Routing in Well-Separated Pair Decomposition Spanners

Fatemeh Baharifard* Majid Farhadi† Hamid Zarrabi-Zadeh†

Abstract

In this paper, we present a local routing scheme for the well-separated pair decomposition (WSPD) spanners. Given a point set P in the plane, a WSPD spanner is a geometric graph whose vertex set is P, and for each pair (A, B) in the well-separated pair decomposition of P, an edge is added to the graph from an arbitrary point $a \in A$ to an arbitrary point $b \in B$. It is well-known that such a graph is a $(1+\varepsilon)$-spanner of P, where $\varepsilon > 0$ is an input parameter used for constructing the well-separated pair decomposition. Our routing scheme assigns to each point $p \in P$ a routing table of size $O(\frac{1}{\varepsilon^2}\log\alpha)$, where α is the ratio of the furthest distance to the closest distance in P. It can then locally route a message from any arbitrary point p to any point q in P along a path whose length is at most $1+\varepsilon$ times the Euclidean distance between the pair of points. The WSPD construction considered in this paper is based on compressed quadtrees. To the best of our knowledge, this is the first time that a local routing scheme with an optimal competitive routing ratio is considered for this famous class of WSPD spanners.

1 Introduction

A geometric graph G is a *t-spanner* for a point set P, if for each pair of points p and q in P, there is a path in G between p and q, whose length is at most t times the Euclidean distance between p and q. The minimum t such that G is a t-spanner of P is called the *spanning ratio* of G.

One of the most important problems in communication networks is to send/route a message from a source point to any other target point in such a way that the total distance traveled by the message is at most a constant times the shortest path or Euclidean distance between the two points. Network routing strategies such as Dijkstra's algorithm [10] require knowledge of the whole network topology in order to compute a short route. In many settings, this assumption is impractical, and the routing algorithm is supposed to work without knowing the full structure of the graph. Therefore, a *local routing* strategy is usually preferred, meaning that the algorithm can route the message to the target using only information stored in the message itself and in the current node [15]. If the information stored in the current node is of size k, we say that the local routing algorithm has a *routing table* of size k. Moreover, a local routing algorithm $\mathcal{A}$ is called *μ-memory* if it uses a memory of size μ stored with the message [4]. The algorithm $\mathcal{A}$ is *c-competitive* if the total distance traveled by the message is not more than c times the Euclidean distance between source and destination. The minimum c such that a routing algorithm $\mathcal{A}$ is c-competitive is called the *routing ratio*.

Related Work. Recently, a stream of research has explored local routing algorithms for some geometric spanners such as Delaunay triangulations and θ-graphs (for definitions, see [5, 9]). Chew [9] was the first to describe a local routing algorithm on the L_1-Delaunay triangulation with a routing ratio of $\sqrt{10}$, using only the information of the target point, the current point, and all neighborhood of the current point. Subsequently, local routing algorithms using the same set of information were presented for TD-Delaunay triangulation by a spanning ratio of $5/\sqrt{3}$ [3], and for the standard Delaunay triangulation by a spanning ratio of 5.90 [1]. In [5], a *θ-routing* algorithm is described which has a constant routing ratio on all θ_k-graphs with $k \geq 7$. Moreover, a deterministic local routing scheme with a routing ratio of 2 is presented for θ_6-graph in [3].

Very recently, Bose *et al.* [2] considered a specific type of WSPD spanners, and presented two near-optimal local routing schemes for this type of spanners. In their settings, the WSPD construction is based on fair split trees, and the WSPD spanner is constructed by selecting a well-chosen edge from each partition of WSPD (the rightmost point in each set) as its representative, rather than picking an arbitrary edge. They showed that their WSPD spanner has an improved spanning ratio of $1+4/s+4/(s-2)$ compared to the original one, which was $1+8/(s-4)$, where $s > 0$ is the separation factor. They presented a 2-local and a 1-local routing algorithm with routing ratios of $1+4/s+6/(s-2)$ and $1+6/(s-2)+6/s+4/(s^2-2s)+8/s^2$, respectively (a routing algorithm on a graph G is called *k-local*, if each vertex v of G stores information about vertices that are at hop distance at most k from v). Their routing scheme did not use a header but required routing tables of total

*School of Computer Science, Institute for Research in Fundamental Sciences (IPM), f.baharifard@ipm.ir

†Department of Computer Engineering, Sharif University of Technology, m_farhadi@ce.sharif.edu, zarrabi@sharif.edu

size $O(s^2 nB)$ bits, where B is the maximum number of bits to store a bounding box.

Competitive local routing algorithms with additional memory have been recently considered for unit disk graphs as popular wireless ad-hoc networks [15, 17]. The unit disk graph connects any two nodes which are within unit distance to each other. Yan *et al.* [17] presented a routing algorithm with low hop (edge) delay, by assigning a label of size $O(\log^2 n)$ to each node, where n is the number of nodes. Subsequently, Kaplan *et al.* [15] discovered a $(1+\varepsilon)$-competitive routing algorithm for unit disk graphs, using a modifiable header (memory) of size $O(\log n \log \Delta)$, where Δ is the diameter of the points, as well as additional polylog bits for each point. Their method is based on the well-separated pair decomposition for unit disk graphs [12].

Our Contribution. In this paper, we focus on an important and well-known class of WSPD spanners whose underlying WSPD is constructed using compressed quadtree. This construction of WSPD is widely used in the literature [8, 11, 14, 16], as it avoids the complexity of fair split trees originally used by Callahan and Kosaraju [7]. We present a competitive $O(\log \alpha)$-memory routing algorithm to route on these WSPD spanners, where α is the ratio of the farthest distance to the closest distance in the input point set. We indeed consider a standard WSPD spanner which is constructed by choosing an arbitrary edge from each pair of the WSPD, and unlike the method used in [2], we do not pose any restriction on choosing the representatives of the pairs when constructing the WSPD spanner. Assuming that we can store a static information (routing table) of size $O(\frac{1}{\varepsilon^2} \log \alpha)$ at each node of the spanner, the proposed algorithm is a $(1+\varepsilon)$-competitive local routing algorithm, which is optimal.

2 Preliminaries

In this section, we briefly describe the notions used throughout the paper.

Well-Separated Pair Decomposition. Let P be a set of n points in the plane, and $s > 0$ be a real number. Two point sets $A, B \subseteq P$ are *well-separated* with respect to a separation factor s, if there are two disjoint disks D_A and D_B with the same radius r, enclosing A and B respectively, such that the distance between D_A and D_B is at least $s \cdot r$. Here, the distance of two subsets A and B is defined as $d(A,B) = \min\{\|a-b\| \mid a \in A, b \in B\}$ where $\|a-b\|$ denotes the Euclidean distance of the points a and b (see Figure 1).

Following the definition in [7], a *well-separated pair decomposition (WSPD)* for P with respect to s is a collection $\mathcal{W} = \{(A_1,B_1),\ldots,(A_m,B_m)\}$ of pairs of non-

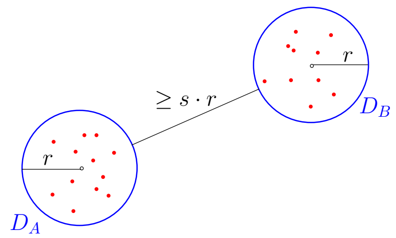

Figure 1: A well-separated pair with separation s

empty subsets of P such that each pair (A_i, B_i) for $1 \leq i \leq m$ is a well-separated pair with respect to s, and for any pair of points $p, q \in P$, there is a unique pair (A_i, B_i) in the collection, such that either $p \in A_i$ and $q \in B_i$, or $q \in A_i$ and $p \in B_i$. The number of well-separated pairs, m, is called the size of the WSPD.

WSPD Construction. A *quadtree* of P is a tree data structure $\mathcal{T}$ in which each internal node has four children, and the points of P are stored in the leaves. The root of $\mathcal{T}$ corresponds to a square bounding box of P, and each internal node $v \in \mathcal{T}$ corresponds to a cell $c(v)$ which is a square formed by splitting the parent cell into four equal-size squares by a horizontal and a vertical cut. A *compressed quadtree* is a quadtree in which any sequence of nodes with degree one are replaced by a single node. A compressed quadtree of a set of n points can be constructed in $O(n \log n)$ time [13].

Given a compressed quadtree $\mathcal{T}$ of P, one can use the following greedy algorithm to build a WSPD of P. The algorithm starts by considering any combination of two children of the root as a pair. If the current pair is not well separated, then the bigger node of the pair is replaced by its children, and the process continues until we reach a well-separated pair decomposition. For a separation factor $s > 0$, this algorithm yields a WSPD of size $O(s^2 n)$ in $O(n \log n + s^2 n)$ time [13].

WSPD Spanners. Callahan and Kosaraju [6] showed how a $(1+\varepsilon)$-spanner can be obtained from a WSPD. They first constructed a WSPD of P with separation factor $s = 4 + 8/\varepsilon$. They then chose an arbitrary point $a_i \in A_i$ and an arbitrary point $b_i \in B_i$ as the *representatives* of A_i and B_i, respectively, and showed that the resulting graph $G = (P, E)$ with $E = \{(a_i, b_i) \mid 1 \leq i \leq m\}$ is a $(1+\varepsilon)$-spanner. We refer to the resulting graph G as a *WSPD spanner* of P throughout the paper. Based on the construction described above, the WSPD spanner has size $O(n/\varepsilon^2)$ and can be computed in $O(n \log n + n/\varepsilon^2)$ time.

3 Routing in WSPD Spanners

Let P be a set of n points in the plane, and let α be the *spread* of P, namely the ratio of the farthest distance to the closest distance in P. In this section, we propose an algorithm to route a message through the WSPD spanner of P, utilizing a small additional memory (stack) along with the message, and a static data (routing tables) in the nodes of the graph.

We first prove an upper bound on the number of WSPD pairs that contain a fixed point. The following packing lemma is an ingredient of our proof.

Lemma 1 (Packing Lemma [7]) *Let D be a disk of radius r in the plane. The number of disjoint quadtree cells of side length at least ℓ overlapping D is at most $(1 + \lceil 2r/\ell \rceil)^2 = O(\max\{2, r/\ell\}^2)$.*

Lemma 2 *For each point $p \in P$, the number of WSPD pairs containing p is upper-bounded by $O(\frac{1}{\varepsilon^2} \log \alpha)$.*

Proof. Let $\mathcal{W}$ be a WSPD of P with separation s as described in Section 2. Let (A, B) be a well-separated pair in $\mathcal{W}$ containing p, and let x and y be the smallest quadtree cells of same length ℓ, enclosing A and B, respectively. Suppose that x and y are in level i of the quadtree. By the construction of WSPD, we know that if $(A, B) \in \mathcal{W}$, then $(P(A), P(B))$ is not in $\mathcal{W}$, where $P(A)$ and $P(B)$ denote the parents of A and B in quadtree, respectively. Then, $d(A, B) \le (s + 2)\sqrt{2}\ell$, because otherwise, the distance between parent cells of $c(x)$ and $c(y)$ with side length at least 2ℓ is more than $s\sqrt{2}\ell$, and hence they are well-separated, which is a contradiction (see Figure 2). Therefore, by packing lemma, at most $O(((s + 2)\sqrt{2})^2) = O(s^2)$ pairs in level i can contain p. Since $s = 4 + 8/\varepsilon$, and there are at most $\log \alpha$ levels in the quadtree, the total number of pairs containing p is $O(\frac{1}{\varepsilon^2} \log \alpha)$. □

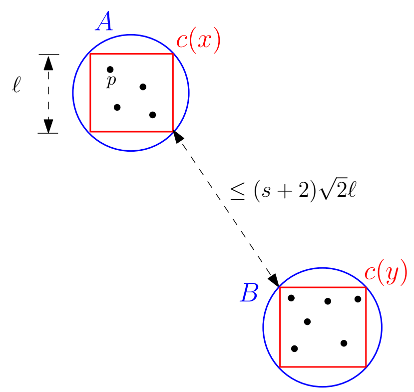

Figure 2: A well-separated pair with bounded distance.

Routing Algorithm. We are now ready to describe our routing algorithm. Let $f_p(q)$ denote a function that searches for a pair (A_i, B_i) in the WSPD such that $(p, q) \in A_i \times B_i$ or $(p, q) \in B_i \times A_i$, and returns their corresponding representatives (a_i, b_i) or (b_i, a_i) in the WSPD spanner. This function must be computable at node p. Therefore, at each node p, we store a list (table) of pairs (A_i, B_i) such that p is a member of either A_i or B_i, and for each such pair (A_i, B_i), we store in the table the boundaries of A_i and B_i (to check membership of an arbitrary point in the set), as well as the representatives of A_i and B_i in the WSPD spanner. Note that Lemma 2 bounds the size of the table stored at each node to $O(\frac{1}{\varepsilon^2} \log \alpha)$. The function is now simply computable at p by trying all pairs including p and checking membership of q in the other side of the pair, using boundaries of the squares corresponding to the sets.

Routing can be performed by simulating the following recursive algorithm, using a stack stored and transmitted along with the message. The inputs are source and destination points, (p, q), and we are at p at the beginning of the algorithm (see Figure 3).

Algorithm 1 ROUTE (p, q)

```
(a, b) ← f_p(q)
ROUTE(p, a)
traverse along edge (a, b)
ROUTE(b, q)
```

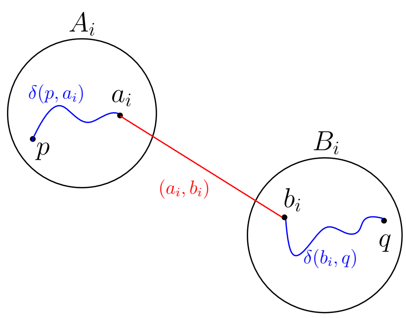

Figure 3: Routing a message from p to q.

Lemma 3 *For any pair of points $p, q \in P$, the path traversed by Algorithm 1 from p to q is at most $1 + \varepsilon$ times the Euclidean distance between p and q.*

Proof. We prove by induction on the Euclidean distance of the points. Fix a pair $p, q \in P$. Suppose by induction that for any pair $x, y \in P$ with $d(x, y) \le d(p, q)$, the traversed path $\delta(x, y)$ has length at most $(1 + \varepsilon)d(x, y)$, where $d(x, y)$ denotes the Euclidean distance between x and y. By construction of the WSPD spanner, there is a pair (A_i, B_i) such that $p \in A_i$ and $q \in B_i$. Therefore, we have:

$$
\begin{aligned}
\delta(p,q) &\leq \delta(p,a_i) + d(a_i,b_i) + \delta(b_i,q) \\
&\leq (1+\varepsilon)d(p,a_i) + [d(p,q)+4r] + (1+\varepsilon)d(b_i,q) \\
&\leq (1+\varepsilon)4r + [d(p,q)+4r] \\
&\leq d(p,q) + (1/s)(8+4\varepsilon) \\
&\leq (1+\varepsilon)d(p,q)
\end{aligned}
$$

□

Lemma 4 *The maximum depth of recursion in Algorithm 1, and thus the maximum size of the stack sent along with the message, is $O(\log \alpha)$.*

Proof. This is easy to see by noting that elements stored in the stack, corresponding to the recursion history for the current call, are monotonically deepening in the quadtree. □

Putting all these together, we get our main theorem.

Theorem 5 *Let P be a set of n points in the plane with spread α, and let S be a WSPD spanner of P with spanning ratio $1+\varepsilon$. We can locally route a message between any two nodes of S using a memory of size $O(\log \alpha)$ stored with the message, and a routing table of size $O(\frac{1}{\varepsilon^2} \log \alpha)$ stored at each node, such that the path traversed between the two nodes has length at most $1+\varepsilon$ times their Euclidean distance.*

4 Conclusion

In this paper, we considered the WSPD spanners based on compressed quadtrees, and proposed an efficient local routing algorithm on these spanners, using a memory of size $O(\log \alpha)$ stored with the message, and a routing table of size $O(\frac{1}{\varepsilon^2} \log \alpha)$ stored in the nodes of the spanner, where α is the spread of the underlying points. The path traveled between any two points by the algorithm is guaranteed to be no longer than $1+\varepsilon$ times the Euclidean distance between the two points. Although we presented our routing algorithm in the plane, the algorithm can be easily extended to any fixed dimension d, at the expense of increasing the routing table size to $O(\frac{1}{\varepsilon^d} \log \alpha)$. It is interesting to see if the size of routing table and/or memory can be improved.

References

[1] N. Bonichon, P. Bose, J. D. Carufel, L. Perković, and A. V. Renssen. Upper and lower bounds for competitive online routing on Delaunay triangulations. *arXiv:1501.01783*, 2015.

[2] P. Bose, J.-L. De Carufel, V. Dujmović, and F. Paradis. Local routing in spanners based on WSPDs. In *Proc. 15th Workshop Algorithms Data Struct.*, pages 205–216, 2017.

[3] P. Bose, R. Fagerberg, A. van Renssen, and S. Verdonschot. Competitive routing in the half-θ_6-graphs. In *Proc. 23rd ACM-SIAM Sympos. Discrete Algorithms*, pages 1319–1328, 2012.

[4] P. Bose, R. Fagerberg, A. van Renssen, and S. Verdonschot. Competitive local routing with constraints. In *Proc. 26th Annu. Internat. Sympos. Algorithms Comput.*, pages 23–34, 2015.

[5] P. Bose, A. van Renssen, and S. Verdonschot. On the spanning ratio of Theta-graphs. In *Proc. 13th Workshop Algorithms Data Struct.*, pages 182–194, 2013.

[6] P. B. Callahan and S. R. Kosaraju. Faster algorithms for some geometric graph problems in higher dimensions. In *Proc. 4th ACM-SIAM Sympos. Discrete Algorithms*, pages 291–300, 1993.

[7] P. B. Callahan and S. R. Kosaraju. A decomposition of multidimensional point sets with applications to k-nearest-neighbors and n-body potential fields. *J. ACM*, 42(1):67–90, 1995.

[8] T. M. Chan. Well-separated pair decomposition in linear time? *Inform. Process. Lett.*, 107(5):138–141, 2008.

[9] L. P. Chew. There is a planar graph almost as good as the complete graph. In *Proc. 2nd Annu. ACM Sympos. Comput. Geom.*, pages 169–177, 1986.

[10] E. W. Dijkstra. A note on two problems in connexion with graphs. *Numerische mathematik*, 1(1):269–271, 1959.

[11] J. Fischer and S. Har-Peled. Dynamic well-separated pair decomposition made easy. In *Proc. 17th Canad. Conf. Computat. Geom.*, volume 5, pages 235–238, 2005.

[12] J. Gao and L. Zhang. Well-separated pair decomposition for the unit-disk graph metric and its applications. *SIAM J. Comput.*, 35(1):151–169, 2005.

[13] S. Har-Peled. *Geometric approximation algorithms.* American Mathematical Society, 2011.

[14] S. Har-Peled and M. Mendel. Fast construction of nets in low-dimensional metrics and their applications. *SIAM J. Comput.*, 35(5):1148–1184, 2006.

[15] H. Kaplan, W. Mulzer, L. Roditty, and P. Seiferth. Routing in unit disk graphs. In *Proc. 12th Latin American Symp. Theoret. Informatics*, pages 536–548, 2016.

[16] E. Park and D. M. Mount. Output-sensitive well-separated pair decompositions for dynamic point sets. In *Proc. 21st*, pages 354–363, 2013.

[17] C. Yan, Y. Xiang, and F. F. Dragan. Compact and low delay routing labeling scheme for unit disk graphs. *Comput. Geom.*, 45(7):305–325, 2012.