Roundtrip Spanners and Roundtrip Routing in Directed Graphs

Liam Roditty * *Mikkel Thorup* † *Uri Zwick* *

Abstract

We introduce the notion of *roundtrip-spanners* of weighted *directed* graphs and describe efficient algorithms for their construction. For every integer $k \geq 1$ and any $\epsilon > 0$, we show that any directed graph on n vertices with edge weights in the range $[1, W]$ has a $(2k + \epsilon)$-roundtrip-spanner with $O(\frac{k^2}{\epsilon} n^{1+1/k} \log(nW))$ edges. We then extend these constructions and obtain compact roundtrip routing schemes. For every integer $k \geq 1$ and every $\epsilon > 0$, we describe a roundtrip routing scheme that has stretch $4k + \epsilon$, and uses at each vertex a routing table of size $\tilde{O}(\frac{k^2}{\epsilon} n^{1/k} \log(nW))$. We also show that any weighted directed graph with *arbitrary* positive edge weights has a 3-roundtrip-spanner with $O(n^{3/2})$ edges. This result is optimal. Finally, we present a stretch 3 roundtrip routing scheme that uses local routing tables of size $\tilde{O}(n^{1/2})$. This routing scheme is essentially optimal. The roundtrip-spanner constructions and the roundtrip routing schemes for directed graphs that we describe are only slightly worse than the best available spanners and routing schemes for undirected graphs. Our roundtrip routing schemes substantially improve previous results of Cowen and Wagner. Our results are obtained by combining ideas of Cohen, Cowen and Wagner, Thorup and Zwick, with some new ideas.

1 Introduction

The main results of this paper are improved roundtrip routing schemes for directed graphs. To obtain a clearer presentation of these schemes, and the techniques used to construct them, we begin with a discussion of spanners and roundtrip-spanners.

A spanning tree is the sparsest subgraph that preserves the connectivity of a graph. A *spanner* is a relatively sparse subgraph that preserves, to a certain extent, the *distances* in a graph. More formally, let $G = (V, E)$ be an undirected graph with a weight (or length) function $w : E \rightarrow \mathbb{R}^+$ defined on its edges. A subgraph $H = (V, E')$ of G is said to be a *t-spanner* of G if and only if $\delta_H(u, v) \leq t \cdot \delta_G(u, v)$, for every $u, v \in V$. (Here $\delta_G(u, v)$ is the distance between u and v in G.) For any integer $k \geq 1$, any weighted undirected graph on n vertices has a $(2k - 1)$-spanner with $O(n^{1+1/k})$ edges. (See Althöfer *et al.* [ADD+93], Peleg [Pel00], Thorup and Zwick [TZ01a].) This tradeoff between the *stretch* of a spanner and its size is known to be tight for $k = 1, 2, 3, 5$, and is believed to be tight for any k. (See, e.g., the discussion in [TZ01a].) Spanners have many applications in distributed computing (see Peleg [Pel00]).

Can the notion of spanners be extended to directed graphs? The definition of spanners still makes sense in the directed setting, but the notion becomes uninteresting. Take a complete directed bipartite graph. Removing any edge changes the reachability in the graph, so no strict subgraph is a t-spanner of this graph, for any finite t. Thus, no general tradeoff between stretch and size is possible for directed graphs.

Cowen and Wagner [CW99, CW00], working on routing problems, made the very nice observation that some of the techniques related to approximated distances and paths in undirected graphs can be used on directed graphs, if *roundtrip distances*, instead of *one way* distances are considered. Here the roundtrip distance from u to v is the one-way distance from u to v plus the one way distance from v to u. Roundtrip distances in directed graphs are very natural generalizations of distances in undirected graphs. In particular, roundtrip distances are symmetric, and hence they define a metric for directed graphs.

Implicit in the work of Cowen and Wagner [CW99, CW00] is the notion of *roundtrip spanners*. We make this notion explicit and obtain vastly improved constructions of roundtrip spanners. Let $G = (V, E)$ be a *directed* graph with a weight (or length) function $w : E \rightarrow \mathbb{R}^+$ defined on its edges. A subgraph $H = (V, E')$ of G is said to be a *t-roundtrip-spanner* of G if and only if $\delta_H(u \rightleftarrows v) \leq t \cdot \delta_G(u \rightleftarrows v)$, for every $u, v \in V$. (Here $\delta_G(u \rightleftarrows v) = \delta_G(u \rightarrow v) + \delta_G(v \rightarrow u)$ is the roundtrip distance between u and v in G.)

*School of Computer Science, Tel Aviv University, Tel Aviv 69978, Israel. E-mail: {liamr,zwick}@tau.ac.il. This research was supported by the ISRAEL SCIENCE FOUNDATION (grant no. 475/98).

†AT&T Labs - Research, 180 Park Avenue, Florham Park, NJ 07932, USA. E-mail address: mthorup@research.att.com.

The results of Cowen and Wagner [CW99, CW00] imply that, for any integer $k \geq 1$, any weighted directed graph on n vertices has a $(2^k - 1)$-roundtrip-spanner of size $\tilde{O}(n^{1+1/k})$. For $k = 2$, this result is very close to being optimal. For larger values of k, the stretch obtained is extremely large. We show that for any $k \geq 1$ and any fixed $\epsilon > 0$, any weighted directed graph on n vertices with edge weights in the range $[1, W]$ has a $(2k + \epsilon)$-roundtrip-spanner of size $O(\frac{k^2}{\epsilon} n^{1+1/k} \log(nW))$. This is very close to the best available, and believed optimal, results for undirected graphs. For $k = 2$ we have a specialized construction that shows that any n-vertex digraph has a 3-roundtrip-spanner of size $O(n^{3/2})$. This improves the corresponding result of Cowen and Wagner [CW99, CW00] by a polylogarithmic factor and makes it completely tight.

There are several different techniques for constructing sparse spanners for undirected graphs. (See Althöfer *et al.* [ADD+93], Awerbuch *et al.* [ABCP99], Cohen [Coh99], Peleg [Pel00], Thorup and Zwick [TZ01a].) Interestingly, some of these techniques can be adapted to the directed setting while others cannot, at least not in an obvious way. The technique of Thorup and Zwick [TZ01a], for example, which gives the fastest way of constructing $(2k - 1)$-spanners of size $O(n^{1+1/k})$ for undirected graphs, extends to the directed case when $k = 2$, but not for larger values of k. Using it we obtain the optimal construction of 3-roundtrip-spanners. To obtain sparser roundtrip-spanners with a larger stretch we use an extension of the nice technique of Cohen [Coh99].

One of the uses of spanners in distributed computing is the design of compact *routing schemes*. Let $G = (V, E)$ be a graph with *port numbers* assigned to its links. (In a directed graph a link is simply a directed edge. In undirected graphs, each undirected edge represent two links, one in each direction.) The port number assigned to each edge is unique. A routing scheme assigns to each vertex $v \in V$ a short label, or name, denoted $label(v)$. Each vertex $v \in V$ is also equipped with a local routing table $RT(v)$. A vertex u that wants to send a packet p to vertex v is assumed to know $label(v)$. It construct a message $(label(v), p)$ that contains $label(v)$ at its *header*, and passes its local routing daemon. (Each vertex of the network is assumed to run a local routing daemon.) When a routing daemon at some vertex w of the network receives a message with $label(v)$ at its header, it has to decide, based on $label(v)$ and on its local routing table $RT(w)$, on which out-going link to forward this message. This out-going link is identified by its port number. For more on routing schemes see Peleg [Pel00], Thorup and Zwick [TZ01b].

The *stretch* of a routing scheme is the worst possible ratio between the length of the route that a message follows from u to v, divided by the distance from u to v in the graph. The *roundtrip-stretch* of a routing scheme is the corresponding quantity for a roundtrip route from u to v and back.

It is easy to obtain stretch 1, i.e., route along shortest paths, if a full routing table of size $O(n)$ is stored at each vertex. (We measure the size of a routing table in $O(\log n)$-bit words.) However, it is unrealistic to expect each vertex of a network to hold a routing table that is almost as big as the network itself. Hence, there is great interest what stretches are attainable with much smaller routing tables. Preferably, we would like *all* routing tables to be small.

Thorup and Zwick [TZ01b], improving many previous results, obtained essentially optimal routing schemes for undirected graphs. For every integer $k \geq 1$, they describe a routing scheme that uses $o(\log^2 n)$-bit labels, local routing tables of size $\tilde{O}(n^{1/k})$, and has stretch $2k - 1$. For $k \geq 3$, these routing schemes rely on an initial *handshaking* process (see [TZ01b] for details). Without such a handshaking process, the stretch increases to $4k - 5$.

Many real communication networks cannot be modeled accurately by undirected graphs. Some of the links in a network may be one-directional, and in some of the bidirectional links, the speed, or capacity, may depend on the direction. Thus, the study of routing schemes for *directed* graphs is well motivated. However, attempts to design compact routing schemes for directed graphs meet essentially the same obstacles met when trying to constructed spanners for directed graphs.

It is in this context that Cowen and Wagner [CW99, CW00] suggest the consideration of *roundtrip routing schemes*. This is also supported by the fact that many of the real-life applications actually require roundtrip routes. The IP protocol, for example, requires an acknowledgment to be sent after each packet is received.

Cowen and Wagner [CW99, CW00] describe, for any integer $k \geq 1$, a roundtrip routing scheme of stretch $2^k - 1$ that uses local routing tables of *average* size $\tilde{O}(n^{1/k})$. The size of some of the routing tables in their schemes may be as large as $\Omega(n)$. They also describe a variant of their scheme in which the *maximum* table size is $\tilde{O}(n^{(3^{k-1}+1)/(2\cdot 3^{k-1})})$ and the stretch is still $2^k - 1$. Note, however, that the maximum table size is still $\Omega(n^{1/2})$, for any $k \geq 1$.

We substantially improve these results and obtain, for any integer $k \geq 1$ and any fixed $\epsilon > 0$, a roundtrip routing scheme of stretch $4k + \epsilon$ and *maximum* table size $\tilde{O}(n^{1/k})$, assuming that the edge weights are polynomially bounded. With routing tables of *total* size $\tilde{O}(n^{1+1/k})$, we get stretch $4k+\epsilon$, instead of 2^k-1. Furthermore, we do that while keeping the maximum table size at $\tilde{O}(n^{1/k})$, instead of $\Omega(n^{1/2})$. Furthermore, for $k=2$, we have an improved roundtrip routing scheme that has stretch 3 and uses routing tables of maximum size $\tilde{O}(n^{1/2})$. This scheme is essentially optimal. Our roundtrip routing schemes do *not* use handshaking.

Our improvements, for general k, over the results of Cowen and Wagner [CW99, CW00] are based on a construction of Cohen [Coh99] that predates the results of Cowen and Wagner [CW99, CW00]. Our improvement for stretch 3 extends a recent work by the last two authors [TZ01a, TZ01b].

Note that the stretch, $4k+\epsilon$, of our roundtrip routing schemes is about twice as large as the stretch of our spanners. This is the price paid for being able to make local routing decisions. On the positive side, note that our roundtrip routing schemes do not lag much behind the best known routing schemes for undirected graphs that do not use handshaking that, for the same table size, have stretch $4k-5$ (see [TZ01b]).

The rest of this paper is organized as follows. In the next section we present efficient constructions of sparse roundtrip-spanners. In Section 3 we present our roundtrip routing schemes. In Section 4 we present an essentially optimal roundtrip routing scheme with stretch 3. We end in Section 5 with some concluding remarks and open problems.

2 Roundtrip-spanners

Let $G=(V,E)$ be an undirected graph with a weight (or length) function $w: E \rightarrow \mathbb{R}^+$ defined on its edges. A subgraph $H=(V,E')$ of G is said to be a t-spanner of G if and only if $\delta_H(u,v) \leq t \cdot \delta_G(u,v)$, for every $u,v \in V$. (Here $\delta_G(u,v)$ is the distance between u and v in G.) For any integer $k \geq 1$, any weighted undirected graph on n vertices has a $(2k-1)$-spanner with $O(n^{1+1/k})$ edges. (See Althöfer *et al.* [ADD+93], Peleg [Pel00], Thorup and Zwick [TZ01a].)

The spanner definition can also be applied to directed graphs. However, it is not difficult to see that for any finite t, there are n-vertex graphs with $\Omega(n^2)$ edges that none of their proper subgraphs is a t-spanner of them. (Consider a complete bipartite graph were all edges are directed in the same direction.)

Cowen and Wagner [CW99, CW00], in the context of routing problems, suggested considering *roundtrip* distances in directed graphs, instead of *one-way* distances. They showed that some of the techniques that were used in connection with distances in undirected graphs could be used in connection with roundtrip distances in directed graphs.

Let $G=(V,E)$ be a weighted directed graph. We let $\delta(u \rightarrow v)$ and $\delta(v \leftarrow u)$ be the *one-way* distance from u to v in the graph G. We let $\delta(u \rightleftarrows v) = \delta(u \rightarrow v) + \delta(u \leftarrow v)$ be the *roundtrip* distance from u to v in G. Clearly $\delta(u \rightleftarrows v) = \delta(v \rightleftarrows u)$. When we want to stress the fact that the distances considered are in the graph G, we add G as a subscript, as in $\delta_G(u \rightarrow v)$ and $\delta_G(u \rightleftarrows v)$. The following definition is now natural:

DEFINITION 2.1. (ROUNDTRIP-SPANNERS) *Let $G=(V,E)$ be a weighted directed graph. A subgraph $H=(V,E')$ is said to be a t-roundtrip-spanner of G if and only if $\delta_H(u \rightleftarrows v) \leq t \cdot \delta_G(u \rightleftarrows v)$, for every $u,v \in V$.*

Note that if $G=(V,E)$ is an undirected graph, then $\delta_G(u \rightleftarrows v) = 2\delta(u,v)$, for every $u,v \in V$, and thus roundtrip-spanners of undirected graphs are just conventional spanners.

Although the explicit definition of roundtrip-spanners is new, the following result is implicit in Cowen and Wagner [CW00]:

THEOREM 2.1. (COWEN AND WAGNER [CW00]) *For every integer $k \geq 1$, every weighted directed graph on n vertices has a (2^k-1)-roundtrip-spanner with $\tilde{O}(n^{1+1/k})$ edges.*

We substantially improve this result and prove:

THEOREM 2.2. *For every integer $k \geq 1$ and every $\epsilon > 0$, every weighted directed graph on n vertices with edge weights taken from $[1,W]$ has a $(2k+\epsilon)$-roundtrip-spanner with $O(\frac{k^2}{\epsilon} n^{1+1/k} \log(nW))$ edges.*

If $W = n^{O(1)}$, i.e., the edge weights in the graph are of polynomial size, then for roughly the same number of edges, the stretch is reduced from 2^k-1 to $2k+\epsilon$, an exponential improvement.

Theorem 2.2 is obtained by adapting an elegant construction of Cohen [Coh99] for the directed case.

Let $G=(V,E)$ be an n-vertex graph with edge weights taken from $[1,W]$. Let $1 \leq R \leq 2nW$. Following Cohen [Coh99], we construct a subgraph $H=(V,E')$ of G, with $|E'| = O(kn^{1+1/k})$, such that for every $u,v \in V$, if $\delta_G(u \rightleftarrows v) \leq R$, then $\delta_H(u \rightleftarrows v) \leq 2k \cdot R$. Clearly, if we take the union of such subgraphs, corresponding to

$R = (1+\epsilon)^i$, for $1 \le i \le \log_{1+\epsilon}(2nW)$, we get a $2k(1+\epsilon)$-roundtrip-spanner of G with $O(kn^{1+1/k}\log_{1+\epsilon}(nW))$ edges.

The subgraph $H = H(R)$ is constructed as follows. Let $V' \subseteq V$ be a subset of vertices. For every $v \in V$ and $r \ge 0$, we let $\text{ball}_{V'}(v, r)$ be the set of vertices that are of roundtrip distance at most r from v in the subgraph $G' = G[V']$ induced by V'. In other words, $\text{ball}_{V'}(v, r) = \{u \in V' \mid \delta_{G'}(v, u) \le r\}$. We say that a set B is a ball centered at v if and only if $B = \text{ball}_{V'}(v, r)$, for some $V' \subseteq V$ and $r \ge 0$. Next, following Cohen [Coh99], we define:

DEFINITION 2.2. ((k, R)-(ROUNDTRIP)-COVER) *A collection $\mathcal{C}$ of balls is a (k, R)-(roundtrip)-cover of a directed graph $G = (V, E)$ if and only if each ball in $\mathcal{C}$ is of radius at most kR, and for every $u, v \in V$ such that $\delta_G(u \rightleftarrows v) \le R$, there is a ball $B \in \mathcal{C}$ such that $u, v \in B$.*

For brevity, we usually refer to (k, R)-roundtrip-covers simply as (k, R)-covers. Figure 1 gives an extremely simple algorithm for constructing a (k, R)-roundtrip-cover of a graph. In the algorithm, **sample**(V_i, p) is a procedure that returns a random subset of V_i. Each element of V_i belongs to the returned set, independently, with probability p. The algorithm chooses collections of balls of decreasing radii. At the first iteration, when $i = k - 1$, about $n^{1/k}$ balls of radius kR are chosen. In the second iteration, with $i = k - 2$, about $n^{2/k}$ balls of radius $(k-1)R$ are chosen, etc. In each iteration, when a ball $B_{V_i}(v, (i+1)R)$ is added to the cover $\mathcal{C}$, its *core* $B_{V_i}(v, iR)$ is removed from the graph. The following theorems are adaptations, and slight improvements of results of Cohen [Coh99]:

THEOREM 2.3. *The collection $\mathcal{C}$ constructed by calling* **cover**(G, k, R) *is a (k, R)-cover of the* directed *graph $G = (V, E)$. For every $v \in V$, the* expected *number of balls in the cover containing v is at most $kn^{1/k}$.*

THEOREM 2.4. *If the sampling probability in the i-th iteration of algorithm* **cover** *is increased from $n^{-i/k}$ to $2(n \ln n)^{-i/k}$, then with very high probability, each vertex $v \in V$ is contained in at most $O(kn^{1/k}(\ln n)^{1-1/k})$ balls of the cover $\mathcal{C}$ returned by the algorithm.*

For completeness, we provide a proof of the first theorem. The second proof is similar and is omitted from this extended abstract. Our proofs are slightly different than the proofs given by Cohen, allowing us to get logarithmic improvements over her results. Our arguments are somewhat reminiscent of arguments used in [TZ01a].

Proof. We first argue that the set $\mathcal{C}$ generated by the algorithm is indeed a (k, R)-cover. Let $u, v \in V$ be such that $\delta_G(u \rightleftarrows v) \le R$. Let c be a closed tour in G of total length at most R that contains u and v. Let i be the largest index for which a vertex u' on c is captured by the core $\text{ball}_{V_i}(w, iR)$ of a ball $\text{ball}_{V_i}(w, (i+1)R)$ that is added to the cover. By the choice of i, all the vertices on c are contained in V_i. Thus $\delta_{G[V_i]}(w \rightleftarrows v) \le \delta_{G[V_i]}(w \rightleftarrows u') + \delta_{G[V_i]}(u' \rightleftarrows v) \le iR + R = (i+1)R$, and $v \in \text{ball}_{V_i}(w, (i+1)R)$. Similarly, $u \in \text{ball}_{V_i}(w, (i+1)R)$, as required.

We now bound the expected number of balls in the cover that contain a given vertex $v \in V$. We show that at each iteration, the expected number of balls containing v that are added to the cover is at most $n^{1/k}$.

Consider $B = \text{ball}_{V_{i+1}}(v, (i+1)R)$, the set of vertices that are at distance at most $(i+1)R$ from v at the beginning of the $(i+1)$-st iteration. (The first iteration is iteration number $k-1$, the second is iteration number $k-2$, etc.) Let $\ell = |B|$ be the number of these vertices. Each vertex $u \in B$ has a probability of $n^{-(i+1)/k}$ of being chosen to S_{i+1}. If a vertex u is chosen, then v would be contained in the core $\text{ball}_{V_{i+1}}(u, (i+1)R)$, and v would be removed from the graph and would not be contained in any ball added to the cover at the i-th iteration. Note that if $v \in \text{ball}_{V_i}(u, (i+1)R)$, then clearly $v \in \text{ball}_{V_{i+1}}(u, (i+1)R)$, and thus $u \in B$.

Thus, for every $u \in B$, the probability that u is chosen at the i-th iteration and that $v \in \text{ball}_{V_i}(u, (i+1)R)$ is at most $(1 - n^{-(i+1)/k})^\ell \cdot n^{-i/k}$. (None of the vertices of B should be chosen at the $(i+1)$-st, i.e., previous, iteration, and u has to be chosen at the i-th, i.e., current, iteration.) The expected number of balls that capture v in the i-th iteration is therefore, at most $\ell(1 - n^{-(i+1)/k})^\ell \cdot n^{-i/k}$.

It is easy to check that the function $x(1-p)^x$, where $0 < p < 1$ is a fixed constant, attains it maximum value of $\frac{1}{\mathrm{e}p}$ when $x = -\frac{1}{\ln(1-p)}$. It follows easily that the expected number of balls containing v that are added to the cover at the i-th iteration is at most $n^{1/k}/\mathrm{e}$, as required.

DEFINITION 2.3. (IN AND OUT TREES) *Let $G = (V, E)$ be a weighted directed graph. Let $V' \subseteq V$, $v \in V$ and $r \ge 0$. Let $B = ball_{V'}(v, r)$. We let* $\text{OutTree}(B, v)$ *be a tree containing directed shortest paths in $G' = G[V']$ from v to all the vertices of B. Similarly, we let* $\text{InTree}(B, v)$ *be a tree containing directed shortest paths from all the vertices of B to v. We let* $\text{InOutTrees}(B, v) = \text{InTree}(B, v) \cup \text{OutTree}(B, v)$. *We refer to* $\text{InOutTrees}(B, v)$ *as a* double-tree. *We*

```
algorithm cover(G, k, R)

C ← φ ; V_{k-1} ← V
for i ← k − 1 downto 0
{
    S_i ← sample(V_i, n^{-i/k})
    C ← C ∪ { ball_{V_i}(v, (i + 1)R) | v ∈ S_i }
    V_{i-1} ← V_i − ∪_{v∈S_i} ball_{V_i}(v, iR)
}
return C ;
```

Figure 1: Cohen's algorithm for constructing a (k, R)-(roundtrip)-cover

sometimes omit the center v from these notations when it is from the context.

Let $B = \text{ball}_{V'}(v, r)$. It is easy to check that if $u \in B$, and w is on a shortest path from u to v, or from v to u in $G' = G[V']$, then $w \in B$. This follows from the fact that $\delta_{G'}(v \rightleftarrows w) \le \delta_{G'}(v \rightleftarrows u)$. Thus, all the vertices in InTree(B) and OutTree(B), and thus in InOutTrees(B), are in the ball B. It follows that the number of edges in InOutTrees(B) is at most $2(|B| - 1)$. (Note that InTree(B) and OutTree(B) may share edges.) We also need the following lemma:

LEMMA 2.1. *Let $B = ball_{V'}(v, r)$, and let $u_1, u_2 \in B$. Then,* InOutTrees(B, v) *contains a closed directed tour containing u_1 and u_2 of length at most $2r$.*

Proof. By definition, InOutTrees(B, v) contained a closed tour of length $\delta_{G'}(v \rightleftarrows u_1)$ containing v and u_1, and a closed tour of length $\delta_{G'}(v \rightleftarrows u_2)$ containing v and u_2. By combining these tours we get a tour of length at most $\delta_{G'}(v \rightleftarrows u_1) + \delta_{G'}(v \rightleftarrows u_2) \le 2r$ containing u_1 and u_2.

We can now present a proof of Theorem 2.2:

Proof. **(of Theorem 2.2)** Let $\epsilon' = \frac{\epsilon}{2k}$. For every $1 \le i \le \log_{1+\epsilon'}(2nW)$, let $\mathcal{C}_i$ be a (k, R_i)-cover of G, where $R_i = (1 + \epsilon')^i$, constructed by calling **cover**(G, k, R_i). Let H be composed of the union of InOutTrees(B), for every $B \in \cup_i \mathcal{C}_i$. Clearly, the obtained graph is a $(2k+\epsilon)$-roundtrip-spanner of G and that the *expected* number of edges in it is $O(\frac{k^2}{\epsilon} n^{1+1/k} \log(nW))$.

Our roundtrip-spanner construction actually proves a stronger result:

DEFINITION 2.4. (DOUBLE-TREE COVERS) *Let $G = (V, E)$ be a weighted directed graph. A collection $\mathcal{T}$ of double-trees of G is said to be a t-double-tree-cover of G if and only if for every $u, v \in V$ there is a double-tree $T \in \mathcal{T}$ such that $u, v \in T$ and $\delta_T(u \rightleftarrows v) \le t \cdot \delta_G(u \rightleftarrows v)$.*

THEOREM 2.5. *For every integer $k \ge 1$ and every $\epsilon > 0$, every weighted directed graph on n vertices with edge weights taken from $[1, W]$ has a $(2k + \epsilon)$-double-tree-cover such that every vertex $v \in V$ is contained in at most $O(\frac{k^2}{\epsilon} n^{1/k} (\log n)^{1-1/k} \log(nW))$ double-trees.*

3 Roundtrip routing schemes

Fraigniaud and Gavoille [FG01] and Thorup and Zwick [TZ01b] describe an extremely efficient way of routing on trees:

THEOREM 3.1. ([FG01],[TZ01B]) *Let $T = (V, E)$ be an* undirected *tree on n vertices with each edge $e \in E$ assigned a unique $O(\log n)$-bit* port number*. Then, it is possible to efficiently assign each vertex $v \in V$ an $O(\log^2 n / \log\log n)$-bit label, denoted label(v), such that if $u, v \in V$, then given label(u) and label(v), and nothing else, it is possible to find, in constant time, the port number assigned to the first edge on the path in T from u to v.*

Let B be a ball centered at v. We can convert each double-tree InOutTrees(B, v) into a standard tree InOutTree(B, v) (note the slight difference in notation) by considering InTree(B, v) and OutTree(B, v) as being trees on different sets of vertices, and joining them at v. For every $u \in B$, $u \ne v$, we let u' be the copy of u in the InTree(B, v), and u'' be the copy of u in OutTree(B, v). We also let $v' = v'' = v$. For every $u_1, u_2 \in B$, there is a directed path in InOutTree(B, v) from u_1' to u_2''. Although Theorem 3.1 was stated for undirected trees, it is easy to extend it to directed trees like InOutTree(B, v), and thus to double-trees like InOutTrees(B, v).

Theorem 2.5 essentially reduces the problem of roundtrip routing in general directed graphs into the problem of routing in double-trees. Routing in double-trees could be done using the techniques of Fraigniaud and Gavoille [FG01] and Thorup and Zwick [TZ01b].

One big problem still remains, however. Suppose that u wants to route a message to v. We know that the double-tree cover $\mathcal{T}$ contains a double-tree T in which $\delta_T(u \rightleftarrows v) \le (2k + \epsilon) \cdot \delta_G(u \rightleftarrows v)$. But, how does u *identifies* this tree?

For any $R_j = (1 + \epsilon)^j$, where $1 \le j \le \log_{1+\epsilon}(nW)$, and any $0 \le i \le k - 1$, the double-tree cover $\mathcal{T}$ of

Theorem 2.5 contains at most $\tilde{O}(n^{1/k})$ double-trees of radius $(i+1)R_j$ containing v. A naive solution to the problem posed above would be to specify all these trees in *label*(v). But *label*(v) would then be of length $\Omega(kn^{1/k}\log_{1+\epsilon}(nW))$, which is too much, as we are aiming for labels of polylogarithmic size.

We can decrease the labels to polylogarithmic size at the price of doubling the resulting stretch. We need the notion of *extended core* and the following lemma:

LEMMA 3.1. *Let $\mathcal{C}$ be a (k,R)-(roundtrip)-cover of a weighted directed graph $G=(V,E)$ constructed using algorithm* **cover** *of Figure 1. Let $v\in V$. Let i be the largest index for which there is a ball $B=ball_{V_i}(w,(i+1)R)\in\mathcal{C}$, such that $v\in B'=ball_{V_i}(w,(i+1/2)R)\in\mathcal{C}$. (We refer to B' as the* extended core *of B.) Then, if $u\in V$ and $\delta_G(u\rightleftarrows v)\le R/2$, then $u\in B$.*

Proof. Let c be a shortest closed tour containing u and v. We show that $c\subseteq V_i$, and therefore, $\delta_{G[V_i]}(w\rightleftarrows u)\le\delta_{G[V_i]}(w\rightleftarrows v)+\delta_{G[V_i]}(v\rightleftarrows u)\le(i+1/2)R+R/2\le(i+1)R$, and thus $u\in B$, as required.

Suppose, therefore, for the sake of contradiction, that there is a vertex u' on c that is not in V_i. This means that there is a vertex $w'\in S_j$, for $j>i$ such that $u'\in\mathrm{ball}_{V_j}(w',jR)$. But then $\delta_{G[V_j]}(w'\rightleftarrows v)\le\delta_{G[V_j]}(w'\rightleftarrows u')+\delta_{G[V_j]}(u'\rightleftarrows v)\le jR+R/2\le(j+1/2)R$, a contradiction to the choice of i.

Let $\epsilon'=\frac{\epsilon}{4k}$. Let $v\in V$ be a vertex. For every $R_j=(1+\epsilon')^j$, where $1\le j\le\ell=\lceil\log_{1+\epsilon}(nW)\rceil$, we choose a *single* ball $\mathrm{ball}_{V_i}(w,(i+1)R_j)\in\mathcal{C}$, such that $v\in\mathrm{ball}_{V_i}(w,(i+1/2)R_j)\in\mathcal{C}$, as in Lemma 3.1, and let $cent_j(v)=w$ be its center. We let

$$label_j(v)\;=\;(cent_j(v),\textit{tree-label}_j(v))\;,$$

where *tree-label*$_j(v)$ is the label of v in the double-tree of the ball centered at $cent_j(v)$, assigned by the tree routing scheme of Theorem 3.1. Finally, we let

$$label(v)\;=\;(label_1(v),label_2(v),\ldots,label_\ell(v))\;.$$

The routing table $RT(u)$ of a vertex $u\in V$ is composed of $\ell=\lceil\log_{1+\epsilon}(nW)\rceil$ (two-level) hash tables. The j-th table $RT_j(u)$ holds the centers of the balls of the (k,R_j)-cover that contain u. (It is possible to combine these tables into a single hash table, but for simplicity, we assume that they are separate.) For each one of these centers, $RT_j(u)$ holds the tree-label of u in the corresponding double-tree.

Suppose now that u wants to send a message to v. It simply finds the smallest j for which $cent_j(v)\in RT_j(u)$. The message can then be routed on the double-tree centered at $cent_j(v)$. The label of u in that tree is extracted from $RT_j(u)$, and the label of v in that tree is extracted from $label_j(v)$.

The initial routing decision takes $O(\log_{1+\epsilon}(nW)$ time. Each subsequent decision takes only constant time, as the correct double-tree was already identified. An implementation of linear space, constant query time, hash tables is described in Fredman *et al.* [FKS84]. (For deterministic constructions of such hash tables, see Alon and Naor [AN96] and Pagh [Pag00].) We thus have:

THEOREM 3.2. *The roundtrip routing scheme described above has stretch $4k+\epsilon$. It uses local routing tables of size $\tilde{O}(\frac{k}{\epsilon}n^{1/k}\log(nW))$ and $o(\frac{k}{\epsilon}\log^2 n\log(nW))$-bit labels. The headers attached to the messages are $o(\log^2 n)$-bit long. The initial routing decision takes $O(\frac{k}{\epsilon}\log(nW))$ time. Each subsequent routing decision takes only constant time.*

Proof. Let $u,v\in V$ and suppose that $\delta_G(u\rightleftarrows v)=R/2$, where $(1+\epsilon')^{j-1}\le R\le(1+\epsilon')^j$. Consider the ball B, centered at $cent_j(v)$ that contains v in its extended core. By Lemma 3.1, this ball also contains u. The radius of this ball is at most $k(1+\epsilon')^j$. The double-tree corresponding to this ball contains a closed tour containing u and v of total length at most $2k(1+\epsilon')^j$. Messages from u to v, and back, would be routed along this tour, giving a stretch of at most $(1+\epsilon')4k=4k+\epsilon$.

4 A roundtrip routing scheme with stretch 3

In this section we describe an essentially optimal roundtrip routing scheme of stretch 3. It uses local routing tables of size $\tilde{O}(n^{1/2})$ and $O(\log n)$-bit labels. Each routing decision takes constant time. This routing scheme combines ideas from Cowen [Cow01], Cowen and Wagner [CW99, CW00], and Thorup and Zwick [TZ01b]. We start with the following definitions:

DEFINITION 4.1. (ROUNDTRIP ORDERING) *Let $G=(V,E)$ be a weighted directed graph and let $v\in V$. We assume, without loss of generality, that $V=\{1,2,\ldots,n\}$. We say that $u_1\prec_v u_2$ if and only if one of the following conditions holds:*

1. $\delta(u_1\rightleftarrows v)<\delta(u_2\rightleftarrows v)$,
2. $\delta(u_1\rightleftarrows v)=\delta(u_2\rightleftarrows v)$ *but* $\delta(u_1\to v)<\delta(u_2\to v)$,
3. $\delta(u_1\rightleftarrows v)=\delta(u_2\rightleftarrows v)$ *and* $\delta(u_1\to v)=\delta(u_2\to v)$ *but* $u_1<u_2$.

We say that $u_1 \preceq u_2$ if and only if $u_1 \prec u_2$ or $u_1 = u_2$. It is easy to verify that $\prec_v$ and $\preceq_v$ are transitive relations.

DEFINITION 4.2. (CENTERS AND CLUSTERS) *Let $G = (V, E)$ be a weighted directed graph. Let $A \subseteq V$ be a set of* centers. *For every $v \in V$, we let $cent_A(v)$ be the element of A that satisfies $cent_A(v) \preceq_v w$, for every $w \in A$. The cluster $C_A(u)$ of a vertex $u \in V$ is defined as follows: $C_A(u) = \{v \in V \mid u \prec_v cent_A(v)\}$. Finally, we let $\delta(A \rightleftarrows v) = \delta(cent_A(v) \rightleftarrows v)$.*

With these careful definitions, we now have:

LEMMA 4.1. *If $v \in C_A(u)$ and w is on a shortest path from u to v, then $v \in C_A(w)$.*

Proof. If $w = u$, then the claim is obvious. Otherwise, as w is on a shortest path from u to v, we have $\delta(w \rightleftarrows v) \leq \delta(u \rightleftarrows v)$ and $\delta(w \to v) < \delta(u \to v)$. (Note that we assume that all edge weights are positive.) Thus, $w \prec_v u$. As $v \in C_A(u)$, we get that $u \prec_v cent_A(v)$. By the transitivity of $\prec_v$, it follows that $w \prec_v cent_A(v)$, and thus $v \in C_A(w)$, as required.

Using a simple adaptation of an iterative sampling technique of Thorup and Zwick [TZ01b] to the directed case, we obtain:

THEOREM 4.1. *Let $G = (V, E)$ be a weighted directed graph with $|v| = n$ and $|E| = m$. It is possible, in $O(mn)$ time, to find a set $A \subseteq V$ of centers such that $|A| = O((n \log n)^{1/2})$ and $|C_A(w)| = O((n \log n)^{1/2})$, for every $w \in V$.*

The stretch 3 roundtrip routing schemes starts by choosing a set of centers A that satisfies the conditions of Theorem 4.1. Each vertex $v \in V$ is then assigned the following label:

$$label(v) \;=\; (v, cent_A(v), port(cent_A(v), v)) \;,$$

where for every $u, v \in V$, we let $port(u, v)$ be the port number corresponding to the first edge on the shortest path from u to v in G.

The routing table $RT(u)$ at a vertex $u \in V$ is a (2-level) hash table that holds for each $v \in A \cup C_A(u)$ the pair $(v, port(u, v))$. The size of this table is $O((n \log n)^{1/2})$ and for each $v \in A \cup C_A(u)$, the pair $(v, port(u, v))$ can be located in *worst-case* constant time. As before, we use the data structure of Fredman *et al.* [FKS84]. (See also Alon and Naor [AN96] and Pagh [Pag00].)

When a message destined for v reaches u, it is routed as follows:

1. If $u = v$, the message reached its destination.
2. Otherwise, if $v \in A \cup C_A(u)$, route the message using the edge with port number $port(u, v)$. (This port number is found, in constant time, by accessing $RT(u)$.
3. Otherwise, if $u = cent_A(v)$, route along the edge with port number $port(cent_A(v), v)$. (The center $cent_A(v)$ and the port number $port(cent_A(v), v)$ are taken from $label(v)$.)
4. Otherwise, route along the edge with port number $port(u, cent_A(v))$. (As $cent_A(v) \in A$, this port number can be obtained, in constant time, by accessing $RT(u)$.)

We now claim:

THEOREM 4.2. *The routing algorithm described above routes a package from u to v on a path whose length is most $\delta(u \rightleftarrows v) + \delta(u \to v)$. The stretch of the roundtrip route from u to v and back is, therefore, at most 3.*

The proof is very similar to proofs given in Cowen [Cow01] and Thorup and Zwick [TZ01b] for the undirected case, and is omitted from this extended abstract. We just note that if the route from u to v is of length at most $\delta(u \rightleftarrows v) + \delta(u \to v)$, and the route from v to u is of length at most $\delta(u \rightleftarrows v) + \delta(v \to u)$, then the roundtrip route is indeed of length at most $3\delta(u \rightleftarrows v)$.

5 Concluding remarks

We have made explicit the notion of *roundtrip-spanners* and obtained greatly improved algorithms for constructing them. In particular, we showed that for any fixed integer $k \geq 1$ and any fixed $\epsilon > 0$, any directed graph on n vertices with edge weights in the range $[1, W]$ has a $(2k+\epsilon)$-roundtrip-spanner with $O(\frac{k^2}{\epsilon} n^{1+1/k} \log(nW))$ edges, and that such a spanner can be constructed in $O(mn)$ time. The dependence on W here can be eliminated. Details would appear in the full version of the paper. Is it possible to reduce the stretch in this construction to $2k-1$, which is believed to be optimal even for undirected graphs?

We have also obtained greatly improved roundtrip routing schemes. To obtain logarithmic size labels, we had to double the stretch of the schemes from $2k+\epsilon$ to $4k+\epsilon$. Is there a way of avoiding this doubling, perhaps using a handshaking process?

Finally, Cohen and Zwick [CZ01] describe an $\tilde{O}(n^2)$ time algorithm for computing stretch 3 distances between all pairs of vertices of an n-vertex *undirected* graph. Is it possible to obtain an analog result for roundtrip distances in *directed* graphs?

References

[ABCP99] B. Awerbuch, B. Berger, L. Cowen, and D. Peleg. Near-linear time construction of sparse neighborhood covers. *SIAM Journal on Computing*, 28:263–277, 1999. A preliminary version appears at the proceedings of FOCS'93.

[ADD+93] I. Althöfer, G. Das, D. Dobkin, D. Joseph, and J. Soares. On sparse spanners of weighted graphs. *Discrete & Computational Geometry*, 9:81–100, 1993.

[AN96] N. Alon and M. Naor. Derandomization, witnesses for Boolean matrix multiplication and construction of perfect hash functions. *Algorithmica*, 16:434–449, 1996.

[Coh99] E. Cohen. Fast algorithms for constructing t-spanners and paths with stretch t. *SIAM Journal on Computing*, 28:210–236, 1999. A preliminary version appeares at the proceedings of FOCS'93.

[Cow01] L.J. Cowen. Compact routing with minimum stretch. *Journal of Algorithms*, 38:170–183, 2001.

[CW99] L.J. Cowen and C. Wagner. Compact roundtrip routing in digraphs. In *Proceedings of the 10th Annual ACM-SIAM Symposium on Discrete Algorithms, Baltimore, Maryland*, pages 885–886, 1999.

[CW00] L.J. Cowen and C. Wagner. Compact roundtrip routing in directed graphs. In *Proceedings of the 19th Annual ACM Symposium on Principles of Distributed Computing, Portland, Oregon*, pages 51–59, 2000.

[CZ01] E. Cohen and U. Zwick. All-pairs small-stretch paths. *Journal of Algorithms*, 38:335–353, 2001.

[FG01] P. Fraigniaud and C. Gavoille. Routing in trees. In *Proceedings of the 28th International Colloquium on Automata, Languages and Programming, Crete, Greece*, pages 757–772, 2001.

[FKS84] M.L. Fredman, J. Komlós, and E. Szemerédi. Storing a sparse table with $O(1)$ worst case access time. *Journal of the ACM*, 31:538–544, 1984.

[Pag00] R. Pagh. Faster deterministic dictionaries. In *Proceedings of the 11th Annual ACM-SIAM Symposium on Discrete Algorithms, San Francisco, California*, pages 487–493, 2000.

[Pel00] D. Peleg. *Distributed computing – A locality-sensitive approach.* Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 2000.

[TZ01a] M. Thorup and U. Zwick. Approximate distance oracles. In *Proceedings of the 33th Annual ACM Symposium on Theory of Computing, Crete, Greece*, pages 183–192, 2001.

[TZ01b] M. Thorup and U. Zwick. Compact routing schemes. In *Porceedings of the 13th Annual ACM Symposium on Parallel Algorithms and Architectures, Crete, Greece*, pages 1–10, 2001.