An Improved Decomposition Theorem for Graphs Excluding a Fixed Minor

Jittat Fakcharoenphol[1] * and Kunal Talwar[2] **

[1] Kasetsart University,
Bangkok, Thailand.
jtf@ku.ac.th
[2] Computer Science Division,
University of California, Berkeley
kunal@cs.berkeley.edu

Abstract. Given a graph G and a parameter δ, we want to decompose the graph into clusters of diameter δ without cutting too many edges. For any graph that excludes a $K_{r,r}$ minor, Klein, Plotkin and Rao [15] showed that this can be done while cutting only $O(r^3/\delta)$ fraction of the edges. This implies a bound on multicommodity max-flow min-cut ratio for such graphs. This result as well as the decomposition theorem have found numerous applications to approximation algorithms and metric embeddings for such graphs.
In this paper, we improve the above decomposition results from $O(r^3)$ to $O(r^2)$. This shows that for graphs excluding any minor of size r, the multicommodity max-flow min-cut ratio is at most $O(r^2)$ (for the uniform demand case). This also improves the performance guarantees of several applications of the decomposition theorem.

1 Introduction

A natural generalization of the s-t flow problem is the multicommodity flow problem, where we want to simultaneously route several commodities. Each commodity has a source and a sink, and the goal is to route the flows so that the total flow on any edge does not exceed its capacity. An optimization version of this problem is the *concurrent flow* problem, first defined by Shahrokhi and Matula [32], where we wish to maximize the *throughput* λ, such that we can feasibly route a λ fraction of each demand.

The sparsity of a cut $(S, \overline{S})$ is defined as $c(S, \overline{S})/d(S, \overline{S})$, where $c(S, \overline{S})$ is the sum of capacities of edges between S to $\overline{S}$ and $d(S, \overline{S})$ is the total demand from some source(sink) in S to a sink(source) in $\overline{S}$. The sparsity of any cut gives a upper bound on the maximum throughput. For the single commodity case, the max-flow min-cut theorem of Ford and Fulkerson [9] and of Elias, Feinstein and

* Reseach done while the author was a graduate student at UC Berkeley. Research partially supported by a DPST scholarship and NSF grant CCR-0105533.
** Research partially supported by NSF via grants CCR-0121555 and CCR-0105533.

Shannon [8], says that the maximum flow equals the value of the sparsest cut, and also gives an algorithm for finding the minimum cut.

The seminal work of Leighton and Rao [18] first considered approximate max-flow min-cut theorems. They showed that for the case of uniform demands, the ratio of sparsest cut to the maximum throughput in any graph is at most $O(\log n)$. Their proof also gives an algorithm to find a cut of sparsity no more than $O(\log n)$ times the maximum throughput (and hence at most $O(\log n)$ times the sparsest cut). This approximation algorithm is a basic subroutine for approximation algorithms for a variety of NP-hard problems.

For arbitrary demands, such an approximate max-flow min-cut theorem was discovered by Klein, Rao, Agrawal and Ravi [16], who showed an upper bound of $O(\log C \log D)$ where C is the sum of all capacities and D is the sum of all demands. This ratio has since been improved and the best currently known bound is $O(\log k)$, where k is the number of commodities, due to Linial, London and Rabinovich [19], and Aumann and Rabani [2] (see the related work section for details). For arbitrary graphs, this is the best(upto constants) that one can do, since an expander graph gives a matching lower bound.

Klein, Plotkin and Rao [15] considered restricted families of graphs, and showed for graphs excluding a minor of size r, the gap is $O(r^3)$ for the uniform demand case and $O(r^3 \log k)$ for the general case. The latter result was improved to $O(r^3 \sqrt{\log k})$ by Rao [26]. In particular, this showed that for planar graphs, which exclude K_5 and $K_{3,3}$ minors, the max-flow min-cut gap is $O(1)$ for the uniform case. Both the aforementioned results use a decomposition lemma proved in [15], which says that given a parameter δ, one can decompose a graph excluding a $K_{r,r}$ minor into clusters of diameter δ, while cutting only $O(r^3/\delta)$ fraction of the edges[3]. Note that any such decomposition of a path graph must cut an $O(\frac{1}{\delta})$ fraction of the edges, and thus the overhead for graphs excluding $K_{r,r}$ was shown to be $O(r^3)$. Not surprisingly, this decomposition lemma has found several other applications to approximation algorithms, distributed computing and embeddings results for such graphs.

In this paper we make some progress towards finding the right relation between the size of the forbidden minor and the overhead of such a decomposition. We show that for any graph excluding a K_r minor, we can find a decomposition into clusters of diameter δ while cutting only $O(r^2/\delta)$ fraction of the edges. This shows that the max-flow min-cut gap for such graphs is $O(r^2)$ for the uniform demands case and $O(r^2\sqrt{\log n})$ for the general case. It also improves the performance guarantees of approximation algorithms and embeddings results for such graphs.

What is the right order of magnitude of the overhead of such a decomposition? An expander graph gives a lower bound of $\Omega(\log r)$, the upper bound we show is $O(r^2)$. Moreover, can we bound this overhead in terms of some other topological/metric properties of the graph? We leave open these intriguing questions.

[3] The second result actually requires the decomposition to have an additional "padding" property, details of which are deferred to the technical sections.

Related Work

As described above, Klein et.al. [16] gave the first non trivial upper bound of $O(\log C \log D)$ for multicommodity max-flow min-cut ratio for arbitrary demands. This was improved to $O(\log k^*)$ through the works of Tragoudas [34], Garg, Vazirani and Yannakakis [11], Plotkin and Tardos [23], Aumann and Rabani [2], Linial, London and Rabinovich [19], and Günlük [12] (k^* here is the size of the smallest vertex cover of the demand graph).

For several special classes of graphs, exact max-flow min-cut theorems have been proved, for example, by Hu [14], Rothschild and Whinston [29], Dinits(see [1]), Seymour [31], Lomonosov [20], Seymour [30] and Okamura and Seymour [22]. See [10] for more on this vein of work.

Network decomposition theorems like this one, are known for other classes of graphs as well. For general graphs, it is known that it suffices to cut an $O(\log n/\delta)$ fraction of the edges to decompose it into clusters of diameter δ, and this is the best one can do for general graphs. For graphs induced by real normed spaces $\mathbf{R}_p^d$, Charikar et.al. [7] show that such decompositions exist with an overhead of $O(d^{\frac{1}{p}})$ for $1 \leq p \leq 2$ and $O(d^{1-\frac{1}{p}})$ for $p > 2$, and that this is tight.

The characterization of planar graphs in terms of forbidden minors is due to Kuratowski [17]. Robertson and Seymour [28] showed that similar charcterizations exist for graphs of genus g for any g. In particular it is known that graphs of genus g exclude $K_{\Omega(\sqrt{g})}$ minor.

The approximate max-flow min-cut theorems have found numerous applications such as Oblivious routing, Data management, small area VLSI layout, efficient simulations of one interconnection network by another, etc. For more details on oblivious routing the reader is referred to the papers by Räcke [25], Azar et.al. [3], Bienkowski, Korzeniowski and Räcke [5], and Harrelson, Hildrum and Rao [13]. Data management applications have been looked at by Maggs et.al.[21]. The reader is referred to Bhatt and Leighton [4] for VLSI layout applications.

The decomposition theorem itself has found applications to approximation algorithms for various NP-hard problems. We mention a few of these applications here. Tardos and Vazirani [33] showed that the decomposition theorem implied an $O(r^3)$ bound on the max (total) flow-min multicut gap and an approximation algorithm for minimum multicut in graphs excluding a $K_{r,r}$ minor.

Rao and Richa [27] gave $O(r^3 \log\log n)$-approximation algorithms for minimum linear arrangement and minimum containing interval graph on graphs excluding K_r minor. Calinescu, Karloff and Rabani [6] gave an $O(r^3)$-approximation algorithm for the 0-extension problem on such graphs and Feige and Krauthgamer gave an $O(r^3 \log n)$-approximation algorithm to minimum bisection on such graphs.

A slight modification of these decompositions have also been used in the area of metric embeddings. Rao [26] showed that graphs excluding K_r minors can be embedded into l_2 with distortion $O(r^3\sqrt{\log n})$. Moreover these embeddings preserve not only distances but also volumes. Recently, Rabinovich [24] showed how to embed a metric excluding K_r into a line with *average distortion* $O(r^3)$. For graphs with tree width r, they further improved the embedding to $O(\log r)$

and left open the question of the correct order for graphs excluding K_r minor. Our results improve the r^3 in all the above applications to r^2.

A note on techniques

The techniques used in this paper borrow generously from those used by Klein, Plotkin and Rao [15]. They showed that if their algorithm of repeatedly shattering BFS trees $O(r)$ times produced a cluster of large diameter, then they could construct a $K_{r,r}$ minor, consisting of r well spaced points in the large diameter cluster and the r roots of the BFS trees. We note that the roots of the BFS trees used were chosen arbitrarily.

Instead, we are somewhat more careful in our choice of the roots. We make sure that the roots of the BFS trees constructed are mutually far apart; this allows us to construct disjoint paths connecting these roots. This allows us to get a better guarantee on the diameter of the clusters.

2 Preliminaries

Let H and G be graphs. Suppose that for every vertex v of H, G contains a connected subgraph $\mathcal{A}(v)$ and for every edge (u, v) in H, there is an edge $\mathcal{E}(uv)$ connecting $\mathcal{A}(u)$ and $\mathcal{A}(v)$ in G. If the $\mathcal{A}(v)$'s are pairwise disjoint, we say that G contains an *H-minor* and call $\cup_v \mathcal{A}(v)$ *an H-minor* of G. We refer to the $\mathcal{A}(v)$'s as *supernodes* and $\mathcal{E}(uv)$'s as *superedges*.

We denote by K_h the complete graph on h nodes. Note that if G contains a K_h minor, it contains every minor on h vertices. Thus if G excludes any minor of size h, it excludes K_h. In particular, excluding a $K_{r,r}$ minor implies excluding a K_{2r} minor. Moreover, a $K_{r,r}$ contains a K_r minor. Thus upto a factor of 2, excluding a K_r minor and excluding a $K_{r,r}$ minor are equivalent.

Given a graph $G = (V, E)$, we can define a natural distance measure on V: $d_G(u, v)$ is the length of the shortest path from u to v. For a subset V' of V, the *weak* diameter of V' is defined to be $\max_{u,v \in V'} \{d_G(u, v)\}$. In this paper, the term diameter will always refer to weak diameter.

A δ-decomposition π of $G = (V, E)$ is a partition of V into subsets $V_1, V_2, \ldots, V_k$ such that each cluster V_i (defined as $\{v \in V : \pi(v) = i\}$) has (weak) diameter at most δ. An edge $e = (u, v)$ is said to be cut by this decomposition if u and v lie in different V_i's.

Let Π be a set of δ-decompositions of G and let $\mathcal{D}$ be a distribution over Π. We say $(\Pi, \mathcal{D})$ is α-padded if for any vertex v, and any $c < \frac{1}{2}$, the probability that v is at distance less than $c\delta$ from any cluster boundary is at most $2c\alpha$. More formally, for a partition π, let $d(v, \pi) = \min_{u:\pi(u) \neq \pi(v)} d(u, v)$. Then we say that $(\Pi, \mathcal{D})$ is α-padded if $\Pr_{\pi \in (\Pi, \mathcal{D})}[d(v, \pi) \leq c\delta] \leq 2c\alpha$. A probabilisitic version of the KPR decomposition was shown to be $O(r^3)$-padded in [26]. We shall show that our decomposition is $O(r^2)$-padded.

For ease of notation in the rest of the paper, we shall give an algorithm to construct an $O(r\delta)$-decomposition of the graph, which cuts $O(r/\delta)$ fraction of the edges, and is $O(r)$-padded. The result claimed in the introduction can of course be derived by scaling δ by a factor of $O(r)$.

3 The decomposition procedure

We decompose the graph recursively $r-2$ times. At each level i, given a cluster G_i, we do the following. We pick, if possible, an *appropriate* node (explained in the next paragraph) a_i in G_i and construct a breadth first search tree rooted at a_i. We say a vertex v is at *level* l if its distance in G_i, from a_i is l. We partition the edges of G_i into δ classes. For $k = 0, 1, \ldots, \delta - 1$, the k^{th} class consists of edges between nodes at level $j\delta + k$ and $j\delta + k + 1$ for some integer $j \geq 0$. We pick an integer $k \in \{0, \ldots, \delta - 1\}$ uniformly at random, and cut the edges in the k^{th} class. We recurse on the resulting clusters.

By *appropriate* above, we mean a node which is at least distance $4r\delta$ far from each of roots of the breadth-first search trees in the higher levels of recursion. In case there is no such node in cluster G_i, we *shatter* the cluster in a different way - each cluster consisting of vertices close to one of the previous level roots.

Finally, we further shatter each resulting cluster G_{r-1} into at most $r-1$ pieces by cutting out clusters of inappropriate nodes; for each of the centers $a_1, \ldots, a_{r-2}$, we cut out a set of vertices close to a_i to form a separate cluster. We redefine G_{r-1} to be the remaining set of nodes C'. The above procedures describe the set of edges that are cut; the final clusters are defined by the connected components of the remaining graph.

Figure 1 show the pseudocode of the procedures. We start by calling the procedure Decompose($G_1 = G, 1, \{\}$).

4 Proof of the decomposition procedure

We first show that the decomposition constructed has the two properties that we needed.

Lemma 1. *The expected number of edges that are cut by the above procedure is* $O(r|E(G)|/\delta)$.

Proof. Note that we have at most r levels of recursion, and at most r cuts made in any shatter procedure. Thus at most $2r$ cuts potentially involve any particular edge. In each call to decompose or shatter, a fixed edge in the cluster has a probability at most $1/\delta$ of being cut (since it is at exactly one level, and we choose one of δ levels u.a.r.). Thus, any fixed edge has a probability at most $2r/\delta$ of being cut. The claim follows by linearity of expectation.

Lemma 2. *The decomposition produced is* $2r$*-padded.*

Proof. From the argument above, each cluster is produced as a result of at most $2r$ random cuts. Fix a vertex v and let Y_i be a random variable denoting its distance from the boundary of the i^{th} cut. Clearly, $d(v, \pi) = \min_i Y_i$. Moreover,

Algorithm Decompose($G_i, i, p = \{a_1, \ldots, a_{i-1}\}$)

1. **if** there exists $v \in G_i$ such that $d_G(a_j, v) \geq 4r\delta$ for all $1 \leq j \leq i-1$ **then**
1.1 $a_i \leftarrow v$.
1.2 Create a BFS tree $\mathcal{T}_i$ in G_i rooted at a_i.
1.3 **if** $\mathcal{T}_i$ contains less than $\delta + 1$ level **then**
1.3.1 **stop**.
1.4 **for** $k = 0, 1, \ldots, \delta - 1$ **do**
1.4.1 Define the k-th cut S^k to be the set of edges between nodes at level $j\delta + k$ and $j\delta + k + 1$ in $\mathcal{T}_i$, for some $j \geq 0$.
1.5 Pick a k randomly in $0, 1, \ldots, \delta - 1$. Let $S = S_k$.
1.6 Cut all edges in S.
1.7 **for** each component G' in $G_i - S$ **do**
1.7.1 **if** $i < r - 2$ **then**
1.7.1.1 Decompose($G', i + 1, \{a_1, \ldots, a_{i-1}, a_i\}$).
1.7.2 **else**
1.7.2.1 Shatter($G', i, \{a_1, \ldots, a_{i-1}, a_i\}$).
2. **else**
2.1 Shatter($G_i, i - 1, p$).

Procedure Shatter($C, k, p = \{a_1, \ldots, a_k\}$)

1. $C' \leftarrow C$.
2. **for** $i = 1, \ldots, k$ **do**
2.1 $C_i \leftarrow$ all nodes v in C' such that $d_G(v, a_i) \leq 4r\delta$.
2.2 Create a breadth-first search tree T_i from nodes in C_i.
2.3 Let T_i' be the first $\delta + 1$ levels of T_i.
2.4 **if** T_i' covers all C' **then**
2.4.1 $C' \leftarrow \emptyset$.
2.5 **else**
2.5.1 Let j be chosen randomly in $0, 1, \ldots, \delta - 1$
2.5.2 Let T_i'' be a subtree of T_i up to level j.
2.5.2 Cut all edges at level j.
2.5.3 $C' \leftarrow C' - (C_i \cup T_i'')$.

Fig. 1. The decomposition procedures.

the i^{th} cut was chosen uniformly at random from δ equispaced cuts, and thus Y_i is uniformly distributed in $[1, \delta/2]$. Hence $\Pr[Y_i \leq c\delta] \leq 2c$ for any $c \leq 1/2$. The claim then follows by a simple union bound.

Having established the required properties of the probabilistic decomposition, we now proceed to show that it is indeed an $O(r\delta)$-decomposition. Note that our decomposition consists of two kinds of clusters - those consisting of vertices close to some root, formed by some call to procedure shatter, and those formed by the procedure decompose. We first show that clusters of the first type have small diameters.

Lemma 3. *The procedure* Shatter *cuts out clusters each of weak diameter at most* $(8r + 2)\delta$.

Proof. For each $j = 1, \ldots, i-1$, we define the set C_j to be the set of all vertices in G_i which are at distance at most $4r\delta$ from a_j. The procedure cuts out cluster T_j'' formed by taking the set of vertices in C_j closer than some randomly chosen threshold $t \leq \delta$ to a_j. Consider any pair of nodes u and v in the same connected component T_j'' in the resulting graph. It must be the case that there is some a_i such that the distance from u and v to a_i is at most $(4r + 1)\delta$ in G_1. Therefore by triangle inequality, the weak diameter of each such component is at most $(8r + 2)\delta$.

We now consider the remaining case. We wish to show that if the graph excludes a K_r minor, then the diameter of each such cluster resulting from our decomposition algorithm is small. We shall show the contrapositive - if the resulting decomposition has some cluster with large diameter, we shall show how to construct a K_r minor in the graph. Let G_{r-1} be a cluster of large diameter and let a_{r-1} and a_r be two vertices in G_{r-1} which are at least distance $4r\delta$ apart. We shall construct a K_r minor, containing a supernode centered at each a_i, for $i = 1, 2, \ldots, r$. We shall use the paths in the bfs trees to find superedges.

Lemma 4. *Suppose that a cluster* G_{r-1} *output by our algorithm has diameter* $4r\delta$. *Then* G_1 *contains a* K_r *minor.*

Proof. As above, denote by a_{r-1} and a_r two nodes in G_{r-1} at distance $4r\delta$ from each other. Note that by our construction, every pair of a_i and a_j is at least distance $4r\delta$ apart.

We shall show how to construct a K_r minor in G_1. We do so by reverse induction - we give a procedure which, for $b = r-2, r-3, \ldots, 1$, constructs a K_{r-b}-minor in G_{b+1}.

Recall that G_{i+1} consists of δ consecutive layers in the bfs tree $\mathcal{T}_i$ rooted at a_i. An *ancestor-path* of v in $\mathcal{T}_i$ is the path in $\mathcal{T}_i$ from v to the root a_i of $\mathcal{T}_i$. We shall construct the minor using suitable ancestor-paths in $\mathcal{T}_i$'s.

Given a K_b-minor in G such that starting at each supernode $\mathcal{A}(g)$ there is path P_g, we say that the paths $\{P_g\}$ are *tails* if each path P_g is disjoint from the other paths and also from all supernodes except $\mathcal{A}(g)$. We shall refer to P_g's ending node (outside of $\mathcal{A}(g)$) as the *tip* of the tail P_g and denote it by $tip(P_g)$.

Klein, Plotkin and Rao [15] show how to construct the minor inductively by also constructing tails which are ancestor-paths of $\mathcal{T}_b$ and special nodes (which they called *middle nodes*) on the tails which are far apart, and using them to further construct disjoint components of the minor. We use a similar approach.

We shall construct a K_{r-b}-minor in G_{b+1}. In addition, we construct $r-b+1$ tails $\{P_i\}$ which are ancestor-paths of $\mathcal{T}_b$ of length exactly 4δ such that for each tail P_i, a middle node h_i of P_i is at distance $4b\delta$ from the other middle nodes h_j's. Moreover, we require that every middle node is at distance at least $4b\delta$ from the root a_b of $\mathcal{T}_b$. This shall be our (reverse) inductive claim.

For the basis step, when $b=r-2$, let P be the shortest path from a_{r-1} to a_r in G_{r-1} (since G_{r-1} is connected, such a path exists). We construct a K_2-minor from the path P. We let $\mathcal{A}(a_r)$ be a path of length $4\delta-1$ on P starting from a_r. The other supernode $\mathcal{A}(a_{r-1})$ is then $P-\mathcal{A}(a_r)$. We construct the tails by taking P_j to be the ancestor-paths in $\mathcal{T}_{r-2}$ of length 4δ from a_j, for $j \in \{r, r-1\}$. It can be checked that they are proper tails and the middle nodes in these tails are at distance at least $4(r-2)\delta$ from each other. Also the middle nodes in these tails are at distance at least $4(r-2)\delta$ from a_{r-2}.

We now show the inductive step. Assuming that the claim is true for $b=i+1$, we want to show that the claim is true for $b=i$, i.e., G_{i+1} contains K_{r-i} as a minor and a new set of tails with the required properties.

We first construct the minor. For $j>i+1$, we create supernodes $\mathcal{A}'(a_j)$ from the supernodes of K_{r-i-1} as follows. We let $\mathcal{A}'(a_j)$ be $\mathcal{A}(a_j) \cup (P_j - \{tip(P_j)\})$. From the inductive assumption, these supernodes are disjoint. This gives us $r-i-1$ supernodes. We let $\mathcal{A}'(a_{i+1})$ be a union of all ancestor-paths in $\mathcal{T}_{i+1}$ starting from the tip of all the tails $\{P_j\}$.

We must show that $\mathcal{A}'(a_{i+1})$ is disjoint from all other new supernodes. Since we create tails of length 4δ from the ancestor-paths in $\mathcal{T}_{i+2}$, the end nodes of the tails lie outside the subgraph G_{i+2}; therefore, the supernodes $\mathcal{A}(a_j)$, lying inside G_{i+2}, and $\mathcal{A}'(a_{i+1})$ are disjoint. Also, the tails $\{P_j\}$ and $\mathcal{A}'(a_{i+1})$ are disjoint by construction. Moreover, the last edges on the paths P_j give us the required additional superedges. This shows that G_{i+1} contains a K_{r-i}-minor.

To finish the inductive claim, we need to construct the tails with the desired properties. For each middle node h_j, let the tail P'_j be the ancestor-paths in $\mathcal{T}_i$ from h_j of length 4δ. These tails are mutually disjoint because h_j's are at distance at least $4(i+1)\delta$ from each other in G_i. We also create another tail P'_i starting from a_i in the same way. It is straightforward to verify that the new middle nodes $\{h'_j\}$ are at the right distance of each other.

We must also show that the tail P'_j are disjoint from all $\mathcal{A}'(a_k)$ where $k \neq j$. Consider any node v in $\mathcal{A}'(a_k)$. From the choice of h_j, the levels of v and h_j in $\mathcal{T}_{i+1}$ differ by more than δ. This implies that v does not lie on the ancestor-paths of h_j in $\mathcal{T}_i$ for any j[4] (since G_{i+1} consists of at most δ consecutive layers of $\mathcal{T}_i$, there is a path of length at most δ from h_j to any $\mathcal{T}_i$-ancestor (say w) of h_j lying in G_{i+1}. Thus h_j and w would be within δ layers of each other in $\mathcal{T}_{i+1}$ and hence v is different from w). Thus for any $j \neq k$, P'_j is disjoint from $\mathcal{A}'(a_k)$. To show

[4] This is exactly the "moat" argument in [15].

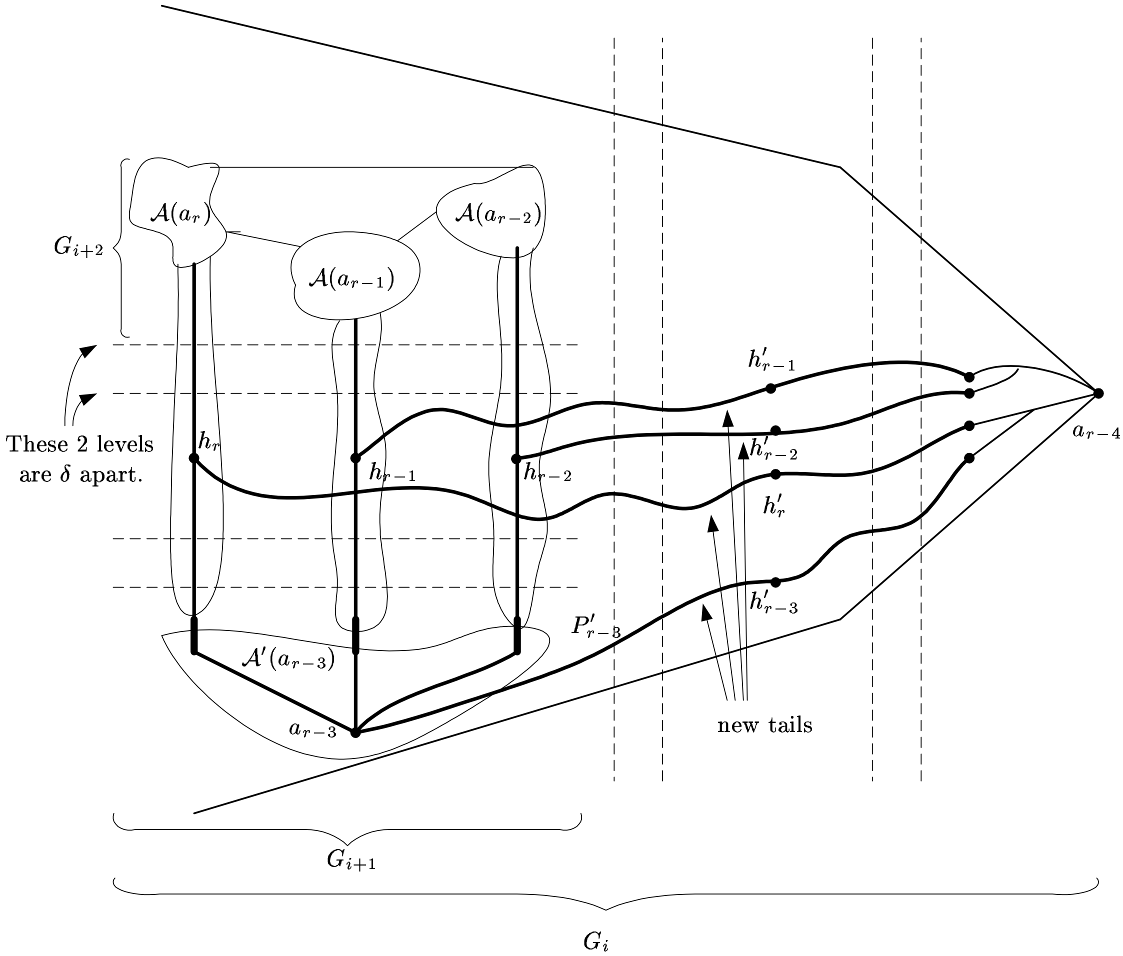

Fig. 2. The inductive step.

that P'_j does not cross any tails P_k, we note that the distance between h_j and h_k is more than 6δ. Finally, since a_i is at distance $4(i+1)\delta$ from all the middle nodes h_j, the path P'_i is also a proper tail.

It only remains to show that the middles node h'_j's are at distance at least $4i\delta$ from a_{i-1}. From our construction a_{i-1} is at distance at least $4r\delta$ from a_j, where $j > i$. We know inductively that the new middle node h'_j are at distance at most $2(r-i)\delta$ from a_j. By triangle inequality then, the distance from h'_j and a_{i-1} is at least $4r\delta - 2(r-i)\delta \geq 4i\delta$. This completes the inductive argument.

Thus, when $b = 1$, the induction claim says that G_2 contains a K_{r-1}-minor and the tails with the appropriate properties. We can construct a K_r-minor in G_1 as in the inductive step. This completes the proof of Lemma 4.

From the above lemmas, we have the main theorem.

Theorem 1. *Given a graph G and parameters δ and r, we can either find a K_r minor in G or find a $O(r)$-padded $O(r\delta)$-probabilistic decomposition of the G which expects to cut at most $O(mr/\delta)$ edges.*

We can also generalize this procedure for graphs with distances and weights on the edges. Moreover, if the padding property is not required, we can easily derandomize the algorithm by picking the best cut at each step.

5 Acknowledgement

We would like to thank Satish Rao for several helpful discussions. We would also like to thank the anonymous referees for several helpful comments.

References

1. G. Adel'son-Vel'ski, E. Dinits, and A. Karzanov. *Flow Algorithms.* Nauka, Moscow, 1975. In Russian.
2. Y. Aumann and Y. Rabani. An $O(\log k)$ approximate min-cut max-flow theorem and approximation algorithm. *SIAM J. Comput.*, 27(1):291–301, 1998.
3. Y. Azar, E. Cohen, A. Fiat, H. Kaplan, and H. Räcke. Optimal oblivious routing in polynomial time. In *Proceedings of the thirty-fifth annual ACM symposium on Theory of computing*, 2003.
4. S. N. Bhatt and F. T. Leighton. A framework for solving VLSI graph layout problems. *Journal of Computer and System Sciences*, 28(2):300–343, Apr. 1984.
5. M. Bienkowski, M. Korzeniowski, and H. Räcke. A practical algorithm for constructing oblivious routing schemes. In *Fifteenth ACM Symposium on Parallelism in Algorithms and Architectures*, June 2003.
6. G. Calinescu, H. Karloff, and Y. Rabani. Approximation algorithms for the 0-Extension problem. In *Proceedings of the Twelfth Annual ACM-SIAM Symposium on Discrete Algorithms (SODA-01)*, pages 8–16, New York, Jan. 7–9 2001. ACM Press.

7. M. Charikar, C. Chekuri, A. Goel, S. Guha, and S. Plotkin. Approximating a finite metric by a small number of tree metrics. In IEEE, editor, *39th Annual Symposium on Foundations of Computer Science: proceedings: November 8–11, 1998, Palo Alto, California*, pages 379–388, 1109 Spring Street, Suite 300, Silver Spring, MD 20910, USA, 1998. IEEE Computer Society Press.
8. P. Elias, A. Feinstein, and C. E. Shannon. A note on the maximum flow through a network. *IEEE Trans. Inform. Th.*, IT-2:117–119, 1956.
9. L. R. Ford and D. R. Fulkerson. *Flows in Networks.* Princeton Univ. Press, Princeton, NJ, 1962.
10. A. Frank. Packing paths, circuits, and cuts - a survey. In B. Korte, L. Lovász, H.-J. Prömel, and A. Schrijver, editors, *Paths, Flows and VLSI-Layouts*, pages 47–100. Springer Verlag, 1990.
11. N. Garg, V. V. Vazirani, and M. Yannakakis. Approximate max-flow min-(multi)cut theorems and their applications. In *Proceedings of the twenty-fifth annual ACM symposium on Theory of computing*, pages 698–707. ACM Press, 1993.
12. O. Günlük. A new min-cut max-flow ratio for multicommodity flows. *Lecture Notes in Computer Science: Integer Programming and Combinatorial Optimization*, 2337:54–66, 2002.
13. C. Harrelson, K. Hildrum, and S. Rao. A polynomial-time tree decomposition to minimize congestion. In *Symposium on Parallel Algorithms and Architectures*, 2003.
14. T. Hu. Multicommodity network flows. *Operations Research*, 11:344–360, 1963.
15. P. Klein, S. A. Plotkin, and S. Rao. Excluded minors, network decomposition, and multicommodity flow. In *Proceedings of the twenty-fifth annual ACM symposium on Theory of computing*, pages 682–690. ACM Press, 1993.
16. P. N. Klein, S. Rao, A. Agrawal, and R. Ravi. An approximate max-flow min-cut relation for unidirected multicommodity flow, with applications. *Combinatorica*, 15(2):187–202, 1995.
17. K. Kuratowski. Sue le problème des courbes gauches en topologie. *Fund. Math.*, 15:217–283, 1930.
18. T. Leighton and S. Rao. An approximate max-flow min-cut theorem for uniform multicommodity flow problems with applications to approximation algorithms. In *29th Annual Symposium on Foundations of Computer Science*, pages 422–431, White Plains, New York, 24–26 Oct. 1988. IEEE.
19. N. Linial, E. London, and Y. Rabinovich. The geometry of graphs and some of its algorithmic applications. *COMBINAT: Combinatorica*, 15, 1995.
20. M. V. Lomonosov. Combinatorial approaches to multiflow problems. *Discrete Applied Math.*, 11:1–94, 1985.
21. B. M. Maggs, F. M. auf der Heide, , B. Vöcking, and M. Westermann. Exploiting locality for data management in systems of limited bandwidth. In *38th Annual Symposium on Foundations of Computer Science*, pages 284–293, Miami Beach, Florida, 20–22 Oct. 1997. IEEE.
22. H. Okamura and P. Seymour. Multicommodity flows in planar graphs. *Journal of Combinatorial Theory, Series B*, 31:75–81, 1981.
23. S. A. Plotkin and É. Tardos. Improved bounds on the max-flow min-cut ratio for multicommodity flows. In *ACM Symposium on Theory of Computing*, pages 691–697, 1993.
24. Y. Rabinovich. On average distortion of embedding metrics into l_1 and into the line yuri rabinovich. In *Proceedings of the thirty-fifth annual ACM symposium on Theory of computing*, 2003.

25. H. Räcke. Minimizing congestion in general networks. In *Proceedings of the 43rd Annual Symposium on the Foundations of Comuter Science*, pages 43–52, Nov. 2002.
26. S. Rao. Small distortion and volume preserving embeddings for planar and euclidean metrics. In *Proceedings of the fifteenth annual symposium on Computational geometry*, pages 300–306. ACM Press, 1999.
27. S. Rao and A. W. Richa. New approximation techniques for some ordering problems. In *Proceedings of the Ninth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 211–218, San Francisco, California, 25–27 Jan. 1998.
28. N. Robertson and P. D. Seymour. Graph minors. VIII. a Kuratowski theorem for general surfaces. *Journal of Combinatorial Theory Series B*, 48(2):255–288, 1990.
29. B. Rothschild and A. Whinston. On two commodity network flows. *Operations Res.*, 14:377–387, 1966.
30. P. Seymour. Matroids and multicommodity flows. *European Journal of Combinatorics*, 2:257–290, 1981.
31. P. D. Seymour. Four-terminus flows. *Networks*, 10:79–86, 1980.
32. F. Shahrokhi and D. W. Matula. The maximum concurrent flow problem. *Journal of the ACM (JACM)*, 37(2):318–334, 1990.
33. É. Tardos and V. Vazirani. Improved bounds for the max-flow min-multicut ratio for planar and $k_{r,r}$-free graphs. *Information Processing Letters*, 47:77–80, August 1993.
34. S. Tragoudas. *VLSI partitioning approximation algorithms based on multicommodity flow and other techniques.* PhD thesis, University of Texas, Dallas, 1991.