n)$, where $\beta = O(\sqrt{\log n \log n})$. After reducing the maximum edge-disjoint paths problem to the maximum vertex-disjoint paths problem by considering the line graph, we apply Theorem 6.1 to obtain $\Omega(\text{OPT}_i^{\frac{1}{4}})$ disjoint paths in each well-linked instance. Then, we can find $\Omega(\text{OPT}^{\frac{1}{4}}/\sqrt{\log n \log n})$ disjoint paths in total. \square

References

- [1] M. Andrews, Approximation algorithms for the edge-disjoint paths problem via Racke decompositions, *Proc. 51st IEEE Symposium on Foundations of Computer Science (FOCS)*, 2010, 277–286.
- [2] M. Andrews, J. Chuzhoy, S. Khanna and L. Zhang, Hardness of the undirected edge-disjoint paths problem with congestion, *Proc. 46th IEEE Symposium on Foundations of Computer Science (FOCS)*, 2005, 226–244.
- [3] B. Bollobas and A. Thomason, Highly linked graphs, *Combinatorica*, **16** (1996), 313–320.
- [4] B. Bollobas and A. Thomason, Proof of a conjecture of Mader, Erdős and Hajnal on topological complete subgraphs, *European Journal of Combinatorics*, **19** (1998), 883–887.
- [5] C. Chekuri and J. Chuzhoy, Polynomial bounds for the grid-minor theorem, *Proc. 46th ACM Symposium on Theory of Computing (STOC)*, 2014, 60–69.
- [6] C. Chekuri, S. Khanna and B. Shepherd, Multicommodity flow, well-linked terminals, and routing problems, *Proc. 37th ACM Symposium on Theory of Computing (STOC)*, 2005, 183–192.
- [7] C. Chekuri, S. Khanna and B. Shepherd, An $O(\sqrt{n})$ approximation and integrality gap for disjoint paths and unsplittable flow, *Theory of Computing*, **2** (2006), 137–146.
- [8] C. Chekuri, S. Khanna and B. Shepherd, The all-or-nothing multicommodity flow problem, *SIAM Journal on Computing*, **42** (2013), 1467–1493.
- [9] J. Chuzhoy and S. Li, A polylogarithmic approximation algorithm for edge-disjoint paths with congestion 2, *Proc. 53rd IEEE Symposium on Foundations of Computer Science (FOCS)*, 2012, 233–242.
- [10] R. Diestel, *Graph Theory, 3rd Edition*, Springer-Verlag, 2005.
- [11] R. Diestel, T.R. Jensen, K.Y. Gorbunov and C. Thomassen, Highly connected sets and the excluded grid theorem, *J. Combin. Theory Ser. B*, **75** (1999), 61–73.
- [12] S. Even, A. Itai and A. Shamir, On the complexity of timetable and multicommodity flow problems, *SIAM Journal on Computing*, **5** (1976), 691–703.

- [13] 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.
- [14] V. Guruswami, S. Khanna, R. Rajaraman, B. Shepherd, and M. Yannakakis, Near-optimal hardness results and approximation algorithms for edge-disjoint paths and related problems, *J. Comp. Syst. Science*, **67** (2003), 473–496.
- [15] K. Kawarabayashi and B. Reed, A nearly linear time algorithm for the half integral disjoint paths packing, *Proc. ACM-SIAM Symposium on Discrete Algorithms (SODA)*, 2008, 446–454.
- [16] K. Kawarabayashi and Y. Kobayashi, An improved algorithm for the half-disjoint paths problem, *SIAM J. Discrete Mathematics*, **25** (2011), 1322–1330.
- [17] K. Kawarabayashi and Y. Kobayashi, Breaking $O(n^{1/2})$ -approximation algorithms for the edge-disjoint paths problem with congestion two, *Proc. 43rd ACM Symposium on Theory of Computing (STOC)*, 2011, 81–88.
- [18] K. Kawarabayashi and Y. Kobayashi, Linear min-max relation between the treewidth of H -minor-free graphs and its largest grid minor, *Proc. 29th Symposium on Theoretical Aspects of Computer Science (STACS)*, 2012, 278–289.
- [19] K. Kawarabayashi and Y. Kobayashi, An $O(\log n)$ -approximation algorithm for the disjoint paths problem in Eulerian planar graphs, *ACM Transactions on Algorithms*, **9** (2013), Article 16.
- [20] K. Kawarabayashi, Y. Kobayashi and B. Reed, The disjoint paths problem in quadratic time, *J. Combin. Theory Ser. B*, **102** (2012), 424–435.
- [21] J. Kleinberg, Decision algorithms for unsplittable flow and the half-disjoint paths problem, *Proc. 30th ACM Symposium on Theory of Computing (STOC)*, 1998, 530–539.
- [22] J. Kleinberg, An approximation algorithm for the disjoint paths problem in even-degree planar graphs, *Proc. 46th IEEE Symposium on Foundations of Computer Science (FOCS)*, 2005, 627–636.
- [23] J. Kleinberg and É. Tardos, Disjoint paths in densely embedded graphs, *Proc. 36th IEEE Symposium on Foundations of Computer Science (FOCS)*, 1995, 52–61.
- [24] J. Kleinberg and É. Tardos, Approximations for the disjoint paths problem in high-diameter planar networks, *J. Comp. Syst. Science*, **57** (1998), 61–73.
- [25] S. Kolliopoulos and C. Stein, Improved approximation algorithms for unsplittable flow problems, *Proc. 38th IEEE Symposium on Foundations of Computer Science (FOCS)*, 1997, 426–435.
- [26] 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.
- [27] S. Kreutzer and S. Tazari, On brambles, grid-like minors, and parameterized intractability of monadic second-order logic, *Proc. ACM-SIAM Symposium on Discrete Algorithms (SODA)*, 2010, 354–364.

- [28] P. Raghavan and C.D. Thompson, Randomized rounding: A technique for provably good algorithms and algorithmic proofs, *Combinatorica*, **7** (1987), 365–374.
- [29] S. Rao and S. Zhou, Edge disjoint paths in moderately connected graphs, *SIAM J. Computing*, **39** (2010), 1856–1887.
- [30] 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.
- [31] B. Reed and D. Wood, Polynomial treewidth forces a large grid-like minor, *European Journal of Combinatorics*, **33** (2012), 374–379.
- [32] N. Robertson and P.D. Seymour, Graph minors. XIII. The disjoint paths problem, *J. Combin. Theory Ser. B*, **63** (1995), 65–110.
- [33] N. Robertson, P.D. Seymour and R. Thomas, Quickly excluding a planar graph, *J. Combin. Theory Ser. B*, **62** (1994), 323–348.
- [34] A. Schrijver, *Combinatorial Optimization: Polyhedra and Efficiency*, number 24 in Algorithm and Combinatorics, Springer-Verlag, 2003.
- [35] A. Srinivasan, Improved approximations for edge-disjoint paths, unsplittable flow, and related routing problems, *Proc. 38th IEEE Symposium on Foundations of Computer Science (FOCS)*, 1997, 416–425.
- [36] A. Thomason, An extremal function for contractions of graphs, *Math. Proc. Cambridge Philos. Soc.*, **95** (1984), 261–265.