Strong-Diameter Decompositions of Minor Free Graphs

Ittai Abraham
Hebrew University of Jerusalem
Jerusalem, Israel
ittaia@cs.huji.ac.il

Cyril Gavoille*
Laboratoire Bordelais de Recherche en Informatique
University of Bordeaux
Bordeaux, France
gavoille@labri.fr

Dahlia Malkhi
School of Computer Science and Engineering
Hebrew University of Jerusalem
and Microsoft Research, Silicon Valley Center
dalia@microsoft.com

Udi Wieder
Microsoft Research, Silicon Valley Center
uwieder@microsoft.com

ABSTRACT

We provide the first sparse covers and probabilistic partitions for graphs excluding a fixed minor that have *strong* diameter bounds; i.e. each set of the cover/partition has a small diameter as an induced sub-graph. Using these results we provide improved distributed name-independent routing schemes. Specifically, given a graph excluding a minor on r vertices and a parameter $\rho > 0$ we obtain the flowing results: (1) a polynomial algorithm that constructs a set of clusters such that each cluster has a strong-diameter of $O(r^2\rho)$ and each vertex belongs to $2^{O(r)}r!$ clusters; (2) a name-independent routing scheme with a stretch of $O(r^2)$ and tables of size $2^{O(r)}r!\log^4 n$ bits; (3) a randomized algorithm that partitions the graph such that each cluster has strong-diameter $O(r6^r\rho)$ and the probability an edge (u,v) is cut is $O(r\,d(u,v)/\rho)$.

Categories and Subject Descriptors: C.2.1 [Computer-Communication Networks]: Network Architecture and Design – *Distributed networks*; G.2.2 [Discrete Mathematics]: Graph Theory – *Network problems, Graph labeling.*

General Terms: Algorithms, Theory.

Keywords: Compact Routing, Minor Free Graphs.

1. INTRODUCTION

As networks grow large and complex, a key approach in managing information and constructing algorithms is to decompose the network into locality-preserving *clusters*. Then, information and/or management can be divided between the clusters, such that every node is responsible only for clusters for which it belongs. Such decompositions into locality sensitive clusters have become key tools in network and graph theory and span a large body of literature.

Consider an undirected weighted graph $G = (V, E, \omega)$, i.e., $E \subseteq V \times V$ and $\omega : E \to \mathbb{R}^+$. Let $d_G(u,v)$ be the cost of a minimum cost path between u and v where the cost of a path is the sum of weights of its edges. Let $\mathrm{diam}(G) = \max_{u,v} d_G(u,v)$. Given $U \subseteq V$, let $G[U]$ be the induced subgraph whose nodes are U and whose edges are the edges in G whose endpoints both belong to U. Let $B_G(u,\rho) = \{v \mid d_G(u,v) \le \rho\}$. For $A \subseteq V$ and $v \in V$ let $d_G(A,v) = \min_{u\in A} d_G(u,v)$. When G is clear from the context we omit the subscript and write $d(u,v)$.

A *sparse cover* is a set of subsets (clusters) of graph nodes introduced by Awerbuch and Peleg in [8] with the following properties.

Definition 1. *A (k,τ,ρ)* sparse cover *is a set of clusters $\mathcal{C} \subset 2^V$ with the following properties:*

1. *[Cover]: $\forall v \in V$, $\exists C \in \mathcal{C}$ such that $B(v,\rho) \subseteq C$.*
2. *[Small strong-diameter]: $\forall C \in \mathcal{C}$, $\mathrm{diam}(G[C]) \le k\rho$.*
3. *[Sparsity]: $\forall u \in V$, $|\{C \in \mathcal{C} \mid u \in C\}| \le \tau$.*

When a (k,τ,ρ) sparse covers exists for any ρ we say that the graphs admits a (k,τ) *sparse cover scheme*.

Sparse covers are used as a building block for a variety of applications. These include distance coordinates, routing with succinct routing tables [8, 3], mobile user tracking [8], resource allocation [6], synchronization in distributed algorithms [7], and others.

For general graphs, the seminal construction in [8] provides a $(2k-1, 2k\cdot n^{1/k})$ sparse cover scheme for any integer $k \ge 1$. This result asymptotically matches known lower bound that arise from dense graphs with high girth [18]. For certain restricted families of graphs, better covers are known to exist. For example, if the graph is α doubling[1] then $(1+\varepsilon, (1+1/\varepsilon)^{O(\log(\alpha))})$ sparse cover scheme can be constructed for any $\varepsilon > 0$.

[1]A graph is α doubling (or of doubling dimension $\log\alpha$) if every ball of radius r can be covered by at most α balls of radius $r/2$.

*Supported by the projects "GeoComp" and "Alpage"of the ACI Masses de Données.

SPAA'07, June 9–11, 2007, San Diego, California, USA.

Minor-free graphs

The *contraction* of an edge $e = (u, v)$ is the replacement of nodes u, v with a new vertex whose incident edges are the edges other than e that were incident to u or v. A graph H is a *minor* of G if H is a subgraph of a graph obtained by a series of edge contractions of G. A celebrated theorem of Robertson and Seymour states that every (possibly infinite) set of graphs $\mathcal{G}$ that is closed under edge contractions and edge removals could be characterized by a finite set of graphs called its *obstruction set*, where a graph G is in the set $\mathcal{G}$ if and only if none of its minors is contained in the obstruction set.

For example it is well known that planar graphs are exactly all the graphs whose set of minors exclude $K_{3,3}$[2] and K_5. It is natural is ask whether graphs that exclude some fixed minor have better sparse covers than general graphs.

A (k, τ) *weak-diameter sparse cover scheme* is defined as above with one important change: the diameter bound is now imposed on distances in the original graph. That is, a short path between cluster nodes may contain nodes which are **outside** the cluster. Previous work had shown that graphs excluding a fixed minor have an improved weak-diameter sparse covers. By iteratively applying the partitions of Klein *et al* [15] (KPR for short) one can obtain a $(O(r^2), O(2^r))$ weak-diameter sparse cover scheme, where r is the maximal number of vertices in the set of excluded minors. In fact, there are simple planar graphs in which the KPR construction yields clusters with arbitrarily high diameter (see Section 2 below). The challenge of providing minor-free graph decompositions whose clusters have strong-diameter bounds remained open, and is addressed by the present work. Our first result is stated in the following theorem:

THEOREM 1. *Every weighted graph excluding a $K_{r,r}$ minor has a $(O(r^2), 2^{O(r)}r!)$ sparse cover scheme constructible in polynomial time.*

During the preparation of this note it has been brought to our knowledge that independent work in progress [10] has achieved a $(4, O(\log n))$ sparse cover scheme for graphs excluding a fixed minor, and a $(O(1), O(1))$ sparse cover scheme for planar graphs.

Compact Routing

As mentioned above, many applications of sparse covers are known. We highlight one in particular in this paper: the classical problem of compact, loop-free routing. In this problem we consider a distributed network of nodes connected via a network in which each node has an arbitrary network identifier. A routing scheme assigns a routing table to each node such that any source node can route messages to any destination node, given the destination's network identifier. The fundamental trade-off in compact routing schemes is between the *space* used to store the routing table on each node and the *stretch* factor of the routing scheme. The stretch factor is defined as the maximum ratio over all pairs between the length of the route induced by the scheme and the length of a shortest-path between the same pair.

In this paper we assume a network with arbitrary node names. This model is referred as the *name-independent* model because the designer of the routing scheme has no control over node names and thus node names cannot encode any topological information. A model which allows the network designer to choose node names is called the *labeled routing* model. In this version of the problem, the designer of a solution may pick node names that contain (polylogarithmic size) information about their location in the network, like for instance the X, Y-coordinates in a geographic network. Labeled routing is useful in many aspects of network theory, but less so in practice. Knowledge of the labels needs to be disseminated to all potential senders, as these labels are not the addresses by which nodes of an *existing* network, e.g. an IP network, are known. Furthermore, if the network may admit new joining nodes, all the labels might need to be re-computed and distributed to any potential sender. Finally, various recent applications pose constraints on nodes addresses that cannot be satisfied by existing labeled routing schemes. E.g., Distributed Hash Tables (DHTs) require nodes names in the range $[1, n]$, or ones that form a binary prefix.

Abraham et al. provide in [3] the following result. For every n-node unweighted graph excluding a fixed $K_{r,r}$ minor, there exists a polynomial time constructible name-independent routing scheme with constant stretch factor, in which every node v requires routing tables of polylog(n) bits and $O(\log^2 n/\log\log n)$-bit headers. Thorup [17] addressed the problem of labeled routing schemes in planar graphs. He shows the existence of a polylog(n) memory $1+\varepsilon$ stretch labeled routing scheme. Abraham *et al* [1] extend this result to any minor free family. These results cannot be extended to the name-independent domain since in that case it is known that a stretch of 3 is required for trees if less than $\Omega(n \log n)$ bits are used [5].

Our next contribution is stated in the following theorem.

THEOREM 2. *For every n-node unweighted graph of diameter D excluding a $K_{r,r}$ minor, there is a polynomial time constructible name-independent routing scheme, in the fixed port model*[3]*, with stretch $O(r^2)$ and using $O(\log n)$-bit headers, in which every node requires tables of $2^{O(r)}r! \cdot \log D \cdot \log^3 n/\log\log n$ bits.*

Probabilistic Sparse Partitions

Another approach for decomposing a graph is to partition it to disjoint clusters by removing a small number of edges. More precisely, we have the following definitions. A *strong-diameter ρ bounded partition of G* is a partition of V into disjoint clusters $C_1, C_2, \ldots$ such that for each cluster C_i, $\mathrm{diam}(G[C_i]) \leq \rho$. Given a partition P and a node u, let $P(u)$ be the unique cluster that contains u.

DEFINITION 2. *A (k, η, ρ)* probabilistic sparse partition *is a distribution $\mathcal{P}$ on partitions with the following properties:*

1. *[Small diameter]: $\forall P \in \mathcal{P}$, P is a strong-diameter $k\rho$ bounded partition of G.*
2. *[Small probability of cutting an edge]: $\forall u, v \in V$,*

$$\Pr_{P\sim\mathcal{P}}[P(u) \neq P(v)] \leq \eta\, d(u, v)/\rho \ .$$

[2] Let $K_{r,r}$ be the complete bipartite graph with r nodes in each set.

[3] I.e., the port number around each node u is an arbitrary permutation of $\{1, \ldots, \deg(u)\}$ that, as well as the node names, cannot be changed during the design of the routing scheme.

When a (k, η, ρ) probabilistic sparse partition exists for any ρ we say that the graphs admits a (k, η) *probabilistic sparse partition scheme*. If the partitions produced has only a weak-diameter bound we say that the resulting scheme is a *weak-diameter probabilistic sparse partition scheme*.

Sparse partitions play a key role in approximation algorithms, such as multi-commodity flow optimization problems [15]. Klein et al. provide a $(O(r^3), O(r))$ weak-diameter probabilistic sparse partition scheme for graphs excluding a $K_{r,r}$ minor. Fakcharoenphol and Talwar improve in [13] to a $(O(r^2), O(r))$ weak-diameter probabilistic sparse partition scheme.

Our improvement is to provide a strong-diameter bound, as follows.

THEOREM 3. *For every weighted graph excluding a $K_{r,r}$ minor there exists a polynomial time sampleable $(O(r6^r), O(r))$ strong-diameter probabilistic sparse partition scheme.*

Furthermore, for any $\rho > 0$, it is possible to find in polynomial time a partition with strong-diameter $O(r6^r\rho)$, where the total weight of edges crossing the partition is $O(\sum_{(u,v)\in E} d_G(u,v)/\rho)$.

We envision that this result may play an important role in further optimization problems and graph embeddings into dominating trees.

1.1 Summary of Contributions

In summary, the paper provides the following results for any $K_{r,r}$-minor-free graph.

- There is a sparse cover of the radius-ρ balls around every node, for every $\rho > 0$, such that each cluster in the cover has strong-diameter $O(r^2\rho)$, and every node belongs to $2^{O(r)}r!$ clusters.
- There is a name-independent routing scheme with $O(r^2)$ stretch and tables of size less than $2^{O(r)}r!\log^4 n$ bits.
- There is a sparse partition where, for every $\rho > 0$, the probability an edge (u, v) is cut is $O(r\, d(u, v)/\rho)$ and the strong-diameter of each cluster $O(r6^r\rho)$.

All the schemes are polynomially constructible, and do not assume that r is known is advance.

2. LARGE DIAMETER KPR CLUSTERS

In this section, we briefly review the KPR [15] algorithm, and exemplify its unbounded diameter.

For planar graphs, KPR performs three recursive tree cuts into *stripes* of height ρ. Each tree is a breadth-first search tree (BFS) of a connected component that is left from the previous cut.

More precisely, initially start with G_1, the whole graph. Select an offset $h_1 \in [0, \rho - 1]$ uniformly at random. Build a BFS tree T_1 on G_1, rooted at an arbitrary node τ_1. Slice T_1 into stripes of height ρ: The i-th stripe contains nodes whose BFS distance from the root of T_1 is between $h_1 + i\rho$ and $h_1 + (i+1)\rho - 1$. Recurse on any connected component G_2 contained within any stripe. The recursion continues for r phases.

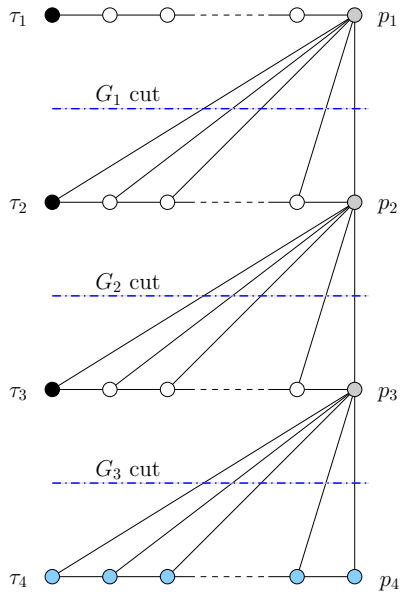

Figure 1: Example graph in which KPR partition has arbitrarily large diameter.

Figure 1 depicts a simple outerplanar graph (so excluding $K_{2,3}$ and K_4) in which the KPR cut for $r = 3$ results in a final cluster containing nodes τ_4 and p_4 whose strong-diameter is arbitrarily large. This example also shows that adding more iterations of KPR-cuts, say an arbitrary $r > 3$, does not remedy the situation even in the planar case.

For arbitrary $r < \rho$ iterations, the graph is composed of $r + 1$ paths $\tau_1 \to p_1$, $\tau_2 \to p_2$, ..., $\tau_{r+1} \to p_{r+1}$ of length respectively $h_i + k\rho - 1$ where $k \geq 1$ is an arbitrary large integer, and $h_i \in [0, \rho - 1]$ is an offset. In addition there edges are between p_i and each node of the path $\tau_{i+1} \to p_{i+1}$, $i \in [1, r]$. We now explain the example in detail.

In the original graph G_1, the BFS tree T_1 is rooted at τ_1. We note that node p_1 has distance less than ρ to all the nodes below the cut line marked 'G_1 cut'. The distance in G_1 from τ_1 to p_1 is $d_{G_1}(\tau_1, p_1) = h_1 + k\rho - 1$. Hence, choosing an offset of h_1, the k-th stripe of T_1 may consists of all nodes under the G_1 cut, which forms a connected component G_2 induced by the nodes of $G_1 \setminus (\tau_1 \to p_1)$.

Note that, despite the fact that nodes τ_2 and p_2 have distance two in G_1 (going through p_1), their distance in G_2 is arbitrarily large ($d_{G_2}(\tau_2, p_2) = h_2 + k\rho - 1$).

Continuing on, we build the next steps in a similar manner. The tree cut of T_2 rooted at τ_2 in G_2 might perform the cut marked as 'G_2 cut' by choosing the offset h_2, leaving all nodes below it as the connected component G_3. And so on. Finally, the distance in G_{r+1} between the nodes of the bottom path $\tau_{r+1} \to p_{r+1}$ ($\tau_4 \to p_4$ on the picture) have arbitrarily large distance: All the nodes $p_1, \ldots, p_r$ that shorten the distance from τ_{r+1} to p_{r+1} have been cut away from the cluster.

Observe that the unbounded strong-diameter of the KPR decomposition occurs for a specific choice of offsets $h_1, \ldots, h_r$. However we can consolidate the counter-example

by rebuilding the previous graph for every offset sequences, and by identifying all the nodes τ_1. The resulting graph is still outerplanar. Clearly, any choice of r offsets during the randomization provides at least an unbounded strong-diameter component.

3. SPARSE COVER WITH STRONG DIAMETER

We provide a graph cover procedure that yields clusters have a bounded strong-diameter. The algorithm uses the KPR [15] paradigm: recursively cutting strips from BFS trees. Unlike KPR, our algorithm takes at each iteration, not only the desired strip, but also grows balls around desired "core" portions of the strip. Our construction and proof can be seen as an enhancement of the arguments in [3]. Intuitively, our construction ensures that each final cluster G_{r+1} will have a "core" denoted H_{r+1}, such that (1) each node in G_{r+1} is "close" to H_{r+1} (2) each node in H_{r+1} is "far" from $G \setminus G_{r+1}$. Hence if G_{r+1} has a long enough path then (1) would imply that there exist r nodes in the H_{r+1} that are far away from each other. Then (2) would imply the exitance of a $K_{r,r}$ minor.

The main result of this section is stated in the following theorem.

THEOREM 1. *Every weighted graph excluding a $K_{r,r}$ minor has a $(O(r^2), 2^{O(r)}r!)$ sparse cover scheme constructible in polynomial time.*

PROOF. The algorithm receives a graph G and a parameter ρ. We begin with some notation. In order to be consistent we fix an arbitrary labeling of the vertices. Given a subgraph $G_i \subseteq G$, let τ_i be the vertex with minimal label in G_i. Let T_i be the unique breadth-first-search (BFS) spanning tree of G_i, rooted at τ_i where parent vertices have the minimal possible label. We can now slice T_i to slices of size ρ, let $S_{i,j}$ be the jth slice of T_i

$$S_{i,j} = \{v \in G_i \mid j\rho \leq d_{G_i}(\tau_i, v) < (j+1)\rho\}$$

For a set U in a graph G and a distance c let $B_G(U,c) = \{v \mid \exists u \in U, d_G(u,v) \leq c\}$.

Cover algorithm:

Initially $G = G_1 = H_1$.

`cover-`(G_i, H_i): If $i = r+1$ then return G_i as one of the clusters of the cover. Otherwise, for every integer j and every connected component G' of the set

$$B_{G_i}(S_{i,j} \cap H_i, i\rho)$$

set $G_{i+1} = G'$, $H_{i+1} := G' \cap S_{i,j} \cap H_i$ and execute `cover-`(G_{i+1}, H_{i+1}).

CLAIM 1. *For every v there exists unique indexes $j_1, \ldots, j_r$ such that $v \in H_{r+1}$.*

PROOF. By induction. Clearly $v \in H_1$. Given $v \in H_i$ there exist a unique index j_i such that $v \in S_{i,j_i}$. Let G_{i+1} be the connected component of $B_{G_i}(S_{i,j_i} \cap H_i, i\rho)$ that contains $v \in H_i$ so that $v \in H_{i+1} = G_{i+1} \cap S_{i,j_i} \cap H_i$ as required. □

So given $v \in H_{r+1}$ this implicitly defines a series of subgraphs $G_1, H_1, \ldots, G_r, H_r, G_{r+1}, H_{r+1}$ induced by the cover algorithm.

Property 1. *(Cover).* From Claim 1 there exists $H_{r+1} \ni v$. By induction, given $B_G(v,\rho) \subseteq G_i$ and $v \in H_{i+1}$ it follows from the definition of G_{i+1} that $B_G(v,\rho) \subseteq G_{i+1}$

Property 2. *(Sparse clusters).* For each graph G_i that v belongs to, it belongs to at most $(2i+1)$ graphs G_{i+1} with different indexes j due to the use of a ball of radius $i\rho$ on each stripe $S_{i,j}$. Hence for each $i \in [1,r]$, by induction, a node belongs to at most $\prod_{1 \leq j \leq i}(2j+1) \leq 2^i(i+1)!$ graphs G_i. Therefore a node belongs to at most $2^{O(r)}r!$ clusters.

Property 3. *(Strong-diameter).* Fix some cluster G_{r+1}. We will now show that if G_{r+1} has a strong diameter of more than $4(r+1)^2\rho$ then G contains a $K_{r,r}$ minor. If there exist two nodes such that $d_{G_{r+1}}(y_1, y_r) > 4(r+1)^2\rho$ then their shortest path in G_{r+1} can be partitioned into $r-1$ segments using r points $y_1, \ldots, y_r \in G_{r+1}$ such that the balls $B_{G_{r+1}}(y_i, 2(r+1)\rho)$ are pairwise disjoint. Let x_i be the closest point in H_{r+1} to y_i (so $d(x_i, y_i) \leq (r+1)\rho$) then by the construction of G_{r+1}, the balls $B_{G_{r+1}}(x_i, (r+1)\rho)$ are pairwise disjoint.

We conclude the theorem by using the following lemma.

LEMMA 1. *If there exists points $x_1, \ldots, x_r \in H_{r+1}$ such that the balls $B_{G_{r+1}}(x_i, (r+1)\rho)$ are pairwise disjoint then G contains a $K_{r,r}$ minor.*

PROOF. Such a minor is composed of r sets called the *left super-nodes*, denoted as $L_1, L_2, \ldots, L_r$, such that $x_i \in L_i$; and from r sets called the *right super-nodes*, denoted $V_1, V_2, \ldots, V_r$. Each super-node is a connected sub-graph of G, all sets are pairwise disjoint and there is an edge connecting each set from the first group with a each set from the second group.

For the analysis we use the following notation: for $u \in G_{i+1}$ let $\text{tail}_i(u)$ be the unique path on $T_i \cap G_{i+1}$ from u towards the root τ_i.

The left super-nodes: For each $i \in [1,r]$, let $L_i = \bigcup_{j \in [1,r]} \text{tail}_j(x_i)$. For any $i,j \in [1,r]$, by construction, $\text{tail}_j(x_i) \subseteq G_{j+1}$ and its length in G_{j+1} is at most $(j+1)\rho$. Observe that each L_i is a set of paths in G that are connected at x_i.

The right super-nodes: For all $i \in [1,r]$ let U_i denote the subtree of T_i formed by the paths on T_i for all $j \in [1,r]$ from each x_j to τ_i. The right super nodes are V_i for $i \in [1,r]$, where $V_i = U_i \setminus G_{i+1}$. Observe that each V_i induces a connected subtree of G.

The super edges are the edges in T_i connecting each V_i with each $\text{tail}_i(x_j) \in L_j$ for each $i,j \in [1,r]$.

We now show that the sets are pairwise disjoint.

For any $i,j \in [1,r]$ and $j < \ell$ we claim that $\text{tail}_j(x_i) \subseteq G_\ell$ and that $\text{tail}_j(x_i)$ is disjoint from the right node $V_{\ell-1}$. The proof is by by induction on ℓ. For $\ell = j+1$ this is true by construction.

For $j+1 < \ell$, by the induction hypothesis we have $\text{tail}_j(x_i) \subseteq G_{\ell-1}$. Since $x_i \in H_\ell$ and the length of $\text{tail}_j(x_i)$ is at most $(j+1)\rho$ it follows that $\text{tail}_j(x_i) \subseteq G_\ell$ and that it is disjoint from $V_{\ell-1}$. This follows since $j+1 \leq \ell-1$ and the fact that $B_{G_{\ell-1}}(x_i, (\ell-1)\rho) \subseteq G_\ell$.

Therefore the set L_i is contained in $B_{G_{r+1}}(x_i, (r+1)\rho)$ so for all $\ell \in [1,r]$ and all $i < j \in [1,r]$, $L_i \cap L_j = \emptyset$ due to the assumptions that the balls $B_{G_{r+1}}(x_i, (r+1)\rho)$ are

pairwise disjoint. From the inductive claim above, for all $i \in [1,r]$ and all $j \leq \ell \in [1,r]$, $\text{tail}_j(x_i) \cap V_\ell = \emptyset$. Finally, for all $i \in [1,r]$ and all $\ell < j \in [1,r]$, the set V_ℓ is clearly disjoint from $\text{tail}_j(x_i)$ and from V_j since by construction $V_\ell = U_\ell \setminus G_{\ell+1}$ and $V_j \cup \text{tail}_j(x_i) \subseteq G_j \subseteq G_{\ell+1}$. □

4. NAME-INDEPENDENT ROUTING

In this part we consider the problem of routing messages between any pair of nodes of an unweighted graph G with precomputed compact routing tables. The performances of the routing scheme is measured in term of the size of the local routing tables and the maximum *stretch*, i.e., the ratio between the length of the route from x to y and the minimum possible route length, $d_G(x,y)$.

We concentrate our attention on *name-independent* routing schemes, that is node names cannot be relabeled to optimize routing tables. *Labeled* routing scheme of stretch $1+\varepsilon$ and with polylogarithmic size routing tables, labels, and headers are known for weighted graphs excluding a fixed minor [1], whereas any name-independent routing scheme on unweighted stars (depth one trees, so excluding K_3) requires a stretch at least 3 if less than $\Omega(n \log n)$ bits per node are used [5].

We assume that G is unweighted, since it has been proved in [4] that there are stars with edge cost 1 or k for which every name-independent routing scheme of stretch $< 2k+1$ requires routing tables of $\Omega((n \log n)^{1/k})$ bits, for every integer $k \geq 1$.

In the remaining of the paper, we will assume that the n node names of G range arbitrary in $\{1, \ldots, n^{O(1)}\}$, i.e., are on $O(\log n)$ bits. The scheme extends easily to longer names by the use of hashing techniques.

Thanks to Theorem 1 we can show:

THEOREM 2. *For every n-node unweighted graph of diameter D excluding a $K_{r,r}$ minor, there is a polynomial time constructible name-independent routing scheme, in the fixed port model*[4]*, with stretch $O(r^2)$ and using $O(\log n)$-bit headers, in which every node requires tables of $2^{O(r)} r! \cdot \log D \cdot \log^3 n / \log\log n$ bits.*

First let us outline the technique of hierarchical routing schemes introduced by Awerbuch and Peleg [8, 9]. Let us assume that there exist k and τ such that, for every $\rho > 0$, the graph G has a (k, ρ, τ) sparse cover, and let $\mathcal{C}_\rho$ denote this cover. Then, routing in G can be done by considering the family of covers $\mathcal{F} = \{\mathcal{C}_1, \ldots, \mathcal{C}_{2^i}, \ldots, \mathcal{C}_{2^{\lceil \log D \rceil}}\}$. More precisely, for each cover $\mathcal{C}_{2^i} \in \mathcal{F}$ and for each cluster $C \in \mathcal{C}$, we root a shortest path spanning tree T_C of $G[C]$, so of depth at most $k2^i$. Let us call $\mathcal{T}$ the collection of all these trees. Roughly speaking, the routing task for a source u consists in seeking the target v in each tree of $\mathcal{T}$ it belongs to.

Actually, u needs to seek v only in $\lceil \log D \rceil + 1$ trees in nondecreasing depth, each tree spanning the ball $B_G(u, 2^i)$ for some $i \in \{0, \ldots, \lceil \log D \rceil\}$. If each try can be done within a route of length proportional to the depth of the tree, then it is not difficult to check that the resulting stretch of the route from u to v is $O(k)$, the cluster covering $B_G(u, 2^i)$ being of diameter at most $k2^i$. Overall, if a tree routing scheme for seeking v can be implemented with M-bit routing tables, then the routing scheme for G uses at most $O(\tau \cdot \log D \cdot M)$ bits, each node participating in at most τ tree routings for each of the $O(\log D)$ covers of $\mathcal{F}$.

By Theorem 1, we have $\tau = 2^{O(r)} r!$ and $k = O(r^2)$. Therefore Theorem 2 can be proved by designing a routing scheme with $M = O(\log^3 n / \log\log n)$-bit routing tables for seeking any destination in a tree along routes of length proportional to its depth. Unfortunately, such tree routing schemes cannot be applied as a black box and plugged to the hierarchical scheme for G. Indeed, routing from u to v in some partial tree T of G inevitably requires to visit some nodes outside T: forcing routing to use only edges of T is equivalent to routing in a weighted tree T' spanning G where edges of $T' \setminus T$ have some large costs. And we have seen that routing trees with edge weights in $\{1,k\}$ requires $\Omega((n \log n)^{1/k})$ bits for stretch $\Theta(k)$ [4]. It follows that routing in a cluster C interact with nodes not in C. This is source of several complications since a node might be concerned with (and its routing table possibly charged for) the routing to some trees in which it does not belong to. Potential a node maybe a neighbor of all the $|\mathcal{T}| = \Omega(n)$ trees. To solve the problem, we will use a modified version of the single-source unweighted tree routing of [2] combined with the low density of minor-free graphs to balance routing information. This last step makes our routing scheme available only for *unweighted* graphs.

An *L-error reporting* routing scheme for a subgraph C of G is a routing scheme such that, for all $u \in C$ and v of G: if $v \in C$, then the route from u to v has cost at most L, and if $v \notin C$, then the routing from u to v reports to u a *failure* mark in the header after a loop of cost at most L. In order to prove Theorem 2, our goal is to construct, for each depth-h tree $T \in \mathcal{T}$, a space efficient $O(h)$-error reporting routing scheme (cf. Lemma 3 below).

An α-orientation of a graph is an orientation of its edges such that every node has out-degree at most α.

LEMMA 2. *Any graph excluding a $K_{r,r}$ minor has an $\Theta(r\sqrt{\log r})$-orientation that can be computed in linear time.*

PROOF. Graph excluding a fixed minor are closed under taking induced subgraphs. It is known that the n-node graphs excluding a K_r minor have no more that $f(r) \cdot n$ edges [16] where $f(r) = \Theta(r\sqrt{\log r})$. Therefore, an $f(r)$-orientation can be easily obtained in linear time by pruning the graph with the minimum degree node. Now, the family of graphs excluding a K_{2r} minor contains all the graphs excluding a $K_{r,r}$ minor. Thus any n-node graph excluding a $K_{r,r}$ minor has no more than $f(2r) \cdot n$ edges, and so an $\Theta(r\sqrt{\log r})$-orientation computable in linear time. □

The *sparsity* of a collection of trees of a graph G is the maximum number of trees a node of G belongs to. The key lemma is the following:

LEMMA 3. *Let $\mathcal{T}$ be a collection of trees of sparsity σ in an n-node graph G with an α-orientation. Then, one can construct in polynomial time for each node of G a routing table of $O(\sigma \cdot \log n \cdot (\alpha + \log^2 n / \log\log n))$ bits, such that each depth-h tree of $\mathcal{T}$ has a $8h$-error reporting routing scheme using $O(\log n)$-bit headers.*

The proof of Theorem 2 is completed thanks to Lemma 3 by observing that the sparsity of the collection $\mathcal{T}$ is $\sigma =$

[4]I.e., the port number around each node u is an arbitrary permutation of $\{1, \ldots, \deg(u)\}$ that, as well as the node names, cannot be changed during the design of the routing scheme.

$2^{O(r)}r! \cdot \log D$, and that $\alpha = O(r\sqrt{\log r})$ (Lemma 2), so providing a $2^{O(r)}r! \cdot \log D \cdot \log^3 n / \log\log n$-bit routing scheme for G as claimed.

PROOF. Let $\mathcal{U}$ be the set of node names for G, $|\mathcal{U}| \leq n^{O(1)}$. Consider a depth-$h$ rooted tree T of $\mathcal{T}$, and let $m \leq n$ be the number of nodes of T, and s be the root for T. Consider a source u of T, and let v be any node of G. For simplicity, we confuse the nodes with their names, that is we assume that $u, v \in \mathcal{U}$.

Let $\psi : \mathcal{U} \to \mathcal{P}$ be some universal hashing function mapping the names of $\mathcal{U}$ to $\mathcal{P} = \{1, \ldots, p\}$ where p is some prime such that $m \leq p < 2m$. Such function ψ can be implemented with a degree-$O(\log p)$ polynomial of the field $\mathbb{Z}_p$ such that there are at most $O(\log p)$ collisions [11], thus using $O(\log^2 m)$ bits.

The outline of the L-error reporting scheme for T from u to v is the following:

1. Node u hashes v in $\psi(v) \in \mathcal{P}$.
2. Node u routes a message to a node w of T, thanks to a labeled tree routing scheme $\mathcal{L}_1$, whose label in $\mathcal{L}_1$ is precisely $\psi(v)$.
3. Node w is in charge of the labels, for a tree routing scheme $\mathcal{L}_2$, of all the nodes z of T such that $\psi(z) = \psi(v)$.
4. If v is not one of such node z of T, then a failure mark is sent back from w to u using routing $\mathcal{L}_1$.
5. Otherwise, w routes to v using routing $\mathcal{L}_2$.

The routing schemes $\mathcal{L}_1, \mathcal{L}_2$ are based on a specific port labeling of T, called *virtual port labeling* w.r.t. T. They requires that ports number i of u leads to the i-th heaviest child of u in T (the port to the parent of u is fixed to 0). So, given the label of the current node x and the label of the destination y, say $\ell_2(x)$ and $\ell_2(y)$, the scheme $\mathcal{L}_2$ computes the virtual port number i of the edge on the path from x to y in T. For the scheme $\mathcal{L}_2$, we will use the label tree routing scheme of [14], that uses $O(\log m)$-bit labels.

Unfortunately, *real* port numbers of u are fixed and range in $\{1, \ldots, \deg_G(u)\}$, and $\deg_G(u) \neq \deg_T(u)$ in general. To overcome this problem we distribute a translation table from virtual to real port numbers over all the neighbors of u, potentially charging some nodes of G that are not in T. To prevent the overload of some nodes, we use the α-orientation of G.

Let $\text{port}(x, y)$ be the real port number of the edge from x to y in G. Consider a node z with directed neighbors (according to the α-orientation) $z^1, \ldots, z^\alpha$, and let z_i be the i-th heaviest child of z.

For the port translation, node z stores: 1) the real port number of its parent for each tree of $\mathcal{T}$ it belongs to; 2) the real port to z_p with $p = \text{port}(z, z^i)$ for each $i \in \{1, \ldots, \alpha\}$, and for each tree of $\mathcal{T}$ node z belongs to; 3) $\text{port}(z^i, z^i_p)$, where $p = \text{port}(z^i, z)$, for each $i \in \{1, \ldots, \alpha\}$, and for each tree of $\mathcal{T}$ node z^i belongs to. We check that overall the memory requirements for the port translation is $O(\sigma \cdot \alpha \cdot \log n)$ bits per node of G.

The port translation in a node x of T is performed as follows: if the virtual port is 0, then the real port of the parent of T is returned. If the virtual port is one of the real port to $x^1, \ldots, x^\alpha$, then the real port is returned. Otherwise, if the virtual port is p, then a message from x on port p, specifying the tree T of $\mathcal{T}$, is sent, and let z be its neighbor. Note that this edge is incoming in x in the orientation, and outgoing from z. Therefore, z knows the real port number of the p-th child of x in the tree T. In other words, the routing from x to its parent is done in 1 step, whereas for a child it is done in at most three steps.

The L-error reporting routing scheme from u to v in T has the following performances: 1) if $v \notin T$, then the route has length $L \leq d_T(u, s) + 3d_T(s, w) + d_T(w, s) + 3d_T(s, u) \leq 8h$. 2) if $v \in T$, then the route has length $L \leq d_T(u, s) + 3d_T(s, w) + d_T(w, s) + 3d_T(s, v) \leq 8h$. Therefore, the scheme is $8h$-error reporting.

It remains to describe the tree routing scheme $\mathcal{L}_1$ (see [2] for details). The scheme is based on two numbers, $c(x)$ and $q(x)$, we assign with each node x of T. The first, called the *charge of* x, represents the total number of values $\psi(v)$ mapped on the nodes of T_x, the subtree T of root x. (So for the root s, $c(s) = |\mathcal{P}| = p$). The second one denotes the number of values $\psi(v)$ that are mapped to node x. These two numbers satisfy that, for every x, $c(x) = \sum_{y \in T_x} q(y)$.

Given the numbers $c(x)$ and $q(x)$ one can then route through a *modified* DFS number $f(x)$ associated with each x and defined by: for the root $f(s) := 1$, and $f(x_i) := f(x) + q(x) + \sum_{j<i} c(x_j)$, where x_i is the i-th child of x. (This matches to the standard DFS numbering if $q(x) = 1$ for every x.)

Now the routing $\mathcal{L}_1$ is done similarly to Interval Routing Scheme. Let w be the node in charge of $\psi(v)$. Assume that w is a descendant of some node x, initially $x = s$. It is easy to see that:

1. either $\psi(v) \in [f(x), f(x) + q(x))$, and $w = x$, i.e., the key of v is stored by x;
2. or w is a descendant of x_i where $\psi(v) \in [f(x_i), f(x_{i+1}))$, and thus the routing in x must answer port i.

So the routing from x to $\psi(v)$ is well defined if x is aware of $f(x)$, $q(x)$, and of the vector $\vec{c}(x) = (c(x_1), c(x_2), \ldots)$ of charges of x's children. Indeed the numbers $f(x_i)$ and $f(x_{i+1})$ can be computed from $f(x)$, $q(x)$, and from $\vec{c}(x)$.

It is proved in [2] that $c(x)$ and $q(x)$ can be polynomially computed in a particular way so that $q(x) = O(\log m / \log\log m)$ and $\vec{c}(x)$ contains at most $O(\log^2 m / \log\log m)$ distinct values, and so coded with $O(\log^3 m / \log\log m)$ bits.

A node x belonging to T stores $q(x)$, $c(x)$, $\vec{c}(x)$, and all the labels and names for which x is in charge: this is $O(q(x) \log m)$ values, since there are $O(\log m)$ nodes of T that can collide in the same node x. Labels and names are on $O(\log m)$ bits, therefore the storage for $\mathcal{L}_1$ in x is $O(\log^3 m / \log\log m)$ bits for T.

Thus a node of G stores $O(\sigma \cdot \log^3 n / \log\log n)$ bits for $\mathcal{L}_1$ plus $O(\sigma \cdot \alpha \cdot \log n)$ bits for the port translation table and the scheme $\mathcal{L}_2$. Hashing functions for each tree represents $O(\sigma \cdot \log^2 n)$ bits. In total, a node of G stores $O(\sigma \cdot \log n(\alpha + \log^2 n / \log\log n))$ bits as claimed.

We check that all the headers are on $O(\log n)$ bits, completing the proof of the lemma. □

5. PROBABILISTIC SPARSE PARTITIONS

In this section we present probabilistic sparse partitions with strong-diameter guarantees. The overall approach is again similar in sprit to KPR. However there are three major differences. First, our phase i stripes are of width $6^i\rho$,

while KPR always chooses width ρ. Second, after the initial cuts we use a cone based approach (see [12]) to "carve out" an appropriate "core" from each stripe. Third and most importantly, some nodes end up associated with clusters that are outside of their stripe! specifically, the nodes of a stripe i that do no get assigned to the ith "core" will be associated with the nodes of the $(i+1)$ stripe.

THEOREM 3. *For every weighted graph excluding a $K_{r,r}$ minor there exists a polynomial time sampleable $(O(r6^r), O(r))$ strong-diameter probabilistic sparse partition scheme.*

Furthermore, for any $\rho > 0$, it is possible to find in polynomial time a partition with strong-diameter $O(r6^r\rho)$, where the total weight of edges crossing the partition is $O(\sum_{(u,v)\in E} d_G(u,v)/\rho)$.

Our proof strategy focuses on an unweighted graph first. We describe the partition procedure which is to be performed r times. The following describes the k-th iteration step by step where $k \in [1, r]$.

1. Denote by G_{k-1} the current graph. $G_0 = (V, E)$ is the base of the recursion. Sample two numbers h_k, ℓ_k uniformly and independently from $[0, \rho - 1]$.

2. Perform BFS from an arbitrary root node $s \in G_{k-1}$: $T_k = \mathrm{BFS}(G_{k-1}, s)$

3. Divide into layers: $L_k(i) = \{u \mid ib_k\rho + \ell_k \leq d_{G_{k-1}}(u,s) < (i+1)b_k\rho + \ell_k\}$, where $b_k = 6b_{k-1} = 6^k$. We omit i in most cases below, and mention it explicitly only when needed.

4. For each layer $L_k(i)$ let $M(L_k(i))$ be the set of nodes at its middle, i.e., $M(L_k(i)) = \left\{u \in L_k(i) \mid d_{G_{k-1}}(u, L_k(i-1)) = (b_k/2)\rho\right\}$. When $L_k(i)$ is clear from the context we may abbreviate notation and write M_k.

5. Define a "cone" distance function $\gamma_k(\cdot,\cdot)$ on the directed edges of G_k with respect to T_k. Specifically for a directed edge $u \to v$ let $\gamma_k(u,v) = 0$ if u is the unique parent of v in the tree T_k and otherwise the distance equals the original distance $\gamma_k(u,v) = \omega(u,v) = d_G(u,v)$. Notice that γ_k is *not symmetric*.

   We can extend γ_k to nodes that are not connected by an edge. Specifically, let $\ell_k(u,v)$ be the cost of the minimal cost directed path from u to v where the cost of a directed path is the sum of the weights of its directed edges according to γ_k. We can also extend the notion of a ball to a cone by defining for a set U and a distance c, $B_G(U, c, \gamma_k) = \{v \mid \exists u \in U, \ell_k(u,v) \leq c\}$.

6. Define $S_k^+(i)$ to be the set of nodes within $L_k(i)$ with γ_k distance of at most $b_k\rho/2 + h_k$ from $M(L_k(i))$. $S_k^+(i) = B_{L_k(i)}(M(L_k(i)), b_k\rho/2 + h_k, \gamma_k) = \{u \mid u \in L_k(i),\ \gamma_k(M(L_k(i)), u) \leq b_k\rho/2 + h_k\}$. Note that $S_k^+(i)$ is a set grown around M_k which is at the middle of $L_k(i)$. It may not include all of $L_k(i)$ but it has the property that if u is included then so are its children in $T_i \cap L_k(i)$.

7. After performing the previous steps to *all* layers in the decomposition, add all **unassigned** nodes from $L(i+1)$ which were not included in $S_k^+(i+1)$ into the set $S_k(i)$.
   $S_k(i) = S_k^+(i) \cup \left\{L_k(i+1) \setminus S_k^+(i+1)\right\}$

8. Note that now the sets $S_k(i)$ partitions G_{k-1}. For each i recurse on every connected component X_i of $S_k(i)$: $G_k \in X_i$.

CLAIM 2. *In any iteration $k \in [1, r]$ of the algorithm, if $u, v \in G_{k-1}$ then the probability that the k-th iteration cuts an edge (u,v) into different clusters is at most $2/\rho$.*

PROOF. In each execution of the procedure there are two ways an edge u, v could be cut. The first is that u, v are cut in stage (3); i.e. $u \in L_k(i)$ while $v \in L_k(i+1)$ for some i. The probability the edge is cut by the layers is at most $1/\rho$ due to the randomness of ℓ_k. The second way in which (u,v) might be cut, given $u, v \in L_k(i)$, is if $u \in S_k^+(i)$ and $v \notin S_k^+(i)$. In other words it must be the case that one of the nodes (say w.l.o.g u) has a small γ_k distance from M_k while node v has a large γ_k distance from M_k. Note however that

$$|\gamma_k(M_k, u) - \gamma_k(M_k, v)| \leq d_G(u,v)$$

The threshold distance for inclusion in $S_k^+(i)$ is $b_k\rho/2 + h_k$ where $h_k \in [0, \rho - 1]$ is chosen uniformly at random. It follows that given that $u, v \in L_k(i)$, the probability (u,v) is cut at most $1/\rho$, which concludes the proof of the lemma. □

There are a r recursive calls so by the union bound the probability of an edge (u,v) being cut is at most $2r \cdot d_G(u,v)/\rho$.

REMARK 1. *The expected total weight of a cut in each iteration k is $2W_k/\rho$ where W_k is the total weight of edges in G_k. There are ρ^2 possibilities for a choice of h_k, ℓ_k. At least one value of h_k, ℓ_k yields a partition where the weight of cut edges is at most $2W_k/\rho$. Thus an exhaustive search would yield a cut with this value. If this is done in every recursive call then the total weight of cut edges is at most $2rW_0/\rho$.*

We are now left with proving that the diameter of each component is $O(\rho)$. We do this by showing that if there are two nodes in G_r such that the distance between any two of them is greater than $12rb_r\rho$ (a constant which depends on r but not on $|V(G)|$) then the graph contains a $K_{r,r}$ minor.

From now one we omit the notation that states which stripe we are talking about (the subscript i in the previous section). The following lemma characterizes the properties we will need in order to show the existence of the $K_{r,r}$ minor. Fix some iteration k.

LEMMA 4. *Each node $u \in G_k$ has an anchor $a_k(u) \in M_k$ such that $a_k(u) \in G_k$ and $d_{G_k}(u, a_k(u)) \leq (3b_k/2 + 2)\rho$.*

PROOF. Consider the construction of G_k out of G_{k-1}. The node u can be assigned to G_k either in Step (6) or Step (7) of the construction. If it were assigned in Step (6) then there is a node $a(u)$ such that $\gamma_k(a(u), u) \leq (b_k/2 + 1)\rho$. The shortest path includes at most $\frac{b_k}{2}\rho + (\frac{b_k}{2} + 1)\rho$ edges of T_k which have a γ_k distance of 0, therefore $d_{G_k}(a(u), u) \leq (b_k + 1 + b_k/2 + 1)\rho = (3b_k/2 + 2)\rho$.

If u was assigned to G_k in Step (7) then all its parents in the BFS tree were also assigned to G_k, therefore the path to the root of T_k reaches a node in M_k after distance at most $b_k\rho$. □

LEMMA 5. *Let $u \in M_k$. For every $v \in G_{k-1}$ such that $d_{G_{k-1}}(u,v) \leq b_k\rho/2$ it holds that $v \in G_k$. In other words a ball around u in G_{k-1} of radius $b_k\rho/2$ is contained in G_k.*

PROOF. Step (6) above includes in S_k^+ all the nodes at distance $b_k\rho/2$, thus $v \in S_k^+$. The lemma then follows since the path between u and v is contained in S_k. □

5.1 The Super-nodes

Assume there are two nodes $x, y \in G_r$ such that $d_{G_r}(x,y) \geq 12rb_r\rho$. There must be therefore r nodes $x = x_1, x_2, \ldots, x_r = y$ in G_r such that $d_{G_r}(x_i, x_j) \geq 12b_r\rho$ for every $i \neq j$. We show that this implies that the graph contains a $K_{r,r}$ minor which contradicts the assumption that G is $K_{r,r}$ free. Such a minor is composed of r sets denoted as $B_1, \ldots, B_r$ such that $x_i \in B_i$ and r sets $R_1, \ldots, R_r$ such that each set is a connected sub-graph of G, all sets are disjoint and there is an edge connecting B_i and R_j for every i, j. This yields a contradiction since each set could be contracted to a single node creating a $K_{r,r}$ minor.

The Set B_i

The node x_i has an anchor in $a_r(x_i) \in M_r$. Call this node a_r (for brevity we omit the subscript i), and denote by A_r the path between x_i and a_r . The node a_r has an anchor $a_{r-1}(a_r) \in M_{r-1}$. Call this node a_{r-1} and define recursively $a_{j-1} = a_{j-1}(a_j)$, and A_j to be the path between a_{j-1} and a_j.

Let $u \in M_k$. Define $\text{tail}_k(u)$ to be the path in T_k which connects u to the upper boundary of L. In other words $\text{tail}_k(u)$ is a path of length at most $b_k/2$ in T_k starting from u towards the root. Now define:

$$B(k) = \bigcup_{j=1}^{k} A_j \cup \text{tail}_j(a_j)$$

The set B_i is now defined as $B(r)$. Clearly the induced graph $G[B_i]$ is connected. This however turns out not to be enough.

LEMMA 6. *All the nodes in B_i belong to G_r. Furthermore $\text{diam}_{G_r}(B_i) \leq 3b_r\rho$.*

PROOF. We prove that $B(k) \subseteq G_k$ by induction on k. For the base case we have $B(1) = A_1 \cup \text{tail}_1(a_1)$ where $A_1 \subseteq G_1$ by Lemma 4. We have that $\text{tail}_1(a_1) \subseteq G_1$ by Lemma 5. By the induction hypothesis we have that $B(r-1) \subseteq G_{r-1}$. Furthermore, Lemma 4 implies that $A_r \subseteq G_r$ and Lemma 5 implies that $\text{tail}_r(a_r) \subseteq G_r$. It remains therefore to show that $B(r-1) \subseteq G_r$. Let $u \in B(r-1)$. By the induction hypothesis $d_{G_{r-1}}(a_r, u) \leq 3b_{r-1}\rho$. We have that $3b_{r-1} \leq b_r/2$ so by Lemma 5 $B(r-1) \subseteq G_r$. Furthermore $\text{diam}_{G_r} B(r) \leq (3b_r/2+1)\rho + 3b_{r-1}\rho \leq 3b_r\rho$. □

The Set R_j

The Set R_j is constructed by pruning the tree T_j at G_j. In other words, $u \in R_j$ if $u \notin G_j$ and there is a node $v \in G_j$ such that u is an ancestor of v in T_j. Clearly the following holds:

LEMMA 7. *The set R_j has the following properties:*

1. *The induced subgraph $G[R_j]$ is connected.*
2. *$R_j \subseteq G_{j-1}$.*
3. *$R_j \cap G_j = \emptyset$.*

5.2 Putting It All Together

We defined $2r$ sets B_i and R_i. In order to complete the construction of the $K_{r,r}$ minor we have to show the following.

LEMMA 8. *If there are two nodes x, y in G_r such that $d_{G_r}(x,y) \geq 12rb_r\rho$ then the $2r$ sets of nodes B_i, R_i, $i \in [1,r]$, have the following properties:*

1. *For every i the subgraph $G[B_i]$ and the subgraph $G[R_i]$ are connected.*
2. *The sets B_i and R_i, $i \in [1,r]$, are all mutually disjoint.*
3. *For every i, j there are nodes $u \in B_i$ and $v \in R_j$ such that (u,v) is an edge in G.*

First we show why Lemma 8 suffices to prove Theorem 3. Since all the sets are connected in G and they are all mutually disjoint, each one of the sets could be contracted into a different single node using only minor operations. Property (3) of the lemma implies that the resulting graph contains a $K_{r,r}$ minor contradicting the fact that G is $K_{r,r}$ free. We conclude that it must be that the *strong*-diameter of each G_r is bounded by $12rb_r\rho$.

PROOF. The first assertion of the Lemma is immediate from the previous Section.

To see why the third Assertion is true consider two sets R_i, B_j. The set B_j contains the path $\text{tail}_i(a_i)$ which is defined to be a BFS path in T_i. The set R_i is the remaining part of T_i thus the last node in $\text{tail}_i(a_i)$ is connected to a node in T_i.

It remains to show that all the sets are mutually disjoint. We do this case by case:

First, for every $i \neq j$ it holds that $R_i \cap R_j = \emptyset$. Assume w.l.o.g that $i \leq j-1$. By Lemma 7 it holds that $R_i \cap G_{j-1} = \emptyset$ while $R_j \subseteq G_{j-1}$. Conclude that $R_i \cap R_j = \emptyset$.

Second, for every i, j it holds that $B_i \cap R_j = \emptyset$. By Lemma 7 the set R_j is disjoint from G_r while by Lemma 6 the set B_i is contained in G_r.

Finally, for every $i \neq j$ it holds that $B_i \cap B_j = \emptyset$. This follows since $x_1, x_2, \ldots, x_r$ are far from one another in G_r, yet each B_i has a small radius in G_r. To be precise, the radius of each B_i is bounded by $3b_r\rho$ while the distance between x_i and x_j is at least $12b_r\rho$. □

5.3 The Weighted Case

We now present the reduction from the weighted graph case to the unweighted construction above. It is first worthwhile illuminating the key aspects of the above construction that are affected by having non-uniform edge weights. First, we need to find a node in M_k, the middle-strip, whenever two connected nodes cross M_k. Second, for two nodes u, v whose distance is larger than r, we need to find r nodes along the shortest path from u to v at distance $d(u,v)/r$ apart. Finally, we need the distances in G to uphold the triangle inequality; without it, the construction above may not yield the required cluster diameter bound.

We address all of these issues with the following reduction. Scale weights so that every edge weighs at least 1. Round up edge weights to the nearest integer. Note that edge weights increase by at most 2 by these transformations. Introduce virtual intermediate nodes along each edge, at intervals of length 1. Remove all weights. Let the new

unweighted graph be denoted G'. It is easy to see that virtual nodes do not change the topological properties of the graph. Hence, if G excludes $K_{r,r}$, then so does G'. Now, perform the probabilistic sparse partition above on G', and let the resulting clusters in G' be $C'_1, ..., C'_m$. Output the set of clusters $G[C'_1], ..., G[C'_m]$ induced by G''s clusters.

To see that the resulting partition satisfies the required properties, first observe that for any $u, v, \in C'$, distances satisfy $d_G(u, v) \leq d_{C'}(u, v)$. Hence, any bound on the diameters of the C' clusters is maintained in the clusters induced on G.

Second, let us consider the probability that an edge $(u, v) \in G$ is cut by the partition. This edge is represented in G' by at most $\lfloor d_G(u, v) + 2 \rfloor$ unweighted edges. By union bound, the probability that (u, v) is cut is at most $\lfloor d_G(u, v) + 2 \rfloor \frac{2}{\rho} \leq \frac{6 d_G(u,v)}{\rho}$.

We remark that the time complexity of the construction does suffer from the transformation, by a factor that is proportional to the aspect ratio of G.

6. OPEN PROBLEMS

The results of this paper could be utilized and optimized in several ways. The work suggests two main open problems.

First, all our theorems have an exponential dependency on the size of the forbidden minor. When weak-diameter is concerned it is possible to achieve a polynomial dependency [15]. It would be interesting to find sparse covers and sparse partitions with strong-diameter and a polynomial dependency in r. Note that the exponential dependency is an artifact of the technique of doubling the width of the cutting stripes each iteration. This is a key ingredient of our approach, thus such an improvement would probably require a different approach.

Finally, can Theorem 3 be extended to star-decompositions (see [12])? Can it be used to improve results in approximation algorithms? Natural candidates are metric embeddings and building spanners.

7. REFERENCES

[1] Ittai Abraham and Cyril Gavoille. Object location using path separators. In 25^{th} *Annual ACM Symposium on Principles of Distributed Computing (PODC)*, pages 188–197. ACM Press, July 2006.

[2] Ittai Abraham, Cyril Gavoille, and Dahlia Malkhi. Routing with improved communication-space trade-off. In 18^{th} *International Symposium on Distributed Computing (DISC)*, volume 3274 of Lecture Notes in Computer Science, pages 305–319. Springer, October 2004.

[3] Ittai Abraham, Cyril Gavoille, and Dahlia Malkhi. Compact routing for graphs excluding a fixed minor. In 19^{th} *International Symposium on Distributed Computing (DISC)*, volume 3724 of Lecture Notes in Computer Science, pages 442–456. Springer, September 2005.

[4] Ittai Abraham, Cyril Gavoille, and Dahlia Malkhi. On space-stretch trade-offs: Lower bounds. In 18^{th} *Annual ACM Symposium on Parallel Algorithms and Architectures (SPAA)*, pages 217–224. ACM Press, July 2006.

[5] Ittai Abraham, Cyril Gavoille, Dahlia Malkhi, Noam Nisan, and Mikkel Thorup. Compact name-independent routing with minimum stretch. In 16^{th} *Annual ACM Symposium on Parallel Algorithms and Architectures (SPAA)*, pages 20–24. ACM Press, July 2004.

[6] Baruch Awerbuch and David Peleg. Locality-sensitive resource allocation. Technical Report CS90-27, Weizmann Institute, November 1990.

[7] Baruch Awerbuch and David Peleg. Network synchronization with polylogarithmic overhead. In *IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 514–522, 1990.

[8] Baruch Awerbuch and David Peleg. Sparse partitions. In 31^{th} *Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 503–513. IEEE Computer Society Press, October 1990.

[9] Baruch Awerbuch and David Peleg. Routing with polynomial communication-space trade-off. *SIAM J. Discret. Math.*, 5(2):151–162, 1992.

[10] Costas Busch, Ryan LaFortune, and Srikanta Tirthapura. Improved sparse covers for graphs excluding a fixed minor. Technical Report 06-16, Department of Computer Science, Rensselaer Polytechnic Institute, November 2006.

[11] J. Lawrence Carter and Mark N. Wegman. Universal hash functions. *Journal of Computer and System Sciences*, 18(2):143–154, 1979.

[12] Michael Elkin, Yuval Emek, Daniel A. Spielman, and Shang-Hua Teng. Lower-stretch spanning trees. In *STOC '05: Proceedings of the thirty-seventh annual ACM symposium on Theory of computing*, pages 494–503, New York, NY, USA, 2005. ACM Press.

[13] Jittat Fakcharoenphol and Kunal Talwar. An improved decomposition theorem for graphs excluding a fixed minor. In *RANDOM-APPROX*, pages 36–46, 2003.

[14] Pierre Fraigniaud and Cyril Gavoille. Routing in trees. In 28^{th} *International Colloquium on Automata, Languages and Programming (ICALP)*, volume 2076 of Lecture Notes in Computer Science, pages 757–772. Springer, July 2001.

[15] Philip Klein, Serge A. Plotkin, and Satish Rao. Excluded minors, network decomposition, and multicommodity flow. In 25^{th} *Annual ACM Symposium on Theory of Computing (STOC)*, pages 682–690. ACM Press, 1993.

[16] Andrew Thomason. The extremal function for complete minors. *Journal of Combinatorial Theory, Series B*, 81(2):318–338, 2001.

[17] Mikkel Thorup. Compact oracles for reachability and approximate distances in planar digraphs. *Journal of the ACM*, 51(6):993–1024, November 2004.

[18] Mikkel Thorup and Uri Zwick. Approximate distance oracles. *J. ACM*, 52(1):1–24, 2005.