Shortest-path metric approximation for random subgraphs

Jan Vondrák *

Abstract

We consider graph optimization problems where the cost of a solution depends only on the shortest-path metric in the graph, such as Steiner Tree or Traveling Salesman. We study a scenario where such a problem needs to be solved repeatedly on random subgraphs of a given graph G. With the goal of speeding up the repeated queries and saving space, we describe the construction of a sparse subgraph $Q \subset G$ which contains an approximately optimal solution for any such problem on a random subgraph of G, with high probability. More precisely, the subgraph Q has the property that after some vertices or edges are removed randomly, Q still contains c-approximate shortest paths between all pairs of vertices with high probability. The number of edges in Q is $O(p^{-c}n^{1+2/c}\log n)$ for edge-induced random subgraphs and $O(p^{-2c}n^{1+2/c}\log^2 n)$ for vertex-induced random subgraphs, where n is the number of vertices in G, p the sampling probability of edges/vertices, and $c \in \mathbb{Z}$, $c \geq 3$ is the desired approximation factor.

1 Introduction

Often, a situation arises where a certain optimization problem needs to be solved repeatedly and only a part of the input is changing. We assume that the dynamic part of the input is random but we have some information about the underlying probability distribution. In such cases, it is useful to perform a precomputation that involves only the static part of the input and our assumptions about the dynamic part. Our goal is to speed up the subsequent solution of instances arriving at random.

For example, the topology of telecommunication networks can be considered fixed but the demands of a given customer may vary over time. The goal is to exploit the topology without knowing the demands. The same situation occurs in performing multicast in telecommunication networks; we need to solve a minimum spanning tree or Steiner tree problem to connect a group of users, but the topology or graph does not change when connecting different groups of users. Yet another example is a delivery company which has to solve daily vehicle routing problems in which the road network does not change but the locations of customers to serve do.

Examples of such *repetitive optimization* problems with both static and dynamic inputs are abundant but it is often unclear whether one can take advantage of the advance knowledge of the static part of the input. One situation, which has been heavily studied from a practical point of view, entails *s-t* shortest path queries in large-scale navigation systems or Geographic Information Systems. In that setting, it is too slow to compute from scratch the shortest path whenever a query

*Department of Mathematics, Princeton University, Princeton, NJ 08540. E-mail: jvondrak@math.princeton.edu.

comes in. Various preprocessing steps have been proposed, often creating a hierarchical view of the network, see for example [5].

It can also be the case that only a random part of the network is available. This situation arises when some vertices or edges fail with certain probabilities and our solution can only use the remaining subgraph. The authors of [3] studied the Minimum Spanning Tree (MST) problem in this scenario. They proved that for any graph G with arbitrary edge weights, there exists a subset of $O(n \log_b n)$ edges which contains almost surely the minimum spanning tree of a random subgraph. The random subgraph can be generated by taking a random subset of either vertices or edges, each independently with probability p, and $b = 1/(1-p)$. Thus, by considering Q instead of G, subsequent MST queries can be speeded up significantly. This is the setting considered in this paper.

Optimization on random subgraphs. We investigate repetitive optimization problems in the formal setting of *random subgraphs* chosen from an arbitrary fixed graph G. We denote by $V(p)$ a random subset of vertices where each vertex is chosen independently with probability p. Similarly, we denote by $E(p)$ a random subset of edges where each edge is present independently with probability p. We are interested in random subgraphs H induced by $V(p)$ or $E(p)$. (We assume that successive instances are generated randomly in this way.) We denote the optimization problem in question by $\mathcal{P}$ and its optimal solution for instance H by $\mathcal{P}(H)$. We start with a fixed weighted graph G; however, rather than solving the optimization problem $\mathcal{P}$ on G itself, we are interested in solving it on a random subgraph $H \subset G$. We would like to preprocess the graph G so that we reduce the amount of time or space needed to solve $\mathcal{P}$ repeatedly for random subgraphs. Our approach here is to find a sparse subgraph $Q \subset G$ which has the property that almost surely (with respect to H), $\mathcal{P}(Q \cap H)$ is a solution which matches or at least approximates $\mathcal{P}(H)$.

Metric-based problems. In this paper, we study a class of problems which are based on the shortest-path metric, in the sense that the cost of a solution depends only on distances between pairs of vertices. (We interpret the edge weights as lengths here.) Examples are the Steiner Tree, where each vertex in a set of terminals T should be connected to a given root, or the Traveling Salesman where we seek a closed walk traversing all vertices. In both cases, the solution can be written as a collection of paths connecting pairs of vertices and the cost of each path is simply its length.

We are interested in solving such a problem repeatedly on random subgraphs of a given graph. It should be stressed that the entire instance of the problem is restricted to a subgraph, e.g. in case of the Steiner Tree problem, we are not choosing the terminals randomly but an entire subgraph H on which an instance of Steiner Tree should be solved. The terminals can be fixed in advance, or even chosen arbitrarily for each subgraph.

Since problems such as Steiner Tree and Traveling Salesman are NP-hard, we are content with finding a constant-factor approximation to the optimum. In our setting, Q should contain constant-factor approximate solutions to such problems on random subgraphs. Therefore it is sufficient to ask that Q, restricted to a random subgraph H, has a shortest-path metric approximating that of H

within a constant factor. Then any solution in H can be approximated by a good solution in $Q \cap H$ as well, so we can work with the graph Q instead of G for all purposes concerning constant-factor approximations to metric-based problems. This motivates the following definition.

Definition 1.1. *Consider a graph G with fixed edge weights. A subgraph $Q \subset G$ is c-metric-approximating for a certain distribution of random subgraphs $H \subset G$, if with high probability, for any u-v path in H there is a u-v path in H, using only edges of Q, of length expanded at most by a factor of c.*

Note. *High probability* means probability tending to 1 as the number of vertices tends to infinity.

Relation to spanners. Our notion of metric approximation for random subraphs is reminiscent of the notion of a *spanner*, a sparse subgraph approximating distances in the original graph. Spanners have been studied in ample scope, under various restrictions - for geometric graphs, general graphs, with constrained degrees, restricted structure, etc. In general, we say that a subgraph $S \subset G$ is a c-spanner if for any path in G of length ℓ there is a corresponding path in S of length at most $c\ell$. See for example [2] for a survey of results about general spanners.

The existence of spanners with a low number of edges is related to the existence of graphs without short cycles. We say the graph G has *girth* g, if the shortest cycle is C_g. A c-spanner for a graph of girth $g \geq c+2$ (with unit edge weights) must be the graph itself, since no edge can be replaced by another path of length at most c. Therefore the number of edges required in general for a c-spanner is at least the extremal number of edges for a graph of girth $g = c+2$. It is known that there are graphs of girth g with $n^{1+1/(g-1)}$ edges (in classical work by Paul Erdős [7]; the proof uses the probabilistic method). On the other hand, a c-spanner can be found by a construction avoiding cycles shorter than $c+1$, which yields a c-spanner of girth at least $c+2$ (see [2, 4]). Thus the extremal number of edges for graphs of given girth provides an upper bound on the size of c-spanners as well. The best known upper bound on the number of edges for a graph of girth g is $O(n^{1+2/(g-2)})$ for $g \geq 4$ even [1]. Thus there always exists a c-spanner with $O(n^{1+2/c})$ edges.

However, our requirements are stronger than those for a spanner: we ask that $Q \cap H$, rather than Q, approximates the metric of H. In other words, we are not allowed to use paths in Q leaving the subgraph H. In fact, we construct a "robust spanner" which is resistant to random failure of vertices or edges. Still, our algorithms will be based on ideas similar to those producing good spanners, and the bounds we obtain are intimately related to extremal graphs of a given girth.

Our results. We describe an efficient construction of c-metric-approximating subgraphs $Q \subset G$ for random subgraphs induced by sampling either vertices or edges independently with probability p. This yields c-approximate solutions for any metric-based optimization problem on such random subgraphs, with high probability. The number of edges in Q is $O(p^{-c} n^{1+2/c} \log n)$ for edge-induced random subgraphs and $O(p^{-2c} n^{1+2/c} \log^2 n)$ for vertex-induced random subgraphs, where $c \in \mathbb{Z}, c \geq 3$ is the desired approximation factor.

The factor of $O(n^{1+2/c})$ comes from a known construction of c-spanners for weighted graphs [4]. This construction forms the basic building block of our algorithms. It produces subgraphs without short cycles and therefore the bound is directly related to known bounds on extremal graphs of girth $c+2$. Should this bound be improved, we would obtain a better bound as well. For constant p and c, we are losing only a polylogarithmic factor compared to constructing a c-spanner.

The rest of the paper is organized as follows. Section 2 presents a lower bound on the size of metric-approximating sets in case we want to preserve a shortest s-t path, or even an s-t path with a low stretch factor. Section 3 explains our basic approach to constructing metric-approximating sets, based on previous approaches to constructing spanners. This turns out to give a good result for edge-induced random subgraphs. For vertex-induced random subgraphs, the solution is slightly more involved, applying the spanner construction to random subgraphs. Our final randomized algorithm for vertex-induced random subgraphs is described in Section 4.

2 A lower bound

We start by showing that unlike in the case of minimum spanning trees [3], we cannot ask here for subgraphs containing the optimal shortest paths when restricted to a random subgraph. This is true even in the case of a single s-t path, for fixed vertices s and t (where the deterministic version of the problem is trivial).

Recall that we denote by $V(p)$ a random subset of V where each vertex is sampled independently with probability p. We denote by $G[W]$ the subgraph induced by W. Interchangeably, we denote by Q a subgraph of G or the corresponding subset of edges. In either case, $Q[W]$ denotes the subgraph of Q induced by the subset of vertices W. In the spirit of [3], we might like to find a subgraph Q such that for a random induced subgraph $G[W], W = V(p) \cup \{s,t\}$, $Q[W]$ contains a shortest s-t path in $G[W]$ with high probability. It turns out that in contrast to the MST problem, such a subgraph Q cannot be very sparse.

Example (see Figure 1). Consider a graph $G = (V, E)$ where

- $V = \bigcup_{i=1}^{k} A_i \cup \bigcup_{i=1}^{k} B_i$
- $E = \bigcup_{i=1}^{k} E(A_i) \cup \bigcup_{i=1}^{k} E(B_i) \cup K$.
- $A_i = \{s, a_{i1}, a_{i2}, \ldots, a_{il}\}$
- $E(A_i)$ is a path s-a_{i1}-a_{i2}-…-a_{il}
- $B_i = \{t, b_{i1}, b_{i2}, \ldots, b_{il}\}$
- $E(B_i)$ is a path t-b_{i1}-b_{i2}-…-b_{il}
- K is a complete bipartite graph $\{(a_{il}, b_{jl}) : 1 \leq i, j \leq k\}$.

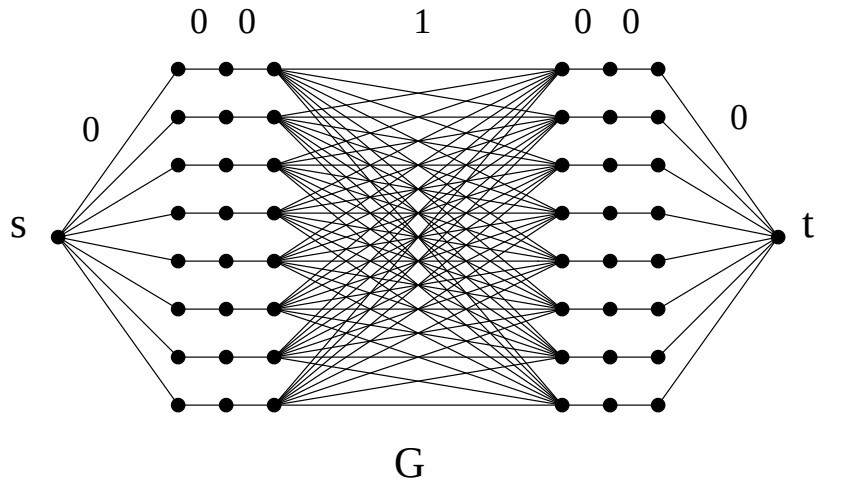

Figure 1: An s-t-shortest path example, for $k = 8$ and $l = 3$.

- The edges in $E(A_i)$ and $E(B_i)$ have zero length, while the edges in K have unit length.

We set $k = 2^l, n = |V| = l2^{l+1} + 2$. Sample $W \subseteq V$, each vertex with probability $1/2$ except s and t which are always in W. Consider the event that A_i is the unique path among $A_1, A_2, \ldots, A_k$ which survives in $G[W]$: this happens with probability $2^{-l}(1 - 2^{-l})^{k-1} > 1/ek$. Independently, B_j is the unique surviving path among $B_1, B_2, \ldots, B_k$, with probability at least $1/ek$. Thus with probability $1/(ek)^2$, there is a unique shortest s-t path (of length 1), using the surviving paths A_i, B_j and the edge $(a_{il}, b_{jl}) \in K$. This holds for every edge in K with the same probability and the corresponding events are disjoint. If $h = |K \setminus Q|$, there is probability at least $h/(ek)^2$ that Q doesn't contain the shortest path. Therefore Q must contain $\Omega(k^2) = \Omega(n^2/\log^2 n)$ edges in K, otherwise the probability of missing the shortest path is at least $1/2e^2$.

Moreover, note that if Q does not contain the shortest path, the next shortest connection between s and t has length at least 3. So in fact, we need $\Omega(n^2/\log^2 n)$ edges even if we want Q to contain some *approximately-shortest* s-t path, for any stretch factor $c < 3$. Therefore, it seems too ambitious to ask for a subgraph Q which contains an s-t path very close to optimal with high probability.

Instead, we ask whether it is possible to find a sparse subgraph Q which contains at least an approximately-shortest s-t path with high probability, for some stretch factor $c \geq 3$. More generally, we seek c-metric approximating subgraphs in the sense of Definition 1.1.

3 A deterministic construction

Let us assume that each vertex is removed independently with probability $1/2$ and denote the set of surviving vertices by W. We want to find a subgraph Q which provides short connections between

vertices even when restricted to $G[W]$.

The core of our algorithm is a procedure previously used to construct c-spanners [4]. In short, this procedure includes an edge (u, v) in Q, unless it is already "covered" by a short u-v path in Q. What we mean by being "covered" is simply that there is already another path connecting u and v in Q. This path serves as a substitute for (u, v) in the spanner. Since we want to maintain this property even when random vertices are removed, we iterate the procedure r times (i.e., we take a union of r edge-disjoint spanners). This ensures that each $(u, v) \notin Q$ is covered by r edge-disjoint paths.

Algorithm 1. (given parameters $c, r \in \mathbb{Z}_+$)

1. Let $E_1 := E(G)$. Repeat the following for $k = 1, 2, \ldots, r$.
2. Let $Q_k := \emptyset$.
3. Process the edges of E_k in the order of increasing edge weights.
4. For each $(u, v) \in E_k$, unless it is covered by a path of $\leq c$ edges in Q_k, include (u, v) in Q_k.
5. If $k < r$, set $E_{k+1} := E_k \setminus Q_k$ and go back to step 2.
6. Finally, let $Q := \cup_{k=1}^{r} Q_k$.

Note that in each stage k, this algorithm maintains two useful properties. Any edge which is *not* in Q_k is covered by a path of at most c edges; this path uses edges of smaller weight than (u, v) , so indeed it approximates the length of (u, v) within a factor of c. Also, the algorithm avoids all cycles shorter than $c + 2$ to be created in Q_k; this serves to bound the size of Q.

First, we use this algorithm with $c = 3$, to get a 3-metric-approximating set.

Lemma 3.1. *For $c = 3$ and $r = \lfloor 12 \ln n \rfloor$, Algorithm 1 finds in polynomial time a subgraph Q of size $|Q| = O(n^{3/2} \ln n)$ which is 3-metric-approximating for uniformly random vertex-induced subgraphs.*

Proof. For each k, Q_k is a C_4-free subgraph. By a well-known result in extremal graph theory [1], $|Q_k| = O(n^{3/2})$. Therefore $|Q| = O(rn^{3/2}) = O(n^{3/2} \log n)$.

Now consider an edge $(u, v) \in E$ that we have not chosen in any stage, i.e. $(u, v) \notin \cup_{k=1}^{r} Q_k$. Call (u, v) a bad edge if $(u, v) \in E \setminus Q$ and none of its covering paths survive in $G[W]$. For each k, there is a covering u-v path of 2 or 3 edges. By construction, these paths are edge-disjoint. Condition on $u, v \in W$, since otherwise the edge does not appear in $G[W]$. For a path of 2 edges (u-w-v), there is conditional probability $1/2$ that it survives in $G[W]$. For a path of 3 edges (u-w-w'-v), the probability is $1/4$.

Note that if two of these paths share a vertex w, then they must both have 3 edges and w must be a common neighbor of u and v. In that case, we remove the two paths and replace them by u-w-v. For two vertex-disjoint paths of 3 edges, the conditional probability that neither of them survives in $G[W]$ is $(1 - 1/4)^2$ which is more than the probability of destruction for a single path of 2 edges.

Therefore we may assume that we have r paths of type u-w-w'-v. (u, v) can be bad only if $u, v \in W$ while none of these paths survive in $G[W]$:

$$\Pr[(u,v) \text{ is bad} \mid u, v \in W] \leq \left(1 - \frac{1}{4}\right)^r < e^{-r/4} \leq \frac{1}{n^3},$$

$$\Pr[\exists \text{ bad } (u,v) \in E] \leq \sum_{(u,v)\in E} \Pr[(u,v) \text{ is bad} \mid u, v \in W]\ \Pr[u, v \in W] < \frac{1}{n}.$$

Therefore with high probability, all edges in $G[W]$ are either in Q or covered by some path in Q of length expanded by at most 3. □

As in spanners, one might try to produce sparser subgraphs at the cost of increasing the approximation factor, by checking for paths longer than 3. However, in the case of vertex-sampling, $c > 3$ creates difficulties because of the positive correlation between overlapping paths. Still, this idea works in the edge-sampling case.

Theorem 3.2. *For any graph G with edge lengths, Algorithm 1 with $r = \lceil 4p^{-c} \ln n \rceil$ and integer $c \geq 3$ finds with high probability a c-metric-approximating subgraph $Q \subseteq E$ (for random subgraphs induced by $F = E(p)$) such that*

$$|Q| = O\left(p^{-c} n^{1+2/c} \log n\right).$$

Proof. Recall the proof of Lemma 3.1. The bound on $|Q|$ follows from the girth of Q_i in each stage which is at least $c + 2$. Therefore, $|Q_i| = O(n^{1+2/c})$ (see [4], Lemma 6).

Every edge $e \in E \setminus Q$ is covered by r edge-disjoint paths whose length approximates that of e by a factor of at most c. Each of them has probability of survival at least p^c. Here, these events are independent because of edge-disjointness. Consequently, the probability that none of these paths survive in F is at most $(1-p^c)^r < 1/n^4$. By the union bound, all edges in $E \setminus Q$ have a c-approximate substitute path in $Q \cap F$ with probability at least $1 - 1/n^2$. □

4 A randomized construction

Now let's turn to the construction of a good metric-approximating subgraph for subgraphs induced by random subsets of vertices $W = V(p)$. We must ensure that every edge in $G[W]$ is either in Q or covered by a short path in $Q[W]$, with high probability. For $p = 1/2$ and $c = 3$, we have a 3-metric-approximating subgraph of size $O(n^{3/2} \log n)$ due to Lemma 3.1. The remaining question is whether we can find sparser c-metric approximating subgraphs for larger values of c, and how their size depends on p for $p \to 0$.

The essential obstacle to applying Algorithm 1 with higher values of c is that the resulting covering paths need not be vertex-disjoint. For instance, there can be arbitrarily many u-v paths of length 4 with a vertex-cut of size 1. This would destroy our objective to guarantee a high probability of survival for at least one path. The solution is to run the same algorithm repeatedly on random induced subgraphs. This can be seen as generating spanners on random subgraphs and then taking

their union. This forces the substitute paths to be vertex-disjoint with a certain probability which means they can be analyzed independently with respect to random vertex failures. A careful tuning of the sampling parameters yields a good bound on the size of a c-metric approximating subgraph.

Algorithm 2. (given parameters $t, c \in \mathbb{Z}_+, q \in (0,1)$)

- Repeat the following steps in stages $i = 1, 2, \ldots, t$.
- Set $Q_i = \emptyset$ and sample a random subset of vertices $S_i = V(q)$.
- Process the edges of $G[S_i]$ in the order of increasing edge lengths.
- Include each edge in Q_i, unless it is covered by a path of at most c edges in Q_i.
- Set $Q = \bigcup_{i=1}^{t} Q_i$.

Theorem 4.1. *Let $c \geq 3$ be a given integer and $n^{-1/(2c)} < p < 1$. For a graph G with given edge weights, Algorithm 2 with $q = 1/\lfloor 8cp^{-c} \log n \rfloor$ and $t = O(c^2\ p^{-3c} \log^3 n)$ finds with high probability a c-metric-approximating subgraph $Q \subseteq E$ (for random subgraphs induced by $W = V(p)$) such that*

$$|Q| = O\left(p^{-2c} n^{1+2/c} \log^2 n\right).$$

Proof. Our goal here is to cover each edge in $E \setminus Q$ by $\Omega(p^{-c} \log n)$ vertex-disjoint paths of length stretched by at most c. For that purpose, we set the following parameters:

- $r = \lfloor 8cp^{-c} \log n \rfloor$.
- $q = 1/r$.
- $t = \lceil 64r^2\ p^{-c} \log n \rceil$.

For $1 \leq j \leq t$, consider an edge $(u, v) \in E \setminus \bigcup_{i=1}^{j} Q_i$ and suppose that in $\bigcup_{i=1}^{j} Q_i$ it has been covered by $k < 8p^{-c} \log n$ vertex-disjoint paths with at most c edges. Define a set of vertices $R_j(u, v)$ which contains the internal vertices of these k paths, does not contain u and v, and contains some additional vertices (for example, the lexicographically smallest) so that the cardinality of $R_j(u, v)$ is always $r = \lfloor 8cp^{-c} \log n \rfloor$. If (u, v) has been already covered by $8p^{-c} \log n$ vertex-disjoint paths, define $R_j(u, v)$ of size r arbitrarily.

Call S_{j+1} "good for (u, v)", if $u, v \in S_{j+1}$ and $R_j(u, v) \cap S_{j+1} = \emptyset$.

$$\Pr[S_{j+1} \text{ is good for } (u, v)] = q^2(1 - q)^r.$$

We choose $q = 1/r$, assuming $r \geq 2$, which implies

$$\Pr[S_{j+1} \text{ is good for } (u, v)] = \frac{1}{r^2}\left(1 - \frac{1}{r}\right)^r \geq \frac{1}{4r^2}.$$

Observe that if S_{j+1} is good for (u,v) then either we include (u,v) in Q, or we cover it by a new vertex-disjoint path, unless it is covered by $8p^{-c}\log n$ paths already. We have a total of t stages. We estimate the number of good stages for (u,v):

$$\mu = \mathbf{E}[\#\text{good stages for } (u,v)] \geq \frac{t}{4r^2}.$$

We have $t = \lceil 64r^2\ p^{-c}\log n \rceil$ so that $\mu \geq 16p^{-c}\log n$. The events of "a stage being good for (u,v)" are independent for different stages. By Chernoff's inequality (see [6], Theorem 4.2),

$$\Pr[\#\text{stages good for } (u,v) < \mu/2] < e^{-\mu/8} \leq n^{-2p^{-c}}$$

$$\Pr[\exists (u,v) \in E;\ \#\text{good stages for } (u,v) < \mu/2] < n^{2-2p^{-c}} = o(1).$$

So there are at least $\mu/2 \geq 8p^{-c}\log n$ good stages for every edge, with high probability with respect to our sampling of $S_1,\ldots,S_t$. Assuming that this happens, since every good stage either includes $(u,v) \in Q$ or yields a new vertex-disjoint path connecting (u,v), in the end every edge in $E \setminus Q$ has at least $8p^{-c}\log n$ vertex-disjoint substitute paths each consisting of at most c edges. Each path survives in $G[W]$, where $W = V(p)$, with probability at least p^c. This implies

$$\Pr[Q[W] \text{ is not a } c\text{-spanner for } G[W]] \leq \sum_{e\in E\setminus Q} (1-p^c)^{8p^{-c}\log n} < \sum_{e\in E\setminus Q} \frac{1}{n^8} < \frac{1}{n^6}.$$

Again, the size of Q is constrained by the absence of short cycles. Each Q_i is a subgraph of girth $g(Q_i) \geq c+2$. Moreover, it is a subgraph on the vertices of S_i where

$$\mathbf{E}[|S_i|] = qn = \frac{n}{r}$$

and by Chernoff's inequality

$$\Pr[|S_i| > 2n/r] < e^{-n/3r},$$

$$\Pr[\exists i \leq t; |S_i| > 2n/r] < t\ e^{-n/3r}$$

which tends to 0 for our choice of parameters (since $p^{-2c} \leq n$, we have $r = O(\sqrt{n}\log n)$ and $t = O(n^{3/2}\log^3 n)$). Thus with high probability, we also have $|S_i| \leq 2n/r$. The size of a subgraph on $|S_i|$ vertices of girth $g(Q_i) \geq c+2$ can be at most

$$|Q_i| \leq |S_i|(1+|S_i|^{2/c}) \leq \frac{2n}{r}(1+n^{2/c})$$

(see [4], Lemma 6). Thus the total number of edges selected by the algorithm is:

$$|Q| \leq \frac{2nt}{r}(1+n^{2/c}) \leq (2^9+o(1))\ c\ p^{-2c}\ n^{1+2/c}\log^2 n$$

with high probability. □

5 Concluding remarks

It is known that for even $c \geq 2$ there exist graphs of girth $c+2$ with $\Omega(n^{1+1/c})$ edges [1]. For such graphs, there is no c-metric-approximating subgraph other than the graph itself and therefore we need at least $\Omega(n^{1+1/c})$ edges in general. For subgraphs induced by $V(p)$, p constant, our construction yields a subgraph with $O(n^{1+2/c}\log^2 n)$ edges, so the gap between the two bounds is $O(n^{1/c}\log^2 n)$. However, the factor of $n^{1/c}$ arises from the lack of tight bounds on graphs of given girth rather than poor performance of our algorithm. Alternatively, we could formulate our bounds in terms of the maximal number of edges for graphs of girth g. Denoting this number by $m(n,g)$, we would obtain that our c-metric-approximating subgraph has size $O(m(n,c+2)\log^2 n)$, while the lower bound is exactly $m(n,c+2)$. Thus the price we pay for having a "robust spanner", resistant to random failures of vertices, is $O(\log^2 n)$. In the case of edge-induced random subgraphs, the gap is $O(\log n)$. We leave it as an open question whether these gaps can be improved.

6 Acknowledgements

I would like to thank Santosh Vempala who suggested the definition of metric-approximating subgraphs, and Michel Goemans for helpful discussions.

References

[1] B.Bollobás. *Extremal graph theory.* Academic Press, London-New York, 1978.

[2] D. Peleg and A.A. Schäffer. Graph spanners. *J. Graph Theory*, 13(1):99–116, 1989.

[3] M.X. Goemans and J. Vondrák. Covering minimum spanning trees of random subgraphs. In *SODA*, pages 927–934, 2004.

[4] I. Althöfer and G. Das and D. Dobkin and D. Joseph and J. Soares. On sparse spanners of weighted graphs. *Discrete and Computational Geometry*, 9:81–100, 1993.

[5] N. Jing, Y.W. Huang, and E.A. Rundensteiner. Hierarchical encoded path views for path query processing: An optimal model and its performance evaluation. *IEEE T. on Knowledge and Data Engineering*, 10(3):409–432, 1998.

[6] R. Motwani and P. Raghavan. *Randomized Algorithms.* Cambridge University Press, 1995.

[7] P. Erdös and H. Sachs. Reguläre Graphen gegebener Taillenweite mit minimaler Knotenzahl. *Wiss Z Martin-Luther University Halle-Wittenberg Math-Natur*, Reihe 12:251–257, 1963.