ABSTRACT

The temporary and unfixed physical topology of a wireless ad-hoc network is determined by the distribution of the wireless nodes as well as the transmission power (range) assignment of each node. This paper studies asymmetric power assignments for which the induced communication graph is $k$-strongly connected, while minimizing the total energy assigned (which is NP-Hard) and maximizing the network lifetime. We show that our power assignment algorithm from [9] achieves a bicriteria approximation of $(O(k), O(k \log n \sqrt{n \log(n)})$ with high probability for the minimal total cost/maximal network (respectively) lifetime problem in the plane in the case of arbitrary battery charges. The same algorithm is an $(O(k), O(1))$-approximation in the case of uniform batteries. To the best of our knowledge, this is the first attempt to provide a bicriteria approximation factor for the total power assignment cost and the network lifetime under the $k$-fault resilience criterion. We provide some results for the linear power assignment algorithm in [30] as well. In addition, we extend the static algorithms above to support dynamic node insert/delete operations in $O(\log n)$ time for the linear case and an expected $O(k \log \log n)$ amortized time in the plane.

Categories and Subject Descriptors

C.2.1 [Computer-Communication Networks]: Network Architecture and Design—Wireless communication; G.2.2 [Discrete Mathematics]: Graph Theory—Network Problems; G.3 [Probability and Statistics]: Probabilistic algorithms (including Monte Carlo)

1. INTRODUCTION

A wireless ad-hoc network consists of several transceivers (nodes) located in the plane, communicating by radio. The underlying physical topology of the network is dependent on the distribution of the wireless nodes (location) as well as the transmission power (range) assignment of each node. The transmission range $r_t$ of node $t$ is determined by the power assigned to that node, denoted by $p(t)$. It is customary to assume that the minimal transmission power required to transmit to distance $d$ is $d^\alpha$, where the distance-power gradient $\alpha$ is usually taken to be in the interval $[2, 4]$ (see [26]). Thus, node $t$ receives transmissions from $s$ if $p(s) \geq d(s,t)^\alpha$, where $d(s,t)$ is the Euclidean distance between $s$ and $t$.

There are two possible models: symmetric and asymmetric. In the symmetric setting, for any two nodes $s,t \in T$, $p(s) \geq d(s,t)^\alpha \Leftrightarrow p(t) \geq d(t,s)^\alpha$, that is node $s$ can reach node $t$ if and only if node $t$ can reach node $s$. We can also refer to it as the undirected model. The asymmetric variant allows directed links between two nodes. Krumke et al. [23] argued that the asymmetric version is harder than the symmetric one. In this paper we examine the asymmetric model.

The most fundamental problem in wireless ad-hoc networks is to find a power assignment which induces a communication graph that satisfies some topology property. Two natural optimization objectives arise, minimizing the total energy consumption and maximizing the network lifetime.

This paper is organized as follows. In what follows, we present the model, discuss the previous work, state our results and describe range assignment algorithms for planar and linear $k$-strong connectivity. The total power/network lifetime bicriteria approximation factor is derived in Section 2. In Section 3 we show a dynamic conversion of these algorithms.

1.1 The model

We are given a set $T$ of $n$ transceivers $t_1, t_2, \ldots, t_n$, positioned in $\mathbb{R}^d$, $d \geq 1$. A power assignment for $T$ is a vector of transmission powers $A = \{p(t) \mid t \in T\}$. The transmission possibilities resulting from a power assignment in-
duce a directed communication graph $H_A = (T, E_A)$, where $E_A = \{(s, t) \mid p(s) \geq d(s, t)^\alpha\}$ is the set of directed edges resulting from the power assignment. The cost of the power assignment is given by $C_A = \sum_{t \in T} p(t)$.

We assume that all nodes share the same frequency band, and time is divided into equal size slots that are grouped into frames. Thus, the study is conducted in the context of TDMA. In TDMA wireless ad-hoc networks, a transmission scenario is valid if and only if it satisfies the following three conditions; (i) A node is not allowed to transmit and receive simultaneously, (ii) A node cannot receive from more than one neighbor at the same time, (iii) A node receiving from a neighbor should be spatially separated from any other transmitter by at least a distance $D$. However, if nodes use unique signature sequences (i.e., joint TDMA/CDMA scheme), then the second and third conditions may be dropped, and the first condition only characterizes a valid transmission scenario. Thus, our MAC layer is based on TDMA scheduling [16, 17, 31] such that collisions and interferences do not occur.

The wireless distribution of communication nodes presumes that each transceiver does not have a constant power supply, but rather has to rely on its own energy sources. Each node $t$ has some initial battery charge $b_t$, which is sufficient for a limited amount of time, depending on the power assigned to $t$. It is common to take the lifetime of a wireless node $t$ to be $l_t = b_t/r_t^\alpha$. Let $B = \{b_t \mid t \in T\}$ be a vector of initial battery charges. The network lifetime is defined as the time it takes the first node to run out of its battery charge. For a given power assignment $A$, the network lifetime of the induced communication graph with respect to the initial battery charges $B$ is

$$l_A(B) = \min_{t \in T} l_t = \min_{t \in T} \frac{b_t}{r_t^\alpha}.$$  

In the special case where all initial battery charges are equal, that is $b_t = b$ for all $t \in T$, we say that $B$ is uniform. Otherwise, $B$ is arbitrary.

For a power assignment $A$, the communication graph $H_A$ is strongly connected if, for any two nodes $s, t \in T$, there exist directed paths from $s$ to $t$ and from $t$ to $s$ in $H_A$. In this paper, we require that $H_A$ remains strongly connected even if up to $k - 1$ nodes are removed. We refer to $k$ as the fault resistance parameter. If $H_A$ satisfies this requirement, $H_A$ is a $k$-fault resistant strongly connected graph. In short, $H_A$ is $k$-strongly connected.

This paper addresses the minimum energy-maximum lifetime $k$-fault resistant strong connectivity problem (MEML$k$SC).

**Problem 1 (MEML$k$SC).**

**Input:** A set of nodes $T$ in $\mathbb{R}^2$, a vector of initial battery charges $B$ and a constant $k \geq 1$.

**Output:** A power assignment $A$, so that $H_A$ is a $k$-fault resistant strongly connected graph, $C_A$ is minimized and $l_A(B)$ is maximized.

In this work we assume $\alpha = 2$ for simplicity, though our results can be easily extended to any constant $\alpha$.

### 1.2 Previous work

In [29], a formal study of controlling the network topology by adjusting the transmission power of the nodes, was initiated. Most of the problems are aimed at computing a low energy power assignment that meets global topological constraints. In this paper we focus on the strong connectivity (all-to-all) topology property. This property is extremely useful in certain applications of wireless networks (e.g., a battlefield or rescue operation).

To the best of our knowledge, no non-trivial results are known for both the total power assignment cost and network lifetime under the $k$-fault resilience criterion. In what follows, we present previous results separately for estimating the cost of the power assignment and for maximizing the network lifetime.

**Total energy consumption.** Kirousis et al. [22] were the first to study the strong connectivity problem while minimizing the total energy consumption. They proved it to be NP-hard for the 3-dimensional Euclidean space for any value of $\alpha$. For the planar case, they provided a 2-approximation algorithm. The NP-hardness for the 2-dimensional Euclidean space for any value of $\alpha$ was proved in [13], and a simple 1.5-approximation algorithm for the case $\alpha = 1$ has been provided in [3]. Further results may be found in [2, 5, 12].

A natural extension to the topology problems above is to impose the constraint of fault resistance. The benefits of a $k$-fault resistant topology is the multi-path redundancy for load balancing and higher transmission reliability. As power-optimal strong connectivity is NP-hard, so is power-optimal $k$-strong connectivity. The best approximation result up to date for planar asymmetric $k$-strong connectivity is due to Carmi et al. [9], with an approximation ratio of $O(k)$. Another possible connectivity property is $k$-edge connectivity, which implies that the removal of any $k$ edges results in a disconnected graph. In [7], Calinescu and Pan presented various aspects of symmetric/asymmetric $k$-strong connectivity for nodes and edges. Hajiaghayi et al. [18] gave an algorithm for symmetric $k$-strong connectivity with $O(k)$-approximation factor in geometric graphs. Jia et al. [20] present various approximation factors (depending on $k$) for the symmetric $k$-strong connectivity. Additional results can be found in [1, 4, 6, 14, 19, 24].

**Network lifetime.** In the case of uniform battery charges, maximizing the network lifetime is equivalent to minimizing the maximal power assigned to any node. The first to study this problem were Ramamathan and Hain [29], who provided an optimal polynomial time algorithm for this problem under the strong connectivity property. A general approach, which leads to polynomial time algorithms for any monotone property, was developed in [24]. In [5], a PTAS for the problem under various network tasks was developed by devising an LP formulation for the problem. For additional results, see [11, 21].

Moscibroda and Wattenhofer [25] tackle the problem for data gathering applications by maximizing the time the network is clustered by a dominating set. They give approximation algorithms with an approximation ratio of $O(\log n)$ for the cases of both uniform and arbitrary battery charges. In the case of uniform batteries, they also add the $k$-fault resilience criterion. A similar problem, maximizing the residual energy of a node after a broadcast operation, the so-called the critical energy problem, was studied in [27], for which an optimal polytime algorithm was designed. Some results for efficient sensor scheduling and sensing range as-
3) We extend the algorithms in [9] and [30] to support dynamic node insert/delete operations. In particular, our main contributions are:

1) For the planar layout of nodes, we show that the algorithm described in [9] achieves a bicriteria approximation of \( O(k), O(k \log n \sqrt{n \varphi(n)}) \) with high probability for the minimal total cost/maximal network (respectively) lifetime problem in the case of arbitrary battery charges. The same algorithm is a \( O(k), O(1) \)-approximation in the case of uniform batteries.

2) In this paper we address the linear layout of nodes as well. The transceivers are positioned on a single line in increasing order from left to right (see Figure 2.1). We show that the algorithm described in [30] results in an optimal network lifetime in the case of uniform \( B \), which guarantees a \( O(1), 1 \)-approximation for the minimal total cost/maximal network (respectively) lifetime problem.

3) We extend the algorithms in [9] and [30] to support dynamic node insert/delete operations in \( O(\log n) \) time for the linear case and an expected cost and network lifetime.

1.4 Range assignments

Next we describe the range assignments for the linear and planar k-strong connectivity developed in [30] and [9] respectively.

1.4.1 Linear k-strong connectivity ([30])

Put \( d_i = d(t_i, t_{i+1}) \), for \( 1 \leq i \leq n - 1 \), and

\[
d^t_{i,k} = d(t_i, t_{\max(1,i-k)}), \quad d^t_{i,k} = d(t_i, t_{\min(n,i+k)}).
\]

Now, for each \( i \), assign \( p(t_i) = \max\{d^t_{i,k}^2, d^t_{i,k}^2\} \). Let \( A_{L,k} \) denote the resulting range assignment.

There are \( k \) node-disjoint paths between any pair of nodes since, if there is a path from \( t_i \) to \( t_j \), where \( i < j \), then there is a path from \( t_i \) to any intermediate node \( t_l \) for \( i < l < j \). The power assignment results in a constant factor approximation.

1.4.2 Planar k-strong connectivity ([9])

For each node \( t \in T \), let \( N_t \subseteq T \) be the set of \( k \) nodes closest to \( t \) and let \( r^*_t = \max_{e \in N_t} d(t, t') \). We now describe the range assignment algorithm. Compute a minimum spanning tree \( \text{mst} \) of the Euclidean graph induced by \( T \). Assign

\[
\text{We say that the network lifetime } l_A(B), \text{ of the communication graph induced by } A, \text{ is an approximation of } O(\lambda) \text{ to the maximal network lifetime, } l^*_A(B), \text{ if } l^*_A(B)/l_A(B) \in O(\lambda).
\]

\[
\text{Note that, if node } t_i \text{ reaches } t_j \text{ in one hop, there is no need to have } k \text{ node-disjoint paths from } t_i \text{ to } t_j, \text{ since any failure of a node other than } t_i \text{ to } t_j \text{ does not interrupt transmission from } t_i \text{ to } t_j.
\]

to each node \( t \in T \) the range \( r^*_t \). Denote this initial range assignment by \( A' \). For each edge \( e = (t, s) \) of \( \text{mst} \), increase the range of the nodes in \( N_t \cup N_s \) (if necessary), so that each node \( t' \in N_t \) will reach all nodes in \( N_s \cup \{s\} \), and vice versa. Let \( A_k \) denote the resulting range assignment.

The idea is rather simple, we construct \( k \) node-disjoint paths along the edges of the \( \text{mst} \). Think of each \( N_t \) as a large intersection, which contains \( k \) intersection points. For every edge \( (t, s) \) in the \( \text{mst} \), all nodes in \( N_s \cup \{s\} \) are made reachable in one hop from the nodes in \( N_t \). The range assignment of each node \( t \) must be at least \( r^*_t \) (otherwise \( k \)-strong connectivity is impossible), and in addition sufficient enough to create the intersections mentioned above. We obtain an approximation ratio of \( O(k) \) times the optimum under the \( L_2 \)-metric in \( O(n \log n) \) time.

2. NETWORK LIFETIME

In this section we analyze the \( k \)-strong connectivity algorithms described in Subsections 1.4.1 and 1.4.2. We show that these algorithms enjoy a good approximation factor for two optimization problems simultaneously – minimizing the total energy consumption and maximizing the lifetime of the network.

Recall that each node \( t \) has some initial battery charge \( b_t \). Its lifetime is determined by the power \( p(t) \) assigned to it. The lifetime of the whole network, resulting from a power assignment \( A \) with respect to initial battery charges \( B \), is defined as the minimal lifetime of all nodes, that is \( l_A(B) = \min_{t \in T} b_t/p(t) \).

2.1 Planar node layout

In what follows, we address the planar layout of nodes \( T \). We start by showing a constant factor approximation in the case of uniform initial battery charges, and then present our analysis for the general case.

**Uniform battery charges**

First we assume that the initial battery charges \( B_U \) are uniform, that is \( b_i = b \) for all \( t \in T \).

**Lemma 2.1** ([9]). Given an \( \text{mst} \) edge \((t, s)\), let \( r^{t,s}_t \) be the range which node \( t' \in N_t \) has to be assigned in order to reach all nodes in \( N_s \cup \{s\} \). Then

\[
r^{t,s}_t < r^*_t + d(t, s) + r^*_s.
\]

Note that the inequality holds if \( t' \) is replaced by any \( s' \in N_s \). Let \( A^*_N \) be a power assignment so that the induced communication graph is \( k \)-strongly connected and the network lifetime \( l^*_A(B_U) \) is maximized. For a given edge \( s = (t, s) \), we denote \( |e| = d(t, s) \). The following two lemmas provide upper bounds for the network lifetime.

**Lemma 2.2.** \( l^*_A(B_U) \leq \min_{t \in T} b_t / r^*_t \).

**Proof.** For a graph to be \( k \)-strongly connected, each node has to have at least \( k \) neighbors. Thus, each node \( t \in T \) has to be assigned \( p(t) \geq r^*_t \).

**Lemma 2.3.** \( l^*_A(B_U) \leq \min_{e \in \text{mst}} |e|^2 / b_t \).

**Proof.** Obviously, the network lifetime decreases as the fault resistance factor increases. That is, the higher the value of \( k \), the lower the maximal possible network lifetime.
This is the case since higher fault resistance requires larger range assignments. Consider the maximal possible lifetime of a 1-strongly connected graph. Let $A_{	ext{MST}}$ be a power assignment in which each node is assigned to reach its neighbors in the MST. It is easy to see that $H_{A_{	ext{MST}}}$ is 1-strongly connected. Let $e_t = (u,v)$ be the longest edge in the MST, so that

$$l_{A_{	ext{MST}}}(B_t) = \min_{e \in \text{MST}} |e| = \frac{b}{|e|}. \quad (1)$$

Suppose by contradiction that there exists some power assignment $A'$ such that the corresponding network lifetime $l'_{A'}(B_t) > l_{A_{	ext{MST}}}(B_t)$ and $H_{A'}$ is a 1-strongly connected graph. Consider the cut $(S,T)$ induced by $e_t$ in MST. Since the graph $H_{A'}$ is 1-strongly connected, there exists an edge $e' = (s',t') \in E_{A'}$ such that $s' \in S$ and $t' \in T$. Clearly, $e'$ is not in the MST. From (1) and the fact that $t' \leq \frac{b}{|e'|}$, we obtain $\frac{b}{|e'|} < \frac{b}{|e|}$. This means that the MST is not a minimum spanning tree of the Euclidean graph induced by $T$, which is a contradiction. \hfill \blacksquare

Let $r_{t_{\text{max}}}^* = \max_{e \in T} r_{t}^*$ and let $e^* \in \text{MST}$ be the longest edge of the MST, so that $|e^*| = \max_{e \in \text{MST}} |e|$. We are ready to state our main result.

**Theorem 2.4.** For the planar k-strong connectivity power assignment $A_{k}$ and uniform initial battery charges $B_u$, it holds $l_{A_{k}}(B_u) \geq l_{A_{k}}(B_t)/9$.

**Proof.** According to the range assignment algorithm, each node $t$ is initially assigned a power $p(t) = r_{t}^{2}$ so as to have at least $k$ neighbors. Then the power of each node is increased if needed according to some MST edge. We distinguish between two cases:

**Case 1:** $p(t) = r_{t}^{2}$. If the power of node $t$ does not increase, then obviously $p(t) \leq r_{t}^{2}$.

**Case 2:** $p(t) > r_{t}^{2}$. Then the power is increased due to some MST edge $e = (u,v)$. According to Lemma 2.1:

$$p(t) = (r_{t}^{u,v})^{2} < (r_{u}^{*} + |e| + r_{v}^{*})^{2}.$$

Consider two possibilities:

(a) If $|e| \leq \max\{r_{u}^{*}, r_{v}^{*}\}$, then

$$p(t) < 9 (\max\{r_{u}^{*}, r_{v}^{*}\})^{2} \leq 9 r_{t_{\text{max}}}^{* 2}.$$

(b) If $|e| > \max\{r_{u}^{*}, r_{v}^{*}\}$, then

$$p(t) < 9 |e|^{2} \leq 9 |e^*|^{2}.$$

We have shown that the power assignment of each node $t$ is $O\left(\max\{r_{t_{\text{max}}}^{*}, |e^*|\}^{2}\right)$. According to Lemmas 2.2 and 2.3, for every $t \in T$ we have $l_{t} \geq l_{A_{k}}(B_u)/9$, and therefore $l_{A_{k}}(B_u) \geq l_{A_{k}}(B_t)/9$. \hfill \blacksquare

**Arbitrary battery charges**

In the case of arbitrary battery charges we use a probabilistic approach, assuming that nodes are placed uniformly in the unit square, and analyze the lifetime for a sufficiently large number of nodes $n$.

Recall that the range increase of some node $t'$ is at most $r_{t'} + d(t,s) + r_{e}$, where $t' \in N_t$ and $e = (t,s)$ is some MST edge adjacent to $t$. Unfortunately, in the case of arbitrary initial battery charges $B_u$ we cannot use previous bounds on maximal network lifetime (Lemmas 2.2 and 2.3). In the following lemma we state a much more general bound on the maximal network lifetime.

**Lemma 2.5.** $l_{A_{k}}(B_u) \leq \min_{t \in T} \frac{b_t}{r_{t}^{*}}$.

We omit the proof; it is similar to the one in Lemma 2.2, but with varying initial battery charges. The difference between the bounds given by Lemmas 2.2 and 2.5 is crucial, since we cannot use them in a similar manner as we did in the proof of Theorem 2.4. For example, node $t'$ can be assigned a range of $O(r_{t'}^{*})$, which will result in an arbitrarily small network lifetime, as depicted in Figure 1. We counter that by proving in Lemma 2.6 that the ratio between $r_{t}^{*}$ and $r_{e}^{*}$ for any two nodes is bounded with high probability under uniform distribution of nodes in the plane for $k < \frac{n}{(1+\gamma) \log n}$, where $\gamma$ is any positive constant. Let $r_{t_{\text{max}}}^{*} = \max_{t \in T} r_{t}^{*}$ and $r_{t_{\text{min}}}^{*} = \min_{t \in T} r_{t}^{*}$.

**Lemma 2.6.** For a set of $n$ points $T$ placed uniformly in the unit square

$$\lim_{n \to \infty} \Pr\left[ \frac{r_{t_{\text{max}}}^{*}}{r_{t_{\text{min}}}^{*}} = O\left(\sqrt{k \log n \sqrt{\varphi(n)}}\right) \right] = 1,$$

where $\varphi(n)$ is any function with $\lim_{n \to \infty} \varphi(n) = \infty$.

To prove the lemma, we will need Lemmas 2.9 and 2.10 below. Before we get there, let us point out an additional difficulty, due to the fact that we cannot use the bound projected by the longest edge in the MST (Lemma 2.3). To cope with that, we use

**Lemma 2.7.** (Penrose [28]). For $n$ points placed uniformly in the unit square, let $M_n$ (respectively, $M'_n$) be the longest edge-length of the nearest neighbor graph (respectively, the minimum spanning tree) on these points. Then, $\lim_{n \to \infty} \Pr[M_n = M'_n] = 1$.

Since $M_n \leq r_{t_{\text{max}}}^{*}$ for any value of $k$, and $e^* = M'_n$, we obtain the following lemma.

**Lemma 2.8.** For a set of $n$ points, placed uniformly in the unit square, $\lim_{n \to \infty} \Pr[|e^*| \leq r_{t_{\text{max}}}^{*}] = 1$.

**Lemma 2.9.** For a set of $n$ points, placed uniformly in the unit square, $\lim_{n \to \infty} \Pr\left[ r_{t_{\text{max}}}^{*} > 2 \sqrt{\frac{(k+1) \log n}{\pi(n-1)}} \right] = 0$.\hfill \blacksquare

Figure 1: Node $t' \in T$ is assigned a range of $O(r_{t'}^{*})$ to reach all nodes in $N_{t'}$. The lifetime of $t'$ is decreased by a factor of $r_{t_{\text{max}}}^{*}/r_{t'}^{*}$, which may be arbitrarily large.
Proof. Let $\varepsilon = 2\sqrt{(k+1)\log n/(n(n-1))}$ Since
\[ \Pr [r^*_t > \varepsilon] = \Pr \left( \bigcup_{t \in T} [r^*_t > \varepsilon] \right) \leq \sum_{t \in T} \Pr [r^*_t > \varepsilon], \] we have
\[ \lim_{n \to \infty} \sup \Pr [r^*_t > \varepsilon] \leq \lim_{n \to \infty} \sup \sum_{t \in T} \Pr [r^*_t > \varepsilon]. \] We will prove that $\lim_{n \to \infty} \sum_{t \in T} \Pr [r^*_t > \varepsilon] = 0$. For any node $t$, the probability that there are at most $k - 1$ out of the other $n - 1$ nodes within a distance $\varepsilon$ from $t$ is maximal when the point $t$ is a corner of the unit square. Therefore,
\[ \sum_{t \in T} \Pr [r^*_t > \varepsilon] \leq \sum_{t \in T} \sum_{k=1}^{k-1} \binom{n-1}{i} \left( \frac{n-i}{n} \right)^i \leq \sum_{i=0}^{k-1} \frac{(1 - \frac{1}{2} \pi \varepsilon^2)^{n-1-i}}{i!} \left( \frac{\pi \varepsilon^2 (n-1)}{1 - \frac{4}{3} \pi \varepsilon^2} \right)^i. \]
\[ = n \left( 1 - \frac{(k+1) \log n}{n-1} \right) \sum_{i=0}^{k-1} \frac{(k+1) \log n}{1 - \frac{(k+1) \log n}{n-1}} \frac{k!}{(k+1)^{k+1}}. \]
\[ \leq \frac{n}{\log n} \sum_{i=0}^{k-1} \frac{(k+1) \log n}{1 - \frac{(k+1) \log n}{n-1}} \frac{k!}{(k+1)^{k+1}}. \]
\[ \leq \frac{k}{n^k (k-1)!} \left( \frac{(k+1) \log n}{n-1} \right)^{k-1} \leq \frac{k}{n^k (k-1)!} \left( \frac{(k+1) \log n}{n-1} \right)^{k-1}. \]
Clearly $\lim_{n \to \infty} \sum_{t \in T} \Pr [r^*_t > \varepsilon] = 0$ for $k < \frac{n}{(1+\gamma) \log n}$, where $\gamma$ is any positive constant. 

Lemma 2.10. For a set of $n$ points, placed uniformly in the unit square
\[ \lim_{n \to \infty} \Pr [r^*_{\text{min}} < \sqrt{\frac{1}{2\pi(n-1)}} / \sqrt{n} \varphi(n)] = 0, \] where $\varphi(n)$ is any function with $\lim_{n \to \infty} \varphi(n) = \infty$.

Proof. Let $\delta = \sqrt{\frac{1}{2\pi(n-1)}} / \sqrt{n} \varphi(n)$. Similarly to the previous lemma,
\[ \Pr [r^*_{\text{min}} < \delta] = \Pr \left( \bigcup_{t \in T} [r^*_t < \delta] \right) \leq \sum_{t \in T} \Pr [r^*_t < \delta], \] so that
\[ \lim_{n \to \infty} \sup \Pr [r^*_t < \delta] \leq \lim_{n \to \infty} \sup \sum_{t \in T} \Pr [r^*_t < \delta]. \]
We will prove that $\lim_{n \to \infty} \sum_{t \in T} \Pr [r^*_t < \delta] = 0$. For any node $t$, the probability that there are at least $k$ out of the other $n - 1$ nodes within a distance $\delta$ from $t$ is maximal when the point $t$ is at a distance of at least $\delta$ from the boundary of the unit square. Let $a_i = \binom{n-1}{i} (\pi^2)^i (1 - \pi^2)^{n-1-i}$, for $0 \leq i \leq n - 1$. It is easy to verify that, for $\delta \leq \sqrt{\frac{1}{2\pi(n-1)}}$, we have $a_{i+1} \leq a_i/2$ for each $i$. Therefore,
\[ \sum_{t \in T} \Pr [r^*_t < \delta] \leq \sum_{i=0}^{n-1} \frac{(n-1)}{i} (\pi^2)^i (1 - \pi^2)^{n-1-i} \leq 2n \frac{(n-1)}{k} (\pi^2)^k (1 - \pi^2)^{n-1-k} \leq 2n(n-1)! \delta^2 \leq 2n(n-1)! \delta^2/k! \]
Clearly $\lim_{n \to \infty} \sum_{t \in T} \Pr [r^*_t < \delta] = 0$ for any value of $k$.

Lemma 2.6 follows easily from Lemmas 2.9 and 2.10. We are ready to state our main result.

Theorem 2.11. For a set of $n$ points, placed uniformly in the unit square, arbitrary initial battery charges $\mathcal{B}$ and a planar $k$-strong connectivity power assignment $A_k$, the network lifetime $l_{A_k}(\mathcal{B})$ is at most $O(k \log n \sqrt{n} \varphi(n))$ times worse than $l_{A_k^*}(\mathcal{B})$ with high probability, where $\varphi(n)$ is any function with $\lim_{n \to \infty} \varphi(n) = \infty$.

Proof. The proof resembles the proof of Theorem 2.4. If the power of node $t$ is increased due to some MST edge $e = (u,v)$, then according to Lemma 2.1 it holds $p(t) = (r^*_u)^2 < (r^*_u + |e| + r^*_v)^2$. In conjunction with Lemma 2.8, we have $p(t) \leq 9r^*_u^2$ with high probability. By Lemma 2.5, $l_{A_k^*}(\mathcal{B})$ is minimal in $T_b/r^*_u^2$. Finally, from Lemma 2.6, for any node $t$,
\[ (r^*_u/r^*_t)^2 \leq (r^*_u/r^*_\text{min})^2 = O(k \log n \sqrt{n} \varphi(n)) \]
with high probability. This rests our proof.

2.2 Linear node layout

In this section we address the linear layout of nodes $T$ and uniform initial battery charges $\mathcal{B}_L$. Let $\forall t \in T$, $b_t = b$ as before. We now show that the power assignment $A_{L,k}$ in 1.4.1 results in an optimal network lifetime for linear $k$-strong connectivity. We use the following lemma from [30].

Observation 2.12 ([30]). For a linear layout of nodes $T$, let $A_L$ be a power assignment so that $H_2$ is a $k$-strongly connected line. Then for each node $t_i \in T$ there are at least $\min\{i-1, k\} \binom{n-i}{k} \min(n-i, k)$ nodes to its left (right) with sufficient range assignment to reach $t_i$ in one hop.

Let $A^*_L$ be a power assignment so that $H_2$ is a $k$-strongly connected line and the network lifetime $l_{A^*_L}(\mathcal{B}_L)$ is maximized. The next theorem shows the optimality of $l_{A^*_L}(\mathcal{B}_L)$.

Theorem 2.13. For a linear layout of nodes $T$, and uniform initial battery charges $\mathcal{B}_L$, the power assignment $A_{L,k}$ is optimal, that is $l_{A_{L,k}}(\mathcal{B}_L) = l_{A^*_L}(\mathcal{B}_L)$.

Proof. From Lemma 2.12 it easily follows that
\[ l_{A^*_L}(\mathcal{B}_L) \leq \min_{i \leq t_i \leq n} \frac{b}{(d_{L,k}(i))^2} \text{ and } l_{A_{L,k}}(\mathcal{B}_L) \leq \min_{i \leq t_i \leq n} \frac{b}{(d_{L,k}(i))^2}, \]
since $t_i$ has to be reachable in one hop by at least $\min\{i-1, k\} \binom{n-i}{k} \min(n-i, k)$ nodes to its left (right). Therefore, $l_{A_{L,k}}(\mathcal{B}_L) \leq l_{A^*_L}(\mathcal{B}_L)$.


3. DYNAMIC ALGORITHMS

In this section we present an efficient dynamic scheme for maintaining k-strong connectivity in both \( \mathbb{R}^1 \) and \( \mathbb{R}^2 \). We extend our static algorithms in [9] and [30], for linear and planar cases, respectively, and make them dynamic.

3.1 Linear node layout

The static algorithm for linear k-strong connectivity in 1.4.1 assigns each node \( t \in T \) with sufficient power to reach \( k \) nodes to its left and \( k \) nodes to its right. Therefore, in order to compute the power assignment of node \( t \), we need to know the distance to \( k \)-th node to its left and \( k \)-th node to its right, that is \( d^L_{i,k} \) and \( d^R_{i,k} \), respectively. We maintain a dynamic data structure that supports insert/delete/query operations in \( O(\log n) \) time. In other words, we dynamically maintain the set of points and efficiently answer the following query: <Given point index \( i \), determine the power of \( t \), in the near optimal power assignment described in [9]>. Moreover, only relative location is needed and not the exact coordinates of each node.

We use a standard balanced binary search tree implemented as a red-black tree [15]. Each node \( t \) is represented by a node in the red-black tree and holds the following information: (a) the number \( n_t \) of nodes in a subtree rooted at \( t \) and (b) for each child \( t_c \), the distance \( d(t, t_c) \). The nodes in the tree are ordered in accordance to their relative positions. The nodes are added or deleted according to the red-black tree operations. We start by describing the insert/delete operations and then address the query operation.

**Insert and Delete** — A new node \( t \) is inserted or deleted according to the red-black insert/delete operations which take \( O(\log n) \) time. The update of node internal information takes place along with the insert/delete of the node along the path from root to \( t \).

**Query** — Given some node \( t_i \), we need to compute its power assignment by computing \( d^L_{i,k} \) and \( d^R_{i,k} \). Without loss of generality we describe the computation of \( d^R_{i,k} \). First we locate the node \( t_i \) which takes \( O(\log n) \) time. Then we traverse the tree starting at \( t_i \) in order to find a node \( k \) places distant from \( t_i \) to the right. Since every node \( t \) holds the number of nodes in a subtree rooted at \( t \) we can find \( t_{i+k} \) in \( O(\log n) \) time by a simple binary search starting at \( t_i \). Once we have located \( t_{i+k} \) we compute the value \( d^R_{i,k} \) as follows\(^5\). Let \( u \) be the least common ancestor of \( t_i \) and \( t_{i+k} \). We compute the distance between \( t_i \) and \( t_{i+k} \) by following the path from \( u \) to \( t_i \) and \( t_{i+k} \). Recall that each node stores the distance to every child. When following a path from \( u \) to \( t_i \) for every left child we add the distance and for every right child we subtract the distance. When following the path from \( u \) to \( t_{i+k} \) we do the same, only this time adding on a right child and subtracting on left. The value \( d^R_{i,k} \) is obtained by adding the distances computed for both paths. As a result the total time it takes to answer a power assignment query for any node is \( O(\log n) \).

For example, in Figure 2.1 there are 8 nodes positioned on a single line. The corresponding red-black tree is presented in Figure 2.2. Each node on the line is represented by a node in the tree. For each node \( t_i \), the value in square brackets is \( n_{t_i} \). For every edge \((t_i, t_j)\) the distance \( d(t_i, t_j) \) is represented by a value along that edge. In Figure 2.3 the calculation of \( d^R_{4,4} \) is presented. That is we look for the distance from \( t_3 \) to \( t_7 \). The least common ancestor of \( t_3 \) and \( t_7 \) is \( t_4 \). We compute as described earlier, \( d(t_3, t_4) = 15 - 9 = 6 \) and \( d(t_4, t_7) = 12 + 8 = 20 \). As a result \( d^R_{4,4} = d(t_3, t_4) + d(t_4, t_7) = 6 + 20 = 26 \).

**3.2 Planar node layout**

The static algorithm for planar k-strong connectivity in 1.4.2 heavily depends on computing \( N_t \) for every node \( t \). That is, for every node we want to be able to find its \( k \)-closest nodes in the plane. In order to maintain the topology property of k-strong connectivity in dynamic settings, where nodes can be added or removed, we need to maintain a dynamic data structure which allows a cheap \( k \)-closest nodes query with every update operation. Next we describe such a data structure with a query in \( O(\log^2 n) \) time under the \( L_\infty \) norm. The preprocessing time is \( O(n \log n) \). Later we discuss the implication of using the \( L_\infty \) norm and its effect on the overall approximation factor.

**Dynamic k-closest nodes**

Below we present a scheme that computes for a given point \( t \), the \( k \) nearest points in the plane under the \( L_\infty \) metric. In other words we aim to find the smallest axis-parallel square centered at \( k \) that contains exactly \( k \) points. We perform a binary search over the sorted values of \( x \)-and \( y \)-coordinates of points, in order to determine the radius of square centered at \( p \). Once we have fixed the radius of the square, we ask how many points are inside of it. The idea is to apply standard orthogonal range searching with fractional cascading technique (see Willard and Lueker [32]), that allows one to count the number of points in a given query axis-parallel rectangle in \( O(\log n) \) time after \( O(n \log n) \) preprocessing. This can be generalized to deal with updates in \( O(\log^2 n) \) time and reporting points in time \( O(k + \log^2 n) \). If the number is exactly \( k \), we found all the \( k \)-nearest neighbors of \( p \). Otherwise, we increase or decrease the radius of square depending on the number of points inside of square. Finally, after we find a square with \( k \) points we can actually report them using a dy-

---

\(^5\) Note that if \( i + k > n \) then \( d^R_{i,k} = d(t_i, t_n) \), which matches the definition.
dynamic orthogonal range tree augmented for reporting (not counting) the points inside of given axis-parallel rectangle. To answer a range reporting query we simply traverse the subtrees determined by the searching procedure, reporting the points of their leaves. Each subtree can be traversed in time proportional to the number of leaves it contains. The query time is $O(k + \log^2 n)$.

Using the $L_\infty$ metric worsens the original $O(k)$ approximation ratio of static $k$-strong connectivity by $\sqrt{2}$ factor as stated in the next observation.

**Observation 3.1.** For any node $t \in T$, let $N^\infty_t$ and $N^2_t$ be the sets of its $k$-closest nodes computed in the $L_\infty$ and $L_2$ metrics, respectively. Then $\max_{t' \in N^\infty_t} d(t, t') \leq \sqrt{2} \max_{t' \in N^2_t} d(t, t')$.

We will now show that node $t'$ can be one of the $k$-closest nodes for at most $8k$ other nodes in $L_\infty$. This will mean that adding or removing a single node will not change more than $O(k)$ $k$-closest node sets, that is the number of all sets $N_t$ so that $t' \in N_t$ is $O(k)$. Let $RN^\infty_t$ be a set of nodes which have $t'$ as their $k$-closest neighbor in $L_\infty$. That is, if $t' \in N^\infty_t$ then $t \in RN^\infty_t$, where $N^\infty_t$ is as defined in Observation 3.1. We prove that $|RN^\infty_t| \leq 8k$, for any $t' \in T$. We will need the following technical lemma.

**Lemma 3.2.** Let $P$ be a set of points in the plane. For any point $q$ in the plane, let $X \subseteq P$ be a set of points that have point $q$ as their nearest neighbor in $P \cup \{q\}$ under the $L_\infty$ metric. Then $|X| \leq 8$.

**Proof.** Without loss of generality, assume that $q$ is the origin, and partition the plane into eight wedges by the four lines $y=0$, $x=0$, $y=x$, and $y=-x$, so that each wedge is open on its clockwise side and closed on its counterclockwise side. We claim that each of these wedges can contain at most one point $a \in P$ whose nearest $L_\infty$-neighbor in $P \cup \{q\}$ is $q$. Without loss of generality, consider the wedge $W_1$, $W_1 = \{(x, y) \mid x \geq 0 \text{ and } 0 < y \leq x\}$. Suppose $|P \cap W_1| \geq 2$ (otherwise the claim is obvious), and let $a$ be the leftmost point in $P \cap W_1$ (see Figure 3). If there is more than one such point, choose $a$ to be the lowest among them. Let $b$ be any other point of $S \cap W_1$ and $d_{\infty}(a, b)$ be the distance between points $a$ and $b$ under the $L_\infty$ metric. Then $d_{\infty}(b, a) < d_{\infty}(b, q)$. Indeed, $d_{\infty}(b, q) = b_x$ and $d_{\infty}(a, b) = \max\{|b_x-a_x|, |b_y-a_y|\}$ Clearly, $b_x-a_x < b_y$. If $b_y < a_y$ then $|b_y-a_y| = b_y-a_y < b_y \leq b_x$; otherwise $|b_y-a_y| = a_y-b_y < a_y \leq a_x \leq b_x$. Hence, in both cases $d_{\infty}(b, a) < d_{\infty}(b, q)$, as claimed. That is, $a$ is the only one point of $P \cap W_1$ in $X$. Since we have eight wedges, $|X| \leq 8$.

The same argument in the proof can be applied when analyzing the maximal number of points for which point $q$ is one of the $k$-nearest points. We show that it is possible to have at most $k$ points in each wedge. The following observation can be easily derived then.

**Observation 3.3.** Let $P$ be a set of points in the plane. For any point $q$ in the plane, let $X_q \subseteq P$ be a set of points that have point $q$ as one of their $k$-nearest neighbors in $P \cup \{q\}$ under the $L_\infty$ metric. Then $|X_q| \leq 8k$.

The following lemma is easily derived from Observation 3.3.

**Lemma 3.4.** $\forall t' \in T$ : $|RN^\infty_t| \leq 8k$.

**Dynamic mst**

We maintain the $mst$ under insert and delete operations by using the results provided by Chan in [10]. The author uses a reduction to bichromatic closest pairs in order to maintain the Euclidean $mst$ of a dynamic 2-$d$ point set in $O(\log^{10} n)$ expected amortized time.

**Observation 3.5.** Let $T$ be a set of $n$ points in the plane. Let $mst_T = (T, E)$ be a Euclidean minimum spanning tree over $T$, with $E$ being the set of edges. A point $p$ removal or addition to $T$ changes at most $O(1)$ edges in $E$.

**Proof.** Suppose a point is deleted from $T$. Since in $mst_T$ the degree of each node is at most 6, the deletion of a point may result in at most 6 unconnected components. To form a new $mst$ at most 5 new edges are needed. As a result there are at most 6 edges removed and 5 added to $E$ when forming a new $mst$. Now suppose a point is added to $T$. In the resulting $mst$ this point might have at most 6 neighbors. We cannot simply connect the point because we might have some cycles. There might be at most $\binom{6}{3}$ cycles created as a result of adding a point. What we need to do is connect the point and then resolve the cycles by deleting the edge with the largest cost from each cycle. If we compare the result with the original $mst$, at most 6 edges might be added and at most $\binom{6}{3}$ removed.

The observation above shows that an addition or removal of a node affects only a constant number of nodes with respect to their $mst$ neighbors. Recall that the power assignment for $k$-strong connectivity is to create $k$ symmetric links between $N_t$ and $N_t$ for every edge $(s, t) \in mst$. Therefore, each addition or removal of a node will force power assignment changes according to the changes in the $mst$. Next we focus on insert/delete operations.

**Insert.** — When a new node $t$ is introduced into the system we need to: (a) compute its $k$-closest nodes, (b) recompute $r^*_w$ of every node $w$, for which $t \in N_w$, (c) update $mst$ and (d) update the assignment alongside the $mst$ edges. We focus on each of the steps:

(a) Given some point $p$ it takes $O(\log^2 n)$ time to find its $k$-closest nodes under the $L_1$ norm.

(b) Due to Lemma 3.4 there are at most 8$k$ nodes for which we need to recompute their $k$-closest nodes, which takes $O(k \log^2 n)$ time.
(c) The $\text{mst}$ update takes $O(\log^{10} n)$ expected amortized time according to [10].

(d) When a new node is added, at most $O(1)$ edges are altered. Each $\text{mst}$ edge addition/removal affects $O(k)$ of nodes that need to change their range assignment.

To summarize we have an expected amortized $O(k \log^2 n + \log^{10} n)$ update time in case of adding a new node to the system.

\textbf{Delete} — When we remove a node $t \in T$ we need to do the same as for the insert operation except (a). We recalculate $k$-closest nodes of all nodes $w$ for which $t \in N_w$, update the $\text{mst}$ and the assignment alongside its edges. As before, the expected amortized node removal time would be $O(k \log^2 n + \log^{10} n)$.

4. REFERENCES


