Geometric Spanners With Small Chromatic Number*

Prosenjit Bose[1] Paz Carmi[1] Mathieu Couture[1]
Anil Maheshwari[1] Michiel Smid[1]
Norbert Zeh[2]

[1]School of Computer Science, Carleton University

[2]Faculty of Computer Science, Dalhousie University

June 22, 2007

Abstract

Given an integer $k \geq 2$, we consider the problem of computing the smallest real number $t(k)$ such that for each set P of points in the plane, a $t(k)$-spanner for P exists that has $O(|P|)$ edges and whose chromatic number is at most k. We prove that $t(2) = 3$, $t(3) = 2$, $t(4) = \sqrt{2}$, and give upper and lower bounds on $t(k)$ for $k > 4$. We also consider an on-line variant of the problem, in which the points of P are given one after another, and the color of a point must be decided at the moment the point is given.

1 Introduction

Let P be a set of n points in the plane. A *geometric graph* with vertex set P is an undirected graph whose edges are line segments that are weighted by their Euclidean length. For a real number $t \geq 1$, such a graph G is called a *t-spanner* if the weight of the shortest path in G between any two vertices p and q does not exceed $t|pq|$, where $|pq|$ is the Euclidean distance between p and q. The smallest t having this property is called the *stretch factor* of the graph G. Thus, a graph with stretch factor t approximates the $\binom{n}{2}$ distances between the points in P within a factor of t. The problem of constructing t-spanners with $O(n)$ edges for any given point set has been studied intensively; see the book by Narasimhan and Smid [6] for an overview.

In this paper, we consider the problem of computing t-spanners whose chromatic number is at most k, for some given value of k. The goal is to minimize the value of t over all finite

*Research partially supported by HPCVL, NSERC, MRI, CFI, and MITACS.

sets P of points in the plane. We call a spanner whose chromatic number is at most k a *k-chromatic spanner*.

Problem 1.1 *Given an integer $k \geq 2$, let $t(k)$ be the infimum of all real numbers t with the property that for every finite set P of points in the plane, a k-chromatic t-spanner for P exists. Determine the value of $t(k)$.*

Observe that in the definition of $t(k)$, there is no requirement on the number of edges of the chromatic spanner. This is not a restriction, because, as shown by Gudmundsson *et al.* [4], any t-spanner for P contains a subgraph with $O(n)$ edges which is a $((1+\epsilon)t)$-spanner for P.

We show that the minimum spanning tree of P can be used to obtain a 2-chromatic 3-spanner, thus showing that $t(2) \leq 3$. We also give an example of a point set P such that any 2-chromatic graph with vertex set P has stretch factor at least three. Thus, we have $t(2) = 3$.

Determining the value of $t(3)$ requires more effort. We show that for any point set P, a planar graph with vertex set P that is triangle-free can be computed. It is known that such graphs have chromatic number at most three. We show that this leads to a 3-chromatic 2-spanner, thereby proving that $t(3) \leq 2$. We also show, by means of an example, that $t(3) \geq 2$. Thus, we obtain that $t(3) = 2$.

Consider the case when $k = 4$. Any plane spanner, such as the Delaunay triangulation [5], has chromatic number at most four. The stretch factor of a plane spanner, however, may not be tight. We show that, nevertheless, the Delaunay triangulation can be used to obtain a 4-chromatic $\sqrt{2}$-spanner; thus, $t(4) \leq \sqrt{2}$. Again by means of an example, we also show that $t(4) \geq \sqrt{2}$. Therefore, we have $t(4) = \sqrt{2}$.

For $k > 4$, we are not able to obtain the exact value of $t(k)$. Inspired by the *ordered Θ-graph* of Bose *et al.* [1], we show that $t(k) \leq 1 + 2\sin\frac{\pi}{2(k-1)}$. We also show that the vertex set of the regular $(k+1)$-gon gives $t(k) \geq 1/\cos\frac{\pi}{k+1}$.

In the second part of the paper, we consider an on-line variant of the problem, in which the points of P are given one after another, and the color of a point must be decided at the moment when it is given; thus, later on, the color of a point cannot be changed.

Problem 1.2 *Given an integer $k \geq 2$, let $t'(k)$ be the infimum of all real numbers t with the property that for every finite set P of points in the plane, which is given on-line, a k-chromatic t-spanner for P exists. Determine the value of $t'(k)$.*

A simple variant of the ordered Θ-graph shows that $t'(k) \leq 1 + 2\sin(\pi/k)$. Thus, we have $t'(2) \leq 3$, $t'(3) \leq 1 + \sqrt{3}$ and $t'(4) \leq 1 + \sqrt{2}$. Since $t'(2) \geq t(2) = 3$, it follows that $t'(2) = 3$. We also give examples showing that $t'(3) \geq 1 + \sqrt{3}$ and $t'(4) \geq 1 + \sqrt{2}$. We finally show that, for $k \geq 5$, $t'(k) \geq 1/\cos\frac{\pi}{k}$.

The rest of this paper is organized as follows: in Section 2, we define the t-ellipse property and show its relationship to our problem. In Section 3, we give upper and lower bounds for the off-line problem (Problem 1.1). In Section 4, we give give upper and lower bounds for

number of colors	$t(k)$ (off-line)		$t'(k)$ (on-line)	
k	lower bound	upper bound	lower bound	upper bound
2	3	3	3	3
3	2	2	$1+\sqrt{3}$	$1+\sqrt{3}$
4	$\sqrt{2}$	$\sqrt{2}$	$1+\sqrt{2}$	$1+\sqrt{2}$
k	$1/\cos\frac{\pi}{k+1}$	$1+2\sin\frac{\pi}{2(k-1)}$	$1/\cos\frac{\pi}{k}$	$1+2\sin\frac{\pi}{k}$

Table 1: Summary of our results.

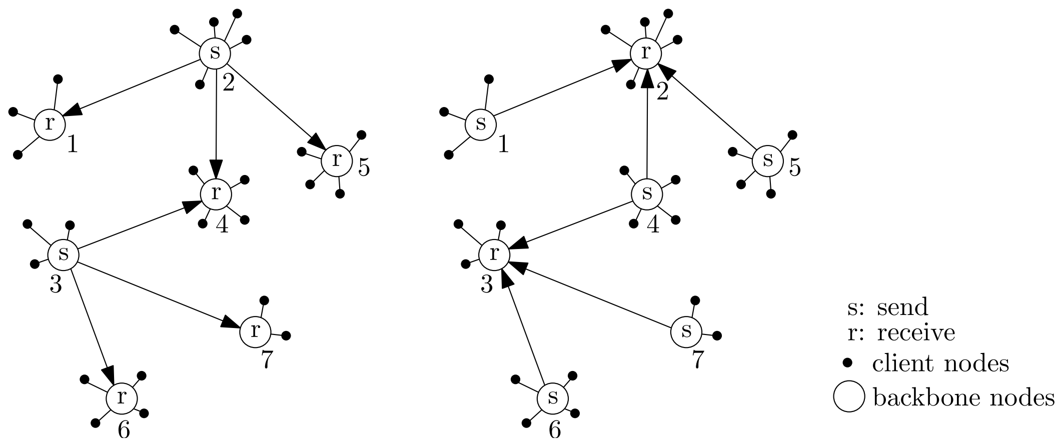


Figure 1: The two possible states of the backbone nodes.

the on-line problem (Problem 1.2). In Section 5, we present simulation results. We conclude in Section 6. In Table 1, we summarize our results. We now motivate our work.

1.1 Motivation

In a recent paper, Raman and Chebrolu [7] proposed a new protocol, called 2P, allowing to address rural Internet connectivity in a low-cost manner using off-the-shelf 802.11 hardware. Since their infrastructure uses several directional antennae at one node rather than one single omnidirectional antenna, simultaneous communications are possible at one node. However, due to restrictions inherent to the 802.11 standard, backbone nodes have to communicate with each other using a single channel. While simultaneous transmissions and simultaneous receptions are possible, it is not physically possible for one node to both transmit and receive at the same time. Therefore, backbone nodes have to alternate between the send and receive states (see Figure 1). This forces the backbone to be a bipartite graph, i.e., to have chromatic number equal to two.

The backbone creation algorithm of Raman and Chebrolu [7] outputs a tree, which is obviously bipartite. However, the tree structure presents the following disadvantage: it is possible that the path that a message has to follow is much longer than the distance (either Euclidean or in terms of hops) between the originating node and its destination. For example, in Figure 1, a message routed from node 1 to node 3 has to go through nodes 2 and 4, whereas a direct link between 1 and 3 could be added while satisfying the bipartition requirement.

Note that the physical constraint preventing nodes to simultaneously receive and transmit can be met even if the graph is not bipartite. In fact, any graph with chromatic number k would meet this requirement: all one has to do is to prevent two nodes that have different colors to transmit simultaneously. A degenerate case is when each node has its own color, in which case at most one node can transmit at any given moment. This case is undesirable, since the amount of time during which a node can transmit decreases as the size of the network increases.

For these reasons, it is desirable to have geometric graphs that have both small chromatic number and small stretch factor.

2 The t-Ellipse Property

In this section, we show that Problem 1.1, i.e., determining the smallest value of t such that a k-chromatic t-spanner exists for any point set P, is equivalent to minimizing the value of t such that any point set can be colored using k colors in a way that satisfies the so-called t-ellipse property.

Definition 2.1 (t-ellipse property) *Let $k \geq 2$ be an integer, let P be a finite set of points in the plane and let $c : P \rightarrow \{1, \ldots, k\}$ be a k-coloring of P. We say that that the coloring c satisfies the* t-ellipse property *if, for each pair of distinct points p and q in P with $c(p) = c(q)$, there exists a point $r \in P$ such that $c(r) \neq c(p)$ and $|pr| + |rq| \leq t|pq|$.*

Thus, if p and q have the same color, then the ellipse $\{x \in \mathbb{R}^2 : |px| + |xq| \leq t|pq|\}$ contains a point r of P whose color is different from that of p and q.

Proposition 2.2 *Let $k \geq 2$, let P be a set of points in the plane, and let G be a k-chromatic t-spanner of P with k-coloring c. Then c satisfies the t-ellipse property.*

Proof: Let $p, q \in P$ such that $c(p) = c(q)$. Since G is a t-spanner, there is a path $p = p_1, \ldots, p_l, p_{l+1} = q$ in G such that $\sum_{i=1}^{l} |p_i p_{i+1}| \leq t|pq|$. Also since G is k-chromatic, we have $l \geq 2$ and $c(p_2) \neq c(p)$. We have

$$t|pq| \geq \sum_{i=1}^{l} |p_i p_{i+1}| = |pp_2| + \sum_{i=2}^{l} |p_i p_{i+1}| \geq |pp_2| + |p_2 q|,$$

where the last inequality follows from the triangle inequality. Thus, if we define $r := p_2$, then $c(r) \neq c(p)$ and $|pr| + |rq| \leq t|pq|$. □

Proposition 2.3 *Let $k \geq 2$, let P be a set of points in the plane, and let $c : P \to \{1, \ldots, k\}$ be a k-coloring of P that satisfies the t-ellipse property. Then, there exists a k-chromatic t-spanner of P.*

Proof: Let $K_c(P)$ be the complete k-partite graph with vertex set P in which there is an edge between two points p and q if and only if $c(p) \neq c(q)$. By definition, $K_c(P)$ is k-colorable. We show that $K_c(P)$ is a t-spanner of P. Let p and q be two distinct points of P such that (p, q) is not an edge in $K_c(P)$. This means that $c(p) = c(q)$. Since c has the t-ellipse property, there exists a point r in P such that $c(r) \neq c(p)$ and $|pr| + |rq| \leq t|pq|$. Since $c(r) \neq c(p)$ (and consequently, $c(r) \neq c(q)$), the edges (p, r) and (r, q) are both in $K_c(P)$. This means that (p, r, q) is a t-spanner path in $K_c(P)$ between p and q. □

From now on, we define the *stretch factor* of a k-coloring of a point set as the stretch factor of the complete k-partite graph that is induced by this coloring. By Propositions 2.2 and 2.3, the problem of determining $t(k)$ is equivalent to determining the minimum stretch factor of any k-coloring of any point set.

We conclude this section by showing why it is sufficient to focus on the coloring problem without worrying about the number of edges in the spanner. The following theorem is due to Gudmundsson *et al.* [4]; its proof is based on the the well-separated pair decomposition of Callahan and Kosaraju [2].

Theorem 2.4 *Let $\epsilon > 0$ and $t \geq 1$ be constants, let P be a set of n points in the plane, and let G be a t-spanner of P. There exists a subgraph G' of G, such that G' is a $((1+\epsilon)t)$-spanner of P and G' has $O(n)$ edges.*

Proposition 2.5 *Let $k \geq 2$, let P be a set of n points in the plane, and let $c : P \to \{1, \ldots, k\}$ be a k-coloring of P that satisfies the t-ellipse property. Then, for any constant $\epsilon > 0$, there exists a k-chromatic $((1 + \epsilon)t)$-spanner of P that has $O(n)$ edges.*

Proof: By Proposition 2.3, there exists a k-chromatic t-spanner G of P. By Theorem 2.4, G contains a subgraph G' with $O(n)$ edges, such that G' is a $((1 + \epsilon)t)$-spanner of P. Since G is k-chromatic, G' is k-chromatic as well. □

3 Upper and lower bounds on $t(k)$

The structure of this section is as follows: For $k = 2, 3$, and 4, we give coloring algorithms whose outputs have bounded stretch factor. Then, we show that these stretch factors are tight by providing point sets for which no coloring algorithm can achieve a better stretch factor. Then we present our upper and lower bounds for $t(k)$, when $k > 4$.

We now give the coloring algorithm for $k = 2$.

Algorithm 1 Off-line 2 Colors

Input: P, a set of points in the plane
Output: c, a 2-coloring of P

Compute a Euclidean minimum spanning tree T of P
$c \leftarrow$ a 2-coloring of T

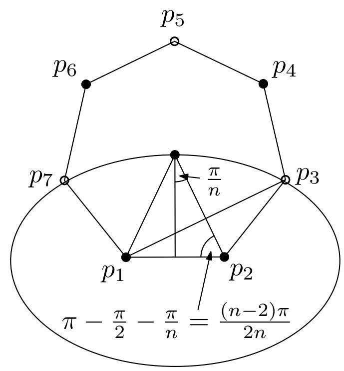


Figure 2: Lower bound of $3 - \epsilon$ for $k = 2$.

Proposition 3.1 *For any point set P, the 2-coloring computed by Algorithm 1 has stretch factor at most 3. Thus, we have $t(2) \leq 3$.*

Proof: It is sufficient to show that the 2-coloring c computed by Algorithm 1 has the 3-ellipse property. Let p and q be two distinct points in P such that $c(p) = c(q)$. Observe that (p, q) is not an edge in the minimum spanning tree T. Let r be the nearest neighbor of p. Since the edge (p, r) is in T, we have $r \neq q$ and $c(r) \neq c(p)$. Since r is nearer to p than q, we have

$$|pr| + |rq| \leq |pr| + |rp| + |pq| = 2|pr| + |pq| \leq 2|pq| + |pq| = 3|pq|.$$

□

Proposition 3.2 *For every $\epsilon > 0$, there exists a point set P such that every 2-coloring of P has stretch factor at least $3 - \epsilon$. Thus, we have $t(2) \geq 3$.*

Proof: Let n be an odd integer, and let $P = \{p_1, \ldots, p_n\}$ be the set of vertices of a regular n-gon given in counter-clockwise order. Let c be an arbitrary 2-coloring of P. By the pigeonhole principle, there are two points in P which are adjacent on the n-gon and that have the same color. We may assume without loss of generality that these two points are p_1 and p_2. Also, we may assume that $|p_1p_2| = 1$ (see Figure 2). Let t be any real number such that c satisfies the t-ellipse property. Then $|p_1p_3| + 1 \leq t$. But $|p_1p_3| = 2\sin((n-2)\pi/2n)$, which tends to 2 as n goes to infinity. □

We now consider the case when $k = 3$. Our strategy is to construct a graph such that any coloring of that graph has the 2-ellipse property. We then show that this graph is 3-colorable.

Algorithm 2 Off-line 3 Colors

Input: P, a set of n points in the plane
Output: c, a 3-coloring of P, and G, a 3-chromatic graph whose vertex set is P
1: Let G be the graph with vertex set P and whose edge set is empty
2: Let $e_1, \ldots, e_{\binom{n}{2}}$ be the pairs of points of P in sorted order of their distances
3: **for** $i = 1$ to $\binom{n}{2}$ **do**
4:     Let $e_i = (p_i, q_i)$
5:     **if** G contains no edge (p, q) where $|p_i p| + |pq_i| \leq 2|p_i q_i|$ and $|p_i q| + |qq_i| \leq 2|p_i q_i|$ **then**
6:         add the edge e_i to G
7:     **end if**
8: **end for**
9: //assertion: G is 3-colorable (see Lemma 3.5)
10: $c \leftarrow$ a 3-coloring of G

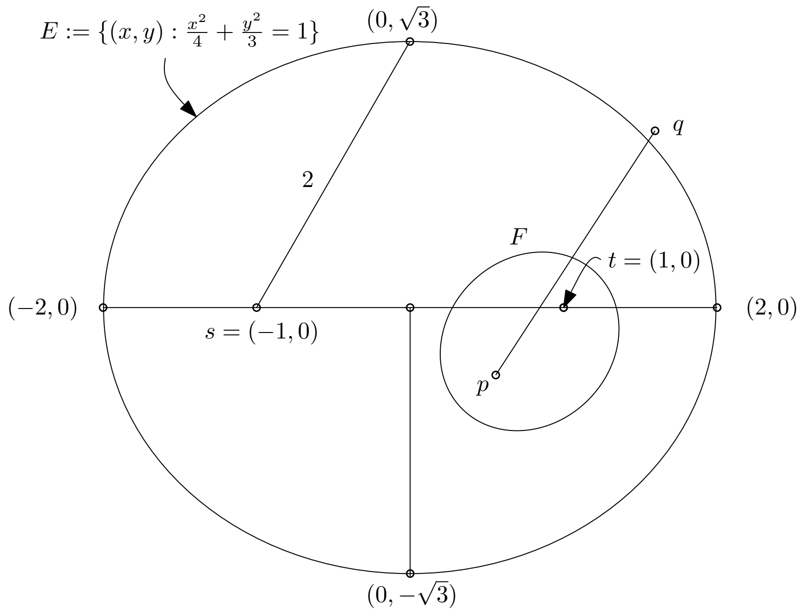
Lemma 3.3 *The graph G computed by Algorithm 2 is triangle-free.*

Proof: Assume that G contains a triangle with vertices p, q, and r. We may assume without loss of generality that (p, r) was the last edge of this triangle that was considered by the algorithm. This means that (p, r) is the longest edge of the triangle. When $e_i = (p, r) = (p_i, q_i)$ in line 4, G already contains the edge (p, q). Since $|p_i p| + |pq_i| = |pp| + |pr| \leq 2|pr|$ and $|p_i q| + |qq_i| = |pq| + |qr| \leq 2|pr|$, the edge (p, r) is not added to G. This is a contradiction and, therefore, G is triangle-free. □

Lemma 3.4 *The graph G computed by Algorithm 2 is plane.*

Proof: Assume that G contains two crossing edges (p, q) and (s, t). We may assume without loss of generality that $s = (-1, 0)$, $t = (1, 0)$, and the pair (s, t) has a larger index than (p, q) after the pairs have been sorted in line 2. Thus, we have $|pq| \leq |st|$. When $e_i = (s, t) = (p_i, q_i)$ in line 4, G already contains the edge (p, q). Let E be the ellipse whose boundary is determined by the set of points e where $|se| + |et| = 2|st|$ (see Figure 3, left). If both p and q are outside E, then $|pq| > |st|$. If both p and q are inside E, then it follows from step 2 of the algorithm that the edge (s, t) is not added to G. Therefore, exactly one point of $\{p, q\}$ is inside E.

Without loss of generality, p is inside E, q is outside E, the pair (p, t) has a smaller index than the pair (p, s) after the pairs have been sorted in line 2, and p is below the x-axis. We will show below that the ellipse F whose boundary is the set of points f such that $|pf| + |ft| = 2|pt|$ is completely contained inside E. Thus, since E does not contain any edge, the same is true for F. It follows that the edge (p, t) is added to G, thus preventing



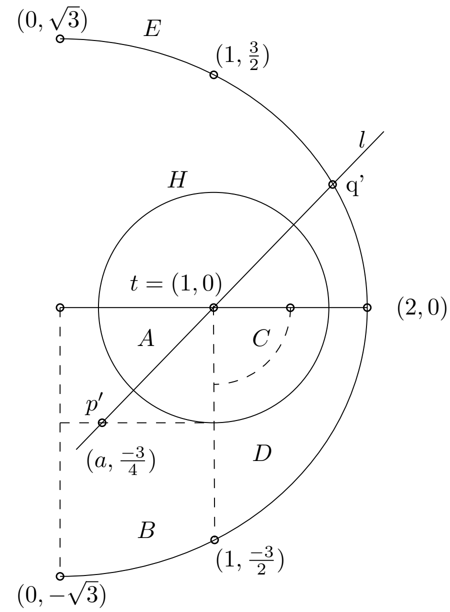


Figure 3: Proof of Lemma 3.4.

the insertion of the edge (s, t) because edge (p, t) is easily seen to have smaller index than (s, t).

It remains to prove that the ellipse F is contained in the ellipse E. The proof considers four cases, depending on the location of p (see Figure 3, right).

Case A: $[0 \leq p_x \leq 1$ **and** $-3/4 \leq p_y \leq 0]$ We show that for any point e on E, we have $|pe| + |et| > 2|pt|$. Note that we only need to check the case when e is below the x-axis and either $p_y = -3/4$ or $p_x = 0$. We consider these two cases separately.
If $p_y = -3/4$, let $a = p_x$. In this case, we have

$$|pe| + |et| - 2|pt| = \sqrt{(a - e_x)^2 + (3/4 + e_y)^2} + \sqrt{(e_x - 1)^2 + e_y^2} - 2\sqrt{(a-1)^2 + 9/16}.$$

Since $e_y = -\sqrt{(12 - 3e_x^2)}/2$, the above expression is completely determined by a and e_x. Elementary algebraic transformations (verified with Maple) show that it always evaluates to a positive value when e_x varies from -2 to 2 and a varies from 0 to 1.

If $p_x = 0$, let $b = p_y$. We have

$$|pe| + |et| - 2|pt| = \sqrt{e_x^2 + (b - e_y)^2} + \sqrt{(e_x - 1)^2 + e_y^2} - 2\sqrt{1 + b^2}.$$

As in the previous case, when e_x varies from -2 to 2 and b varies from 0 to -3/4, elementary algebraic manipulations (which we verified with Maple) show that the above expression evaluates to a positive number.

Case B: $[0 \leq p_x \leq 1$ **and** $p_y < -3/4]$ In this case, we show that $|pq| > |st|$. Let a be the x-coordinate of p, let p' be the point $(a, -3/4)$, and let q' be the intersection of the line l through t and p' with the ellipse E. Since $|pq| \geq |p'q'|$, it is sufficient to show that $|p'q'| > |st| = 2$. The line l is given by the equation

$$y = \frac{3(x-1)}{4(1-a)}.$$

Since E is given by the equation $3x^2 + 4y^2 = 12$, the intersection between E and l is given by:

$$4(1-a)^2x^2 + 3(x-1)^2 - 16(1-a)^2 = 0.$$

For a fixed value of a, let $x(a)$ be the largest root of the above polynomial, and let $y(a)$ be the y-coordinate of l at $x = x(a)$. Then

$$|p'q'| = \sqrt{(x(a)-a)^2 + (y(a)+3/4)^2}.$$

When a varies from 0 to 1, this expression always evaluates to strictly more than 2.

Case C: [$p_x > 1$ **and** $|pt| \leq 1/2$] In this case, the ellipse F is completely contained in the circle H centered at t whose radius is $3/2$. Since H is contained in E, F is also contained in E.

Case D: [$p_x > 1$ **and** $|pt| > 1/2$] We separate this case into two subcases, depending on whether q has a positive or negative x-coordinate. If it is positive, then the part of $\overline{pq}$ that is above the x-axis has length at least $3/2$ and the part of $\overline{pq}$ that is below the x-axis has length more than $1/2$, which means that $|pq| > 2 = |st|$. If the x-coordinate of q is negative but greater than -1, then the same reasoning applies. If the x-coordinate of q is smaller than -1, then the projection of $\overline{pq}$ on the x-axis is larger than 2, which means that $|pq| > 2 = |st|$. □

Lemma 3.5 *The graph G computed by Algorithm 2 is 3-colorable.*

Proof: By Lemmas 3.3 and Lemma 3.4, G is plane and triangle-free. It is known that such a graph is 3-colorable; see [3],[8]. □

Proposition 3.6 *For any point set P, the 3-coloring computed by Algorithm 2 has stretch factor at most 2. Thus, we have $t(3) \leq 2$.*

Proof: It is sufficient to show that the 3-coloring c produced by Algorithm 2 has the 2-ellipse property. Let p and q be points in P such that $c(p) = c(q)$. Let E be the ellipse whose boundary is the set of points e such that $|pe| + |eq| = 2|pq|$. Since (p, q) is not an edge in G, G must contain an edge (s, t) whose two endpoints are inside E. Since $c(s) \neq c(t)$, at least one of s and t has a different color than p and q. Without loss of generality, s is that point. Since s is inside E, we have that $|ps| + |sq| \leq 2|pq|$. □

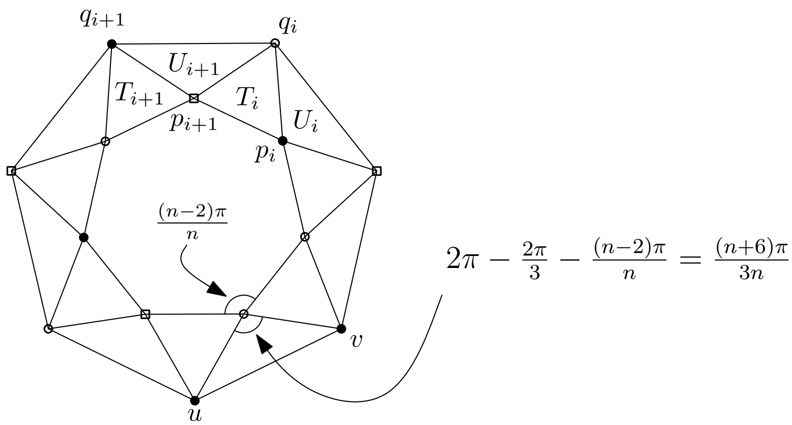


Figure 4: Lower bound of $2 - \epsilon$ for $k = 3$.

Proposition 3.7 *For every $\epsilon > 0$, there exists a point set P such that every 3-coloring of P has stretch factor at least $2 - \epsilon$. Thus, we have $t(3) \geq 2$.*

Proof: Let n be an odd integer, and let $P = \{p_1, \ldots, p_n, q_1, \ldots, q_n\}$ where the p_i's are the vertices of a regular n-gon given in counter-clockwise order, and the q_i's are such that the triangles $T_i = (q_i, p_i, p_{i+1})$ are equilateral with interior lying outside the n-gon (indices are taken modulo n); see Figure 4. Now consider the set of triangles $\mathcal{T} = \{T_1, \ldots, T_n, U_1, \ldots, U_n\}$, where $U_i = (q_{i-1}, p_i, q_i)$. A simple parity argument shows that, for any 3-coloring of P, there is at least one triangle of $\mathcal{T}$ that has two vertices u and v that are assigned the same color. If this triangle is a T_i, then the stretch factor between u and v is at least 2. If this triangle is a U_i, then the stretch factor between u and v is at least $2/2\sin((n+6)\pi/6n)$, which tends to 2 when n goes to infinity. $\square$

Next, we consider the case when $k = 4$. For this case, we simply use the Delaunay triangulation to find a 4-coloring. We then show that this coloring satisfies the $\sqrt{2}$-ellipse property.

Algorithm 3 Off-line 4 Colors

Input: P, a set of points in the plane
Output: c, a 4-coloring of P
Compute the Delaunay triangulation D of P
$c \leftarrow$ a 4-coloring of D

Proposition 3.8 *For any point set P, the coloring computed by Algorithm 3 has stretch factor at most $\sqrt{2}$. Thus, we have $t(4) \leq \sqrt{2}$.*

Proof: It is sufficient to show that the coloring c computed by Algorithm 3 has the $\sqrt{2}$-ellipse property. Let p and q be points of P such that $c(p) = c(q)$. Since (p, q) is not an edge

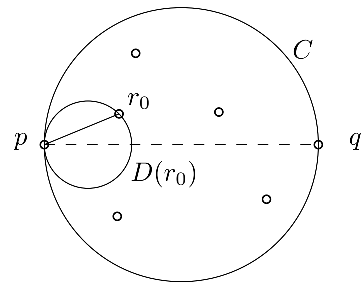


Figure 5: Upper bound of $\sqrt{2}$ for $k = 4$.

in the Delaunay triangulation, the circle C whose diameter is pq contains at least one point of P. For a point r inside C, let $D(r)$ be the circle through p and r whose center is on pq (see Figure 5). Let r_0 be the point inside C such that $D(r_0)$ has minimum diameter. Then, $D(r_0)$ is an empty circle with p and r_0 on its boundary, which means that (p, r_0) is a Delaunay edge. Therefore, $c(r_0) \neq c(p)$, and since r_0 is inside C, we have $|pr_0| + |r_0q| \leq \sqrt{2}|pq|$. □

Proposition 3.9 *For every $\epsilon > 0$, there exists a point set P such that every 4-coloring of P has stretch factor at least $\sqrt{2} - \epsilon$. Thus, we have $t(4) \geq \sqrt{2}$.*

Proof: Let n be an odd integer, and let $P = \{p_1, \ldots, p_n, q_1, \ldots, q_n\}$, where the p_i's are the vertices of a regular n-gon, the q_i's are the vertices of a larger regular n-gon with the same center, and $|q_ip_i| = |p_ip_{i+1}|$ for all i; refer to Figure 6. Let Q_i be the quadrilateral $(p_i, p_{i+1}, q_{i+1}, q_i)$. A simple parity argument shows that for any 4-coloring of P, there is at least one Q_i that has two vertices u and v that are assigned the same color. The stretch factor between u and v is then at least $2/2\sin((n+2)\pi/4n)$, which tends to $\sqrt{2}$ when n goes to infinity. □

Our general algorithm for values $k > 4$ uses ideas from the ordered Θ-graph of Bose *et al.* [1]. We take advantage of the fact that we are in an off-line context, so that we can sort the points according to their y-coordinates. We process the points one by one from the lowest to the highest, splitting the half-plane below the current point p being processed into $k-1$ cones of angle $\pi/(k-1)$ and having their apex at p. For each such cone c_j, we take the point r_j in c_j that is closest to p. Then we assign p a color that has not been assigned to any of the r_j's. The fact that this algorithm uses at most k colors is straightforward, since there are at most $k-1$ such r_j.

Proposition 3.10 *For $k > 4$, we have $t(k) \leq 1 + 2\sin(\pi/(2k-2))$.*

Proof: Let p and q be points of P such that $c(p) = c(q)$. We may assume without loss of generality that $|pq| = 1$ and $q_y \leq p_y$. Let c be the cone with apex at p that contains q in line 4 of Algorithm 4, let r the nearest neighbor of p in c, let r' the intersection between

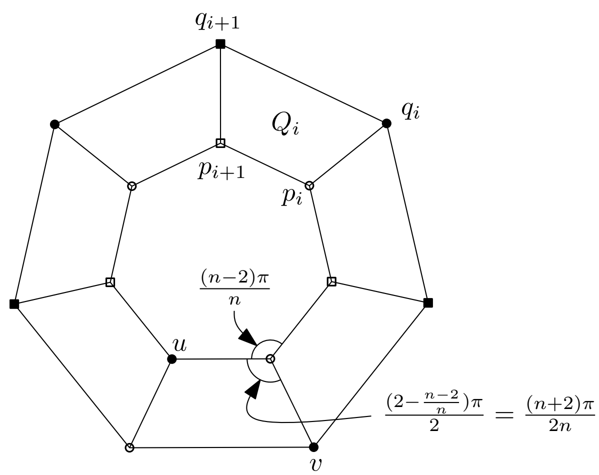


Figure 6: Lower bound of $\sqrt{2} - \epsilon$ for $k = 4$.

Algorithm 4 Off-line k Colors

Input: P, a set of points in the plane
Output: c, a k-coloring of P

```
Let p_1, ..., p_n be the points of P sorted in non-decreasing order of y-coordinates
for i = 1 to n do
    partition the half-plane below p_i into k − 1 cones of angle θ = π/(k − 1) and apex p_i
    for each cone c_j, let r_j be the point in c_j that is closest to p_i
    c(p_i) ← min{l > 0 : ∀r_j, c(r_j) ≠ l}
end for
```

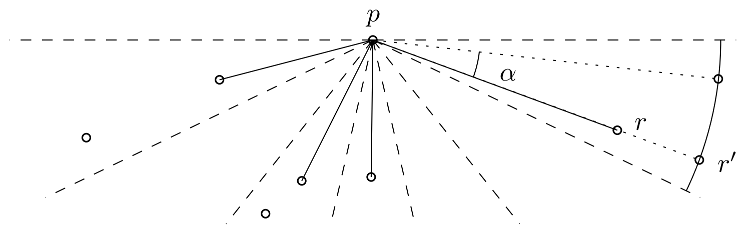


Figure 7: Upper bound of $1 + 2\sin(\pi/(2k-2))$ for $k > 4$.

the line through p and r and the circle centered at p with radius $|pq|$, and let $\alpha = \angle rpq$ (see Figure 7). Then,

$$|pr| + |rq| \leq |pr| + |rr'| + |r'q| = |pq| + |r'q| = |pq| + 2\sin\frac{\alpha}{2}|pq| \leq (1 + 2\sin\frac{\pi}{2(k-1)})|pq|.$$

It follows that the coloring computed by Algorithm 4 has the $(1 + 2\sin(\pi/(2k-2))$-ellipse property. The results follows from the fact that $c(r) \neq c(p)$ and that Algorithm 4 uses at most k colors. □

Proposition 3.11 *For $k > 4$, we have $t(k) \geq 1/\cos(\frac{\pi}{k+1})$.*

Proof: Let $P = \{p_1, \ldots, p_{k+1}\}$ be the vertex set of a regular $(k+1)$-gon. For any three distinct points p, q, and r in P, the ratio $(|pr| + |rq|)/|pq|$ is at least $1/\cos(\frac{\pi}{k+1})$; this value is achieved when p, r, and q are consecutive vertices.

By the pigeonhole principle, any k-coloring of P has to assign the same color to at least two points, say p and q. By the argument above, the stretch factor between p and q is at least $1/\cos(\frac{\pi}{k+1})$. □

4 Upper and lower bounds on $t'(k)$

In the on-line setup, we provide a general algorithm that is the same for all values of $k \geq 2$. Although it is similar to Algorithm 4, there are at least two important differences. First, since we are in an on-line setup, we cannot process the points in the order of their y-coordinates. Therefore, we have to use cones with a greater angle. If we choose the cones a priori as we do in Algorithm 4, we obtain cones whose angle is $2\pi/(k-1)$. However, by aligning the cone's bisectors on the points that are chosen to be neighbors, we can get a slightly better stretch factor, since the angle becomes $2\pi/k$.

Proposition 4.1 *For $k \geq 2$, Algorithm 5 computes a k-coloring that has the t-ellipse property for $t = 1 + 2\sin(\pi/k)$. Thus, we have $t'(k) \leq 1 + 2\sin(\pi/k)$.*

Algorithm 5 On-line k Colors

Input: P, an arbitrarily ordered list of points in the plane

Output: c, a k-coloring of P

1: Let $p_1, \ldots, p_n$ be the points of P in the given order
2: **for** $i = 1$ to n **do**
3: $\quad P_i \leftarrow \{p_1, \ldots, p_{i-1}\}$
4: $\quad j \leftarrow 0$
5: $\quad$ **while** $P_i \neq \emptyset$ **do**
6: $\quad\quad j \leftarrow j + 1$
7: $\quad\quad r_j \leftarrow$ a nearest neighbor of p_i in P_i
8: $\quad\quad P_i \leftarrow P_i \setminus \{r_j\}$
9: $\quad\quad$ **for** each $q \in P_i$ **do**
10: $\quad\quad\quad$ **if** $\angle q p_i r_j \leq 2\pi/k$ **then**
11: $\quad\quad\quad\quad P_i \leftarrow P_i \setminus \{q\}$
12: $\quad\quad\quad$ **end if**
13: $\quad\quad$ **end for**
14: $\quad$ **end while**
15: $\quad c(p_i) \leftarrow \min\{l > 0 : \forall r_j, c(r_j) \neq l\}$
16: **end for**

Proof: Algorithm 5 produces a k-coloring, because each p_i selects at most $k-1$ points r_j. If there were more than $k-1$ such points, then two of them would form an angle of $2\pi/k$ or less around p_i. However, this situation cannot occur because of lines 10 and 11. The proof on the stretch factor is the same as for Proposition 3.10. □

Corollary 4.2 *We have $t'(2) \leq 3$, $t'(3) \leq 1 + \sqrt{3}$ and $t'(4) \leq 1 + \sqrt{2}$.*

Since an off-line lower bound also provides an on-line lower bound, we have $t'(2) \geq t(2) = 3$. It follows that $t'(2) = 3$. We now prove that Algorithm 5 is also optimal for $k = 3$ and 4.

Proposition 4.3 *Let $\mathcal{A}$ be an arbitrary on-line coloring algorithm that guarantees a 3-coloring that has the t-ellipse property. Then $t \geq 1 + \sqrt{3}$.*

Proof: The proof is by an adversarial argument, where the adversary forces a stretch factor of at least $1+\sqrt{3}$. The main objective of the adversary is to force $\mathcal{A}$ to assign different colors to the vertices of an equilateral triangle. Then, the next point is placed in the center of this triangle (see Figure 8(a)). This results in a stretch factor of $1 + \sqrt{3}$.

Consider Figure 8(b), where the points are numbered by the order of insertion. Up to symmetry, there is only one way to assign colors to points p_1 to p_6 such that $t < 1 + \sqrt{3}$. Next, consider Figure 8(c), where the point set of Figure 8(b) is reproduced twice. After the insertion of p_{11}, points p_3, p_7 and p_{11} have to be assigned different colors. Otherwise, the stretch factor would already be greater than $1 + \sqrt{3}$. Then, p_3 has to have the same color as either p_3, p_7 or p_{11}. In any case, the stretch factor between p_{12} and that point is $1+\sqrt{3}$. □

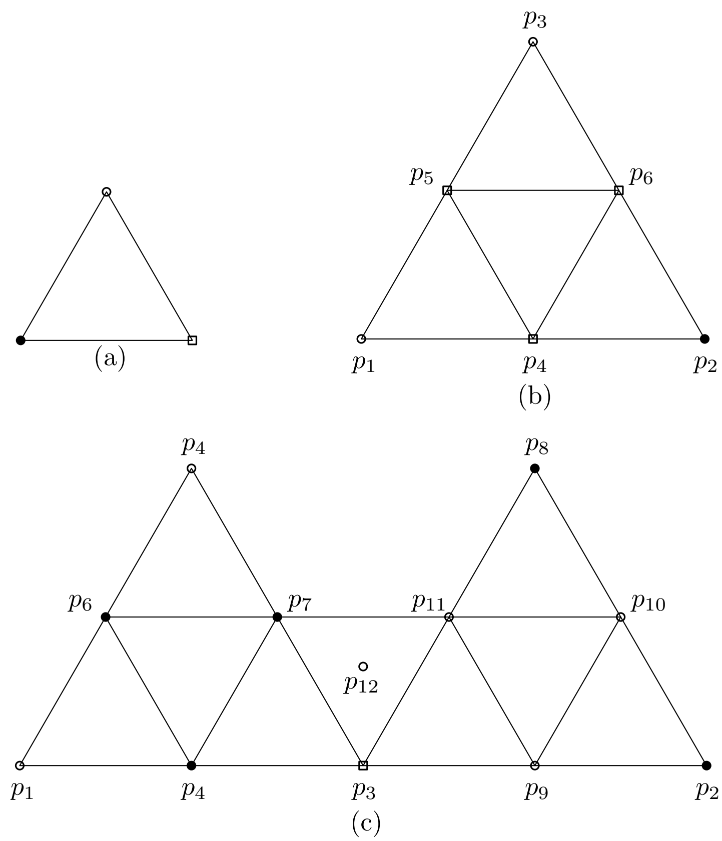


Figure 8: On-line lower bound of $1+\sqrt{3}$ for $k=3$.

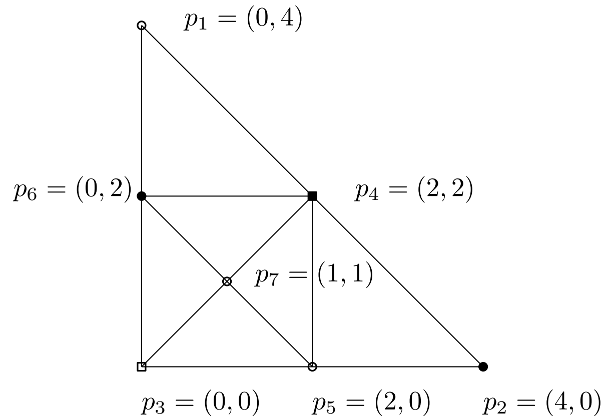


Figure 9: On-line lower bound of $1+\sqrt{2}$ for $k=4$.

Proposition 4.4 *Let $\mathcal{A}$ be an arbitrary on-line coloring algorithm that guarantees a 4-coloring that has the t-ellipse property. Then $t \geq 1+\sqrt{2}$.*

Proof: Consider the point set depicted in Figure 9. Using an adversarial argument, it is easy to show that $\mathcal{A}$ must assign different colors to p_3, p_4, p_5 and p_6. Upon introduction of p_7, $\mathcal{A}$ must assign it the same color as either p_3, p_4, p_5 and p_6. The stretch factor between p_7 and that point is $1+\sqrt{2}$. □

Proposition 4.5 *Let $\mathcal{A}$ be an arbitrary on-line coloring algorithm that guarantees a k-coloring that has the t-ellipse property. Then $t \geq 1/\cos(\frac{\pi}{k})$.*

Proof: Let $P = \{p_1, \ldots, p_k, q\}$, where the p_i' are the vertices of a regular k-gon K and q is the center of the circumcircle of K. If, after processing p_1 to p_k, $\mathcal{A}$ assigned the same color to two points, then as in Proposition 3.11, the stretch factor is $1/\cos(\frac{\pi}{k})$. Otherwise, all p_i are assigned different colors. When q is introduced, the color $\mathcal{A}$ assigns to it has already been assigned to some other point p. In that case, the stretch factor for the edge (q,p) is $1+4\sin(\pi/2k) > 1/\cos(\frac{\pi}{k})$. □

5 Simulation Results

Using simulation, we now provide estimates of the average stretch factor of the colorings produced by Algorithm 4 and Algorithm 5. Using a uniform distribution, we generated 200 sets of 50 points and 200 sets of 200 points. For each point set, we computed the stretch factor for k ranging from 2 to 10. Figure 10 and Figure 11 show the results we obtained for the stretch factor. The 95% confidence interval for these values is ± 0.0365.

The general behavior of the average case performance ratio of these algorithms is not much different than what can be expected from the worst case analysis. In particular, the off-line algorithm performs significantly better than the on-line algorithm. Also, in both cases, as k increases, the stretch factor reduction becomes less and less important. Another interesting observation is that for k large enough ($k > 6$ for 50 points and $k > 3$ for 200 points), the average case stretch factor of the on-line algorithm is worse than the worse case stretch factor of the off-line algorithm.

For $k = 2, 3$ and 4, in the off-line case, we used the algorithm for general values of k. It is important to notice that for $k = 4$, the average stretch factor that we obtained is greater than the upper bound on the the stretch factor obtained by using Algorithm 3. This means that in that case, Algorithm 4 performs worse than Algorithm 3. However, Algorithm 3 is less practical, since we have to compute a 4-coloring of a planar graph.

6 Conclusion

In this paper, we investigated the problem of computing a spanner of a point set that has a linear number of edges and a small chromatic number k. To the best of our knowledge, this problem has not been considered before. For small values of k ($k \leq 4$), we provided tight upper and lower bounds on the smallest possible stretch factor of such spanners. For larger values of k, we provided general upper and lower bounds which, unfortunately, are not tight. We also considered an on-line variant of this problem. Our algorithm for this variant is truly on-line only for the coloring part. Upon insertion of a point, edge selection has to be completely redone from the beginning. Therefore, one way to improve our work would be to provide on-line algorithms that are on-line both on the coloring and the edge selection parts.

References

[1] Prosenjit Bose, Joachim Gudmundsson, and Pat Morin. Ordered theta graphs. *Comput. Geom. Theory Appl.*, 28(1):11–18, 2004.

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

[3] H. Grötzsch. Ein Dreifarbensatz für dreikreisfreie Netze auf der Kugel. *Wiss. Z. Martin-Luther-Univ. Halle-Wittenberg Math.-Natur. Reihe*, 8:109–120, 1959.

[4] J. Gudmundsson, C. Levcopoulos, G. Narasimhan, and M. Smid. Approximate distance oracles for geometric graphs. In *Proceedings of the 13th ACM-SIAM Symposium on Discrete Algorithms*, pages 828–837, 2002.

k	off-line	on-line
2	2.2383	2.5208
3	1.7219	2.1111
4	1.4907	1.8608
5	1.3631	1.7300
6	1.2877	1.6098
7	1.2329	1.5456
8	1.1947	1.4778
9	1.1658	1.4175
10	1.1384	1.3765

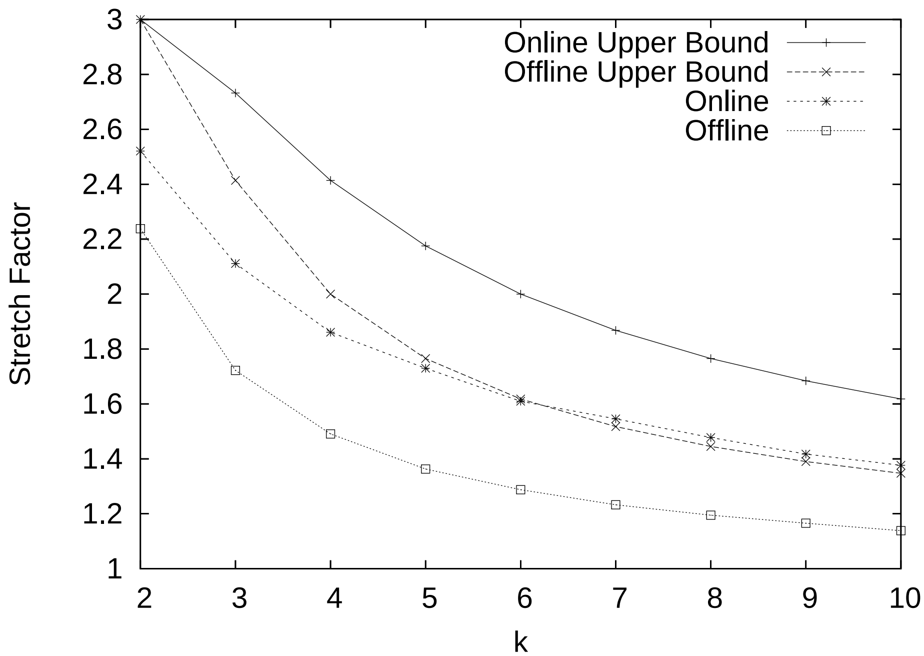


Figure 10: Simulation results for 50 nodes using Algorithm 4 and Algorithm 5.

k	off-line	on-line
2	2.5390	2.7844
3	1.9245	2.3743
4	1.6377	2.0866
5	1.4831	1.9062
6	1.3809	1.7579
7	1.3079	1.6563
8	1.2579	1.5833
9	1.2283	1.5149
10	1.1945	1.4677

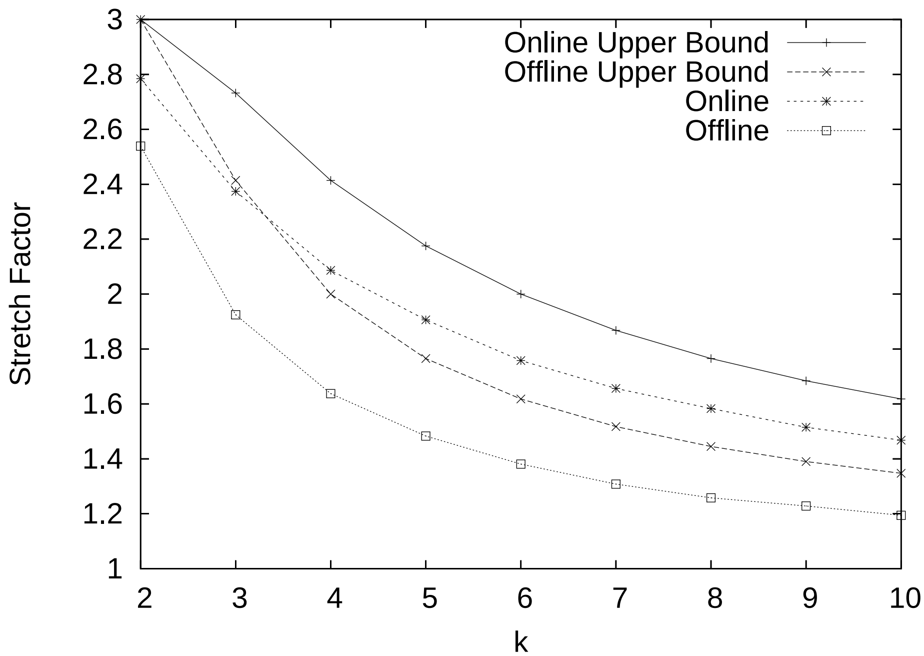


Figure 11: Simulation results for 200 nodes using Algorithm 4 and Algorithm 5.

[5] J. Mark Keil and Carl A. Gutwin. Classes of graphs which approximate the complete Euclidean graph. *Discrete Comput. Geom.*, 7(1):13–28, 1992.

[6] Giri Narasimhan and Michiel Smid. *Geometric Spanner Networks.* Cambridge University Press, New York, NY, USA, 2007.

[7] Bhaskaran Raman and Kameswari Chebrolu. Design and evaluation of a new mac protocol for long-distance 802.11 mesh networks. In *MobiCom '05: Proceedings of the 11th annual international conference on Mobile computing and networking*, pages 156–169, New York, NY, USA, 2005. ACM Press.

[8] Carsten Thomassen. A short list color proof of Grotzsch's theorem. *Journal of Combinatorial Theory B*, 88:189–192, 2003.