Roundtrip Spanners with $(2k-1)$ Stretch

Ruoxu Cen[*1] and Ran Duan[†1]

[1]Institute for Interdisciplinary Information Sciences, Tsinghua University

December 2, 2019

Abstract

A roundtrip spanner of a directed graph G is a subgraph of G preserving roundtrip distances approximately for all pairs of vertices. Despite extensive research, there is still a small stretch gap between roundtrip spanners in directed graphs and undirected spanners. For a directed graph with real edge weights in $[1, W]$, we first propose a new deterministic algorithm that constructs a roundtrip spanner with $(2k-1)$ stretch and $O(kn^{1+1/k}\log(nW))$ edges for every integer $k \geq 1$, then remove the dependence of size on W to give a roundtrip spanner with $(2k-1+o(1))$ stretch and $O(kn^{1+1/k}\log n)$ edges. While keeping the edge size small, our result improves the previous $2k+\epsilon$ stretch roundtrip spanners in directed graphs [Roditty, Thorup, Zwick'02; Zhu, Lam'18], and almost match the undirected $(2k-1)$-spanner with $O(kn^{1+1/k})$ edges [Althöfer et al. '93] which is optimal under Erdös conjecture.

*cenbo@aliyun.com

†duanran@mail.tsinghua.edu.cn

1 Introduction

A t-spanner of a graph G is a subgraph of G in which the distance between every pair of vertices is at most t times their distance in G, where t is called the stretch of the spanner. Sparse spanner is an important choice to implicitly presenting all-pair distances [18], and spanners also have application background in distributed systems and communication networks. For undirected graphs, $(2k-1)$-spanner with $O(n^{1+1/k})$ edges is proposed and conjectured to be optimal [2,16]. However, directed graphs may not have sparse spanners with respect to normal distance measure. For instance, in a bipartite graph with two sides U and V, if there is an directed edge from every vertex in U to every vertex in v, then removing any edge (u,v) in this graph will destroy the reachability from u to v, so its only spanner is itself, which has $O(n^2)$ edges. To circumvent this obstacle, one can approximate the optimal spanner in terms of edge size (e.g. in [3,9]), or can define directed spanners on different distance measures. This paper will study directed sparse spanners on roundtrip distances.

Roundtrip distance is a natural metric with good property. Cowen and Wagner [7,8] first introduce it into directed spanners. Formally, roundtrip distance between vertices u,v is defined as $d_G(u \leftrightarrows v) = d_G(u \to v) + d_G(v \to u)$, where $d_G(u \to v)$ is the length of shortest path from u to v. For a directed graph $G=(V,E)$, a subgraph $G'=(V,E')$ $(E' \subseteq E)$ is called a t-roundtrip-spanner of G if $\forall u,v \in G, d_{G'}(u \leftrightarrows v) \le t \cdot d_G(u \leftrightarrows v)$. t is called the stretch of the roundtrip spanner.

In directed graph $G=(V,E)$ $(n=|V|, m=|E|)$ with real edge weights in $[1,W]$, Roditty et al. [15] provides a $(2k+\epsilon)$-spanner with $O(\min\{(k^2/\epsilon)n^{1+1/k}\log(nW),\ (k/\epsilon)^2 n^{1+1/k}(\log n)^{2-1/k}\})$ edges. Recently, Zhu and Lam [17] derandomizes it and improves the size to $O((k/\epsilon)n^{1+1/k}\log(nW))$ edges, while the stretch is also $2k+\epsilon$. We make a step further based on these works and reduce the stretch to $2k-1$. Formally, we state our main results in the following theorem.

Theorem 1. *For any directed graph G with maximum edge weight W and integer $k \ge 1$, there exists a $(2k-1)$-roundtrip spanner of G with $O(kn^{1+1/k}\log(nW))$ edges, and a $(2k-1+O(1/n^c))$-roundtrip spanner of G with $O(kn^{1+1/k}\log n)$ edges for any constant c. Both spanners can be constructed in $\tilde{O}(kmn\log W)$ time.*

Actually, our result almost matches the lower bound following girth conjecture. The girth conjecture, implicitly mentioned by Erdös [11], says that for any k, there exists a graph with n vertex and $\Omega(n^{1+1/k})$ edges whose girth (minimum cycle) is at least $2k+2$. This conjecture implies that no algorithm can construct a spanner of $O(n^{1+1/k})$ size and less than $2k-1$ stretch for all undirected graph with n vertices [16]. This lower bound also holds for roundtrip spanners on directed graphs.

Our approach is based on the scaling constructions of the $(2k+\epsilon)$-stretch roundtrip spanners in [15,17]. To reduce the stretch, we construct inward and outward shortest path trees from vertices in a hitting set [1,10] of size $O(n^{1/k})$, and carefully choose the order to process vertices in order to make the stretch exactly $2k-1$. To further make the size of the spanner strongly subquatratic, we use a similar approach as in [15] to contract small edges in every scale.

1.1 Related Works

Like many previous works aiming at constructing sparse roundtrip spanners with small stretch, this paper also has $\Omega(nm)$ construction time. Pachoci et al. [13] proposes an algorithm which can construct $O(k\log n)$-roundtrip-spanner with $O(n^{1+1/k}\log^2 n)$ edges. Its construction time is

$O(mn^{1/k}\log^5 n)$, which breaks the cubic time barrier. Very recently, Chechik et al. [6] give an algorithm which constructs $O(k\log\log n)$-roundtrip-spanner with $\tilde{O}(n^{1+1/k})$ edges in $\tilde{O}(m^{1+1/k})$ time.

For spanners defined with respect to normal directed distance, researchers aim to approximate the k-spanner with minimum number of edges. Dinitz and Krauthgamer [9] achieve $\tilde{O}(n^{2/3})$ approximation in terms of edge size, and Bermen et al. [3] improves the approximation ratio to $\tilde{O}(n^{1/2})$.

Another type of directed spanners is transitive-closure spanner, introduced by Bhattacharyya et al. [5]. In this setting the answer may not be a subgraph of G, but a subgraph of the transitive closure of G. In other words, selecting edges outside the graph is permitted. The tradeoff is between diameter (maximum distance) and edge size. One of Bhattacharyya et al.'s results is spanner with diameter k and $O((n\log n)^{1-1/k})$ approximate to optimal edge size [5], using a combination of linear programming rounding and sampling. Berman et al. [4] improves the approximation ratio to $O(n^{1-1/[k/2]}\log n)$. We refer to Raskhodnikova [14] as a review of transitive-closure spanner.

1.2 Organization

In Section 2, the notations and basic concepts used in this paper will be discussed. In Section 3 we describe the construction of the $(2k-1)$-roundtrip spanner with $O(kn^{1+1/k}\log(nW))$ edges, then in Section 4 we improve the size of the spanner to $O(kn^{1+1/k}\log n)$ and the stretch becomes $(2k-1+o(1))$.

2 Preliminaries

In this paper we consider a directed graph $G=(V,E)$ with non-negative real edge weights w where $w(e)\in[1,W]$ for all $e\in E$. Denote $G[U]$ to be the subgraph of G induced by $U\subseteq V$, i.e. $G[U]=(U,E\cap(U\times U))$. A roundtrip path between nodes u and v is a cycle (not necessarily simple) passing through u and v. The roundtrip distance between u and v is the minimum length of roundtrip paths between u and v. Denote $d_U(u\leftrightarrows v)$ to be the roundtrip distance between u and v in $G[U]$. (Sometimes we may also use $d_U(u\leftrightarrows v)$ to denote a roundtrip shortest path between u,v in $G[U]$.) It satisfies:

- For $u,v\in U$, $d_U(u\leftrightarrows u)=0$ and $d_U(u\leftrightarrows v)=d_U(v\leftrightarrows u)$.
- If $d_U(u\to v)$ is the distance from u to v in $G[U]$, then $d_U(u\leftrightarrows v)=d_U(u\to v)+d_U(v\to u)$.
- For $u,v,w\in U$, $d_U(u\leftrightarrows v)\le d_U(u\leftrightarrows w)+d_U(w\leftrightarrows v)$.

Here $d_U(u\to v)$ is the one-way distance from u to v in $G[U]$. Of course $d(u\leftrightarrows v)$ is just the roundtrip distance between u and v in the original graph $G=(V,E)$.

In G, a t-roundtrip spanner of G in a subgraph H of G on the same vertex set V such that the roundtrip distance between any pair of $u,v\in V$ in H is at most $t\cdot d(u\leftrightarrows v)$. t is called the *stretch* of the spanner.

Given a center $u\in U$ and a radius R, define roundtrip ball $Ball_U(u,R)$ to be the set of vertices whose roundtrip distance on $G[U]$ to center u is strictly smaller than the radius R. Formally, $Ball_U(u,R)=\{v\in U: d_U(u\leftrightarrows v)<R\}$. Then the size of the ball, denoted by $|Ball_U(u,R)|$, is the number of vertices in it. Similarly we define $\overline{Ball}_U(u,R)=\{v\in U: d_U(u\leftrightarrows v)\le R\}$.

Subroutine $\texttt{InOutTrees}(U, u, R)$ calculates the edge set of an inward and an outward shortest path tree centered at u spanning vertices in $Ball_U(u, R)$ on $G[U]$. (That is, the shortest path tree from u to all vertices in $Ball_U(u, R)$ and the shortest path tree from all vertices in $Ball_U(u, R)$ to u.) It is easy to see that the shortest path trees will not contain vertices outside $Ball_U(u, R)$:

Lemma 2. *The inward and outward shortest path trees returned by* $\texttt{InOutTrees}(U, u, R)$ *only contains vertices in* $Ball_U(u, R)$.

Proof. For any $v \in Ball_U(u, R)$, let C be a cycle containing u and v such that the length of C is less than R. Then for any vertex $w \in C$, $d_U(u \leftrightarrows w) < R$, so w must be also in the trees returned by $\texttt{InOutTrees}(U, u, R)$. □

For all notations above, we can omit the subscript V when the roundtrip distance is considered in the original graph $G = (V, E)$. Our algorithm relies on the following well-known theorem to calculate hitting sets deterministically.

Theorem 3. *(Cf. Aingworth et al. [1], Dor et al. [10]) For universe V and its subsets $S_1, S_2, \ldots, S_n$, if $|V| = n$ and the size of each S_i is greater than p, then there exists a hitting set $H \subseteq V$ intersecting all S_i, whose size $|H| \le n \ln n/p$, and such a set H can be found in $O(np)$ time deterministically.*

3 A $(2k-1)$-Roundtrip-Spanner Algorithm

In this section we introduce our main algorithm constructing a $(2k-1)$-roundtrip-spanner with $O(kn^{1+1/k}\log(nW))$ edges for any G. We may assume $k \ge 2$ in the following analysis, since the result is trivial for $k = 1$.

Our approach combines the idea of [15] and [17]. In [17], given a length L, we pick an arbitrary vertex u and find the smallest integer h such that $|\overline{Ball}(u, (h+1)L)| < n^{1/k}|\overline{Ball}(u, h \cdot L)|$, then we include the inward and outward shortest path tree centered at u spanning $\overline{Ball}(u, (h+1)L)$ and remove vertices in $\overline{Ball}(u, h \cdot L)$ from V. We can see that $h \le k$, so the stretch is $2k$ for u, v with roundtrip distance L, and by a scaling approach the final stretch is $2k + \epsilon$. We observe that if $h = k - 1$, $|\overline{Ball}(u, (k-1)L)| \ge n^{(k-1)/k}$, so by Theorem 3 we can preprocess the graph by choosing a hitting set H with size $O(n^{1/k}\log n)$ and construct inward and outward shortest path trees centered at all vertices in H, then we do not need to include the shortest path trees spanning $\overline{Ball}(u, (h+1)L)$. The stretch can then be decreased to $2k - 1 + \epsilon$. To make the stretch equals $2k - 1$, instead of arbitrarily selecting u each time, we carefully define the order to select u.

3.1 Preprocessing

We first define a radius $R(u)$ for each vertex u. It is crucial for the processing order of vertices.

Definition 4. *For all $u \in V$, we define $R(u)$ to be the maximum length R such that $|Ball(u, R)| < n^{1-1/k}$, that is, if we sort the vertices by their roundtrip distance to u in G by increasing order, $R(u)$ is the roundtrip distance from u to the $\lceil n^{1-1/k} \rceil$-th vertex.*

For any $u \in V$, $|\overline{Ball}(u, R(u))| \ge n^{1-1/k}$. By Theorem 3, we can find a hitting set H intersecting all sets in $\{\overline{Ball}(u, R(u)) : u \in V\}$, such that $|H| = O(n^{1/k}\log n)$. For all $t \in H$, we build an inward and an outward shortest path tree of G centered at t, and denote the set of edges of these trees by E_0 and include them in the final spanner. This step generates $O(n^{1+1/k}\log n)$ edges in total, and it is easy to obtain the following statement:

Lemma 5. *For $u, v \in V$ such that $d(u \leftrightarrows v) \geq R(u)/(k-1)$, the roundtrip distance between u and v in the graph (V, E_0) is at most $(2k-1)d(u \leftrightarrows v)$.*

Proof. Find the vertex $t \in H$ such that $t \in \overline{Ball}(u, R(u))$, that is, $d(u \leftrightarrows t) \leq R(u)$. Then the inward and outward shortest path trees from t will include $d(u \leftrightarrows t)$ and $d(t \leftrightarrows v)$. By $R(u) \leq (k-1)d(u \leftrightarrows v)$, we have $d(t \leftrightarrows v) \leq d(t \leftrightarrows u) + d(u \leftrightarrows v) \leq k \cdot d(u \leftrightarrows v)$. So the roundtrip distance of u and v in E_0 is at most $d(u \leftrightarrows t) + d(t \leftrightarrows v) \leq (2k-1)d(u \leftrightarrows v)$. □

3.2 Approximating a Length Interval

Instead of approximating all roundtrip distances at once, we start with an easier subproblem of approximating all pairs of vertices whose roundtrip distance is within an interval $[L/(1+\epsilon), L]$. Parameter ϵ is a real number in $(0, 1/(2k-2)]$. The procedure `Cover`(G, k, L, ϵ) described in Algorithm 1 will return a set of edges which gives a $(2k-2+\epsilon)$-approximation of roundtrip distance $d(u \leftrightarrows v)$ if $R(u)/(k-1) > d(u \leftrightarrows v)$ for $d(u \leftrightarrows v) \in [L/(1+\epsilon), L]$.

Algorithm 1 Cover$(G(V, E), k, L, \epsilon)$

```
1: U ← V, Ê = ∅
2: while U ≠ ∅ do
3:     u ← arg max_{u∈U} R(u)
4:     step ← min{R(u)/(k − 1), L}
5:     h ← minimum positive integer satisfying |Ball_U(u, h · step)| < n^{h/k}
6:     Add InOutTrees(U, u, h · step) to Ê
7:     Remove Ball_U(u, (h − 1)step) from U
8: end while
9: return Ê
```

Note that in this algorithm, initially $U = V$ and the balls are considered in $G[U] = G$. In the end of every iteration we remove a ball from U, and the following balls are based on the roundtrip distances in $G[U]$. However, $R(u)$ does not need to change during the algorithm, which can still be based on roundtrip distances in the original graph G. The analysis for the size of the returned set $\hat{E}$ and the stretch are as follows.

Lemma 6. *The returned edge set of* `Cover`(G, k, L, ϵ) *has $O(n^{1+1/k})$ size.*

Proof. When processing a vertex u, by the selection of h in line 5, $|Ball_U(u, h \cdot step)| < n^{h/k}$ and $|Ball_U(u, (h-1)step)| \geq n^{(h-1)/k}$. When $h \geq 2$ it is because of h's minimality, and when $h = 1$ it is because $u \in Ball_U(u, 0)$. So each time `InOutTrees` is called, the size of ball to build shortest path trees is no more than $n^{1/k}$ times the size of ball to remove. During an execution of `Cover`(G, k, L, ϵ), each vertex is removed once from U. Therefore the total size of edges added in $\hat{E}$ is $O(n^{1+1/k})$. □

We can also see the if the procedure `Cover`$(G[U], k, L, \epsilon)$ is run on a subgraph $G[U]$ induced on a subset $U \subseteq V$, then the size of $\hat{E}$ is bounded by $O(|U|n^{1/k})$. It is also easy to see that h is at most $k-1$ by our choice of u:

Lemma 7. *The h selected at line 5 in* `Cover`(G, k, L, ϵ) *satisfies $h \leq k-1$.*

Proof. In $G[U]$, the ball $Ball_U(u,(k-1)step)$ must have size no greater than $Ball(u,(k-1)step)$ since the distance in $G[U]$ cannot be decreased and some vertices may be removed. Since $|Ball(u,R(u))| < n^{1-1/k}$ and $step \leq R(u)/(k-1)$, we get $|Ball_U(u,(k-1)step)| \leq |Ball(u,(k-1)step)| < n^{1-1/k}$, thus $h \leq k-1$. □

Next we analysis the roundtrip distance stretch in $\hat{E}$. Note that in order to make the final stretch $2k-1$, for the roundtrip distance approximated by edges in $\hat{E}$ we can make the stretch $(2k-2)(1+\epsilon)$, but for the roundtrip distance approximated by E_0 we need to make the stretch at most $2k-1$ as E_0 stays the same.

Lemma 8. *For any pair of vertices u, v such that $d(u \leftrightarrows v) \in [L/(1+\epsilon), L]$, either* `Cover`$(G,k,L,\epsilon)$*'s returned edge set $\hat{E}$ can form a cycle passing through u, v with length at most $(2k-2)(1+\epsilon)d(u \leftrightarrows v)$, or $R(u) < (k-1)d(u \leftrightarrows v)$, in which case the E_0 built in Section 3.1 can form a detour cycle with length at most $(2k-1)d(u \leftrightarrows v)$ by Lemma 5.*

Proof. Consider any pair of vertices u, v with roundtrip distance $d = d(u \leftrightarrows v) \in [L/(1+\epsilon), L]$, and a shortest cycle P going through u, v with length d.

During `Cover`(G,k,L,ϵ), consider the vertices on P that are first removed from U. Suppose w is one of the first removed vertices, and w is removed as a member of $Ball_{U_c}(c,(h_c-1)step_c)$ centered at c. This is to say $d_{U_c}(c \leftrightarrows w) < (h_c-1)step_c$.

Case 1: $step_c \geq d$. Then

$$d_{U_c}(c \leftrightarrows u) \leq d_{U_c}(c \leftrightarrows w) + d_{U_c}(w \leftrightarrows u) < (h_c-1)step_c + d \leq h_c step_c,$$

and $u \in Ball_{U_c}(c, h_c step_c)$. The second inequality holds because U_c is the remaining vertex set before removing w, so by definition of w, all vertices on P are in U_c. Symmetrically $v \in Ball_{U_c}(c, h_c step_c)$. `InOutTrees`$(U_c, c, h_c step_c)$ builds a detour cycle passing u, v with length $< 2h_c step_c$. By Lemma 7, we have $h_c \leq k-1$. Also $step_c \leq L \leq (1+\epsilon)d$, therefore we build a detour of length $< 2(k-1)step_c \leq (2k-2)(1+\epsilon)d$ in $\hat{E}$.

Case 2: $step_c < d$. Because $d \leq L$, this case can only occur when $step_c = R(c)/(k-1)$. Because c is chosen before u, $R(u) \leq R(c) = (k-1)step_c < (k-1)d$. By Lemma 5, E_0 can give a $(2k-1)$-approximation of d.

□

3.3 Main Construction

Now we can proceed to prove the first part of the main theorem based on a scaling on lengths of the cycles from 1 to $2nW$.

Theorem 9. *For any directed graph G with real edge weights in $[1, W]$, there exists a polynomial time constructible $(2k-1)$-roundtrip-spanner of G with $O(kn^{1+1/k}\log(nW))$ edges.*

Proof. Note that the roundtrip distance between any pair of vertices must be in the range $[1, 2(n-1)W]$. First do the preprocessing in Section 3.1. Then devide the range of roundtrip distance $[1, 2nW]$ into intervals $((1+\epsilon)^{p-1}, (1+\epsilon)^p]$, where $\epsilon = 1/(2k-2)$. Call `Cover`$(G,k,(1+\epsilon)^p,\epsilon)$ for $p = 0, \cdots, \lceil \log_{1+\epsilon}(2nW) \rceil$, and merge all returned edges to form a spanner.

First we prove that the edge size is $O(kn^{1+1/k}\log(nW))$. Preprocessing adds $O(n^{1+1/k}\log n)$ edges. `Cover`$(G,k,(1+\epsilon)^p,\epsilon)$ is called for $\log_{1+1/(2k-2)}(2nW) = O(k\log(nW))$ times. By Lemma 6, each call generates $O(n^{1+1/k})$ edges. So the total number of edges in the roundtrip spanner is $O(kn^{1+1/k}\log(nW))$.

Next we prove the stretch is $2k-1$. For any pair of vertices u, v with roundtrip distance d, let $p = \lceil \log_{1+\epsilon} d \rceil \in [0, \log_{1+\epsilon}(2nW)]$, then $d \in ((1+\epsilon)^{p-1}, (1+\epsilon)^p]$. By Lemma 8, either the returned edge set of $\texttt{Cover}(G, k, (1+\epsilon)^p, \epsilon)$ can form a detour cycle passing through u, v of length at most $(2k-2)(1+\epsilon)d = (2k-1)d$, or the edges in E_0 can form a detour cycle passing through u, v of length at most $(2k-1)d$.

In conclusion this algorithm can construct a $(2k-1)$-roundtrip-spanner with $O(kn^{1+1/k}\log(nW))$ edges. □

3.4 Construction Time

The running time of the algorithm in the proof of Theorem 9 is $O(kn(m + n\log n)\log(nW))$, in which the bottleneck is the $\tilde{O}(n)$ single-source shortest paths calls. It is also easy to see that the algorithm is deterministic. Next we analysis construction time in detail.

In preprocessing, for any $u \in V$, $R(u)$ can be calculated by running Dijkstra searches with Fibonacci heap [12] starting at u, so the calculating $R(\cdot)$ takes $O(n(m+n\log n))$ time. Finding H takes $O(n^{2-1/k})$ time by Theorem 3. Building E_0 takes $O(n^{1/k}\log n \cdot (m + n\log n))$ time.

A `Cover` call's while loop runs at most n times since each time at least one node is removed. In a loop, u can be found in $O(n)$, and all other operations regarding roundtrip balls can be done in linear time by Dijkstra searches starting at u on $G[U]$. Therefore a `Cover` call takes $O(n(m + n\log n))$ time.

`Cover` is called $O(k\log(nW))$ times. Combined with the preprocessing time, the total construction time is $O(kn(m + n\log n)\log(nW))$.

4 Removing the Dependence on W

The size of the roundtrip spanner in Section 3 is dependent on maximum edge weight W. In this section we can remove this dependence by designing the scaling approach more carefully, then the stretch will be $(2k - 1 + O(1/n^c))$ for any constant c.

Our idea is similar to that in [15]. When we consider the roundtrip distances between $[L/(1+\epsilon), L]$, all cycles with length $\le L/n^3$ have little affect so we can contract them into one node, and all edges with length $> (2k-1)L$ cannot be in any $(2k-1)L$ detour cycles, so it can be deleted. Thus, an edge with length l can only be in the $O(\log_{1+\epsilon} n)$ iterations for L between $l/(2k-1)$ and $l \cdot n^3$ (based on the girth of this edge). We can also show that the number of strongly connected components in the graphs considered in all iterations is bounded by $O(kn\log n)$, thus we can get an $O(kn^{1+1/k}\log n)$-size spanner. However, to get the $(2k-1+o(1))$-stretch in the presence of contracted nodes, we need a careful analysis.

First we define the girth of an edge, and study its properties:

Definition 10. *We define the* girth of an edge *e in G to be the length of minimum directed cycle containing e, and denote it by $g(e)$.*

It is easy to see that for $e = (u, v)$, $d(u \leftrightarrows v) \le g(e)$. In $O(n(m+n\log n))$ time we can compute $g(e)$ for all edges e in G [12]. We can get the following observation:

Lemma 11. *Given a length L, the edge set $\hat{E}$ returned by* `Cover`(G, k, L, ϵ) *does not include edges e with $g(e) > (k-1)L$.*

Proof. Since in `Cover`(G, k, L, ϵ) we include in-out-trees (in a subgraph of G) from u with radius $h \cdot step$, where $h \leq k-1$ by Lemma 7 and $step \leq L$, so the radius is at most $(k-1)L$. Any edge with girth greater than $(k-1)L$ cannot be in such in-out-trees. □

The main algorithm is shown in Algorithm 2.

Algorithm 2 Spanner$(G(V,E), k)$

1: Do the preprocessing in Section 3.1. Let E_0 be the added edges
2: $\hat{E} \leftarrow E_0$
3: $\epsilon \leftarrow 1/(2k-2)$
4: **for** $p \leftarrow 0$ to $\lceil \log_{1+\epsilon}(2nW) \rceil$ **do**
5:     $L \leftarrow (1+\epsilon)^p$
6:     Contract all edges e with $g(e) \leq L/n^3$ in G to form a graph G_p
7:     (Delete edges e with $g(e) > 2(k-1)L$ form G_p)
8:     For all edges $e = (u,v)$ in G_p, if v is contracted from q original vertices in G, let $w(u,v) \leftarrow w(u,v) + (q-1) \cdot L/n^3$ in G_p
9:     Recalculate $R(\cdot)$ with respect to G_p, and call it $R_p(\cdot)$
10:     $\hat{E} \leftarrow \hat{E} \cup$ Cover(G_p, k, L, ϵ)
11: **end for**
12: **return** $H(V, \hat{E})$

We can check that Lemma 8 still holds when calling `Cover`(G_p, k, L, ϵ) w.r.t. $R_p(\cdot)$ and the weights adjusted in line 8 in the current graph G_p. We adjust the weight in G_p in order to give an upper bound of the roundtrip distance in the original graph G:

Lemma 12. *In the graph G_p during the algorithm, if a contracted vertex v in G_p contains original vertices $U = \{u_1, u_2, \cdots, u_q\}$ of G, then every $u_i \in U$ is in a cycle in the subgraph $G[U]$ of length at most $(1+\epsilon)^p/n^3$. So the roundtrip distances between vertices in U are at most $(q-1) \cdot (1+\epsilon)^p/n^3$.*

Proof. If an edge e is contracted in G_p, then $g(e) \leq (1+\epsilon)^p/n^3$. Every edge f in the cycle with length $g(e)$ must have $g(f) \leq g(e)$, so such edges are also contracted to the same vertex in G_p. It is also easy to see that the roundtrip distance between $u_i, u_j \in U$ is at most $(q-1) \cdot (1+\epsilon)^p/n^3$ since every contracted edge corresponds to a cycle of length $(1+\epsilon)^p/n^3$. □

Lemma 13. *For u, v in the original graph, let the contracted vertices in G_p containing them are u', v', respectively. If u' and v' are different and strongly connected in G_p, then their roundtrip distance d' in the weight adjusted graph G_p satisfies $d(u \leftrightarrows v) \leq d' \leq d(u \leftrightarrows v) + (1+\epsilon)^p/n^2$.*

Proof. In the minimum cycle containing u and v in the original graph G, for the vertices in the cycle which is contracted to a vertex w' in G_p, find the first and last vertices contracted to w' in the cycle, and call them w_1 and w_2. Then by Lemma 12, $d(w_1 \leftrightarrows w_2) \leq (q-1) \cdot (1+\epsilon)^p/n^3$ if w' contains q original vertices. When we contract w_1 and w_2 to w' in G_p and adjust the weights in line 8, we add $(q-1) \cdot (1+\epsilon)^p/n^3$ to every edge pointing to w', which is at least the distance from w_1 to w_2, thus we have $d' \geq d(u \leftrightarrows v)$. Also in every roundtrip distance in G_p, the amount of adjustment is at most $(1+\epsilon)^p/n^2$, so $d' \leq d(u \leftrightarrows v) + (1+\epsilon)^p/n^2$. □

Since $G_{p'}$ can be seen as a graph contracted from G_p if $p' > p$, we have the following corollary:

Corollary 14. *During the algorithm* `Spanner`*(G, k), when more and more edges are contracted, the roundtrip distance between every pair of vertices cannot decrease in the weight adjusted G_p.*

By Lemma 11, `Cover`(G_p, k, L, ϵ) will not include edges with girth longer than $(k-1)L$ in the weight adjusted G_p, thus will not include edges with girth longer than $2(k-1)L$ in G. We include line 7 for our analysis, but it is not necessary.

Lemma 15. *The subgraph returned by algorithm* `Spanner`*(G, k) has $O(kn^{1+1/k} \log n)$ edges.*

Proof. Preprocessing adds $O(n^{1+1/k} \log n)$ edges as in Section 3.1. Next we count `Cover`(G, k, L, ϵ)'s returned edges.

We remove the directions of all edges in G to get an undirected graph G', and remove the directions of all edges in every G_p to get an undirected graph G'_p, but define the weight of an edge e in G' and every G'_p to be the girth $g(e)$ in G. Let F be a minimum spanning forest of G' w.r.t. the girth $g(e)$. We can see that in iteration p, if we remove edges in F with $g(e) > 2(k-1)(1+\epsilon)^p$ and contract edges e with $g(e) \leq (1+\epsilon)^p/n^3$ in F, then the connected components in F will just be the connected components in G'_p, which are the strongly connected components in G_p. This is because of the cycle property of MST: If an edge $e = (u, v)$ in G'_p has $g(e) \leq (1+\epsilon)^p/n^3$, then in F all edges in the path connecting u, v have girth $\leq (1+\epsilon)^p/n^3$, thus u, v are already contracted; If an edge $e = (u, v)$ in G'_p has $g(e) \leq 2(k-1)(1+\epsilon)^p$, then in F all edges in the path connecting u, v have girth $\leq 2(k-1)(1+\epsilon)^p$, so u, v are in the same component in F.

So the total size of connected components $\{C : |C| \geq 2\}$ in G'_p is at most 2 times the number of edges e in F with $(1+\epsilon)^p/n^3 < g(e) \leq 2(k-1)(1+\epsilon)^p$, and every edge in F can be in at most $\log_{1+\epsilon} 2(k-1)n^3 = O(k \log n)$ number of different G'_p. Thus, the total number of connected components with size at least 2 in all G'_p is bounded by $O(kn \log n)$. By Lemma 6, after calling `Cover`(G_p, k, L, ϵ), $\hat{E}$ will have $|C|n^{1/k}$ new edges for every connected component C with $|C| \geq 2$ in G'_p. Thus the total number of edges in the subgraph returned by `Spanner`(G, k) is bounded by $O(kn^{1+1/k} \log n)$. ☐

We can now prove the stretch of this roundtrip spanner. Note that in the algorithm $L = (1+\epsilon)^p$ is an upper bound for the roundtrip distance d' in G_p, but not for the original distance in G, so we need to analyze it carefully.

Lemma 16. *Algorithm* `Spanner`*(G, k) constructs a $(2k-1+O(1/n))$-roundtrip-spanner of G.*

Proof. We prove it by the induction on p. When p is small, there is no contract vertices in G_p, then it is the same as Lemma 8, so we get the $(2k-1)$-stretch. Next we assume that the distances between vertices of G contracted in the same vertex in G_p have a $(2k-1)$-approximation in $\hat{E}$. (Thus, they are strongly connected in $\hat{E}$.)

For any pair of vertices u, v with roundtrip distance $d = d(u \leftrightarrows v)$ on G. By Lemma 13 and Corollary 14, there exists a p such that in G_p the roundtrip distance d' between u' and v' satisfies $(1+\epsilon)^{p-1} < d' \leq (1+\epsilon)^p$, where u', v' are the contracted vertices in G_p containing u, v, respectively. (Suppose $p' = \lceil \log_{1+\epsilon} d \rceil$, then $d \in ((1+\epsilon)^{p'-1}, (1+\epsilon)^{p'}]$, and $(1+\epsilon)^{p'-1} < d \leq d_{G_{p'}}(u \leftrightarrows v) \leq d + (1+\epsilon)^{p'}/n^2 \leq (1+1/n^2)(1+\epsilon)^{p'}$. And in $G_{p'+1}$ the roundtrip distance cannot be smaller than in $G_{p'}$ and $d_{G_{p'+1}}(u \leftrightarrows v) \leq d + (1+\epsilon)^{p'+1}/n^2 \leq (1+(1+\epsilon)/n^2)(1+\epsilon)^{p'} < (1+\epsilon)^{p'+1}$, when $\epsilon >> 1/n^2$. So p can be p' or $p'+1$.)

By Lemma 8, either $\texttt{Cover}(G_p, k, (1+\epsilon)^p, \epsilon)$'s returned edge set $\hat{E}$ can form a detour cycle with length at most $(2k-2)(1+\epsilon)d'$ on graph G_p, or $R_p(u') < (k-1)d'$. In the latter case since in weight adjusted G_p distance cannot be decreased, and some vertices are contracted in G_p, $R(u)$ in the original graph G must also be smaller than $(k-1)d'$, by Lemma 5, in E_0 the roundtrip distance between u, v is at most $(2k-2)d' + d \le (2k-1+O(k/n^2))d \le (2k-1+O(1/n))d$. (This is because we have $d' \le d + (1+\epsilon)^p/n^2$ by Lemma 13.) In the former case, by inductive hypothesis, the original vertices contracted in one vertex in G_p are strongly connected in $\hat{E}$, so the roundtrip distance d'' between u and v in final $\hat{E}$ is at most

$$(2k-2)(1+\epsilon)d' + (1+\epsilon)^p/n^2 \le (2k-1+O(1/n^2))d' \le (2k-1+O(1/n))d$$

($k \ge 2$ and n is not too small.) □

Note that we can make the stretch to be $2k-1+O(1/n^c)$ for any constant c if we contract edges with girth $\le L/n^{c+2}$ in line 6.

4.1 Construction Time

The analysis of `Spanner`'s running time is similar to Section 3.4. Compared with the primary version, `Spanner` adds operations of building G_p and recalculating weights and $R(\cdot)$ before each `Cover` call. We also need to calculate $g(\cdot)$ in preprocessing, which can done by n Dijkstra searches. G_p can be built in $O(m)$ time. Recalculating $R(\cdot)$ takes $O(n(m + n\log n))$ and recalculating weights takes $O(m)$ time. `Cover` is called $\log_{1+\epsilon'}(2nW) = O(k\log(nW))$ times. Therefore the total construction time is still $O(kn(m+n\log n)\log(nW))$.

References

[1] Donald Aingworth, Chandra Chekuri, Piotr Indyk, and Rajeev Motwani. Fast estimation of diameter and shortest paths (without matrix multiplication). *SIAM Journal on Computing*, 28(4):1167–1181, 1999.

[2] Ingo Althöfer, Gautam Das, David Dobkin, Deborah Joseph, and José Soares. On sparse spanners of weighted graphs. *Discrete & Computational Geometry*, 9(1):81–100, Jan 1993.

[3] Piotr Berman, Arnab Bhattacharyya, Konstantin Makarychev, Sofya Raskhodnikova, and Grigory Yaroslavtsev. Improved approximation for the directed spanner problem. In *International Colloquium on Automata, Languages, and Programming*, pages 1–12, Berlin, Heidelberg, 2011. Springer.

[4] Piotr Berman, Sofya Raskhodnikova, and Ge Ruan. Finding sparser directed spanners. In *IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science*, pages 424–435, Dagstuhl, 2010. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik.

[5] A. Bhattacharyya, E. Grigorescu, K. Jung, S. Raskhodnikova, and D. Woodruff. Transitive-closure spanners. *SIAM Journal on Computing*, 41(6):1380–1425, 2012.

[6] Shiri Chechik, Yang P. Liu, Omer Rotem, and Aaron Sidford. Improved Girth Approximation and Roundtrip Spanners. *arXiv e-prints*, page arXiv:1907.10779, Jul 2019.

[7] Lenore J Cowen and Christopher G Wagner. Compact roundtrip routing for digraphs. In *Proceedings of the 10th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 885–886, Philadelphia, 1999. SIAM.

[8] Lenore J Cowen and Christopher G Wagner. Compact roundtrip routing in directed networks. In *Proceedings of the 19th Annual ACM Symposium on Principles of Distributed Computing*, pages 51–59, New York, 2000. ACM.

[9] Michael Dinitz and Robert Krauthgamer. Directed spanners via flow-based linear programs. In *Proceedings of the 43rd Annual ACM Symposium on Theory of Computing*, pages 323–332, New York, 2011. ACM.

[10] Dorit Dor, Shay Halperin, and Uri Zwick. All-pairs almost shortest paths. *SIAM Journal on Computing*, 29(5):1740–1759, 2000.

[11] Paul Erdös. Extremal problems in graph theory. In *Theory of Graphs and Its Applications (Proc. Sympos. Smolenice, 1963)*, pages 29–36, Prague, 1964. Publ. House Czechoslovak Acad. Sci.

[12] Michael L Fredman and Robert Endre Tarjan. Fibonacci heaps and their uses in improved network optimization algorithms. *Journal of the ACM*, 34(3):596–615, 1987.

[13] Jakub Pachocki, Liam Roditty, Aaron Sidford, Roei Tov, and Virginia Vassilevska Williams. Approximating cycles in directed graphs: Fast algorithms for girth and roundtrip spanners. In *Proceedings of the 29th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1374–1392, Philadelphia, 2018. SIAM.

[14] Sofya Raskhodnikova. Transitive-closure spanners: A survey. In *Property Testing*, pages 167–196. Springer, Berlin, Heidelberg, 2010.

[15] Liam Roditty, Mikkel Thorup, and Uri Zwick. Roundtrip spanners and roundtrip routing in directed graphs. *ACM Trans. Algorithms*, 4(3):29:1–29:17, July 2008.

[16] Mikkel Thorup and Uri Zwick. Approximate distance oracles. In *Proceedings of the 33rd Annual ACM Symposium on Theory of Computing*, pages 183–192, New York, 2001. ACM.

[17] Chun Jiang Zhu and Kam-Yiu Lam. Deterministic improved round-trip spanners. *Information Processing Letters*, 129:57 – 60, 2018.

[18] Uri Zwick. Exact and approximate distances in graphs: a survey. In *European Symposium on Algorithms*, pages 33–48, Berlin, Heidelberg, 2001. Springer.