Analysis of an $O(\sqrt{n})$-Approximation Algorithm for the Maximum Edge-Disjoint Paths Problem with Congestion Two

Nabiha Asghar
Department of Combinatorics & Optimization
University of Waterloo, Ontario, Canada

1. BACKGROUND

1.1 Edge Disjoint Paths Problem

The edge-disjoint paths problem (EDPP) has been studied for a long time by graph theorists and algorithm developers for combinatorial optimization. In its most basic form, the EDPP is as follows: given a graph $G = (V, E)$ and a set of l pairs of vertices/terminals (s_i, t_i) for $i \in [l]$ in G, decide whether or not G has l edge-disjoint paths connecting the given pairs of terminals. This problem is equivalent to finding vertex-disjoint paths connecting a given set of terminal pairs in a graph, because it is 'dual' to EDPP (by considering the line graph).

The generic EDPP is NP-complete but it has many variants, some of which are solvable to optimality in polynomial time (for example, an $O(n^3)$ algorithm is known through Robertson and Seymour for the case where l is fixed and is not part of the input. Here n is the number of vertices in the graph). One spinoff of EDPP is the maximum edge-disjoint paths problem, which has many useful applications in network flow theory such as vehicle routing and data transmission modeling. The maximum edge-disjoint paths problem will be the focus of this report.

1.2 Maximum Edge Disjoint Paths Problem

The maximum edge disjoint paths problem (MEDPP) is as follows:

Given an undirected graph $G = (V, E)$ and a set of l pairs of vertices (s_i, t_i) for $i \in [l]$ in $V(G)$, find the maximum number of pairs that can be routed by edge-disjoint paths (abbreviated EDPs).

This problem is also NP-complete, as expected. Hence the main target of recent research on this problem is to develop efficient polynomial time approximation algorithms for it. For restricted classes of graphs, such as planar graphs, trees, meshes and highly connected graphs, there exist algorithms with constant factor and poly-logarithmic approximation guarantees. However for the generic case, the best possible approximation to date is due to a seminal work by Chekuri, Khanna and Shepherd [6], who have given an $O(\sqrt{n})$ approximation algorithm.

There exist some important hardness results for MEDPP. For directed graphs, it is known that no polynomial time algorithm can achieve an approximation guarantee of $O(m^{\frac{1}{2}-\varepsilon})$ for any $\varepsilon > 0$, where m denotes the number of edges in the given graph, unless $P = NP$ [10]. For the undirected case, the strongest result known so far says that no polynomial time algorithm can achieve an approximation guarantee of $\Omega((\log m)^{\frac{1}{2}-\varepsilon})$ for any $\varepsilon > 0$ [1].

Given these hardness results, many researchers are now focusing on MEDPP with the allowance of having a non-unitary 'congestion' on the edges of G. A graph G is said to have congestion γ if each edge can be used up to γ times in the EDPs. This scenario occurs in real world applications where there is a need to find a unit unsplittable flow between each terminal pair, given that each edge in between has the capacity of carrying a load of γ. As far as the approximation of MEDPP with congestion γ is concerned, there exists an $O(n^{\frac{1}{\gamma}})$- approximation algorithm for congestion γ [17]. There also exists an $O(1)$-approximation algorithm via randomized rounding if the congestion is allowed to be $O(\frac{\log n}{\log(\log n)})$ [15]. However, a hardness result has been shown which asserts that no polynomial time algorithm can achieve an approximation guarantee of $O(log^{\frac{1}{\gamma+1}-\varepsilon} n)$ for any $\varepsilon > 0$ with congestion γ (up to $o(\frac{\log (\log \text{n})}{\log(\log (\log \text{n}))})$), unless $P = NP$ [1].

1.3 Goal

Up to 2011, the best known algorithm for MEDPP with congestion two was an $O(\sqrt{n})$ approximation algorithm proposed in 2006 [6]. The goal of this report is to present an $O(n^{\frac{3}{7}}\, poly(\log n))$-approximation algorithm for MEDPP with congestion two (recently proposed by Kawarabayashi and Kobayashi [12]) which breaks this previously known bound of $O(\sqrt{n})$. For the ease of readability, the problem and the goal are restated below:

MEDPP with Congestion Two: Given an undirected graph $G = (V, E)$ and a set of l pairs of vertices in $V(G)$, find the maximum number of pairs that can be routed by edge-disjoint paths (EDPs) with congestion two.

Goal: Present an $O(n^{\frac{3}{7}}\, poly(\log n))$-approximation algorithm for MEDPP with congestion two.

1.4 Report Layout

The proof of the claimed approximation-ratio consists of two theorems and a corollary, stated below.
Let OPT be the cost of optimal solution of MEDPP when the congestion is at most 1.

Theorem 1: Given an instance of MEDPP, we can find $\Omega\left(\frac{OPT^{\frac{1}{4}}}{\beta(n)}\right)$ EDPs between the terminal pairs with congestion two in polynomial time, where β is a poly-logarithmic function.

(Idea: Decompose the graph into highly connected subgraphs. For each subgraph, since it is highly connected, we can find a complete graph as a minor and route the paths through this clique minor with congestion two.)

Theorem 2: Given a randomized polynomial time algorithm for finding $\Omega\left(\frac{OPT^{\frac{1}{p}}}{\beta(n)}\right)$ EDPs between given terminal pairs for some $p > 1$, where β is a poly-logarithmic function, there is a polynomial time randomized $O(n^{\frac{p-1}{2p-1}} poly(\log n))$- approximation algorithm for the MEDPP.

(Idea: Reduce the graph till achieves a certain level of edge-connectivity. Then apply a well-known result by Rao and Zhou [16] to find $\frac{OPT}{O(log^{10} n)}$ edge-disjoint paths in the reduced graph. Use these paths to find $\frac{OPT}{O(n^{\frac{p-1}{2p-1}} \text{poly}(\log n))}$ edge-disjoint paths with congestion two in the original graph.)

Given these two theorems, the main result of this report is a straight-forward consequence:

Corollary 3: There exists a randomized polynomial time $O(n^{\frac{3}{7}} poly(\log n))$-approximation algorithm for MEDPP with congestion two.

Proof: Given any instance of MEDPP, Theorem 1 guarantees the existence of $\Omega(OPT^{\frac{1}{4}}/\beta(n))$ EDPs with congestion two, by giving a polynomial time algorithm to do so. Given this algorithm, Theorem 2 guarantees the existence of a randomized polynomial time $O(n^{\frac{3}{7}} poly(\log n))$-approximation algorithm for MEDPP. This proves the claimed approximation-ratio. ■

The layout of the report is as follows: Section 2 gives the necessary definitions and previously known results needed for proofs of Theorem 1 and 2. Section 3 and section 4 give the detailed proofs of Theorem 1 and Theorem 2 respectively. Section 5 gives some concluding remarks.

2. DEFINITIONS & KNOWN RESULTS

This section states the definitions and known results needed for the proofs of Theorem 1 and Theorem 2.

2.1 Definitions

- Given a graph G, its line graph is a graph $L(G)$ such that each vertex of $L(G)$ represents an edge of G, and two vertices of $L(G)$ are adjacent if and only if their corresponding edges in G share a common endpoint in G.
- A set of vertices Z in G is well-linked if, for every set S containing at most half of Z, $|N(S)| \geq |S \cap Z|$. Here, $N(S)$ denotes the set of vertices of G which share an edge with a vertex of S.
- A subcubic tree is a tree which has maximum degree at most three.
- A separation (A, B) of a graph G is a pair of disjoint induced subgraphs A and B 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 $|V(A) \cap V(B)|$.
- A k-web of order h in a given graph G is a set of h disjoint trees $T_1, \dots\dots, T_h$ such that for any distinct i, j, there is a set of k vertex-disjoint paths connecting T_i and T_j.

- The Cartesian Product $G_1 \boxdot G_2$ of two graphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ is the graph with vertex set $V^* = \{(v_1, v_2) \mid v_1 \in V_1, v_2 \in V_2\}$ and an edge between $(u_1, u_2) \in V^*$ and $(v_1, v_2) \in V^*$ exists if and only if either $u_1 = v_1$ and $u_2 v_2 \in E_2$, or $u_2 = v_2$ and $u_1 v_1 \in E_1$. In particular, $G \boxdot K_2$ consists of $G = (V, E)$, its copy $G' = (V', E')$, and edges each connecting one vertex in V and its corresponding vertex in V'.
- A graph G has a grid-like minor of order r if the graph $G \boxdot K_2$ contains a K_r as a minor (called a clique minor of order r). *Note: This definition varies slightly from the one presented in the original paper. This is to avoid the discussion on 'half integral minors' introduced by the authors in the beginning of Section 5.2.*

2.2 Known Results

CKS-Theorem (Chekuri, Khanna and Shepherd [5]): For an input graph G with the set of terminal pairs $\mathcal{T}$, one can compute vertex-disjoint subgraphs $G_1, \dots, G_r$ and their corresponding disjoint sets of vertex pairs $\mathcal{T}_1, \dots, \mathcal{T}_r$ of $\mathcal{T}$ such that the following hold:

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

KT-Theorem (Kreutzer-Tazari [14]): 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 K_p minor in $G \boxdot K_2$ or a K_h minor in G. Furthermore, if each of the k vertex-disjoint paths between T_i and T_j contains a terminal for any distinct i, j, then every node of the obtained minor contains a terminal or its copy.

RS-Theorem (Robertson-Seymour [9]): Let $s_1, \dots, s_q$ and $t_1, \dots, t_q$ 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 vertex-disjoint paths P_i with two ends in s_i, t_i for $i = 1, \dots, q$. Furthermore, given the above clique minor, the desired disjoint paths can be found in $O(qm)$ time.

Rao-Zhou Theorem (Rao-Zhou [16]): For some constant c, a randomized $O(log^{10}n)$-approximation algorithm exists for the MEDPP in a graph with edge-connectivity at least $clog^5 n$.

3. PROOF OF THEOREM 1

In this section, for the sake of ease in readability and understanding, we prove the following weaker version of Theorem 1:

Theorem 1.1: Given an instance of MEDPP, we can find $\Omega\left(\frac{OPT^{1/5}}{\beta(n)}\right)$ EDPs connecting the terminal pairs with congestion two in polynomial time.

Section 3.1 gives the main idea and sketch of the proof of Theorem 1.1. Section 3.2 gives the full proof of Theorem 1.1. Section 3.3 gives an insight into how the proof of Theorem 1.1 can be modified to give the proof of Theorem 1. Section 3.4 gives an overview of the algorithm behind the CKS-Theorem.

3.1. Proof Sketch of Theorem 1.1

Suppose we are given an instance of MEDPP. The goal is to find $\Omega\left(\frac{OPT^{1/5}}{\beta(n)}\right)$ EDPs with congestion two in polynomial time. Since the MEDPP can be reduced to the Maximum Vertex-Disjoint Paths Problem (MVDPP) by considering the line graph of the given instance, it suffices to give an algorithm to find $\Omega\left(\frac{OPT^{\frac{1}{5}}}{\beta(n)}\right)$ vertex-disjoint paths (abbreviated VDPs) in polynomial time such that each vertex is used in at most two paths. We can assume that our graph is free of loops and multi edges since they do not affect the number of VDPs in a graph.

We first prove the following proposition:

Proposition 1.1: In a well-linked instance (G, Z) of MVDPP (this means Z is the set of terminal pairs in G and is a well-linked set in G), we can find $\Omega\left(OPT^{\frac{1}{5}}\right)$ VDPs with congestion two in polynomial time.

<u>Proof sketch of Proposition 1.1:</u>

1) Let h, k be positive integers such that $h = \Theta(|\mathcal{T}_i|^{1/5})$, $k = \Theta(|\mathcal{T}_i|^{4/5}$ and let $|Z| \geq 8hk.$ Construct a k-web of order h (i.e. a set of disjoint trees $T_1, \ldots\ldots, T_h$ such that each T_i contains at-least k vertices in Z and there are k VDPs connecting $T_i \cap Z$ and $T_j \cap Z$ in G for any i, j). (Lemma 3.1, 3.2 and 3.3)
2) Use the k-web of order h to construct a grid-like minor M of order $\Omega(|Z|^{1/5})$, which is attached to terminals in Z in polynomial time. (KT-Theorem)
3) Now construct VDPs (with congestion two) between the pairs of terminals in Z via M in $G.$ (RS-Theorem).

This completes the proof of Proposition 1.1.

Given that Proposition 1.1 holds, Theorem 1.1 is proved as follows: Decompose the graph G into well-linked instances $(G_i, \mathcal{T}_i)$ via CKS-Theorem (as depicted in the figure below). Apply Proposition 1.1 to each $(G_i, \mathcal{T}_i)$ and then 'combine' the result for the original graph G by using the inequality $\sum_i OPT_i \geq \frac{OPT}{\beta(n)}$ given to us by CKS-Theorem (where OPT_i is the optimal value for $(G_i, \mathcal{T}_i)$). With the help of some easy calculus, it follows that we can find $\Omega\left(\frac{OPT^{1/5}}{\beta(n)}\right)$ VDPs in G.

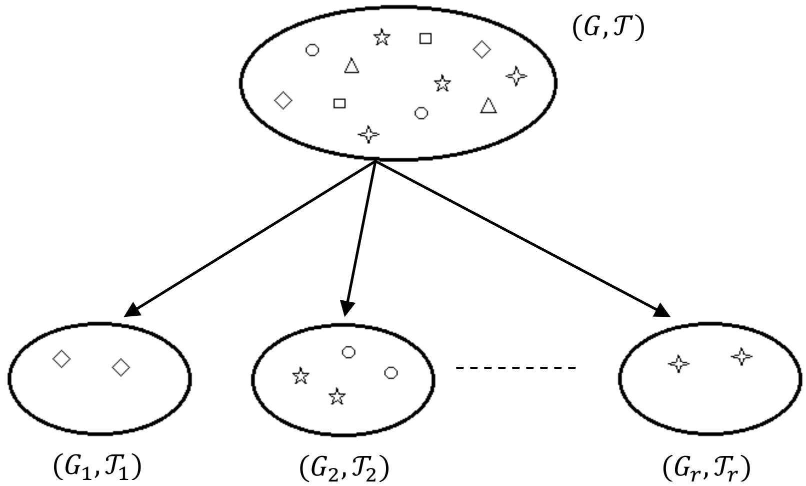


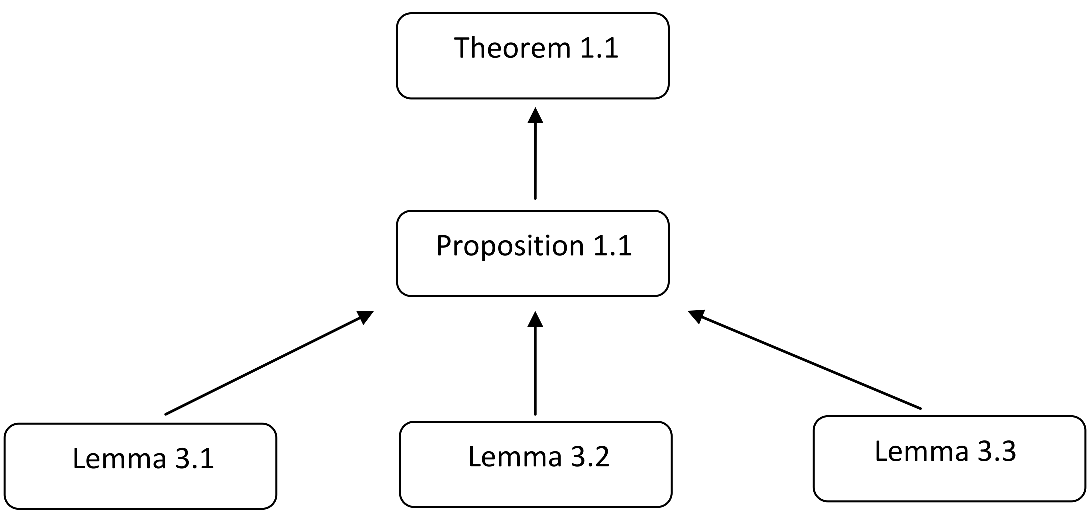
Fig: CKS Decomposition

The graph G with the set of terminal pairs $\mathcal{T}$ (each pair of similar shapes depicts one terminal pair) is decomposed into disjoint subgraphs $(G_i, \mathcal{T}_i)$ such that for each i, $\mathcal{T}_i$ is a well-linked set in G_i. Moreover, $\sum_{i=1}^{r} |\mathcal{T}_i| \geq \frac{OPT}{\beta(n)}$.

This completes the proof sketch of Theorem 1.

3.2. Proof of Theorem 1.1

To prove Theorem 1.1, we first give three lemmas on the properties of well-linked sets. Then using KT-Theorem and RS-Theorem, we give the proof of Proposition 1.1. This is followed by proof of Theorem 1.1. This is depicted in the flow chart below:



Lemma 3.1: Let q be an integer and Z be a well-linked set with $|Z| \geq 4q$. For disjoint subsets $X_1, X_2 \subseteq Z$ with $|X_1| = |X_2| = q$, there exist q VDPs connecting X_1 and X_2.

Proof: By way of contradiction, suppose there are no q VDPs connecting X_1 and X_2. Then there exists a separation (A, B) of order at most $q-1$ in G such that $X_1 \subseteq V(A)$ and $X_2 \subseteq V(B)$.

Sub-claim: Both $A - B$ and $B - A$ contain at most $\frac{|Z|}{2}$ vertices of Z.

Proof: Suppose $A - B$ contains more than $\frac{|Z|}{2}$ vertices of Z. Let $\overline{Z}$ be exactly half of the Z vertices in $A - B$ and let $\overline{A} = A - B - Z\backslash\overline{Z}$. Now $X_2 \subseteq V(B)$ implies $\left|(Z\backslash\overline{Z}) \cap V(B)\right| \geq q$ and the order of separation (A, B) implies $|V(A \cap B)| \leq q - 1$. Putting these two inequalities together, we have $\left|(Z\backslash\overline{Z}) \cap\right.$ *VB≥q≥VA∩B+1.* Now *A* contains exactly half of the vertices of *Z*, hence its neighborhood must have size at least $\frac{|Z|}{2}$, due to the well-linkedness of Z. However,

$$\begin{aligned}
\left|N(\overline{A})\right| &\leq \left|(Z\backslash\overline{Z}) \cap V(A-B)\right| + |V(A \cap B)| \\
&\leq \left|(Z\backslash\overline{Z}) \cap V(A-B)\right| + \left|(Z\backslash\overline{Z}) \cap V(B)\right| - 1 \\
&\leq |(Z\backslash\overline{Z})| - 1 \\
&= \frac{|Z|}{2} - 1,
\end{aligned}$$

which is a contradiction to the fact that Z is a well-linked set. So $A - B$ contains at most $\frac{|Z|}{2}$ vertices of Z. By symmetry, the same holds for $B - A$. This completes the proof of sub-claim. □

We may assume that one of $A - B$ and $B - A$ contains at least $\frac{|Z|}{2} - q$ vertices of Z. (If not, then both contain less than $\frac{|Z|}{2} - q$ vertices of Z. So $|V(A-B)| + |V(B-A)| < |Z| - 2q$, which implies at least $2q$ vertices of Z lie in $V(A \cap B)$. This contradicts the fact that $|V(A \cap B)| \leq q - 1$.) So say $B - A$ has at least $\frac{|Z|}{2} - q$ and at most $\frac{|Z|}{2}$ vertices of Z. Since Z is a well-linked set, the neighborhood of $B - A$ should have size at least $\frac{|Z|}{2} - q$. However,

$$\begin{aligned}
|N(B-A)| &\leq q - 1 \\
&< \frac{4q}{2} - q \\
&\leq \frac{|Z|}{2} - q,
\end{aligned}$$

a contradiction. ■

Lemma 3.2: Given a well-linked set Z in a given graph G, there exists a subcubic tree T which contains a subset Z' of Z such that $|Z'| \geq \frac{|Z|}{4}$, and there are at most $|Z'| - 1$ vertices of degree three in T. Moreover, given a well-linked set Z, there is a polynomial time algorithm to construct such a subcubic tree T and a vertex set $Z' \subseteq Z$.

Proof: The set Z' and tree T are constructed inductively, by constructing sets $Z_i \subseteq Z$ and trees T_i such that $Z_1 \subset Z_2 \subset \cdots \subset Z_i \subseteq V(T_i)$ and there are at most $|Z_i| - 1$ vertices of degree three in T_i. The process is stopped when for some i we get $|Z_i| \geq \frac{|Z|}{4}$. We set $Z' = Z_i$ and output $T = T_i$.

Base Case: Choose $u, v \in Z$. Since Z is well-connected, by Lemma 3.1 there exists a path between u and v. Set T_1 to be this path and $Z_1 = T_1 \cap Z$. Clearly there are no vertices of degree 3 in T_1. Hence the conditions are satisfied so the base case holds.

Inductive Step: Suppose Z_i and T_i have been found, with $|Z_i| < \frac{|Z|}{4}$. Let C_i be the set of vertices of degree 3 in T_i, so $|C_i| \leq |Z_i| - 1$. We need to construct Z_{i+1} and T_{i+1} and show that $|C_{i+1}| \leq |Z_{i+1}| - 1$.

Since $|Z_i| < \frac{|Z|}{4}$, we have that $|Z - Z_i| > |Z_i|$. By Lemma 3.1, there exist $|Z_i|$ VDPs between Z_i and $Z - Z_i$. One of these paths, call it P, avoids the vertices in C_i, since $|C_i| \leq |Z_i| - 1$. Follow P from $w \in Z - Z_i$ to the first vertex in $V(T_i \cap P)$ that it hits, say x. Let Q be the subpath of P between w and x. Set $T_{i+1} = T_i \cup Q$ and $Z_{i+1} = Z_i \cup (Q \cap Z)$. It is clear that T_{i+1} is a tree because we did not create any cycle by adding the path Q to T_i. Also, Q did not hit any vertices of C_i so there are no vertices of degree ≥ 4 in T_{i+1}. Moreover, since x is the only vertex in T_{i+1} that could have degree 3 in $T_{i+1} - V(C_i)$, it follows that $C_{i+1} \leq |C_i| + 1 \leq |Z_i| \leq |Z_{i+1}| - 1$. Thus T_{i+1} and Z_{i+1} satisfy our conditions.
This procedure takes polynomial time because there are at most $\frac{|Z|}{4} = O(n)$ iterations, and in each iteration we need to find $|Z_i|$ disjoint paths which can be accomplished in $O(m)$ time. Hence it is a $O(nm)$-time procedure. ■

Lemma 3.3: Let h and k be positive integers. Given a well-linked set Z with $|Z| \geq 8hk$, 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 VDPs connecting $T_i \cap Z$ and $T_j \cap Z$ in G for any distinct i, j.

Proof: By virtue of Lemma 3.2, we obtain a subcubic tree T and a vertex set $Z' \subseteq Z$ such that $Z' \subseteq V(T)$ and $|Z'| \geq \frac{|Z|}{4} \geq 2hk$.
Consider the following well-known text book result [8, Lemma 12.4.6]:

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.
Proof: If $|X| < 2k$, then taking $F = \phi$ satisfies our requirements so we are done. So assume $|X| \geq 2k$. Choose $e \in E(T)$ such that some component T' of $T - e$ contains at least k vertices of X and $|T'|$ is as small as possible. Finding such an edge e takes polynomial time. Then the end of e in T' has degree at most two in T' because T is subcubic. The minimality of T' implies $|X \cap V(T')| < 2k$. Add e to F and recursively apply this procedure to $T - T'$. Since the number of edges is finite, the iterations stop and we obtain our required set F in polynomial time.

If we let $X = Z$ in this result, then the above procedure takes at least h iterations. Hence in polynomial time we can find a set F of edges in T such that there are h subtrees $T_1, \dots, T_h$ of $T - F$ and $k \leq |V(T_i) \cap Z| \leq 2k - 1$ for all $i \in [h]$. This means $V(T_i) \cap Z$ contains at most a quarter of the vertices of $|Z|$ (since $h \geq 1$). So by Lemma 3.1, there exist k VDPs between $V(T_i) \cap Z$ and $V(T_j) \cap Z$ for any distinct i, j. ■

We now present the proof of Proposition 1.1.

Proof of Proposition 1.1:

Take the well-linked instance (G, Z). Set $k = \Theta(|Z|^{\frac{4}{5}})$ and $p = h = \Theta(|Z|^{\frac{1}{5}})$. Apply Lemma 3.2 to get a k-web of order h. By KT-Theorem, we have a K_p minor in $G \boxdot K_2$ or a K_h minor in G. Since a K_h minor in G is also a K_h minor in $G \boxdot K_2$ and $p = h$, we can say that we have a K_p minor, say M, in $G \boxdot K_2$.

Moreover, there are k VDPs between $V(T_i) \cap Z$ and $V(T_j) \cap Z$ for any distinct i, j, so by virtue of KT-Theorem each node of M contains a terminal or its copy.
Let $q = \left\lfloor \frac{p}{6} \right\rfloor$ and let Y be the set of these q terminal pairs in Z. Let G' be the copy of G in $G \boxdot K_2$.

Claim 3.4: In $G \boxdot K_2$, there is no separation (A, B) of order at most $2q - 1$ such that A contains Y and $B - A$ contains at least one node of M.
Proof: By way of contradiction, suppose such a separation (A, B) exists. Since $B - A$ contains at least one node v of M and v has $p - 1$ neighbors in M, v can have at most $2q - 1$ of these neighbors in $V(A \cap B)$. Hence, $p - 1 - (2q - 1) = p - 2q \geq 6q - 2q = 4q$ vertices of M are contained in $B - A$. Since each node of M contains a terminal or its copy, it follows that $|(Z \cup Z') \cap V(B - A)| \geq 4q$, where Z' is the copy of Z in G'. Let $u \in V(G)$ and u' its corresponding vertex in $V(G')$. By definition, u and u' are connected by an edge in $G \boxdot K_2$. Observe that $u' \in Z'$ implies $u \in Z$ and $u' \in V(B - A)$ implies $u \in V(B)$. So $u' \in Z' \cap V(B - A)$ implies $u \in Z \cap V(B)$. Therefore, $|Z \cap V(B)| \geq 2q$. 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 A_1 contains q terminals in Z (since it contains Y), and B_1 contains $2q$ terminals in Z. By Lemma 3.1, there exist q VDPs connecting A_1 and B_1. But this is not possible, because order of (A_1, B_1) is at most $2q - 1$ (since order of (A, B) is at most $2q - 1$). Hence we have arrived at a contradiction. This completes the proof of Claim 3.4 □

Now apply RS-Theorem to $G \boxdot K_2$ with Y as the set of terminals. In polynomial time, we obtain $q = \Omega\left(|Z|^{\frac{1}{5}}\right)$ VDPs connecting Y in $G \boxdot K_2$. Clearly these paths correspond to $\Omega\left(|Z|^{\frac{1}{5}}\right)$ VDPs in G with congestion two (because a vertex w in G and its corresponding vertex w' in G' may be used in two different VDPs in $G \boxdot K_2$). So we have found $\Omega\left(|Z|^{\frac{1}{5}}\right) \geq \Omega\left(OPT^{\frac{1}{5}}\right)$ VDPs between the terminal pairs in G with congestion two in polynomial time.
This completes the proof of Proposition 1.1. ■

Finally, we present the proof of Theorem 1.1.

Proof of Theorem 1.1:

Take the input instance of MVDPP, and apply the CKS-Theorem to obtain vertex-disjoint subgraphs $G_1, \ldots, G_r$ and their corresponding disjoint sets of terminal pairs $\mathcal{T}_1, \ldots, \mathcal{T}_r$. Now, each $(G_i, \mathcal{T}_i)$ is a well-linked instance of size $\geq 8hk$ for positive integers $h = \Theta(|\mathcal{T}_i|^{1/5})$ and $k = \Theta(|\mathcal{T}_i|^{4/5})$. Let OPT_i be the optimal value of $(G_i, \mathcal{T}_i)$. Then we have $\sum_i OPT_i \geq \frac{OPT}{\beta(n)}$ for $i \in [r]$, where β is a poly-logarithmic function.

Claim 3.5: $(\sum_{i=1}^{r} OPT_i)^{\frac{1}{5}} \leq \sum_{i=1}^{r} (OPT_i^{\frac{1}{5}})$
Proof: We proceed by strong induction.
Base case: For $i = 1$ the claim is trivially true. For $i = 2$:
$(OPT_1^{\frac{1}{5}} + OPT_2^{\frac{1}{5}})^5 = OPT_1 + OPT_2 +$ some non-negative terms (since $OPT_1, OPT_2 \geq 0$)
$\geq OPT_1 + OPT_2$
Hence $(OPT_1 + OPT_2)^{\frac{1}{5}} \leq (OPT_1^{\frac{1}{5}} + OPT_2^{\frac{1}{5}})$. So the base case holds.
Inductive Step: Suppose the inequality is true for $i \leq r$. We need to prove it for $i = r + 1$.

$\left(\sum_{i=1}^{r+1} OPT_i\right)^{\frac{1}{5}} \leq \left(\sum_{i=1}^{r} OPT_i\right)^{\frac{1}{5}} + \left(OPT_{r+1}\right)^{\frac{1}{5}}$, by base case

$$\leq \sum_{i=1}^{r}\left(OPT_i^{\frac{1}{5}}\right) + \left(OPT_{r+1}\right)^{\frac{1}{5}} \text{ , by inductive hypothesis}$$
$$= \sum_{i=1}^{r+1}\left(OPT_i^{\frac{1}{5}}\right)$$

This completes the inductive step, and completes the proof of Claim 3.5. □

We can find $\Omega(OPT_i^{1/5})$ VDPs in each instance $(G_i,\ \mathcal{T}_i)$ by Proposition 1.1. So in all the instances $(G_i,\ \mathcal{T}_i)$, we can find $\sum_i \Omega(OPT_i^{\frac{1}{5}})$ VDPs in total.

Now $\sum_i \Omega(OPT_i^{\frac{1}{5}}) \geq (\sum_i \Omega(OPT_i))^{\frac{1}{5}} \geq \frac{OPT^{\frac{1}{5}}}{\beta(n)}$ by Claim 3.5, hence we can find $\Omega\left(\frac{OPT^{1/5}}{\beta(n)}\right)$ VDPs in total. This completes the proof of Theorem 1.1. ■

3.3. Extending Theorem 1.1 to Theorem 1

We briefly discuss here how to extend Theorem 1.1 to Theorem 1 (i.e. improving the number of paths from $\Omega(OPT^{1/5}/\beta(n))$ to $\Omega(OPT^{1/4}/\beta(n))$). The idea is to use a graph with large minimum degree instead of a clique minor. We begin with some definitions.

Definitions:

- Let $\mathcal{P}_1$ and $\mathcal{P}_2$ be a set of disjoint connected subgraphs in a given graph G. 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 $\mathcal{P}_2$ defined, 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.
- The minimum degree $\delta(G)$ of a graph G is the degree of the vertex which has the least number of edges incident to it.

Proof Sketch of Theorem 1:

The idea is to first use the following lemma to obtain a minor with large minimum degree which can be used to do the routing:

Lemma 3.6: Let G is a graph. Given a k-web of order h in G with $k \geq ch^2p$ for some constant c, there exists a randomized polynomial time algorithm to find either a half-integral H- minor, where H is a graph satisfying $2\delta(H) \geq |V(H)| + 4p - 2$, or a K_h minor in G. Furthermore, if each of the k disjoint paths between T_i and T_j contains a terminal for distinct i, j, then every node of the obtained minor contains a terminal.

The proof of Lemma 3.6 mostly relies on a well known result by Bollobás & Thomason [3, Lemma 3] and we will not say more about it here due to brevity of space.

To prove Theorem 1, first the following proposition is proved:

Proposition 1.2: In a well-linked instance (G, Z) of MVDPP, we can find $\Omega\left(OPT^{\frac{1}{4}}\right)$ VDPs with congestion two in polynomial time.

Proof Sketch: Let Z be the terminal set. Apply Lemma 3.3 and Lemma 3.6 with $p = h = \Theta(|Z^{\frac{1}{4}}|)$ and $k = \Theta(|Z^{\frac{3}{4}}|)$. Then we have an H- minor with $2\delta(H) \geq |V(H)| + 4p - 2$ in $G \boxdot K_2$ 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 in G, then we can connect h terminal pairs by the same arguments as in the proof of Proposition 1.1. Hence we may assume that we have an H- minor with $2\delta(H) \geq |V(H)| + 4p - 2$ in $G \boxdot K_2$.

Take $q = \left\lceil\frac{2p}{3}\right\rceil$ terminals in Z and let Y be the set of these terminals. It can be shown (by way of contradiction) that 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 (if such a separation exists, it contradicts Lemma 3.1 in a way very similar to the one we saw in Claim 3.4).

Now use the following theorem (in contrast to the RS-Theorem in the proof of Proposition 1.1):

BT-Theorem (Bollobás-Thomason [2, Theorem 3]): Let $s_1, \ldots \ldots, s_q$ and $t_1, \ldots \ldots, t_q$ be terminals in a given graph G. If G contains H as a minor, where H is some graph satisfying $2\delta(H) \geq |H| + 4p - 2$, 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 VDPs P_i with two ends in s_i, t_i for $i = 1, \ldots, q$.

So, we can connect Y by q VDPs in $G \boxdot K_2$ in polynomial time, which correspond to q VDPs in G with congestion two. □

Given that Proposition 1.2 holds, Theorem 1 is proved in the same fashion as Theorem 1.1 was proved via Proposition 1.1 (by CKS-Theorem).

3.4. Brief Discussion on CKS-Theorem

In this section, we give a brief sketch of the algorithm behind the CKS-Theorem because it is an essential ingredient of the proof of Theorem 1.1 and it is worthwhile to look into its details. The sketch we provide is fairly technical, and only highlights the main steps of the algorithm. Here, we consider a slightly different and more general form of the theorem which considers well-linkedness of a set in terms of (multi-commodity) flow, instead of cuts or size of neighborhoods. (It is safe to do so, because the algorithms for the two cases are quite similar). We begin with some definitions and then state the theorem to be proved, followed by a sketch of the decomposition algorithm.

Definitions

- Let $G = (V, E, c)$ be a capacitated graph, where c is an integer capacity function on nodes (in the context of this report, $c = 1$). Let $\mathcal{T} = \{(s_1, t_1), (s_2, t_2), \ldots, (s_k, t_k)\}$ be a set of k source-sink pairs, where the i'th pair has a non-negative demand d_i associated to it. Let $d =$

$[d_1, \dots, d_k]$. Then the multicommodity flow problem is to find the maximum number of paths that can route flow between the sources and their respective sinks such that the demand at each sink is satisfied, flow is conserved at every node and node capacities are not violated. The maximum concurrent flow for a given instance is the largest λ such that λd can be feasibly routed in G. The sparsity of a cut is the ratio of the capacity of the cut to the demand separated by the cut. The maxflow-mincut gap is the worst case ratio between these two quantities.

- A product multicommodity flow instance is a special case, where d is induced by a weight function $w: V \to \mathbb{R}^+$ on the nodes of V: for $uv, d(uv) = w(u)w(v)$.
- Let G and $\mathcal{T}$ be as defined above. Let $\mathcal{P}_i, \forall i,$ denote the set of paths joining s_i and t_i in $\mathcal{G}$ and let $\mathcal{P} = \bigcup_i \mathcal{P}_i$. The LP relaxation given below, called the 'Multicommodity Flow Relaxation', is used to obtain an upper bound on the number of pairs from $\mathcal{T}$ that can be routed in $\mathcal{G}$. For each path $P \in \mathcal{P}$, we have a variable $f(P)$ which is the amount of flow sent on P. Let x_i denote the total flow sent on paths for pair i. We let $\overline{f}$ denote the flow vector with a component for each path P. The LP relaxation is:

$$\max \sum_{i=1}^{k} x_i$$

subject to:

$$x_i - \sum_{P \in \mathcal{P}_i} f(P) = 0, \forall\, 1 \leq i \leq k$$

$$\sum_{P:\, v \in P} f(P) \leq c(v), \forall v \in V$$

$$x_i, f(P) \in [0,1], \forall 1 \leq i \leq k, P \in \mathcal{P}$$

- Given a non-negative weight function $\pi: \mathcal{T} \to \mathbb{R}^+$ on a set of nodes $\mathcal{T}$ in G, $\mathcal{T}$ is $\vec{\pi}$-flow-linked in G if there is a feasible multicommodity flow for the problem with demand $\vec{\pi}(u)\vec{\pi}(v)/\vec{\pi}(\mathcal{T})$ between every unordered pair of terminals $u, v \in \mathcal{T}$.

Theorem

Let OPT be a solution to the LP given above for a given instance $(G, \mathcal{T})$ of MVDPP in a graph G. Let $\beta(G) \geq 1$ be an upper bound on the worst case maxflow-mincut gap for product multicommodity flow problems in G. Then there is a partition of G, computable in polynomial time, into vertex-disjoint induced subgraphs $G_1, \dots, G_r$ and weight function $\vec{\pi}: V(G_i) \to \mathbb{R}^+$ with the following properties. Let $\mathcal{T}_i$ be the induced pairs of $\mathcal{T}$ in G_i and let X_i be the set of terminals of $\mathcal{T}_i$.

1. $\vec{\pi}_i(u) = \vec{\pi}_i(v)$ for $uv \in \mathcal{T}_i$
2. X_i is $\vec{\pi}_i$-flow-linked in G_i
3. $\sum_{i=1}^{r} \vec{\pi}_i(X_i) = \Omega(OPT/\beta(G)\log OPT)$

Decomposition Algorithm Overview

Without loss of generality, we can assume that all source and sink nodes in $\mathcal{T}$ are distinct. Start with a multi-commodity flow $\overline{f}$ for $\mathcal{T}$ in G with total flow value $\gamma(G) = OPT$. View the flow for each pair as being decomposed into flow paths. Given a node-induced subgraph $H = (V(H), E(H))$ of G, we let $\gamma(H)$

be the total flow induced in H by the original flow $\overline{f}$. This means that $\gamma(H)$ counts flow only on flow paths from the original flow path decomposition that are completely contained in H. Given a node $u \in X \cap V(H)$, let $\gamma(u, H)$ denote the flow in H for u. By definition, $\gamma(H) = \frac{1}{2}\sum_{u \in V(H)} \gamma(u, H)$.
Then the goal of the algorithm, given H as input, is to output a node-induced subgraph partition of H into $H_1, H_2, \ldots$ with associated weight functions $\vec{\pi}_1, \vec{\pi}_2, \ldots$. The algorithm is as follows:

1. If $0 < \gamma(H) \leq \beta(G) \log OPT$, let uv be some pair with positive flow in H. Define $\vec{\pi}$ on $V(H)$ by $\vec{\pi}(u) = \vec{\pi}(v) = 1$ and $\vec{\pi}(y) = 0$ for $y \neq u, v$. Stop and output H along with $\vec{\pi}$.
2. Otherwise, construct an instance of the product multicommodity flow problem on G with $w(u) = \gamma(u, H)/\sqrt{\gamma(H)}$ for $u \in V$. Let λ be the maximum concurrent flow for this instance.
   (i) if $\lambda \geq 1/(10\beta(G) \log OPT)$, stop the recursive procedure. Let $\vec{\pi}(u) = \gamma(u, H)/10\beta(G) \log OPT$. Output H and $\vec{\pi}$.
   (ii) Otherwise find a vertex cut S such that its size is at most $\beta(G)\lambda w(S) w(V \backslash S)$. Recurse on the induced graphs $H[S]$ and $H[V \backslash S]$.

It can be shown that properties 1 and 2 stated in the theorem are met in step 1 and step 2 (i) of the algorithm. To see why property 3 holds, note that the partitioning procedure defines a recursion tree, whose leaves are the graphs where we stop the recursion, either because the flow is sufficiently small or the concurrent flow for the product multicommodity flow that we set up is large enough. It can be proved that the flow *lost* in all the recursive step is at most $\gamma(G)/2$, from which it follows that $\sum_{i=1}^{r} \gamma(G_i) \geq \gamma(G)/2$. Then from the termination condition it follows that $\pi Gi \geq \gamma(Gi)/10\beta G \log OPT)$. From here, property 3 follows. This proves the theorem.

4. PROOF OF THEOREM 2

In this section we return to the paradigm of EDPs (instead of VDPs). Section 4.1 gives the main idea and sketch of proof of Theorem 2. Section 4.2 gives the complete proof of Theorem 2. Section 4.3 gives an insight into the details of Rao-Zhou Theorem.

4.1. Proof Sketch of Theorem 2

Given an instance of MEDPP, assume that there exists a randomized polynomial time Algorithm A that finds $\Omega\left(\frac{OPT^{\frac{1}{p}}}{\beta(n)}\right)$ EDPs for some $p > 1$. The idea is to make use of Rao-Zhou Theorem and prove Theorem 2 in four main steps given below:

1) In order to be able to use Rao-Zhou Theorem, we require our graph to have edge-connectivity at least $clog^5 n$ (i.e. the minimum cut in G should have size $\Omega(log^5 n)$). To achieve this, we do the following:
   (i) if there exists a cut of size $< clog^5 n$ which separates G into two parts A and B such that both A and B contain a terminal pair, recursively apply Rao-Zhou Theorem to A and B respectively. Combine the solutions to A and B to get a solution in original graph G. (Lemma 4.1, part (i)).
   (ii) if there exists a cut of size $< clog^5 n$ which seperates G into two parts A and B such that $|A| \geq 2$ and A contains no terminal pair, reduce G to G'' by applying some edge

contractions and adding a few dummy edges to 'small-degree vertices' such that the resulting graph G'' is $\Omega(log^5 n)$-edge-connected. (Lemma 4.1 part (ii), Lemma 4.2).

2) Obtain $\frac{OPT}{O(log^{10} n)}$ EDPs in G''. (Rao-Zhou Theorem)
3) Construct a polynomial time Algorithm B which finds $\frac{OPT}{O(n^{\frac{p-1}{2p-1}} poly(\log n))}$ paths in $\mathcal{P}$ that are edge-disjoint in the original graph G.
4) Compare the number of paths found by Algorithm B to that found by Algorithm A. Output the higher number.

4.2. Proof of Theorem 2

Consider the given instance of MEDPP and assume that there exists a randomized polynomial time Algorithm A that finds $\Omega\left(\frac{OPT^{\frac{1}{p}}}{\beta(n)}\right)$ EDPs for some $p > 1$. For some fixed constants a and t, there are two possibilities for the value of OPT:

$$\mathrm{OPT} \leq a\, n^{\frac{p}{2p-1}} log^t n \; (i.e. \mathrm{OPT} = O\left(n^{\frac{p}{2p-1}} log^t n\right))$$

OR

$$\mathrm{OPT} > a\, n^{\frac{p}{2p-1}} log^t n \; (i.e. \mathrm{OPT} = \Omega\left(n^{\frac{p}{2p-1}} log^t n\right))$$

Consider the first case. Since $\Omega\left(\frac{OPT^{\frac{1}{p}}}{\beta(n)}\right)$ EDPs can be found through Algorithm A, the approximation ratio is

$$= \frac{OPT}{\Omega\left(\frac{OPT^{\frac{1}{p}}}{\beta(n)}\right)}$$

$$\leq h OPT^{1-\frac{1}{p}} \beta(n), \text{ for some constant } h$$

$$= h\, \beta(n) OPT^{\frac{p-1}{p}}$$

$$\leq \; h\, \beta(n) (a^{\frac{p-1}{p}}\; n^{\frac{p-1}{2p-1}}\, log^{\frac{t(p-1)}{p}}\; n^{\frac{p}{2p-1}}), \text{ since OPT } \leq a\, n^{\frac{p}{2p-1}} log^t n$$

$$= \; h^* \beta(n) (n^{\frac{p-1}{2p-1}} log^{\frac{t(p-1)}{p}} n), \text{ where } h^* \text{ is a constant}$$

$$= O(n^{\frac{p-1}{2p-1}} poly(\log(n)))$$

This means that if $\mathrm{OPT} = O\left(n^{\frac{p}{2p-1}} log^t n\right)$, the proof of theorem is complete. Therefore, for the remainder of this discourse, we assume that $\mathrm{OPT} = \Omega\left(n^{\frac{p}{2p-1}} log^t n\right)$ for any t. In particular, we assume that $\mathrm{OPT} = \Omega\left(n^{\frac{p}{2p-1}} log^{10} n\right)$.

We now prove some lemmas first, followed by the proof of Theorem 2.

Lemma 4.1: Let there be a partition (A, B) of $V(G)$ such that $|A| \geq 2, |B| \geq 2$, and $|\partial(A)| \leq c log^5 n$, where c is the constant as in Rao-Zhou Theorem. Let OPT_A and OPT_B be the optimal values of the MEDPP when we restrict the problem to $G[A]$ and $G[B]$ respectively. Let $n_A = |A|$ and $n_B = |B|$.

(i) If $OPT_A \geq 1, OPT_B \geq 1$, and $OPT \geq n^\alpha$ for some $0 < \alpha < 1$, then $\frac{OPT_A}{n_A{}^\alpha} + \frac{OPT_B}{n_B{}^\alpha} \geq \frac{OPT}{n^\alpha}$ holds for sufficiently large n. It follows that combining an $O(n^\alpha)$-approximation solution in $G[A]$ and an $O(n^\alpha)$-approximation solution in $G[B]$ gives an $O(n^\alpha)$-approximation solution in G.

(ii) If $OPT_A = 0$ and $G[A]$ is connected, then G can be reduced to a smaller graph which contains at least OPT EDPs.

Proof: Note that by virtue of $|\partial(A)| \leq c log^5 n$, we have that $OPT \leq OPT_A + OPT_B + c log^5 n$ (since each edge in the cut $\partial(A)$ may result in a new EDP between a terminal pair having one end in A and the other in B).

(i) Without loss of generality, we may assume that $n_A \leq n_B$. Let λ be a constant such that $0 < \lambda < \alpha$. There can be two cases:

Case 1: $n^\lambda \leq n_B/n_A$. Then

$$\frac{OPT_A}{n_A^\lambda} + \frac{OPT_B}{n_B^\lambda} = \frac{OPT_A}{n_A^\lambda} + \frac{OPT_A}{n_B^\lambda} - \frac{OPT_A}{n_B^\lambda} + \frac{OPT_B}{n_B^\lambda}$$

$$= \frac{n_B^\alpha\, OPT_A}{n_A^\alpha n_B^\alpha} - \frac{OPT_A}{n_B^\alpha} + \frac{OPT_A}{n_B^\alpha} + \frac{OPT_B}{n_B^\alpha}$$

$$= \frac{(n_B^\alpha/n_A^\alpha - 1)OPT_A}{n_B^\alpha} + \frac{OPT_A + OPT_B}{n_B^\alpha}$$

$$\geq \frac{(n^{\alpha\lambda} - 1)OPT_A}{n_B^\alpha} + \frac{OPT - c log^5 n}{n_B^\alpha}$$

$$\geq \frac{(n^{\alpha\lambda} - 1)OPT_A - c log^5 n}{n_B^\alpha} + \frac{OPT}{n^\alpha} \text{ (since } n_B^\alpha \leq n^\alpha \rightarrow OPT/_{n_B^\alpha} \geq OPT/_{n^\alpha})$$

$$\geq \frac{OPT}{n^\alpha} \text{ (for sufficiently large } n\text{, the first term is non-negative)}$$

Case 2: $n^\lambda \geq n_B/n_A$. Then $n_B \leq n_A n^\alpha = (n - n_B)n^\alpha = n^{\alpha+1} - n_B n^\alpha$. Hence $n_B + n_B n^\alpha \leq n^{\alpha+1}$ and so we have $n_B \leq \frac{n^{1+\alpha}}{1+n^\alpha}$. So:

$$\frac{OPT_A}{n_A^\lambda} + \frac{OPT_B}{n_B^\lambda}$$

$$\geq \frac{OPT_A+OPT_B}{n_B^\alpha} \quad \text{since } n_A \leq n_B \text{ implies } \frac{OPT_A}{n_A^\alpha} \geq \frac{OPT_A}{n_B^\alpha}$$

$$\geq \frac{OPT}{n_B^\alpha} - \frac{clog^5 n}{n_B^\alpha}$$

$$= \frac{OPT}{n^\alpha} - (\frac{n_B}{n})^\alpha \left(\frac{OPT}{n_B^\alpha}\right) + \frac{OPT}{n_B^\alpha} - \frac{clog^5 n}{n_B^\alpha}$$

$$\geq \frac{OPT}{n^\alpha} + \frac{\left(1-(\frac{n^\lambda}{1+n^\lambda})^\alpha\right)OPT - clog^5 n}{n_B^\alpha} \quad \text{since } (\frac{n_B}{n})^\alpha \leq (\frac{n^\lambda}{1+n^\lambda})^\alpha$$

$$\geq \frac{OPT}{n^\alpha} + \frac{\frac{\alpha}{2}\left(1-(\frac{n^\lambda}{1+n^\lambda})\right)OPT - clog^5 n}{n_B^\alpha}$$

$$\geq \frac{OPT}{n^\alpha} \text{ for sufficiently large } n.$$

To go from the third last step to the second last step, we make use of the following fact from calculus: for a positive integer t with $\frac{1}{2^{t-1}} > \alpha \geq \frac{1}{2^t}$ and for $y = \frac{n^\lambda}{1+n^\lambda}$, we have:

$$1 - y^\alpha$$

$$\geq 1 - y^{\frac{1}{2^t}} \text{ since } \alpha < 1 \text{ and } y < 1 \text{ implies } y^\alpha \leq y^{\frac{1}{2^t}}$$

$$= \frac{1-y}{\left(1+y^{\frac{1}{2}}\right)\left(1+y^{\frac{1}{4}}\right)\cdots\left(1+y^{\frac{1}{2^t}}\right)} \quad \text{since } 1-y = \left(1+y^{\frac{1}{2}}\right)\left(1-y^{\frac{1}{2}}\right) = \left(1+y^{\frac{1}{2}}\right)\left(1+y^{\frac{1}{4}}\right)\cdots\left(1+y^{\frac{1}{2^t}}\right)\left(1-y^{\frac{1}{2^t}}\right)$$

$$\geq \frac{1-y}{2^t}$$

$$> \frac{\alpha(1-y)}{2}.$$

From Case 1 and Case 2, we see that $\frac{OPT_A}{{n_A}^\alpha} + \frac{OPT_B}{{n_B}^\alpha} \geq \frac{OPT}{n^\alpha}$ holds for sufficiently large n. From this inequality and the fact that $n_A^\alpha = O(n^\alpha)$ and $n_B^\alpha = O(n^\alpha)$, it follows that combining an $O(n^\alpha)$-approximation solution in $G[A]$ and an $O(n^\alpha)$-approximation solution in $G[B]$ gives an $O(n^\alpha)$-approximation solution in G.

(ii) Contract A to a single vertex by repeated edge-contractions. Since G has OPT EDPs and edge-contractions do not decrease the number of EDPs in a graph, it follows that the reduced graph contains at least OPT EDPs. ∎

We now show that G can be reduced to a graph G'' with edge-connectivity at least $clog^5 n$, so that the conditions for the Rao-Zhou Theorem are met. To achieve this, we present Lemma 4.2.

Lemma 4.2: G can be reduced to a graph G', where G' has the following properties:

- If any partition (A, B) of $V(G')$ satisfies $|\partial(A)| \leq clog^5 n$ (where c is the constant as in Rao-Zhou Theorem) and both $G'[A]$ and $G'[B]$ are connected, then one of A or B is a single vertex
- Each contracted vertex has degree at most $clog^5 n$.

Furthermore, G' can be 'augmented' (via edge additions) to G'', where G'' has edge-connectivity at least $clog^5 n$.

Proof: We first give a reduction from G to G'.

<u>Step 1</u>: Find a partition (A, B) of $V(G)$ such that $|A| \geq 2, |B| \geq 2$, and $|\partial(A)| \leq clog^5 n$. If such partition exists, go to Step 2. Otherwise go to Step 3.

<u>Step 2</u>: If $OPT_A \geq 1$ and $OPT_B \geq 1$, divide the problem in the original graph G into two smaller sub-problems on $G[A]$ and $G[B]$ and solve the sub-problems recursively. Then by virtue of Lemma 4.1 part (i), obtain a solution of the original instance by combining the solutions of the two sub-problems.
Otherwise, we may assume without loss of generality that $OPT_A = 0$. Contract each connected component of $G[A]$ to a single vertex.
Go to Step 1.

<u>Step 3</u>: Find a partition (A, B) of $V(G)$ such that $|A| = 1$, $|\partial(A)| \leq clog^5 n$, $OPT_A \geq 1$ and $OPT_B \geq 1$. If such a partition exists, then divide the problem in the original graph G into two smaller sub-problems on $G[A]$ and $G[B]$ and solve the sub-problems recursively. Then by virtue of Lemma 4.1 part (i), obtain a solution of the original instance by combining the solutions of the two sub-problems.

Let the graph obtained at the end be called G'. It is clear from this reduction that if there exists a cut of size $\leq clog^5 n$ in G', then we must have a single vertex on one of the sides of the cut (because we must have contracted one side of the cut in either step 2 or step 3). Also, each contracted vertex in G' must have degree at most $clog^5 n$ (otherwise we would not have contracted it). It follows, from Lemma 4.1 part (ii), that G' has at least OPT EDPs.

Now we prove the second part of the lemma. Let V^* be the set of vertices of G' with degree at most $clog^5 n$. For each $v \in V^*$, arbitrarily pick an edge $e \in \delta(v)$ and add $clog^5 n - |\delta(v)|$ edges parallel to e. Let G'' be the obtained graph. Then clearly G'' has edge-connectivity at least $clog^5 n$. Also note that any vertex in V^* has degree at most $c^2 log^{10} n$ in G'' (it may happen that for a vertex $u \in V^*$, every edge incident to it gets replicated while replicating the edges of u's neighbors. Since u has degree at most $clog^5 n$ and each of its incident edges gets replicated at most $clog^5 n$ times, the claim follows).

Since G' has at least OPT EDPs and adding edges does not decrease the number of EDPs, it follows that G'' has at least OPT EDPs. ■

We now apply Rao-Zhou Theorem to G'', which has the required edge-connectivity, and obtain $\frac{OPT}{O(log^{10} n)}$ EDPs $\mathfrak{R}_0$ in G'' with high probability. The task now is to find the set $\mathfrak{R}'$ of $\frac{OPT}{O(n^{\frac{p-1}{2p-1}} \operatorname{poly}(\log n))}$ EDPs in G'' such that no two paths in $\mathfrak{R}'$ share a vertex of V^* (because two paths sharing a contracted vertex of V^* in G'' may share an edge in G). We present Algorithm B to find $\mathfrak{R}'$. Claim 4.3 shows that the set $\mathfrak{R}'$ found by Algorithm B has size at least $\frac{c_1\, OPT}{O(n^{\frac{p-1}{2p-1}} \operatorname{poly}(\log n))}$ for some constant c_1 and sufficiently large n.

Algorithm B

Input: $\frac{OPT}{O(log^{10}n)}$ EDPs $\mathfrak{R}_0$ in G''

Goal: find the set $\mathfrak{R}'$ of $\frac{OPT}{O(n^{\frac{p-1}{2p-1}}\text{poly}(\log n))}$ EDPs in G'' such that no two paths in $\mathfrak{R}'$ share a vertex of V^*.

Step 0: Set $\mathfrak{R} = \mathfrak{R}_0$ and $\mathfrak{R}' = \emptyset$.

Step 1: If $\mathfrak{R} = \emptyset$, output $\mathfrak{R}'$ and stop the algorithm. Otherwise take a path $p \in \mathfrak{R}$ through the minimum number of vertices in V^*. Add p to $\mathfrak{R}'$.

Step 2: Remove from $\mathfrak{R}$ all paths sharing a vertex in V^* with p, and go to Step 1.

Claim 4.3: For some constant c_1, the output $\mathfrak{R}'$ of Algorithm B satisfies $|\mathfrak{R}'| \geq c_1 \frac{OPT}{(n^{\frac{p-1}{2p-1}}log^{31}n)}$ for sufficiently large n. Moreover, Algorithm B runs in polynomial time.

Proof: Suppose n is sufficiently large.

While $|\mathfrak{R}| \geq \frac{|\mathfrak{R}_0|}{2}$, we can find a path $p \in \mathfrak{R}$ through at most $\frac{nc^2log^{10}n}{|\mathfrak{R}|}$ vertices in V^*. Why is this so? If every path in $\mathfrak{R}$ went through more than $\frac{nc^2log^{10}n}{|\mathfrak{R}|}$ vertices in V^*, then all the paths in $\mathfrak{R}$ together would go through more than $nc^2log^{10}n$ vertices in V^*. Since each vertex in V^* has degree at most $c^2log^{10}n$, this would mean that somewhere in the paths in $\mathfrak{R}$, an edge is repeated. This contradicts the fact that $\mathfrak{R}$ is a set of EDPs.

Now

$$\frac{nc^2log^{10}n}{|\mathfrak{R}|}$$

$$\leq \frac{2nc^2log^{10}n}{|\mathfrak{R}_0|} \text{ , since } |\mathfrak{R}| \geq \frac{|\mathfrak{R}_0|}{2}$$

$$= \frac{O(log^{10}n)2nc^2log^{10}n}{OPT} \text{ , since} |\mathfrak{R}_0| = \frac{OPT}{O(log^{10}n)}$$

$$\leq \frac{c_2nlog^{20}n}{OPT} \text{ , for some } c_2 > 0.$$

So we can find a path $p \in \mathfrak{R}$ through at most $\frac{c_2nlog^{20}n}{OPT}$ vertices in V^*. Each of these vertices has degree at most $c^2log^{10}n$, hence p shares a vertex in V^* with at most $c^2log^{10}n(\frac{c_2nlog^{20}n}{OPT}) = \frac{c^2c_2nlog^{30}n}{OPT}$ paths in $\mathfrak{R}$. Now $\text{OPT} = \Omega\left(n^{\frac{p}{2p-1}}\, log^{10}n\right)$ implies $|\mathfrak{R}_0| = \frac{OPT}{O(log^{10}n)} = \Omega(n^{\frac{p}{2p-1}})$. Since $\frac{c_2c^2nlog^{30}n}{OPT}$ paths are removed in every iteration, there exists a constant $c_1 > 0$ such that

$$\left(c_1 \frac{OPT}{(n^{\frac{p-1}{2p-1}}log^{31}n)}\right)\left(\frac{c_2c^2nlog^{30}n}{OPT}\right)$$

$= c_1 c_2 c^2 n^{\frac{p}{2p-1}} log^{-1} n$

$< \frac{|\mathfrak{R}_0|}{2}$

This means that the iterations must be carried out $\left(c_1 \frac{OPT}{(n^{\frac{p-1}{2p-1}} log^{31} n)}\right)$ times before $|\mathfrak{R}|$ falls below $\frac{|\mathfrak{R}_0|}{2}$. This means $\left(c_1 \frac{OPT}{(n^{\frac{p-1}{2p-1}} log^{31} n)}\right)$ paths are added to $\mathfrak{R}'$ while $|\mathfrak{R}| \geq \frac{|\mathfrak{R}_0|}{2}$.

To prove that Algorithm B runs in polynomial time, notice that $|\mathfrak{R}_0| = \frac{OPT}{O(log^{10} n)}$ is polynomial in size and the size of V^* is $O(n)$. Step 1 takes at most $|\mathfrak{R}_0|$ time and step 2 takes at most $|\mathfrak{R}_0||V^*|$ time, and the number of iterations is at most $|\mathfrak{R}_0|$. So the overall running time is polynomial. This completes the proof of Claim 4.3. □

So we have $\left(c_1 \frac{OPT}{(n^{\frac{p-1}{2p-1}} log^{31} n)}\right)$ EDPs in G''. Let $\mathfrak{R}_1$ be their corresponding EDPs in G''. Also, let $\mathfrak{R}_2$ be the $\Omega\left(\frac{OPT^{\frac{1}{p}}}{\beta(n)}\right)$ EDPs found by Algorithm A that we assume to exist in the statement of Theorem 2. If $|\mathfrak{R}_1| \geq |\mathfrak{R}_2|$, output $\mathfrak{R}_1$. Otherwise output $\mathfrak{R}_2$.

This completes the proof of Theorem 2. ■

4.3. Brief Discussion on Rao-Zhou Theorem

This section gives brief insight into the proof of Rao-Zhou Theorem used in the proof of Theorem 2. We begin with some definitions and then give the overview of the proof.

Note: Some of the notions described here have been defined and used in Section 3.4. However, to keep these sections independent of each other, we redefine everything for the sake of easy readability.

Definitions:

- Given a non-negative weight function $\pi: X \to \mathbb{R}^+$ on a set of nodes X in G, X is π-cut-linked in G if $\forall S$ such that $\pi(S \cap X) = \sum_{x \in S \cap X} \pi(x) \leq \pi(X)/2$, $|\delta(S)| \geq \pi(S \cap X)$. (G, X) is called a <u>π-cut-linked instance</u>.
- For a cut $(S, \overline{S} = V \backslash S)$ in a graph G, let $\delta(S)$ denote the set of edges with exactly one endpoint in S. Let $cap(S, \overline{S}) = |\delta(S)|$ denote the total capacity of edges in a cut. Then the edge expansion of a cut $(S, \overline{S})$, where $|S| \leq |V|/2$, is $\phi(S) = \frac{cap(S,\overline{S})}{|S|}$. The <u>expansion of a graph</u> G is the minimum expansion over all cuts in G.
- A graph G is an <u>expander</u> if its expansion is at least a constant.

Proof Overview:

The idea is to first solve a fractional relaxation of the MEDPP, and based on its solution, the graph $\mathcal{G}$ is decomposed into disjoint subgraphs such that each subgraph is well-connected with respect to the set of terminals it contains. The key point of this decomposition is that only a constant number of terminal pairs are 'lost' this way, and the rest are still routable. To route them, an expander graph is constructed for each subgraph and it is embedded into $\mathcal{G}$. Then a greedy algorithm is applied to route the set of terminal pairs in this embedded expander graph. A more detailed summary is given below.

Let an instance of MEDPP consist of a graph $\mathcal{G} = (V, E)$ and a set of k terminal pairs $\mathcal{T} = \{(s_1, t_1), (s_2, t_2), \dots, (s_k, t_k)\}$. Let $\mathcal{P}_i, \forall i,$ denote the set of paths joining s_i and t_i in $\mathcal{G}$. Consider the fractional relaxation of the MEDPP, where each terminal pair can route a real-valued amount of flow between 0 and 1, and this flow can be split fractionally across a set of distinct paths. The LP relaxation given below is used to obtain an upper bound on the number of pairs from $\mathcal{T}$ that can be routed in $\mathcal{G}$. For each path p in some $\mathcal{P}_i$, we have a variable $f(p)$ which is the amount of flow sent on p. We let x_i denote the total flow sent on paths for pair i. We let $\overline{f}$ denote the flow vector with a component for each path p. The LP relaxation is:

$$\max \textstyle\sum_{i=1}^{k} x_i$$

subject to:

$$x_i - \sum_{p \in \mathcal{P}_i} f(p) = 0, \forall\ 1 \le i \le k$$

$$\sum_{p:\ e \in p} f(p) \le 1, \forall e \in E$$

$$x_i, f(p) \in [0,1], \forall 1 \le i \le k, \forall p$$

Let $OPT^*(\mathcal{G}, \mathcal{T})$ be the value of this LP for the optimal fraction solution $\overline{f}$ of the LP. The Rao Zhou Algorithm routes a poly-logarithmic fraction of this value using EDPs via the following steps:

1. Based on the value $OPT^*,$ the graph $\mathcal{G}$ is decomposed into subgraphs such that within each subgraph, the subset of terminal pairs is 'well-connected'.

   Sketch of Decomposition: The decomposition is done via the following main theorem:
   Given an MEDPP instance $(\mathcal{G}, \mathcal{T})$ which has minimum cut $\Omega(log^5 n)$ and a solution $\overline{f}$ to the fractional MEDPP problem with x_i, $\forall i$ as in the LP given above, there is a polynomial time algorithm which produces a set of disjoint subgraphs and a weight function $\pi: V(\mathcal{G}) \to \mathbb{R}^+$ on $V(\mathcal{G})$, where:
   - $\forall s_i t_i \in \mathcal{T}, x_i$ contributes the same amount of weight to $\pi(s_i)$ and $\pi(t_i)$
   - The set of nodes $V(H)$ in each subgraph H is π-cut-linked in H
   - Each subgraph H has a minimum cut of $\Omega(log^3 n)$
   - $\forall u$ in a subgraph H such that $\pi(H) \ge \Omega(log^3 n), \pi(u) \le \sum_{i:\ s_i = u,\ t_i \in H} \frac{x_i}{O(log^3 n)}$
   - $\pi(\mathcal{G}) = \Omega(\frac{OPT^*}{O(log^3 n)})$

To put it in simple words, this decomposition theorem simply says that if we sum across all subgraphs G of $\mathcal{G}$, we get a sufficient fraction of terminal pairs in $\mathcal{T}$, i.e. we lose only a constant fraction of the terminal pairs by assigning a zero weight to these lost terminals of $\mathcal{T}$. Moreover, each subgraph G is well-connected with respect to X, the set of induced terminals of $\mathcal{T}$ in G i.e. (G, X) is a π-cut-linked instance.

2. For each π-cut-linked instance (G, T) of $\mathcal{G}$, construct an expander graph H that can be embedded into G using its terminal set.
Construction: First split the graph G into Z subgraphs $G^1, G^2, \dots, G^Z$, each with the same weight function π on its vertex set $V(G^j) = V$ for all j, by extending a uniform sampling scheme from Karger [11] (Karger's scheme requires that the minimum cut should have size at least $\Omega(log^3 n)$. This is where the main requirement of Rao-Zhou's result, i.e. having $\Omega(log^5 n)$-edge-connectivity in the original graph, comes into play). This gives us a set of cut-linked instances $(G^j, X), \forall j,$ such that X is $\frac{(1-\epsilon)\pi}{Z}$-cut-linked in G^j for some $\epsilon < 1$. Now obtain a set $\mathcal{X} = \{X_1, X_2, \dots, X_r\}$ of vertex-disjoint 'superterminals', where each superterminal $X_i \in \mathcal{X}$ consists of a subset of terminals in X and gathers a weight between W and $2W - 1$, where W is a parameter dependent on the number Z. (The idea of making superterminals is that each superterminal, which is a set of clustered terminals, is better connected than individual terminals). These superterminals are now used as vertices in the expander graph H. The edges of H are defined using a technique ([13]) that builds an expander using $O(log^2 n)$ matchings. This expander H is embedded into G without any congestion by routing each matching in one of the split graphs using a maximum flow computation.

3. Route terminal pairs in H greedily via short disjoint paths (which are abundant in an expander graph). The greedy method routes $\Omega(\frac{K}{log^5 n})$ pairs of terminals, where $K = |V(H)| = \Omega(\pi(G)/W)$ in H.
Greedy Algorithm: While there exists a pair (s, t) in $T \subseteq \mathcal{T}$ whose path length is less than a predetermined parameter D in $H = (V, E)$, remove both nodes and edges from H along the path through which we connect a pair of terminals in T.

Clearly this algorithm induces no congestion because we delete a path between a terminal pair in each iteration. To show that the number of iterations (i.e. the number of paths routed by EDPs) is large enough, use the fact that when a path is taken away, all remaining terminal pairs in the expander must have distance at least D. This completes the proof sketch of Rao-Zhou Theorem.

5. CONCLUDING REMARKS

This report has discussed, in detail, the $O(n^{\frac{3}{7}} poly(\log n))$- approximation algorithm for the maximum edge-disjoint paths problem with congestion two, given by Kawarabayashi and Kobayashi [12]. The claimed approximation guarantee is a direct consequence of Theorem 1 and Theorem 2, as stated in Section 1.4. The result relies heavily on two results, given by Rao Zhou [16] and Chekuri et al. [4, 5]; this report has also discussed these two briefly.

One possible way of getting the same approximation guarantee with congestion one is to prove that Theorem 1 holds with congestion one. However, the methods (based on well-linked sets) adopted by the authors to prove Theorem 1 do not hold for the case with congestion one. The reason is that their proof relies on the LP relaxation given by Chekuri et al. [4, 5], which has integrality gap of $O(\sqrt{n})$. So this LP relaxation would not work for an example instance which has $O(\sqrt{n})$ number of terminals.

Recent results & open problems: In 2012, a randomized algorithm with an approximation guarantee of $\Omega(\text{poly}\ (\log\ l))$ for l terminal pairs and congestion at most 14 has been proposed [7]. This is a significant improvement over the previously known approximation guarantee of $n^{O(\frac{1}{c})}$ for congestion at most a constant c. It remains to be seen whether this algorithm can be improved further. Also, for congestion two, the question of whether the approximation guarantee of $O(n^{\frac{3}{7}}\, poly(\log n)$ can be improved remains to be open.

****

REFERENCES

[1] 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.

[2] B. Bollobás & A. Thomason, Highly linked graphs, Combinatorica, 16 (1996), 313–320.

[3] B. Bollobás & A. Thomason, Proof of a conjecture of Mader, Erdös and Hajnal on topological complete subgraphs, European Journal of Combinatorics, 19 (1998), 883–887

[4] C. Chekuri, S. Khanna & B. Shepherd, The all-or-nothing multicommodity flow problem, Proc. 36th ACM STOC, 2004, 156-165

[5] C. Chekuri, S. Khanna & B. Shepherd, Multicommodity flow, well-linked terminals & routing problems, Proc. 37th STOC, 2005, 183-192.

[6] C. Chekuri, S. Khanna & B. Shepherd, An $O(\sqrt{n})$ approximation and integrality gap for disjoint paths and unsplittable flow, Theory of Computing, 2 (2006), 137-146

[7] J. Chuzhoy, Routing in Undirected Graphs with Constant Congestion CoRR abs/1107.2554: (2011)

[8] R. Diestel, Graph Theory, 3rd Edition, Springer-Verlag, 2005.

[9] N. Robertson & P.D. Seymour, Graph Minors. XIII. The disjoint paths problem, J. Combin. Theory Ser. B, 63 (1995), 65-110

[10] 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. Also, Proc. 31st ACM Symposium on Theory of Computing (STOC), 1999, 19-28.

[11] D.R. Karger, Random sampling in cut, flow, and network design problems, Proc. 26th ACM Symposium on Theory of Computing (STOC), 1994.

[12] K. Kawarabayashi & Y. Kobayashi, Breaking $O(n^{\frac{1}{2}})$- Approximation Algorithms for the Edge-Disjoint Paths Problem with Congestion Two, Proc. 43rd ACM Symposium on Theory of Computing (STOC), 2011, 81-88.

[13] R. Khandekar, S. Rao, & U. Vazirani, Graph partitioning using single commodity flows, Proc. 38th ACM Symposium on Theory of Computing (STOC), 2006

[14] S. Kreutzer & S. Tazari, On brambles, grid-like minors & parameterized intractability of monadic second-order logic, SODA 2010, 354-364

[15] P. Raghavan & C.D. Thompson, Randomized rounding: A technique for provably good algorithms and algorithmic proofs, Combinatorica, 7 (1987), 365-374.

[16] S. Rao & S. Zhou, Edge Disjoint paths in moderately connected graphs, SIAM J. Computing, 39(2010),1856-87

[17] A. Srinviasan, 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