

## Strong-Diameter Decompositions of Minor Free Graphs

Ittai Abraham · Cyril Gavoille · Dahlia Malkhi ·  
Udi Wieder

Published online: 27 August 2010  
© Springer Science+Business Media, LLC 2010

**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 following 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  $O(r^2)$ , headers of  $O(\log n + r \log r)$  bits, and tables of size  $2^{O(r)}r! \log^4 n / \log \log n$  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  $O(r d(u, v)/\rho)$ .

**Keywords** Strong-diameter decompositions · Minor free graphs · Compact routing

---

C. Gavoille is supported by the projects ‘GeoComp’ and ‘Alpage’ of the ACI Masses de Données.

I. Abraham  
Hebrew University of Jerusalem, Jerusalem, Israel  
e-mail: [ittaia@cs.huji.ac.il](mailto:ittaia@cs.huji.ac.il)

C. Gavoille (✉)  
University of Bordeaux, Bordeaux, France  
e-mail: [gavoille@labri.fr](mailto:gavoille@labri.fr)

D. Malkhi · U. Wieder  
Microsoft Research, Silicon Valley Center, Mountain View, CA, USA

D. Malkhi  
e-mail: [dalia@microsoft.com](mailto:dalia@microsoft.com)

U. Wieder  
e-mail: [uwieder@microsoft.com](mailto:uwieder@microsoft.com)

## 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  $\omega : E \rightarrow \mathbb{R}^+$ . 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  $\text{diam}(G) = \max_{u,v} d_G(u, v)$ . 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  $CC(G)$  be the set of connected components of  $G$ . Let  $B_G(u, \rho) = \{v \mid d_G(u, v) \leq \rho\}$ . For  $U \subseteq V$  and  $v \in V$ , let  $d_G(U, v) = \min_{u \in U} d_G(u, v)$ , and let  $B_G(U, \rho) = \{v \mid d_G(U, v) \leq \rho\}$ . When  $G$  is clear from the context we omit the subscript and write  $d(u, v)$ .

We continue by defining the various combinatorial objects which play a role in the paper.

### *Sparse Covers*

Sparse covers were introduced by Awerbuch and Peleg in [8] and serve as a building block for a variety of applications. These include distance coordinates, routing with succinct routing tables [4, 8], mobile user tracking [8], resource allocation [6], synchronization in distributed algorithms [7], and others.

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

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

When a  $(k, \tau, \rho)$  sparse covers exists for any  $\rho$  we say that the graphs admits a  $(k, \tau)$  *sparse cover scheme*.

For general graphs, the seminal construction in [16] provides a  $(2k - 1, 2k \cdot n^{1/k})$  sparse cover scheme for any integer  $k \geq 1$ . This result asymptotically matches known lower bound that arise from dense graphs with high girth [16]. For certain restricted families of graphs, better covers are known to exist. For example, if the graph is  $\alpha$  doubling<sup>1</sup> then for any  $\rho, \varepsilon > 0$  one can greedily choose an  $\varepsilon\rho$ -net  $N$  and take the balls  $\{B(u, (1 + \varepsilon)\rho) \mid u \in N\}$ . It is easy to see that this forms a  $(1 + \varepsilon, (1 + 1/\varepsilon)^{O(\log \alpha)})$  sparse cover scheme.

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

## 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  $K_{3,3}$  and  $K_5$  (where  $K_{r,r}$  is the complete bipartite graph with  $r$  nodes in each set). It is natural to ask whether graphs that exclude some fixed minor have better sparse covers than general graphs.

Recall that a  $(k, \tau)$  *weak-diameter sparse cover scheme* is defined as a sparse cover scheme 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 indeed allow for improved weak-diameter sparse covers. Klein, Plotkin and Rao [13] (KPR for short) show 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. The requirement that the bound is over weak-diameters is necessary. We show in Sect. 2 a simple planar graph for which the KPR construction yields clusters with arbitrarily high diameter. The challenge of providing minor-free graph decompositions with clusters of bounded strong-diameter remained open and is addressed by the present work. During the preparation of the previous version of this paper it had been brought to our knowledge that independent work [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.

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.*

## 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 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.

Thorup [15] addressed the problem of labeled routing schemes in planar graphs (in fact in planar weighted digraphs). He shows the existence of a  $\text{polylog}(n)$  memory  $1 + \varepsilon$  stretch labeled routing scheme. Abraham and Gavoille [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 unweighted trees if less than  $\Omega(n \log n)$  bits are used [3]. For name-independent routing scheme, Abraham et al. provide in [4] 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 requires routing tables of  $\text{polylog}(n)$  bits and  $O(\log^2 n / \log \log n)$ -bit headers. The latter scheme, based on a particular weak-diameter sparse cover, is rather involved and provides routing tables of at least  $\log^6 n$  bits.

Our strong-diameter sparse cover scheme leads to a considerably simplification over previous solutions [4], moreover with improved performance. 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,<sup>2</sup> with stretch  $O(r^2)$  and using  $O(\log n + r \log r)$ -bit headers, in which every node requires tables of  $2^{O(r)} r! \cdot \log D \cdot \log^3 n / \log \log n$  bits.*

### *Sparse Partitions and Probabilistic Partitions*

Another approach for graph decomposition is to consider *partitions* of a graph into disjoint clusters. More precisely, we have the following definitions. A *strong-diameter  $\rho$  bounded partition* of  $G$  is a partition of  $V$  into disjoint clusters  $C_1, C_2, \dots$  such that for each cluster  $C_i$ ,  $\text{diam}(G[C_i]) \leq \rho$ . Given a partition  $P$  and a node  $u$ , let  $P(u)$  be the unique cluster that contains  $u$ .

<sup>2</sup>I.e., the port number around each node  $u$  is an arbitrary permutation of  $\{1, \dots, \text{deg}(u)\}$  that, as well as the node names, cannot be changed during the design of the routing scheme.

We can now define the notion of a probabilistic partition that is used in network approximations and metric embeddings.

**Definition 2** A  $(k, \eta, \rho)$  probabilistic partition is a distribution  $\Pr$  on a set of partitions  $\mathcal{P}$  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]:  $\forall u, v \in V$ ,

$$\Pr[P(u) \neq P(v)] \leq \eta d(u, v) / \rho.$$

When a  $(k, \eta, \rho)$  probabilistic partition exists for any  $\rho$  we say that the graph admits a  $(k, \eta)$  probabilistic partition scheme. If the partitions produced have only a weak-diameter bound we say that the resulting scheme is a weak-diameter probabilistic partition scheme.

Closely related to the notion of probabilistic partitions is that of a sparse partition.

**Definition 3** A  $(k, \eta, \rho)$  sparse partition is a partition  $P$  with the following properties:

1. [Small diameter]:  $P$  is a strong-diameter  $k\rho$  bounded partition of  $G$ .
2. [Sparse]:

$$|\{(u, v) \in E \mid P(u) \neq P(v)\}| \leq \eta |E| / \rho.$$

When a  $(k, \eta, \rho)$  sparse partition exists for any  $\rho$  we say that the graph admits a  $(k, \eta)$  sparse partition scheme. If the partition has only a weak-diameter bound we say that the resulting scheme is a weak-diameter sparse partition scheme.

Probabilistic partitions and sparse partitions play a key role in approximation algorithms, such as multi-commodity flow optimization problems [13]. Klein et al. provide a  $(O(r^3), O(r))$  weak-diameter sparse partition scheme for graphs excluding a  $K_{r,r}$  minor. Fakcharoenphol and Talwar improve in [12] to a  $(O(r^2), O(r))$  weak-diameter sparse partition scheme. Both results can be transformed into a weak-diameter probabilistic 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 partition scheme and a  $(O(r6^r), O(r))$  strong-diameter sparse partition scheme.

We envision that this result may play a 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- $\rho$  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 at most  $2^{O(r)}r!$  clusters.

- There is a name-independent routing scheme with  $O(r^2)$  stretch,  $O(\log n + r \log r)$ -bit headers, and tables of size  $2^{O(r)} r! \log^4 n / \log \log n$  bits.
- There is a probabilistic partition scheme 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  $O(r 6^r \rho)$ .

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

## 2 Large Diameter KPR Clusters

In this section, we briefly review the KPR [13] algorithm, and give an example where the clusters created have an unbounded strong diameter.

For planar graphs, KPR performs three recursive tree cuts into *stripes* of height  $\rho$ . Each tree is a breadth-first search tree (BFS) of a connected component that remains from the previous cut. In this section we will use the probabilistic partitions version of the KPR algorithm. We note that the same example can be given for the sparse partition version.

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  $\tau_1$ . Slice  $T_1$  into stripes of height  $\rho$ : 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 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  $\tau_4$  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.

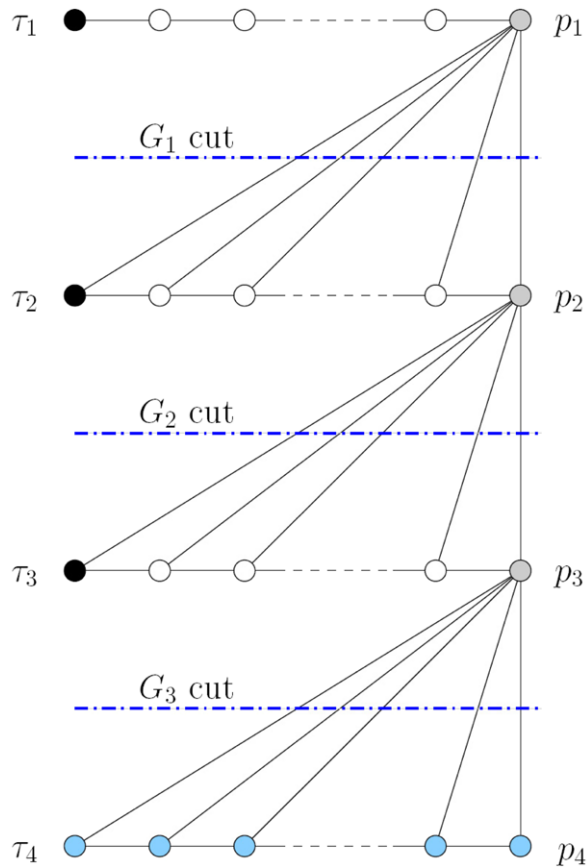
For arbitrary  $r < \rho$  iterations, the graph is composed of  $r + 1$  paths  $\tau_1 \rightarrow p_1$ ,  $\tau_2 \rightarrow p_2, \dots, \tau_{r+1} \rightarrow p_{r+1}$  of length respectively  $h_i + k\rho - 1$  where  $k \geq 1$  is an arbitrary large integer, and  $h_i \in [0, \rho - 1]$  is an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

In the original graph  $G_1$ , the BFS tree  $T_1$  is rooted at  $\tau_1$ . We note that node  $p_1$  has distance less than  $\rho$  to all the nodes below the cut line marked ‘ $G_1$  cut’. The distance in  $G_1$  from  $\tau_1$  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 consist of all nodes under the  $G_1$  cut, which forms a connected component  $G_2$  induced by the nodes of  $G_1 \setminus (\tau_1 \rightarrow p_1)$ .

Note that, despite the fact that nodes  $\tau_2$  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  $\tau_2$  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  $\tau_{r+1} \rightarrow p_{r+1}$  ( $\tau_4 \rightarrow p_4$

**Fig. 1** Example graph in which KPR partition has arbitrarily large diameter



on the picture) have arbitrarily large distance: All the nodes  $p_1, \dots, p_r$  that shorten the distance from  $\tau_{r+1}$  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  $h_1, \dots, h_r$ . However we can consolidate the counterexample by rebuilding the previous graph for all offset sequences, and by identifying all the nodes  $\tau_1$ . The resulting graph is still outerplanar. Clearly, any choice of  $r$  offsets during the randomization provides at least one unbounded strong-diameter component.

### 3 Sparse Cover with Strong Diameter

We now provide a graph cover procedure that yields clusters with a bounded strong-diameter. The algorithm uses the KPR [13] paradigm: recursively cutting strips from BFS trees. Unlike the KPR procedure, at each iteration our algorithm does not cut the exact strip, but pads it by growing balls around central portions of the strip, referred to as “cores”. Our construction and proof can be seen as an enhancement of the arguments in [4]. Intuitively, our construction ensures that each final cluster  $G_{r+1}$  has

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  $G \setminus G_{r+1}$ . Given these two properties we show that each cluster  $G_{r+1}$  has a small strong-diameter in the following way: If  $G_{r+1}$  has a long enough path then by (1) we know that there exist  $r$  nodes in  $H_{r+1}$  that are far away from each other. Then, (2) implies the existence of a  $K_{r,r}$  minor.

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

**Theorem 4** *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* Our proof strategy focuses first on probabilistic partitions for an unweighted graph. Later, in Sect. 6, we detail the reduction from the weighted case to this construction. Henceforth, assume  $G$  is unweighted.

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  $\tau_i$  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  $\tau_i$  where parent vertices have the minimal possible label. We can now slice  $T_i$  to slices of height  $\rho$ . Let  $S_{i,j}$  be the  $j$ th slice of  $T_i$ , so

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

*cover algorithm:*

We fix two global parameters  $\rho$  and  $r$ . The algorithm  $C = \text{cover}(G)$  receives a graph  $G$  and returns a set of clusters  $C \subset 2^V$ . The `cover` algorithm invokes the algorithm  $C = \text{cut-stripe}(G, G, 1)$  and returns the set of clusters output by it.

*cut-stripe algorithm:*

The algorithm  $\text{cut-stripe}(G_i, H_i, i)$  receives a graph  $G_i$ , a subgraph  $H_i \subseteq G_i$  and a parameter  $i$  (implicitly it also has access to the global parameters  $\rho, r$ ) and returns a set of clusters  $C_i \subset 2^{G_i}$ :

- If  $i = r + 1$  then return  $\{G_i\}$ .
- Otherwise, for every integer  $j$  let

$$\mathcal{G}_j = CC(G [B_{G_i}(S_{i,j} \cap H_i, i\rho)])$$

return

$$\bigcup_{j, G' \in \mathcal{G}_j} \text{cut-stripe}(G', G' \cap S_{i,j} \cap H_i, i + 1).$$

Note that  $G' \in \mathcal{G}_j$  may contains nodes *outside*  $S_{i,j}$ . It is this padding which is the main difference from the KPR procedure.

From now on we will fix some node  $v \in G$  and prove that the cover produced by the algorithm has all the properties with respect to  $v$ .



**Claim 1** There exists unique indexes  $j_1, \dots, j_r$  and subgraphs  $G_1, H_1, \dots, G_r, H_r, G_{r+1}, H_{r+1}$  such that for all  $1 \leq i \leq r$ :  $G_{i+1}$  is a connected component of  $G[B_{G_i}(S_{i,j_i} \cap H_i, i\rho)]$  and  $v \in H_{i+1} = G_{i+1} \cap S_{i,j_i} \cap H_i$ .

*Proof* The proof is by induction. Clearly  $v \in H_1 = G$ . 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.  $\square$

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

**Property 1 (Cover)** We show that for the cluster  $G_{i+1}$ ,  $B = B(v, \rho) \subseteq G_{i+1}$ . The proof is by simple induction. In the base case  $B \subseteq G_1$ . Given  $B \subseteq G_i$  and  $v \in H_{i+1}$  it follows from the definition of  $G_{i+1}$  that  $B \subseteq G_{i+1}$ .

**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 each 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  $y_1, y_r$  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  $y_1, \dots, y_r \in G_{r+1}$  such that the balls  $B_{G_{r+1}}(y_i, 2(r + 1)\rho)$  are pairwise disjoint. Let  $x_i$  be a 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 5** *If there exist points  $x_1, \dots, x_r \in H_{r+1}$  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  $L_1, L_2, \dots, L_r$ , such that  $x_i \in L_i$ ; and from  $r$  sets called the *right super-nodes*, denoted  $V_1, V_2, \dots, V_r$ . 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 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  $\tau_i$ .

**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  $\tau_i$ . The right super nodes are  $R_i$  for  $i \in [1, r]$ , where  $R_i = U_i \setminus G_{i+1}$ . Observe that each  $R_i$  induces a connected subtree of  $G$ .

The super edges are the edges in  $T_i$  connecting each  $R_i$  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  $\ell > j$  we claim that  $\text{tail}_j(x_i) \subseteq G_\ell$  and that  $\text{tail}_j(x_i)$  is disjoint from the right node  $R_{\ell-1}$ . The proof is by induction on  $\ell$ . For  $\ell = j + 1$  this is true by construction.

For  $\ell > j + 1$ , 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  $R_{\ell-1}$ . 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  $j \leq \ell \in [1, r]$ ,  $\text{tail}_j(x_i) \cap R_\ell = \emptyset$ . Finally, for all  $i \in [1, r]$  and all  $\ell < j \in [1, r]$ , the set  $R_\ell$  is clearly disjoint from  $\text{tail}_j(x_i)$  and from  $R_j$  since by construction  $R_\ell = U_\ell \setminus G_{\ell+1}$  and  $R_j \cup \text{tail}_j(x_i) \subseteq G_j \subseteq G_{\ell+1}$ .  $\square$

## 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 terms 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 [3].

We assume that  $G$  is unweighted, since it has been proved in [5] 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 remainder of this section, we will assume that the  $n$  node names of  $G$  have arbitrary names in  $\{1, \dots, n^{O(1)}\}$ , 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 5** *For every  $n$ -node unweighted graph of diameters  $D$  excluding a  $K_{r,r}$  minor, there is a polynomial time constructible name-independent routing scheme, in*

the fixed part model,<sup>3</sup> with stretch  $O(r^2)$  and using  $O(\log n + r \log r)$ -bit headers, in which every node requires tables of  $2^{O(r)} r! \cdot \log D \cdot \log^3 v / \log \log n$  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  $\tau$  such that, for every  $\rho > 0$ , the graph  $G$  has a  $(k, \tau, \rho)$  sparse cover, and let  $\mathcal{C}_\rho$  denote this cover. Then, routing in  $G$  can be done by considering the family of covers  $\mathcal{F} = \{\mathcal{C}_1, \dots, \mathcal{C}_{2^i}, \dots, \mathcal{C}_{2^{\lceil \log D \rceil}}\}$ . 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. For any scale  $i$ , although  $u$  belongs to many clusters  $C \in \mathcal{C}_{2^i}$ , it suffices to look at only one cluster that contains  $B_G(u, 2^i)$ . Roughly speaking, the routing task for a source  $u$  consists of seeking the target  $v$  in trees of  $\mathcal{T}$  that correspond to clusters that contain  $B_G(u, 2^i)$ .

Specifically,  $u$  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  $i \in \{0, \dots, \lceil \log D \rceil\}$ . 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  $O(k)$ , since the tree of the cluster covering  $B_G(u, 2^i)$  is of depth at most  $k2^i$ . Overall, if a tree routing scheme for seeking  $v$  can be implemented with  $M$ -bit routing tables, and  $B$ -bit headers, then the routing scheme for  $G$  uses at most  $O(\tau \cdot \log D \cdot M)$  bits for routing tables and  $B + O(\log |\mathcal{T}|)$  bits for headers. Indeed, each node participates in at most  $\tau$  tree routings for each of the  $\lceil \log D \rceil + 1$  covers in  $\mathcal{F}$ , and the headers need to be enlarged by a tree identifier.

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 use for each depth- $h$  tree  $T \in \mathcal{T}$ , a space efficient  $O(h)$ -error reporting routing scheme. We will use a modified<sup>4</sup> version of the single-source unweighted tree routing of [2] combined with the low density of minor-free graphs to balance routing information. Recall that an  $\alpha$ -orientation of a graph is an orientation of its edges such that every node has out-degree at most  $\alpha$ .

**Lemma 6** [2] *Let  $\mathcal{T}$  be a collection of trees in an  $n$ -node graph  $G$  with an  $\alpha$ -orientation. Then, one can construct in polynomial time for each node  $u$  of  $G$  a routing table of  $O(\log^3 n / \log \log n + \alpha \log n)$  bits per tree of  $\mathcal{T}$  containing  $u$ , such that each depth- $h$  tree of  $\mathcal{T}$  has a  $8h$ -error reporting routing scheme using  $O(\log n)$ -bit headers. Moreover, the first header construction takes  $O(\log n)$  time at the source, and all the other routing decisions take  $O(\log \log n)$  time.*

We also need of the following well-known fact:

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

<sup>4</sup>The modified version appears in the full version of [2].

**Lemma 7** Any graph excluding a  $K_{r,r}$  minor has a  $\Theta(r\sqrt{\log r})$ -orientation that can be computed in linear time.

*Proof* Graphs 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 than  $f(r) \cdot n$  edges [14] 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 a  $\Theta(r\sqrt{\log r})$ -orientation computable in linear time.  $\square$

The proof of Theorem 2 is completed thanks to Lemma 6 by observing that each node belongs to at most  $\tau \cdot \log D = 2^{O(r)} r! \cdot \log D$  trees of the collection  $\mathcal{T}$ , and that  $\alpha = O(r\sqrt{\log r})$  (Lemma 7). From the above discussion the name-independent routing scheme uses tables of  $2^{O(r)} r! \cdot \log D \cdot \log^3 n / \log \log n$  bits and  $O(\log n + r \log r)$ -bit headers (since  $\log |\mathcal{T}| = O(r \log r + \log \log D)$ ). The whole construction of the routing scheme takes a polynomial time since the sparse cover (Theorem 1), the orientation (Lemma 6), and the error-reporting routing scheme (Lemma 7) take a polynomial time.

## 5 Probabilistic Partitions and Sparse Partitions

In this section we present probabilistic partitions and Sparse Partitions with strong-diameter guarantees. The overall approach is again similar in spirit to KPR. However there are three major differences. First, our phase  $i$  stripes are of width  $6^i \rho$ , while KPR always chooses width  $\rho$ . Second, after the initial cuts we use a cone based approach (see [11]) 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 not get assigned to the  $i$ th “core” will be associated with the nodes of the  $(i + 1)$  stripe.

**Theorem 6** For every weighted graph excluding a  $K_{r,r}$  minor there exists a polynomial time sampleable  $(O(r6^r), O(r))$  strong-diameter probabilistic partition scheme and a  $(O(r6^r), O(r))$  strong-diameter sparse partition scheme.

Our proof strategy focuses first on probabilistic partitions for an unweighted graph. Later, in Sect. 6, we detail the reduction from the weighted case to this construction. Henceforth, assume  $G$  is unweighted.

partition *algorithm*:

We fix two global parameters  $\rho$  and  $r$ . The algorithm  $P = \text{partition}(G)$  receives a graph  $G$  and returns a partition  $P$  of  $V$ . The partition algorithm invokes the algorithm  $P = \text{cut-and-merge}(G, 1)$  and returns the partition output by it.

cut-and-merge *algorithm*:

The algorithm  $P = \text{cut-and-merge}(G_{k-1}, k)$  receives a base graph  $G_{k-1}$ , and a parameter  $k$  (implicitly it also has access to the global parameters  $\rho, r$ ) and returns a partition of  $G_{k-1}$ :

1. if  $k = r + 1$  then return  $\{G_{k-1}\}$ . Otherwise, sample two integers  $h_k, \ell_k$  uniformly and independently from  $[0, \rho - 1]$ .
2. Perform BFS from an arbitrary root node  $s \in G_{k-1}$ :  $T_k = \text{BFS}(G_{k-1}, s)$
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.,  $M(L_k(i)) = \{u \in L_k(i) \mid d_{G_{k-1}}(u, L_k(i - 1)) = (b_k/2)\rho\}$ . 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  $u \rightarrow v$  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  $\gamma_k(u, v) = d_G(u, v)$ . Notice that  $\gamma_k$  is *not symmetric*.

We can extend  $\gamma_k$  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  $\gamma_k$ .

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  $\gamma_k$  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  $T_i \cap L_k(i)$ .

We say that a node  $u \in L_k(i)$  is **assigned** if  $u \in S_k^+(i)$ .

7. After performing the previous steps to *all* layers in the decomposition, add all **unassigned** nodes from  $L(i + 1)$  (that were not included in  $S_k^+(i + 1)$ ) into the set  $S_k(i)$ .  
 $S_k(i) = S_k^+(i) \cup \{L_k(i + 1) \setminus S_k^+(i + 1)\}$ . See Figs. 2 and 3.
8. Note that now the sets  $S_k(i)$  partition  $G_{k-1}$ . For each  $i$  recurse on every connected component  $G'$  of  $CC(G[S_k(i)])$ .

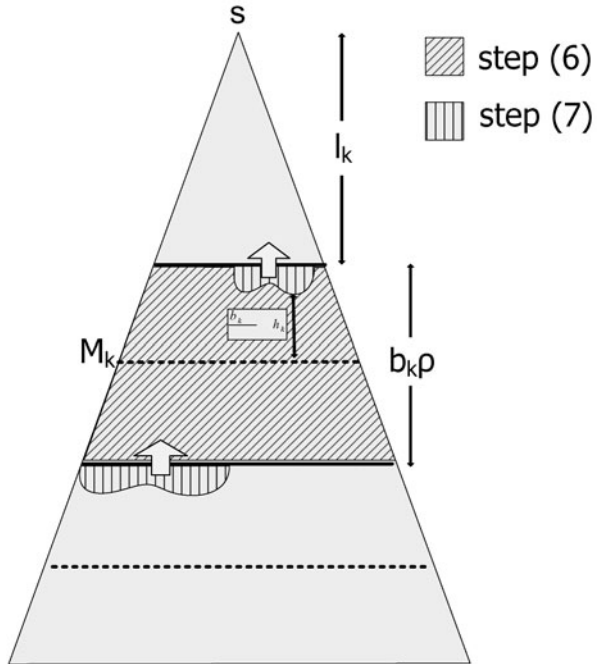
Return

$$\bigcup_{i, G' \in CC(G[S_k(i)])} \text{cut-and-merge}(G', k + 1).$$

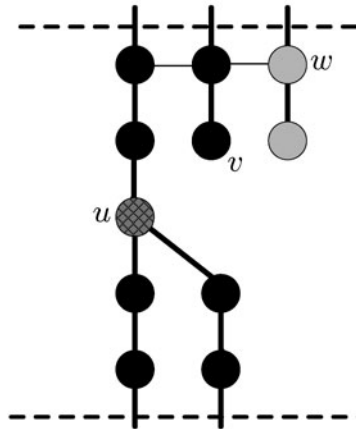
See Fig. 2.

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

**Fig. 2** Steps (6) and (7) in the Sparse Partition procedure. Figure 3 depicts step (6) in greater detail



**Fig. 3** The bold edges are tree edges which are oriented down, and node \$u\$ is in the center. The black nodes indicate a ball around \$u\$ of cone distance 3. Note that \$v\$ is of cone distance 3 from \$u\$, while \$w\$ is of distance 4



*Proof* In each execution of the procedure there are two ways an edge \$(u, v)\$ could be separated. The first is that \$u, v\$ are separated 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 \$\ell\_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 \$\gamma\_k\$ distance from \$M\_k\$ while node \$v\$ has a large \$\gamma\_k\$ distance from \$M\_k\$. Note however that

$$|\gamma_k(M_k, u) - \gamma_k(M_k, v)| \leq d_G(u, v) = 1.$$

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 claim.  $\square$

**Claim 3** For all  $u, v \in V$ ,

$$\Pr[P(u) \neq P(v)] \leq 2r d_G(u, v)/\rho.$$

*Proof* The probability that  $u$  and  $v$  are first separated in the  $k$ th iteration is bounded by the probability that any edge on the shortest path between  $u$  and  $v$  is cut. By union bound on Claim 2 this probability is at most  $2d_G(u, v)/\rho$ . Since there are at most  $r$  recursive calls then a union bound concludes the proof of the claim.  $\square$

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 a cluster  $G_r$  such that the distance between 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.

**Lemma 10** *If there are two nodes  $x, y$  in a cluster  $G_r$  such that  $d_{G_r}(x, y) \geq 12rb_r\rho$  then there exist  $2r$  sets of nodes  $B_i, R_i, i \in [1, r]$ , that 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 10 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 cluster  $G_r$  is bounded by  $12rb_r\rho$ .

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

**Claim 4** Each node  $u \in G_k$  has an *anchor* node  $a_k(u) \in M_k$  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  $\gamma_k$  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  $b_k\rho$ .  $\square$

**Claim 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  $S_k$ .  $\square$

We now proceed to prove Lemma 10. 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  $x = x_1, x_2, \dots, x_r = y$  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  $B_1, \dots, B_r$  such that  $x_i \in B_i$  and  $r$  sets  $R_1, \dots, R_r$  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:

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

The set  $B_i$  is now defined as  $B_i = B_i(r)$ .

**Lemma 13** *The set  $B_i$  has the following properties:*

1. *The induced subgraph  $G[B_i]$  is connected.*
2.  $B_i \subseteq G_r$ .
3.  $\text{diam}_{G_r}(B_i) \leq 3b_r\rho$ .

*Proof* Clearly the induced graph  $G[B_i]$  is connected. We omit the index  $i$  when clear from context and 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 Claim 4. We have that  $\text{tail}_1(a_1) \subseteq G_1$  by Claim 5. By the induction hypothesis we have that  $B(r - 1) \subseteq G_{r-1}$ . Furthermore, Claim 4 implies that  $A_r \subseteq G_r$  and Claim 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 Claim 5  $B(r - 1) \subseteq G_r$ . Furthermore  $\text{diam}_{G_r} B(r) \leq (3b_r/2 + 1)\rho + 3b_{r-1}\rho \leq 3b_r\rho$ .  $\square$



## The Set $R_i$

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

**Claim 14** The set  $R_i$  has the following properties:

1. The induced subgraph  $G[R_i]$  is connected.
2.  $R_i \subseteq G_{i-1}$ .
3.  $R_i \cap G_i = \emptyset$ .

*Proof of Lemma 10* We have defined  $2r$  sets  $\{B_i \mid i \in [1..r]\}$  and  $\{R_i \mid i \in [1..r]\}$ . The first assertion of the Lemma stating that each set induces a connected subgraph follows from Lemma 13(1) and Claim 14 (1).

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 the second assertion, that all the sets are mutually disjoint. We do this by considering three different cases:

First, for every  $i \neq j$  consider the sets  $R_i, R_j$ . Assume w.l.o.g that  $i \leq j - 1$ . By Claim 14 it follows 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 Claim 14 (3) the set  $R_j$  is disjoint from  $G_r$  while by Lemma 13 (2) 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  $x_1, x_2, \dots, x_r$  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  $12b_r\rho$ .  $\square$

## 5.1 Sparse Partitions

The randomized algorithm `cut-and-merge` can easily be modified into a deterministic algorithm that returns one sparse partition. In each iteration  $k$ , instead of choosing  $h_k, \ell_k$  at random, we choose  $h_k, \ell_k$  out of the  $\rho^2$  possible values such that a minimal number of edges are cut.

If the total edge weight at the beginning of the  $k$ th iteration is  $W_k$  then there must exist a choice of  $h_k, \ell_k$  that yields a partition where the weight of cut edges is at most  $2W_k/\rho$ . As this is done in every recursive call then the total weight of cut edges is at most  $2rW/\rho$  where  $W$  is the initial edge weight. This concludes the last part of Theorem 3.

## 6 The Weighted Case

We reduce the weighted case to the unweighted case in the following standard way: Scale weights so that every edge weights 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  $C'_1, \dots, C'_m$ . Output the set of clusters  $G[C'_1], \dots, G[C'_m]$  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  $G$ .

For Theorem 3, 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  $\lfloor d_G(u, v) + 2 \rfloor \frac{2}{\rho} \leq \frac{6d_G(u, v)}{\rho}$  since  $d_G(u, v) \geq 1$ .

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  $G$ .

## 7 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 [13]. 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 [11])? Can it be used to improve results in approximation algorithms? Natural candidates are metric embeddings and building spanners.

## References

1. Abraham, I., Gavoille, C.: Object location using path separators. In: 25th Annual ACM Symposium on Principles of Distributed Computing (PODC), pp. 188–197. ACM Press, New York (2006)
2. Abraham, I., Gavoille, C., Malkhi, D.: Routing with improved communication-space trade-off. In: 18th International Symposium on Distributed Computing (DISC). Lecture Notes in Computer Science, vol. 3274, pp. 305–319. Springer, Berlin (2004)
3. Abraham, I., Gavoille, C., Malkhi, D., Nisan, N., Thorup, M.: Compact name-independent routing with minimum stretch. In: 16th Annual ACM Symposium on Parallel Algorithms and Architectures (SPAA), pp. 20–24. ACM Press, New York (2004)
4. Abraham, I., Gavoille, C., Malkhi, D.: Compact routing for graphs excluding a fixed minor. In: 19th International Symposium on Distributed Computing (DISC). Lecture Notes in Computer Science, vol. 3724, pp. 442–456. Springer, Berlin (2005)

5. Abraham, I., Gavoille, C., Malkhi, D.: On space-stretch trade-offs: Lower bounds. In: 18th Annual ACM Symposium on Parallel Algorithms and Architectures (SPAA), pp. 217–224. ACM Press, New York (2006)
6. Awerbuch, B., Peleg, D.: Locality-sensitive resource allocation. Technical Report CS90-27, Weizmann Institute, November 1990
7. Awerbuch, B., Peleg, D.: Network synchronization with polylogarithmic overhead. In: IEEE Symposium on Foundations of Computer Science (FOCS), pp. 514–522 (1990)
8. Awerbuch, B., Peleg, D.: Sparse partitions. In: 31th Annual IEEE Symposium on Foundations of Computer Science (FOCS), pp. 503–513. IEEE Computer Society Press, Los Alamitos (1990)
9. Awerbuch, B., Peleg, D.: Routing with polynomial communication-space trade-off. *SIAM J. Discrete Math.* **5**(2), 151–162 (1992)
10. Busch, C., LaFortune, R., Tirthapura, S.: Improved sparse covers for graphs excluding a fixed minor. In: 26th Annual ACM Symposium on Principles of Distributed Computing (PODC), pp. 61–70. ACM Press, New York (2007)
11. Elkin, M., Emek, Y., Spielman, D.A., Teng, S.-H.: Lower-stretch spanning trees. In: STOC '05: Proceedings of the Thirty-Seventh Annual ACM Symposium on Theory of Computing, pp. 494–503. ACM Press, New York (2005)
12. Fakcharoenphol, J., Talwar, K.: An improved decomposition theorem for graphs excluding a fixed minor. In: *RANDOM-APPROX*, pp. 36–46 (2003)
13. Klein, P., Plotkin, S.A., Rao, S.: Excluded minors, network decomposition, and multicommodity flow. In: 25th Annual ACM Symposium on Theory of Computing (STOC), pp. 682–690. ACM Press, New York (1993)
14. Thomason, A.: The extremal function for complete minors. *J. Comb. Theory, Ser. B* **81**(2), 318–338 (2001)
15. Mikkil, T.: Compact oracles for reachability and approximate distances in planar digraphs. *J. ACM* **51**(6), 993–1024 (2004)
16. Thorup, M., Zwick, U.: Approximate distance oracles. *J. ACM* **52**(1), 1–24 (2005)