

NEAR OPTIMAL BOUNDS FOR STEINER TREES IN THE HYPERCUBE*

TAO JIANG[†], ZEVI MILLER[†], AND DAN PRITIKIN[†]

Abstract. Given a set S of vertices in a connected graph G , the classic *Steiner tree problem* asks for the minimum number of edges of a connected subgraph of G that contains S . We study this problem in the hypercube. Given a set S of vertices in the n -dimensional hypercube Q_n , the *Steiner cost* of S , denoted by $\text{cost}(S)$, is the minimum number of edges among all connected subgraphs of Q_n that contain S . We obtain the following results on $\text{cost}(S)$. Let ϵ be any given small, positive constant, and set $k = |S|$. **(1)** [upper bound] For every set S we have $\text{cost}(S) < (\frac{1}{3}k + 1 + \frac{1}{2} \ln k)n$. In particular, there is a constant c_1 depending only on ϵ such that if $k > c_1$, then $\text{cost}(S) < (\frac{1}{3} + \epsilon)kn$. **(2)** We develop a randomized algorithm of running time $O(kn)$ that produces a connected subgraph H of Q_n containing S such that with probability approaching 1 as $k, n \rightarrow \infty$ we have $|E(H)| < (\frac{1}{3} + \epsilon)kn$. **(3)** [lower bound] There are constants c_2 and b (with $1 < b < 2$) depending only on ϵ such that if $c_2 < k < b^n$, then as $n \rightarrow \infty$ almost all sets S of size k in Q_n satisfy $\text{cost}(S) > (\frac{1}{3} - \epsilon)kn$. Thus for k in this range with $k \rightarrow \infty$, the upper bound (1) is asymptotically tight. We also show that for fixed k , as $n \rightarrow \infty$, almost always a random family of k vertices in Q_n satisfies $[\frac{k}{3} + \frac{2}{9}(-1 + (-\frac{1}{2})^k)]n - \sqrt{n \ln n} \leq \text{cost}(S) \leq [\frac{k}{3} + \frac{2}{9}(-1 + (-\frac{1}{2})^k)]n + \sqrt{n \ln n}$.

Key words. Steiner trees, hypercube

AMS subject classifications. 05C85, 05C90

DOI. 10.1137/100797473

1. Introduction. Given a metric space Y having metric (or distance function) $\mu : Y \times Y \rightarrow \mathbb{R}$ and a set of points $S \subseteq Y$, the Steiner problem on Y is to find a tree in Y whose vertex set contains S having the minimum possible total length among all such trees. We let $\text{cost}(S)$ be this minimum, where the underlying Y and μ are understood by context. When $Y = \mathbb{R}^2$ and μ is the ℓ_2 norm (the usual geometric distance), the case $|S| = 3$ was studied by Torricelli, and the problem for arbitrary sets S in \mathbb{R}^2 was shown to be NP-complete in [12]. For the same Y with the ℓ_1 metric (i.e., rectilinear distance) the problem was proved to be NP-complete in [13]. Applying the theory of MAX-SNP hardness [23], Trevisan showed [28] that when $Y = \mathbb{R}^n$ and μ is the ℓ_1 metric, the Steiner problem is MAX-SNP hard. This implied that a polynomial time approximation scheme for this problem (a polynomial time algorithm with ratio less than $1 + \epsilon$ for any fixed ϵ) is not possible unless $P = NP$.

We now consider the case when Y is a connected edge-weighted graph $G = (V, E)$ with nonnegative edge weight $w(e)$ for each edge $e \in E(G)$. Thus the metric here is $\mu(x, y) = d_G(x, y)$, the distance between x and y in G . So for any $S \subseteq V$, the *graph Steiner problem*, which we abbreviate as the *Steiner problem* when the underlying graph G is understood, is to determine $\text{cost}(S)$, the minimum over all subtrees T of G containing S , of the quantity $\sum_{e \in E(T)} w(e)$ (called the *total weight* of T), and, if possible, to produce a tree achieving this minimum. We call S the set of *terminals* and a tree achieving this minimum a *Steiner tree* for S . The graph Steiner problem is NP-hard, even in the special case of the hypercube (see below). A polynomial time

*Received by the editors June 3, 2010; accepted for publication (in revised form) June 14, 2011; published electronically September 22, 2011.

<http://www.siam.org/journals/sicomp/40-5/79747.html>

[†]Department of Mathematics, Miami University, Oxford, OH 45056 (jiangt@muohio.edu, millerz@muohio.edu, pritikd@muohio.edu). The first author's research was partially supported by the National Security Agency under grant H98230-07-1-027.

heuristic with approximation ratio roughly 1.55 is given in [27]. See the texts [17] and [25] (among others) for thorough presentations of graph Steiner tree problems.

Of particular interest because of connections to biological applications is the Steiner problem on the Hamming graph, as follows. Let $A_i, 1 \leq i \leq n$, be a set of n finite alphabets, say with $|A_i| = t_i$. The *Hamming graph* $H = H(t_1, t_2, \dots, t_n)$ has $A_1 \times A_2 \times \dots \times A_n$ as its vertex set, where two strings (vertices) x and y in H are joined by an edge precisely when they disagree in exactly one coordinate. All edges are given weight 1. Thus $d_H(x, y)$ is the number of coordinates in which x and y disagree. We note that H can also be viewed as the graph Cartesian product $K_{t_1} \times K_{t_2} \times \dots \times K_{t_n}$ of complete graphs. The case $t_i = 2$ for all i yields the familiar binary hypercube Q_n of dimension n whose vertices are binary strings of length n (over the alphabet $\{0, 1\}$), two vertices joined by an edge when they disagree in exactly one coordinate. A *phylogeny* for S in H is a subtree of H whose leaves are all the points in S , and the *Steiner problem in phylogeny* (STP) is to find the minimum number of edges in any phylogeny for S .

Some results on the Steiner problem and on STP for the Hamming graph are as follows. A polynomial time algorithm solution to the Steiner problem on H in the case $n = 2$ appeared in [5], with a simplified proof for $n = 2$ and lower bound for the case $n = 3$ in [3]. For the next such set of results, let T be a subtree of some graph G containing the terminals $S \subseteq V(G)$. A subtree T' of T is a *full component* of T if no internal point of T' is a point of S . A *k-restricted tree* T for S is one for which every full component of T has at most k points of S . Let $L_k(S)$ be the minimum total weight of a k -restricted subtree of G containing S . Then the *k-Steiner ratio* in G is

$$\rho_k(G) = \min_{S \subseteq V(G)} \frac{\text{cost}(S)}{L_k(S)}.$$

In [9] it is shown that the STP problem for k -restricted phylogeny is APX-complete for $k \geq 4$, even for binary characters, that is, for Q_n . The authors also showed that $\rho_k(Q_n)$ achieves a lower bound for arbitrary metric spaces proved in [4].

A related alignment problem is the following. Take A to be an alphabet of size t , and take $X = \{x_1, x_2, \dots, x_s\}$ to be a set of strings of arbitrary lengths over A . Further, let $p \geq \max\{|x_i|\}$ be a positive integer, and let $T = (V, E)$ be an abstract tree on at least s vertices. We assume here a one-one correspondence between X and some s -subset of V , say letting $x_i \in X$ correspond to some $v_{x_i} \in V$. Now let $A' = A \cup \{\Delta\}$ be an augmented alphabet, where Δ is a new symbol called an ‘‘indel.’’ Consider a one-to-one map $\ell : V \rightarrow (A')^p$ of V into the p -fold Cartesian product $(A')^p$, i.e., the set of length p strings over A' . For each $x_i \in X$ the string $\ell(v_{x_i})$ represents a genome or a vector of characteristics for the species x_i . This $\ell(v_{x_i})$ is a ‘‘padded’’ copy of x_i , consisting of a length p string over A' in which x_i appears as a subsequence, with $p - |x_i|$ indel symbols Δ in the remaining entries. This tree T represents an evolutionary tree leading to the set of species X , in which certain positions (the ones with entry Δ) are unknown (or have been lost in extinct ancestors). We view $\ell(V)$ as a $|V| \times p$ matrix with rows being the images of V under ℓ , and we call this matrix a *tree alignment* T of X . Now let M be a symmetric $(t + 1) \times (t + 1)$ matrix indexed by the letters in A' , where the entry $M(\alpha, \beta), \alpha, \beta \in A'$, is a transition cost (perhaps evolutionary distance) between α and β . Then define the cost $c(T, X, M)$ of this tree alignment by

$$c(T, X, M) = \sum_{uv \in E(T)} \sum_{i=1}^p M(\ell(u)_i, \ell(v)_i),$$

where $\ell(u)_i$ is the entry in the i th coordinate of $\ell(u)$, and similarly for $\ell(v)_i$. The problem is then, given X , to minimize $c(T, X, M)$ over all trees T and order of insertions of $p - |x_i|$ indels for each $1 \leq i \leq s$ in obtaining a padded copy of x_i .

This problem is known as the *multiple sequence tree alignment problem*. In [29] and its preliminary version [19] this problem was proved MAX-SNP hard, and NP-hard even under the restriction that T must be a binary tree. Simplified proofs of these results were given in [30]. For a discussion of these and related problems, see the excellent texts [14] and [24] on computational biology emphasizing connections to discrete mathematics.

We now turn to the focus of this paper, the graph Steiner problem in the hypercube Q_n with terminal set $S \subseteq V(Q_n)$. Recall that Q_n has for its vertices all the 2^n binary strings of length n , two such vertices joined by an edge of weight 1 when they disagree in exactly one coordinate. So the distance $\text{dist}(x, y)$ between two vertices x and y of Q_n is the number of coordinates in which x and y disagree. For this case, the connection with evolutionary trees in biology is made by viewing each terminal $v \in S$ as a description of an individual or species. The i th coordinate is 1 if the species possesses the i th trait and is 0 otherwise. Thus a rooted Steiner tree of Q_n containing S is an efficient branching diagram of how a set of species (the set S) may have evolved from a common ancestor (the root), each edge of the tree indicating an evolutionary transition between possessing or not possessing a particular trait. Thus the Steiner problem for Q_n is known in computational biology as the “maximum parsimony” problem for binary coordinates (see [14]).

This Steiner (or maximum parsimony) problem also includes the well-known “perfect phylogeny” problem (see again [14]) as follows. To avoid trivial reductions, assume that in each of our n coordinates, at least one pair of vertices in S disagree. So trivially $\text{cost}(S) \geq n$. The problem is to determine if there exists a rooted tree τ satisfying the following conditions:

- (1) The leaves of τ are in one-one correspondence with the vertices of S .
- (2) There are a subset E' of $E(\tau)$ and a one-one labeling of E' with the integers $i, 1 \leq i \leq n$, so that for each leaf x of τ , the set of labels appearing on the path from x to the root is precisely the set of coordinates of x (viewed as a vertex in Q_n) having value 1.

Such a tree P , called a *phylogenetic tree* for S , is a possible history of how the set S arose from a common ancestor, the root r . The biological assumptions here are that r has none of the n traits (i.e., $r = 0^n$) and that once a trait appears at a vertex of P it remains a trait in all its descendants. When we contract unlabeled edges of P , we obtain a tree T that corresponds naturally to a subtree of Q_n containing the set S and having n edges. Thus the existence of a phylogenetic tree for S implies $\text{cost}(S) = n$, achieving the trivial lower bound, and the corresponding subtree of Q_n must be a Steiner tree for S . Gusfield [15] found an $O(kn)$ time algorithm for determining whether $S \subseteq V(Q_n)$ has a perfect phylogeny, where $|S| = k$. Consider the natural generalization to the Hamming graph $H = H(r, r, \dots, r)$ with n coordinates, where each coordinate has some value $j, 1 \leq j \leq r$, and $S \subseteq V(H)$. We ask whether there is a subtree T of H containing S for which every leaf of T is in S , such that for every $1 \leq j \leq r$ and $1 \leq i \leq n$ the set of vertices $v \in V(T)$ for which $v_i = j$ induces a subtree of T . A polynomial time algorithm for determining this for any fixed r was found in [1], having running time $O(2^{3r}(kn^3 + n^4))$.

Previous work on the Steiner problem in the hypercube includes the following. Proofs of NP-hardness were given independently in [20, 8, 10, 16] for the weighted

version of the problem, where in [20] NP-hardness was shown even for the special case where all vertices in S have weight 2, using a reduction from the vertex cover problem. Also in [20] an exact formula for $cost(S)$ was given when $|S| \leq 5$. In [21], a result of Frankl and Rödl [11] on the hypergraph Turán problem was applied to show that if S is the set of all weight $r + 1$ points of Q_n , normalized so that $r + 1 \leq \frac{n}{2}$, then $\binom{n}{r+1} + \frac{1}{r+1} \binom{n}{r} \leq cost(S) \leq \binom{n}{r+1} + (1 + o(1)) \frac{\ln r}{r} \binom{n}{r}$ as $r \rightarrow \infty$.

As for arbitrary sets S of any given size k , prior to this report (as pointed out in [20]) there was the simple upper bound $cost(S) \leq \min\{\lfloor \frac{k}{2} \rfloor n, 2^n - 1\}$ for k odd, and $cost(S) \leq \min\{(k - 1 + \frac{1}{k-1}) \frac{n}{2}, 2^n - 1\}$ for k even. This bound follows from joining the points of S by paths to the *centroid* of S , i.e., the point of Q_n whose value in any coordinate is the majority value in that coordinate among the points of S . An improvement upon this simple bound, decreasing the coefficient of kn from $\frac{1}{2}$ to $\frac{9}{20}$, is implicit in another result of [20], though not stated there explicitly, as follows. Let $L_5(n)$ be the maximum of $cost(S)$ over all vertex sets S in Q_n with $|S| = 5$. It was shown in [20] that

$$L_5(n) = 2n - \left\lceil \frac{n}{10} \right\rceil - \left\lceil \frac{n-4}{10} \right\rceil.$$

Thus $L_5(n) \leq \frac{9}{5}n$. Proceeding inductively, assume that we have a subgraph of Q_n containing S having c components. Then select one vertex from each component, group the selected vertices into at most $\lceil \frac{c}{5} \rceil \leq \frac{c}{5} + 1$ blocks each of size at most 5, and by the formula for $L_5(n)$, interconnect the vertices in each block using at most $L_5(n) \leq \frac{9}{5}n$ additional edges and possibly additional vertices for that block. The new subgraph of Q_n thereby formed contains S , has at most $\frac{9}{5}n(\frac{c}{5} + 1)$ additional edges, and has at most $\lceil \frac{c}{5} \rceil \leq \frac{c}{5} + 1$ components. After at most $\lceil \log_5(k) \rceil$ such steps, we obtain a connected subgraph F of Q_n containing S such that

$$|E(F)| \leq \frac{9kn}{25} \left(1 + \frac{1}{5} + \frac{1}{25} + \dots \right) + O(\log(k)) \frac{9n}{5} = \frac{9kn}{20} (1 + o(1))$$

as $k \rightarrow \infty$.

In this paper, we substantially improve upon earlier results as follows. Let S be a set of k vertices in Q_n , and let $\epsilon > 0$ be any small constant. We show that $cost(S) < (\frac{1}{3}k + 1 + \frac{1}{2} \ln k)n$. In particular, there is a constant c_1 depending only on ϵ such that if $k > c_1$, then $cost(S) < (\frac{1}{3} + \epsilon)kn$. Further, we show that this bound is asymptotically tight for moderately sized k in the following strong sense. There are constants c_2 and b (with $1 < b < 2$) depending only on ϵ such that if $c_2 < k < b^n$, then as $n \rightarrow \infty$ almost all sets S of size k in Q_n satisfy $cost(S) > (\frac{1}{3} - \epsilon)kn$. We also give a randomized algorithm of running time $O(kn)$ that produces a connected subgraph H of Q_n containing S such that with probability approaching 1 as $k, n \rightarrow \infty$ we have $|E(H)| < (\frac{1}{3} + \epsilon)kn$.

The paper is organized as follows. We begin by generalizing the problem to subcubes. Let \mathcal{F}_k be a collection of k many subcubes (of various dimensions) of Q_n . Then define $cost(\mathcal{F}_k)$ to be the minimum number of edges in any connected subgraph of Q_n containing at least one vertex from each subcube in the family \mathcal{F}_k . This is an instance of the *group Steiner problem*, where we are given a collection of vertex sets S_i in some weighted graph G and are asked to find the minimum weight tree in G containing at least one vertex from each set S_i in the collection. This problem was introduced in [26] motivated by issues in VLSI design. See [6] for just one recent example of the extensive literature on the group Steiner problem.

We introduce the “merge” of two subcubes D and D' . Intuitively speaking, this is the replacement of these two subcubes by a subcube consisting of possible vertices of Q_n through which D and D' can be efficiently joined with the remaining subcubes in the family \mathcal{F}_k . The size of our family is thereby reduced by 1. A “merging pattern” π is the successive application of such pairwise merges. We define a cost function $\text{cost}(\mathcal{F}_k, \pi)$ for any merging pattern π applied to a family \mathcal{F}_k which upper bounds $\text{cost}(\mathcal{F}_k)$, and show that there is always a connecting subgraph H_π for \mathcal{F}_k such that $\text{cost}(\mathcal{F}_k) \leq |E(H_\pi)| \leq \text{cost}(\mathcal{F}_k, \pi)$. We derive an upper bound on the expected value of $\text{cost}(\mathcal{F}_k, \pi)$ over all merging patterns, yielding our upper bound result for $\text{cost}(\mathcal{F}_k)$.

For the lower bound we show that for almost every family \mathcal{F}_k of “moderate” size, every small subfamily \mathcal{F}' of it is hard to connect, in that $\text{cost}(\mathcal{F}', \pi)$ is high for every merging pattern π . From this we deduce that $\text{cost}(\mathcal{F}_k)$ satisfies a lower bound close to the upper bound previously obtained. Thus for such families \mathcal{F}_k the previously derived upper bounds are nearly optimal. Here we prove and then use the crucial fact there is always some merging pattern π for which $\text{cost}(\mathcal{F}_k) = \text{cost}(\mathcal{F}_k, \pi)$, that is, a merging pattern which achieves $\text{cost}(\mathcal{F}_k)$.

Finally we show by martingale methods that almost any merging pattern π yields a value of $\text{cost}(\mathcal{F}_k, \pi)$ similar to the upper bound for $\text{cost}(\mathcal{F}_k)$. From this we obtain a randomized algorithm of running time $O(kn)$ for producing a connecting subgraph for \mathcal{F}_k which is near optimal, in effect a randomized $1 + \epsilon$ approximation scheme, for almost all moderately sized families \mathcal{F}_k as $k \rightarrow \infty$.

2. The Steiner problem for subcubes. In this section we generalize the Steiner problem to subcubes of Q_n , as described in the introduction. Thus we are given a family \mathcal{F}_k of k many subcubes of Q_n , and the goal is to estimate $\text{cost}(\mathcal{F}_k)$, the minimum number of edges in any connected subgraph H of Q_n containing at least one vertex from each subcube in the family. We say that such an H “connects” \mathcal{F}_k . Our method is to develop a “merge” operation on pairs of subcubes, and by iterating this merge consider a “merging pattern” on the family \mathcal{F}_k . An algorithm is then given for using any merging pattern to construct a subgraph which connects \mathcal{F}_k . By upper bounding the expected value of a certain cost function defined on merging patterns, we obtain our upper bound for $\text{cost}(\mathcal{F}_k)$. To facilitate the statement of various definitions and the algorithm, we need the following notation.

For each $s \in \{0, 1, \dots, n\}$, if we fix the values in a given set of $n - s$ coordinates but allow values in the other s coordinates to vary, then we obtain a set D of 2^s vertices that induces an s -dimensional subcube. We represent D by a $(0, 1, *)$ -string, using the symbol $*$ in each of the s coordinates where the value is allowed to vary. For instance, $D = 1*000*$ represents the set $\{100000, 100001, 110000, 110001\}$. For convenience, we also use D to denote the s -dimensional subcube it induces. So, the same letter D will refer to a $(0, 1, *)$ -string, the set of vertices it represents, and the subcube that set induces. Such flexibility allows us to simplify our presentation.

Given a $(0, 1, *)$ -string D of some length n and $i \in [n]$, we let D_i denote the i th coordinate of D . We define a (symmetric) distance function dist on pairs from $\{0, 1, *\}$ by letting $\text{dist}(0, 1) = \text{dist}(1, 0) = 1$ and $\text{dist}(a, b) = 0$ for all other pairs $a, b \in \{0, 1, *\}$. We extend this distance function to pairs D, D' of $(0, 1, *)$ -strings of any given length n by letting $\text{dist}(D, D') = \sum_{i=1}^n \text{dist}(D_i, D'_i)$. Given two such $(0, 1, *)$ -strings and $i \in [n]$, we say that D and D' have *opposite values* in coordinate i if $\{D_i, D'_i\} = \{0, 1\}$. We see that $\text{dist}(D, D')$ equals the number of coordinates in which D, D' have opposite values, which is also the length of a shortest path in Q_n with one endpoint in D and the other in D' . For instance, $\text{dist}(10*11*, *11101) = 2$,

with a corresponding path of length 2 being 101111, 111111, 111101. When viewed as sets we have $D \cap D' \neq \emptyset$ if and only if $dist(D, D') = 0$.

We now define a (symmetric) operator \wedge on pairs from $\{0, 1, *\}$ by the rule that $0 \wedge 1 = 1 \wedge 0 = *$ and that for each $a \in \{0, 1, *\}$, $a \wedge a = a$ and $a \wedge * = * \wedge a = a$. We then extend \wedge to pairs of $(0, 1, *)$ -strings by performing the operation \wedge coordinate-wise; for instance, $10*11* \wedge *11101 = 1*11*1$. See Figure 1. We refer to $D \wedge D'$ as the *merge* of D and D' . Given two subcubes D and D' , a (D, D') -*path* is a path in Q_n with one end in D and the other in D' .

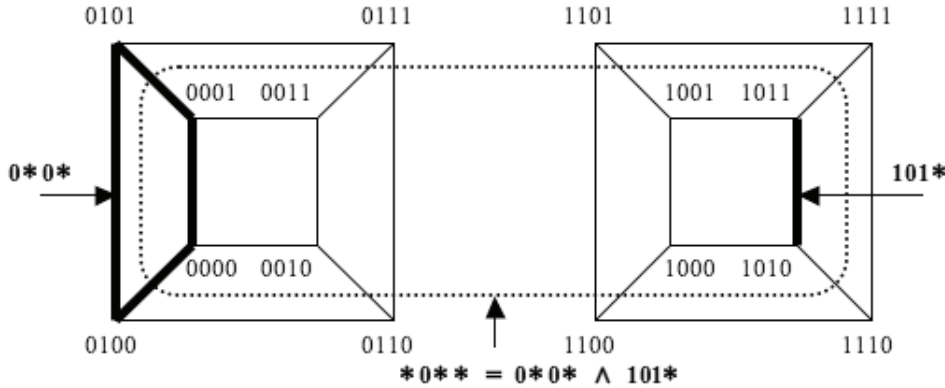


FIG. 1. The merge operation.

PROPOSITION 2.1. Let D and D' be two subcubes in Q_n .

(a) If $D \cap D' \neq \emptyset$, then $D \cap D' = D \wedge D'$.

(b) Let u be any vertex in $D \wedge D'$. Then there exists an (D, D') -path of length $dist(D, D')$ that contains u .

Proof. (a) Suppose $D \cap D' \neq \emptyset$. Then $D \cap D'$ induces a subcube D'' of Q_n . Fix an $i \in [n]$. Then $D''_i = 1$ if and only if $(D_i, D'_i) = (1, 1), (1, *)$, or $(*, 1)$, which is precisely when $D_i \wedge D'_i = 1$. Similarly $D''_i = 0$ if and only if $D_i \wedge D'_i = 0$. Since $D \cap D' \neq \emptyset$, $\{D_i, D'_i\} \neq \{0, 1\}$. Thus, $D''_i = *$ if and only if $D_i = D'_i = *$, which holds if and only if $D_i \wedge D'_i = *$. So, $D \cap D' = D \wedge D'$.

(b) If $dist(D, D') = 0$, then $D \cap D' \neq \emptyset$ and $D \cap D' = D \wedge D'$. We take the single point $\{u\}$ to be the desired path. For the general case, we define bit strings x, x' as follows. For each of the $d = dist(D, D')$ coordinates i for which D, D' have opposite values, let $x_i = D_i$ and $x'_i = D'_i$. For all other coordinates i , let $x_i = x'_i = u_i$. Then we have $x \in D, x' \in D'$, and $dist(x, x') = d$. It suffices to verify that some x, x' -path of length d contains u . Let P be a shortest x, u -path and P' a shortest u, x' -path. Then their concatenation W is an x, x' -walk through u . Since entries of u match those of both x and x' in all but d coordinates, and since u_i matches one of x_i, x'_i in the other d coordinates, W has length $d = dist(D, D')$, and so it must be a path. \square

As before, consider a family $\mathcal{F}_k = (D^1, D^2, \dots, D^k)$ of k subcubes of Q_n , the D^i 's not necessarily distinct. For each $i \in [k]$, we designate one vertex in $D^i \cap V(H)$ as the *representative* of D^i in H . A vertex may represent more than one member of \mathcal{F}_k . A subgraph H that connects \mathcal{F}_k and has the fewest possible edges is necessarily a tree, and we call it a *Steiner tree* for \mathcal{F}_k in Q_n . We let $cost(\mathcal{F}_k)$ denote the number of edges in a Steiner tree for \mathcal{F}_k and call it the *Steiner cost* of \mathcal{F}_k . It is convenient to view \mathcal{F}_k as a $k \times n$ matrix with entries from $\{0, 1, *\}$ such that for each $i \in [k]$ the i th row of \mathcal{F}_k is D^i .

Let n be a positive integer. Consider a $k \times n$ matrix M with entries from $\{0, 1, *\}$ and integers a, b with $1 \leq a < b \leq k$. By the a, b -merge of M we mean the $(k - 1) \times n$ matrix resulting from M by replacing its a th row by the merge of rows a and b , and then deleting the b th row. Thus we also speak of the a, b -merge of a family \mathcal{F}_k of subcubes by viewing that family as a matrix (as noted above). For every row j of M , if $j > b$, then in the new matrix this row becomes row $j - 1$, while if $j < b$, it remains row j . A k -merging pattern is a sequence $\pi = (a_k, b_k), (a_{k-1}, b_{k-1}), \dots, (a_2, b_2)$ of ordered pairs of indices satisfying $1 \leq a_i < b_i \leq i$ for each i . Given such a π and letting $M = M(k)$, recursively define matrices $M(k - 1), M(k - 2), \dots, M(1)$ by letting $M(i - 1)$ be the a_i, b_i -merge of $M(i)$. So for $1 \leq i \leq k$, $M(i)$ is the $i \times n$ matrix resulting from M by applying in succession the first $k - i$ merging pairs $(a_k, b_k), (a_{k-1}, b_{k-1}), \dots, (a_{k-i+1}, b_{k-i+1})$ of π to M . (We suppress any reference to π in the notation $M(i)$ since π will be understood by context.) Now let $cost_M(i)$ be the distance between rows a_i and b_i of $M(i)$ (defined earlier as the number of coordinates in which one of the two strings has a 0 and the other has a 1). Then define $cost(M, \pi) = \sum_{i=2}^k cost_M(i)$, which we call the *merging cost of M relative to the merging pattern π* .

For example, let $M = M(4)$ be the matrix shown at left below, and consider the merging pattern $\pi = (2, 4), (1, 2), (1, 2)$. The display shows the forming of the families $M(3), M(2), M(1)$ from the choice of $M = M(4)$ and π . In this example, $cost_M(4) = 2, cost_M(3) = 2, cost_M(2) = 1$, so $cost(M, \pi) = 5$:

$$\begin{aligned} \begin{bmatrix} 1 & 1 & 0 & * & 1 & 1 \\ 0 & * & * & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 0 & * & 1 & 0 & 0 \end{bmatrix} &\rightarrow \begin{bmatrix} 1 & 1 & 0 & * & 1 & 1 \\ * & 0 & * & 1 & 0 & * \\ 0 & 0 & 0 & 1 & 1 & 1 \end{bmatrix} \\ &\rightarrow \begin{bmatrix} 1 & * & 0 & 1 & * & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 \end{bmatrix} \rightarrow [* & 0 & 0 & 1 & 1 & 1]. \end{aligned}$$

Given a family \mathcal{F}_k of k subcubes of Q_n (viewed as a $k \times n$ matrix) and a k -merging pattern π_k , the following recursive algorithm produces a graph which connects \mathcal{F}_k , having at most $cost(\mathcal{F}_k, \pi_k)$ edges.

ALGORITHM 2.2. Connect (\mathcal{F}_k, π_k) .

Input: a family $\mathcal{F}_k = (D^{k,1}, D^{k,2}, \dots, D^{k,k})$ of subcubes in Q_n and a k -merging pattern $\pi_k = (a_k, b_k), (a_{k-1}, b_{k-1}), \dots, (a_2, b_2)$.

Output: a subgraph H_{π_k} of Q_n that connects \mathcal{F}_k .

Algorithm:

If $k = 2$ then do the following:

Let H_{π_k} be a path of length $dist(D^{k,1}, D^{k,2})$ that intersects both $D^{k,1}$ and $D^{k,2}$.

Return H_{π_k} .

If $k > 2$, then do the following:

Let $\mathcal{F}_{k-1} = (D^{k-1,1}, D^{k-1,2}, \dots, D^{k-1,k-1})$ be obtained from \mathcal{F}_k by deleting its b_k th term and replacing its a_k th term with $D^{k,a_k} \wedge D^{k,b_k}$. Let π_{k-1} be obtained from π_k by deleting (a_k, b_k) .

Let $H_{\pi_{k-1}}$ be the graph returned by **Connect** $(\mathcal{F}_{k-1}, \pi_{k-1})$. Let u_k be a representative of $D^{k,a_k} \wedge D^{k,b_k}$ in $H_{\pi_{k-1}}$. Let P_k be a path of length $dist(D^{k,a_k}, D^{k,b_k})$ through u_k that intersects both D^{k,a_k} and D^{k,b_k} , whose existence is guaranteed by Proposition 2.1. Let $H_{\pi_k} = H_{\pi_{k-1}} \cup P_k$. Return H_{π_k} .

It is straightforward to see that the subgraph H_{π_k} constructed by Algorithm 2.2 connects \mathcal{F}_k and $|E(H_{\pi_k})| \leq cost(\mathcal{F}_k, \pi_k)$. Hence we have the following proposition.

PROPOSITION 2.3. *Let \mathcal{F}_k be a family of $k \geq 2$ subcubes of Q_n , and let π be a k -merging pattern. Then the output H_π of Algorithm 2.2 is a subgraph of Q_n connecting \mathcal{F}_k , satisfying $E(H_\pi) \leq \text{cost}(\mathcal{F}_k, \pi)$. Thus $\text{cost}(\mathcal{F}_k) \leq \text{cost}(\mathcal{F}_k, \pi)$.*

Our next goal is to prove the crucial fact that there always exists a merging pattern π for \mathcal{F}_k for which $\text{cost}(\mathcal{F}_k) = \text{cost}(\mathcal{F}_k, \pi)$. For that we need the following simple structural lemma.

LEMMA 2.4. *Let $q \geq 2$ be an integer. Suppose T is a tree and W is a subset of $V(T)$ with $|W| \geq q$. Then T has a subtree T^* such that $q \leq |V(T^*) \cap W| \leq 2q - 2$ and that $T - E(T^*)$ has a component that contains all of $W - (V(T^*) \cap W)$.*

Proof. Make T a rooted tree by choosing any vertex r to be the root. For each vertex v , let T_v denote the subtree of T rooted at v . Among all vertices u satisfying $|V(T_u) \cap W| \geq q$, choose one to minimize $|V(T_u) \cap W|$, noting that u exists since $|V(T_r) \cap W| \geq q$. Let v_1, \dots, v_m denote the children of u in T . For each $i \in [m]$, let $s_i = |V(T_{v_i}) \cap W|$. By our choice of u , for each $i \in [m]$, we have $s_i \leq q - 1$. Since $|W \cap \{u\}| + \sum_{i=1}^m s_i = |V(T_u) \cap W| \geq q$, there exists a smallest index c such that $|W \cap \{u\}| + \sum_{i=1}^c s_i \geq q$. Our choice of c and the fact that $s_c \leq q - 1$ also imply that $|W \cap \{u\}| + \sum_{i=1}^c s_i = (|W \cap \{u\}| + \sum_{i=1}^{c-1} s_i) + s_c \leq (q - 1) + (q - 1) = 2q - 2$. Let T^* denote the subtree of T formed by u , $\bigcup_{i=1}^c T_{v_i}$, and the edges uv_1, \dots, uv_c . Then $q \leq |V(T^*) \cap W| \leq 2q - 2$, and the subtree of T induced by $V(T) - [V(T^*) - u]$ is the required component of $T - E(T^*)$ containing all of $W - (V(T^*) \cap W)$. \square

The general case of Lemma 2.4 will be used in section 4. For now, we will only need the $q = 2$ case of Lemma 2.4. In this case, the subtree T^* of T which we obtain contains two vertices x, y of W , and $T - E(T^*)$ has a component that contains all of $W - \{x, y\}$. Let L denote the unique x, y -path in T^* (and in T). By the proof of Lemma 2.4, $T - E(L)$ has a component F that contains all of $W - \{x, y\}$. We will use this fact in the following key lemma on merging patterns realizing $\text{cost}(\mathcal{F}_k)$.

LEMMA 2.5. *Let $\mathcal{F}_k = (D^1, \dots, D^k)$ be a family of subcubes of Q_n , $k \geq 2$.*

(a) *There are integers i, j with $1 \leq i < j \leq k$, a Steiner tree T_k for \mathcal{F}_k , a Steiner tree T_{k-1} for \mathcal{F}_{k-1} (the i, j -merge of \mathcal{F}_k) such that $T_k = T_{k-1} \cup P$, where P is a (D^i, D^j) -path of length $\text{dist}(D^i, D^j)$ that goes through a representative of $D^i \wedge D^j$ in T_{k-1} .*

(b) *There exists a merging pattern π for \mathcal{F}_k for which $\text{cost}(\mathcal{F}_k) = \text{cost}(\mathcal{F}_k, \pi)$.*

Proof. (b) follows from (a) by iteratively applying (a) to develop such a π . Concerning (a), let T be a Steiner tree for \mathcal{F}_k , for each m letting x_m denote the vertex of T representing D^m . Let W denote the multiset $\{x_1, \dots, x_k\}$.

First suppose that $x_i = x_j$ for some $i \neq j$. Then $\text{dist}(D^i, D^j) = 0$ and $D^i \cap D^j = D^i \wedge D^j$. Let T_{k-1} be any Steiner tree for \mathcal{F}_{k-1} , let $T_k = T_{k-1}$, and let P be any representative of $D^i \wedge D^j$ in T_{k-1} . All requirements in (a) are then easily satisfied. Thus we assume throughout the rest of the proof that x_1, \dots, x_k are distinct.

By the remarks after Lemma 2.4, we can find $x_i, x_j \in W$ such that, with L denoting the x_i, x_j -path in T , the forest $T - E(L)$ has a component F that contains all of $W - \{x_i, x_j\}$. Since T is connected and acyclic, L intersects F in exactly one vertex w . Since L goes through w , $|E(L)| \geq \text{dist}(w, x_i) + \text{dist}(w, x_j) \geq \text{dist}(w, D^i) + \text{dist}(w, D^j)$. Note that L connects the family (w, D^i, D^j) . Next, we describe a connecting subgraph for (w, D^i, D^j) having at most $|E(L)|$ edges.

We let u be a vertex in the subcube $D^i \wedge D^j$ such that $\text{dist}(w, u) = \text{dist}(w, D^i \wedge D^j)$. Let P' be a shortest w, u -path. By Lemma 2.1, there exists a (D^i, D^j) -path P of length $\text{dist}(D^i, D^j)$ that goes through u . Now, $P' \cup P$ connects (w, D^i, D^j) and $|E(P \cup P')| \leq |E(P')| + |E(P)| = \text{dist}(w, D^i \wedge D^j) + \text{dist}(D^i, D^j)$. Let l be a coordinate

that contributes to $dist(D^i, D^j)$. Then l contributes 1 to $dist(w, D^i) + dist(w, D^j)$, since D_l^i and D_l^j have opposite values and $w_l \in \{0, 1\}$. Further, l does not contribute to $dist(w, D^i \wedge D^j)$ since $(D^i \wedge D^j)_l = *$. Let l be a coordinate that contributes to $dist(w, D^i \wedge D^j)$. Then at least one of D^i and D^j has opposite value from w in coordinate l , so l contributes at least one to $dist(w, D^i) + dist(w, D^j)$. Thus, $dist(D^i, D^j) + dist(w, D^i \wedge D^j) \leq dist(w, D^i) + dist(w, D^j)$. This, combined with our earlier arguments, shows that $|E(P \cup P')| \leq |E(L)|$.

We let $T_k = F \cup (P \cup P')$, and we show that T_k is the tree required in part (a). First, we have that T_k connects \mathcal{F}_k since $P \cup P'$ connects (w, D^i, D^j) . But $|E(T_k)| \leq |E(T)| - |E(L)| + |E(P \cup P')| \leq |E(T)|$. Since T is a Steiner tree for \mathcal{F}_k , T_k must also be a Steiner tree for \mathcal{F}_k . As for the required decomposition of T_k , let $T_{k-1} = F \cup P'$, so that $T_k = T_{k-1} \cup P$. Since F connects $\mathcal{F}_k - \{D^i, D^j\}$ and $u \in D^i \wedge D^j$, we have that T_{k-1} connects \mathcal{F}_{k-1} , the i, j -merge of \mathcal{F}_k . If some tree T' with fewer edges than T_{k-1} also connects \mathcal{F}_{k-1} , then T' together with a (D^i, D^j) -path Q of length $dist(D^i, D^j)$ through a representative w' of T' in $D^i \wedge D^j$ would be a connecting subgraph of \mathcal{F}_k with fewer edges than T_k . This contradicts T_k being a Steiner tree for \mathcal{F}_k . So T_{k-1} must be a Steiner tree for \mathcal{F}_{k-1} . \square

Given a family \mathcal{F}_k of $k \geq 2$ subcubes of Q_n , by Proposition 2.3, for any merging pattern π , $cost(\mathcal{F}_k, \pi)$ provides an upper bound on $cost(\mathcal{F}_k)$. Furthermore, by Lemma 2.5, there exists a merging pattern π such that $cost(\mathcal{F}_k, \pi)$ is exactly $cost(\mathcal{F}_k)$. In the next few sections, we derive lower and upper bounds on $cost(\mathcal{F}_k)$ based on these facts.

3. Upper bound on the Steiner cost. By Proposition 2.3, $cost(\mathcal{F}_k) \leq cost(\mathcal{F}_k, \pi)$ for any merging pattern π of \mathcal{F} . In this section we derive an upper bound for the expectation of $cost(\mathcal{F}_k, \pi)$ over all π , thus giving an upper bound for $cost(\mathcal{F}_k)$. In section 4, we will see that when k is in a certain range, then with probability approaching 1 as $n \rightarrow \infty$, for almost every such \mathcal{F}_k this upper bound is near optimal.

First, we define the probability space of our random merging patterns. For each positive integer $k \geq 2$, let Ω_k denote the set of all $\prod_{t=2}^k \binom{t}{2}$ many k -merging patterns $\pi = (a_k, b_k), (a_{k-1}, b_{k-1}), \dots, (a_2, b_2)$, where $1 \leq a_t < b_t \leq t$ for each $t = 2, \dots, k$. We turn Ω_k into a probability space by using the uniform probability distribution \mathbb{P}_k on Ω_k , letting $\mathbb{P}_k(\pi) = \frac{1}{\prod_{t=2}^k \binom{t}{2}}$ for each $\pi \in \Omega_k$. Equivalently, we may view π as being produced in $k - 1$ steps by making independent choices for (a_t, b_t) , in the order $t = k, k - 1, \dots, 2$. In particular, $\mathbb{P}((a_t, b_t) = (i, j)) = \frac{1}{\binom{t}{2}}$ whenever $1 \leq i < j \leq t$.

Let n be a positive integer and M a $k \times n$ matrix with entries from $\{0, 1, *\}$. Let $\mathbb{E}_k(M)$ denote $\mathbb{E}(cost(M, \pi))$ over (Ω_k, \mathbb{P}_k) , the expected cost of M relative to a random $\pi \in \Omega_k$ with uniform probability. Given an event A in (Ω_k, \mathbb{P}_k) , with slight abuse of notation, we let $\mathbb{E}_k(M|A)$ denote $\mathbb{E}(cost(M, \pi)|A)$ in (Ω_k, \mathbb{P}_k) , the expectation of $cost(M, \pi)$ conditioned on $\pi \in A$. We continue to view a given family \mathcal{F}_k of k subcubes of Q_n as a $k \times n$ matrix over $\{0, 1, *\}$. So $\mathbb{E}_k(\mathcal{F}_k)$ is the expected cost of over (Ω_k, \mathbb{P}_k) of the corresponding matrix.

We turn now to estimating $\mathbb{E}_k(\mathcal{F}_k)$. For each $j \in [n]$, let $\mathcal{F}_{k,j}$ denote the j th column of \mathcal{F}_k ; thus $\mathcal{F}_{k,j}$ is a $k \times 1$ matrix. By the way we define the merging cost of a matrix relative to a merging pattern, it is easy to see that $cost(\mathcal{F}_k, \pi) = \sum_{j=1}^n cost(\mathcal{F}_{k,j}, \pi)$ for each $\pi \in \Omega_k$. By linearity of expectation, we have $\mathbb{E}_k(\mathcal{F}_k) = \sum_{j=1}^n \mathbb{E}_k(\mathcal{F}_{k,j})$. We now develop a few useful lemmas on $\mathbb{E}_k(M)$, where M is a $k \times 1$ matrix with entries from $\{0, 1, *\}$. For convenience, for any 1×1 matrix M with entries from $\{0, 1, *\}$, define $\mathbb{E}_1(M) = 0$.

For inductive arguments, it is useful to consider certain $(k - 1)$ -merging patterns derived from a given k -merging pattern $\pi = (a_k, b_k), (a_{k-1}, b_{k-1}), \dots, (a_2, b_2)$. We let $\pi' = (a_{k-1}, b_{k-1}), (a_{k-2}, b_{k-2}), \dots, (a_2, b_2)$ be the $(k - 1)$ -merging pattern obtained from π by deleting the first merging pair (a_k, b_k) from π . Thus π acts on any $k \times 1$ matrix M as above, while π' acts on any such $(k - 1) \times 1$ matrix M' , independent of M .

Recall that for each $1 \leq t \leq k$, $M(t)$ is the $t \times 1$ matrix obtained from M by applying the first $k - t$ merging pairs $(a_j, b_j), k - t + 1 \leq j \leq k$, in succession (i.e., in decreasing j) to M . There must be an integer $i, 2 \leq i \leq k$, such that the pair (a_i, b_i) satisfies $b_i = i$. That is, (a_i, b_i) merges the last entry of $M(i)$ with some other entry of $M(i)$. Such a pair $(a_i, b_i) = (a_i, i)$ is the $(i - 1)$ st merging pair from the right in π . Let i_π be the maximum such integer i . Thus (a_{i_π}, b_{i_π}) is the first pair in π (reading these pairs from left to right) which merges the original last entry of M with some other entry in the process of executing π . As an example, the 6-merging pattern $\pi = (4, 5)(2, 4)(3, 4)(1, 3)(1, 2)$ has $i_\pi = 4$ and $(a_{i_\pi}, b_{i_\pi}) = (3, 4)$. Now we let π'' be the merging pattern obtained from π by deleting the pair (a_{i_π}, b_{i_π}) from π , so here we have $\pi'' = (4, 5)(2, 4)(1, 3)(1, 2)$. In general π'' is a $(k - 1)$ -merging pattern which can be applied to any $(k - 1) \times 1$ matrix M' independent of M . For example, note that the index k will not appear in a merging pair of π'' . Suppose to the contrary that a pair of π'' contains the index k . Then since every pair of π'' is also a pair of π , a pair of π'' containing index k would imply that $i_\pi = k$. Hence that pair would be deleted in obtaining π'' , a contradiction. So k could not appear in π'' .

We note a simple fact used in both this section and section 5, omitting its straightforward proof.

PROPOSITION 3.1. *Let $k \geq 2$ be a positive integer. Let M, M^* be two $k \times 1$ matrices from $\{0, 1, *\}$. Suppose M^* is obtained from M by permuting its rows. Then $\mathbb{E}_k(M^*) = \mathbb{E}_k(M)$.*

The next lemma will be used in inductive arguments for estimating expectations.

LEMMA 3.2. *Let $\pi = (a_k, b_k), (a_{k-1}, b_{k-1}), \dots, (a_2, b_2)$ be a k -merging pattern with $k \geq 2$, and let M be a $k \times 1$ matrix with entries from $\{0, 1, *\}$.*

(a) *Let π'' be (as above) the $(k - 1)$ -merging pattern obtained from π by deleting the pair (a_{i_π}, b_{i_π}) . Then the map $f : \Omega_k \rightarrow \Omega_{k-1}$ given by $f(\pi) = \pi''$ is surjective and satisfies $|f^{-1}(\pi'')| = \binom{k}{2}$ for all $\pi'' \in \Omega_{k-1}$.*

(b) *If M contains a $*$ -entry, then $\mathbb{E}_k(M) = \mathbb{E}_{k-1}(M')$, where M' is the $(k - 1) \times 1$ matrix obtained from M by deleting that $*$ -entry.*

Proof.

(a) For each $i, 2 \leq i \leq k$, there are $i - 1$ possible $\pi \in f^{-1}(\pi'')$ for which $i_\pi = i$, these consisting of any π in which $(a_{i_\pi}, b_{i_\pi}) = (a_{i_\pi}, i)$, where a_{i_π} is any integer satisfying $1 \leq a_{i_\pi} \leq i - 1$. Now summing over all $2 \leq i \leq k$, we get $|f^{-1}(\pi'')| = \sum_{i=2}^k (i - 1) = \binom{k}{2}$, as claimed.

(b) By Proposition 3.1, we may assume without loss of generality that the last entry of M is $*$. So we have $cost_M(i_\pi) = 0$, and the (a_{i_π}, b_{i_π}) -merge of $M(i_\pi)$ leaves the entry a_{i_π} unchanged. Also, each remaining merging pair in π (acting on M) merges the same row pair containing the same entry pair as its corresponding merging pair in π'' (acting on M'). It follows that $cost(M, \pi) = cost(M', \pi'')$ for all $\pi \in f^{-1}(\pi'')$.

The expectations can now be computed as follows:

$$\begin{aligned}
 \mathbb{E}_k(M) &= \frac{1}{\prod_{t=2}^k \binom{t}{2}} \sum_{\pi \in \Omega_k} \text{cost}(M, \pi) \\
 &= \frac{1}{\prod_{t=2}^k \binom{t}{2}} \sum_{\pi'' \in \Omega_{k-1}} \sum_{\pi \in f^{-1}(\pi'')} \text{cost}(M, \pi) \\
 &= \frac{1}{\prod_{t=2}^k \binom{t}{2}} \sum_{\pi'' \in \Omega_{k-1}} \binom{k}{2} \text{cost}(M', \pi'') \\
 &= \frac{1}{\prod_{t=2}^{k-1} \binom{t}{2}} \sum_{\pi'' \in \Omega_{k-1}} \text{cost}(M', \pi'') = \mathbb{E}_{k-1}(M'). \quad \square
 \end{aligned}$$

The sequence $e_k = \frac{k}{3} + \frac{4}{9}(1 - (-\frac{1}{2})^k)$ plays an important role in our subsequent arguments. We mention without proof some properties of this sequence in the following lemma.

LEMMA 3.3. *The sequence $e_k = \frac{k}{3} + \frac{4}{9}(1 - (-\frac{1}{2})^k)$ is the solution to the recurrence $e_k = \frac{1}{2}(e_{k-1} + e_{k-2} + 1)$ with initial conditions $e_0 = 0, e_1 = 1$. Further, it satisfies $e_{k-1} \leq e_k \leq e_{k-1} + 1$ and $e_k + \frac{\ln k}{2} \leq e_{k-1} + \frac{\ln(k-1)}{2} + 1$ for each integer $k \geq 2$.*

The next lemma proves that for a $k \times 1$ matrix M , the expected cost over all merging patterns $\mathbb{E}_k(M)$ is asymptotically $e_k \approx \frac{1}{3}k$. We first give some intuition as to why this should be so, though the reader may choose to go directly to that lemma for the precise statement and proof. Now, intuition suggests that $\mathbb{E}_k(M)$ is maximized when M has an equal number (up to a difference of 1) of 0's and 1's and no *'s. In this case, the first merging step will either merge two opposite entries, i.e., one of them 0 and the other 1 (call this Case 1) with probability roughly $\frac{1}{2} + o(1)$, or merge two equal entries, both 0 or both 1 (call this Case 2), also with probability roughly $\frac{1}{2} + o(1)$. In Case 1 a cost of 1 is incurred, and by Lemma 3.2 the * arising from the merge can be deleted without affecting the expected cost. So here we incur a cost of 1 and reduce the number of entries by 2. In Case 2 we incur a cost of 0 and reduce the number of entries by 1. So we reduce the number of entries in M by an average of $\frac{3}{2}$ and incur an average cost of $\frac{1}{2} \cdot 0 + \frac{1}{2} \cdot 1 = \frac{1}{2}$. Then we define a *step* to be either a Case 1 merge followed by deleting a *, or a Case 2 merge alone. Since the average cost per step is upper bounded when a given step begins with a balanced number of 0's and 1's, we might guess that $\mathbb{E}_k(M)$ is upper bounded by the sum of average costs of a succession of such steps. With this assumption it takes an average of $\frac{2}{3}k$ steps to complete the merging pattern since we reduce the number of entries by an average of $\frac{3}{2}$ per step. So the total average cost is roughly $\frac{1}{2} \cdot \frac{2}{3}k = \frac{1}{3}k$. Of course multiplying these two averages to get the total expected cost is not in general valid, but these ideas can still be formed into a proof in the following lemma.

LEMMA 3.4. *Let $k \geq 2$ be a positive integer and M a $k \times 1$ matrix with entries from $\{0, 1, *\}$. Then $\mathbb{E}_k(M) \leq e_k + \frac{\ln k}{2}$, where e_k is defined in Lemma 3.3.*

Proof. We use induction on k . When $k = 2$, we need to prove $\mathbb{E}_2(M) \leq e_2 + \frac{\ln 2}{2} = 1 + \frac{\ln 2}{2}$, which holds trivially. Assume that $k \geq 3$ and that the claim holds for values smaller than k . Let $\pi = (a_k, b_k), (a_{k-1}, b_{k-1}), \dots, (a_2, b_2)$ be a random merging pattern from (Ω_k, \mathbb{P}_k) . Let π' be the subpattern $(a_{k-1}, b_{k-1}), \dots, (a_2, b_2)$. Suppose M has s many 1's and t many 0's. Then $s + t = k$ and $st = s(k - s) \leq \frac{k^2}{4}$. Let $M' = M(k - 1)$ denote the (a_k, b_k) -merge of M .

Let B be the event that the first step (a_k, b_k) of π merges a 1 with a 0. We have

$\mathbb{P}(B) = st/\binom{k}{2} \leq \frac{k^2/4}{k(k-1)/2} = \frac{1}{2} + \frac{1}{2(k-1)}$. Consider $\mathbb{E}_k(M|B)$. Given B , the first merge has a cost of 1, and M' contains a $*$ that results from the merge. Let M'' be obtained from M' by deleting that $*$ -entry. By Lemma 3.2 and the induction hypothesis we have

$$\mathbb{E}_k(M|B) = 1 + \mathbb{E}_{k-1}(M') = 1 + \mathbb{E}_{k-2}(M'') \leq 1 + e_{k-2} + \frac{\ln(k-2)}{2}.$$

Given \overline{B} , the first step has 0 cost. Each element of Ω_{k-1} is equally likely for π' . Thus,

$$\mathbb{E}_k(M|\overline{B}) = \mathbb{E}_{k-1}(M') \leq e_{k-1} + \frac{\ln(k-1)}{2}.$$

Hence

$$\begin{aligned} \mathbb{E}_k(M) &= \mathbb{P}(B) \cdot \mathbb{E}_k(M|B) + \mathbb{P}(\overline{B}) \cdot \mathbb{E}_k(M|\overline{B}) \\ &\leq \mathbb{P}(B) \left[1 + e_{k-2} + \frac{1}{2} \ln(k-2) \right] + \mathbb{P}(\overline{B}) \left[e_{k-1} + \frac{1}{2} \ln(k-1) \right]. \end{aligned}$$

By Lemma 3.3, in the last expression above, the coefficient of $\mathbb{P}(B)$ is at least as large as the coefficient of $\mathbb{P}(\overline{B}) = 1 - \mathbb{P}(B)$. So the expression is increasing in $\mathbb{P}(B)$. Since $\mathbb{P}(B) \leq \frac{1}{2} + \frac{1}{2(k-1)}$, we have

$$\begin{aligned} \mathbb{E}_k(M) &\leq \left(\frac{1}{2} + \frac{1}{2(k-1)} \right) \left[1 + e_{k-2} + \frac{1}{2} \ln(k-2) \right] \\ &\quad + \left(\frac{1}{2} - \frac{1}{2(k-1)} \right) \left[e_{k-1} + \frac{1}{2} \ln(k-1) \right] \\ &\leq \frac{1}{2} \left[1 + e_{k-2} + e_{k-1} + \frac{1}{2} \ln(k-2) + \frac{1}{2} \ln(k-1) \right] \\ &\quad + \frac{1}{2(k-1)} \quad (\text{using } e_{k-2} \leq d_{k-1}) \\ &= e_k + \frac{1}{2} \left[\frac{1}{2} \ln(k-2) + \frac{1}{2} \ln(k-1) + \frac{1}{k-1} \right] \leq e_k + \frac{1}{2} \ln k. \end{aligned}$$

To see why the last inequality holds, note that $\ln(1-x) < -x$ for $|x| < 1$. So $\frac{1}{2} \ln(k-2) + \frac{1}{2} \ln(k-1) = \ln k + \frac{1}{2} \ln(1 - \frac{2}{k}) + \frac{1}{2} \ln(1 - \frac{1}{k}) \leq \ln k + \frac{1}{2}(-\frac{2}{k} - \frac{1}{k}) = \ln k - \frac{3}{2k} \leq \ln k - \frac{1}{k-1}$ for $k \geq 3$. This completes the proof. \square

Now, we are ready to give our general upper bound on $\text{cost}(\mathcal{F}_k)$.

THEOREM 3.5. *Let $n, k \geq 2$ be integers. Let \mathcal{F}_k be a family of k subcubes in Q_n . Then $\text{cost}(\mathcal{F}_k) \leq (e_k + \frac{\ln k}{2})n < (\frac{1}{3}k + 1 + \frac{1}{2} \ln k)n$.*

Proof. View \mathcal{F}_k as a $k \times n$ matrix with rows representing members of \mathcal{F}_k . For each $j \in [n]$, let \mathcal{F}_k^j denote the j th column. Over all merging patterns π in (Ω_k, \mathbb{P}_k) , we have $\mathbb{E}_k(\mathcal{F}_k) = \sum_{j=1}^n \mathbb{E}_k(\mathcal{F}_k^j)$. Applying Lemma 3.4 to each $\mathbb{E}_k(\mathcal{F}_k^j)$, we get $\mathbb{E}_k(\mathcal{F}_k) \leq (e_k + \frac{\ln k}{2})n$. Hence, in particular, there exists a merging pattern π such that $\text{cost}(\mathcal{F}_k, \pi) \leq (e_k + \frac{\ln k}{2})n$. By Proposition 2.3, $\text{cost}(\mathcal{F}_k) \leq \text{cost}(\mathcal{F}_k, \pi)$ and the theorem follows. \square

Theorem 3.5 immediately yields the following.

COROLLARY 3.6. *Let $n, k \geq 2$ be integers. Let S be a set of k vertices in Q_n . Then $\text{cost}(S) \leq (e_k + \frac{\ln k}{2})n < (\frac{1}{3}k + 1 + \frac{1}{2} \ln k)n$.*

4. Lower bound and a result on the random family. Consider any small $\epsilon > 0$. In this section, we show that if one randomly and independently selects (with repetition allowed) a multiset S of k points in Q_n and k is of moderate size as a function of n , then almost always $\text{cost}(S)$ is at least $(\frac{1}{3} - \epsilon)kn$. Combining this with our previous upper bounds, this shows $\text{cost}(S)/kn$ is approximately $\frac{1}{3}$ for such k and n . To establish this fact, we need the following notion.

DEFINITION 4.1. Let r be a positive integer and A an $r \times n$ $(0, 1)$ -matrix. For each $0, 1$ -vector \mathbf{c} of length r , let $N_{\mathbf{c}}(A)$ denote the number of columns of A that match \mathbf{c} . Given a small $\epsilon > 0$, we say that A is ϵ -good if for each $0, 1$ -vector \mathbf{c} of length r , $|N_{\mathbf{c}}(A) - \frac{n}{2^r}| \leq \epsilon \frac{n}{2^r}$. A set S of r distinct vertices in Q_n is ϵ -good if the $r \times n$ matrix whose rows are the $(0, 1)$ -strings corresponding to the vertices of S (in any order) is ϵ -good. We note here that if an $r \times n$ binary matrix A is ϵ -good, then its rows must be distinct. If not, consider two rows of A which are the same. If \mathbf{c} is an $r \times 1$ binary column vector having opposite values in these two rows, then since any column of A has identical values in these two rows, it follows that \mathbf{c} cannot appear as a column of A . Thus $N_{\mathbf{c}}(A) = 0$, which violates the inequality defining ϵ -good for $\epsilon < 1$. So any ϵ -good binary matrix must have distinct rows.

A $k \times n$ matrix M with $k \geq q$ is called (q, ϵ) -uniform if every $r \times n$ submatrix of M is ϵ -good for each $r \in [q, 2q - 2]$. The rows of such a matrix must also be distinct, since any two of them are contained in an ϵ -good submatrix on q rows, so these two rows are distinct by the discussion above. A family S of k vertices in Q_n is (q, ϵ) -uniform if the $k \times n$ matrix with vertices of S as row vectors (in any order) is (q, ϵ) -uniform. In particular the vertices of such a family are distinct.

Consider a large n and suitably chosen values of k and q . We now show that when a set S of k vertices is selected randomly and independently with repetition allowed from $V(Q_n)$, this multiset will almost always be (q, ϵ) -uniform. Note that if a set S resulting from this selection is (q, ϵ) -uniform, then the vertices of S are actually distinct as just observed. We need the well-known Chernoff bound, using the simplified version given in [22].

LEMMA 4.2 (the Chernoff inequality). Let X_i be independent, identically distributed random variables, where $X_i = 1$ with probability p and $X_i = 0$ with probability $1 - p$. Consider the random variable $X = \sum_{i=1}^n X_i$. Then whenever $0 \leq t \leq np$,

$$\mathbb{P}(|X - np| > t) < 2e^{-\frac{t^2}{3np}}.$$

For simplicity of presentation, our estimates are sometimes quite rough.

THEOREM 4.3. Let q, k , and n be positive integers, and let $\epsilon > 0$ be given. Let $B = e^{\frac{2\epsilon^2}{3q \cdot 2^{2q-1}}}$ and $\alpha = \frac{1}{\epsilon} q^{1 - \frac{1}{2q}}$. If $k < \alpha B^n$, then there exists a (q, ϵ) -uniform family S of k vertices of Q_n . If $k/(\alpha B^n) \rightarrow 0$, then as $n \rightarrow \infty$ almost always a family S of k vertices randomly and independently selected with repetition allowed from $V(Q_n)$ is (q, ϵ) -uniform.

Proof. Randomly and independently select (with repetition allowed) a set S of k vertices from Q_n . Let M be the associated $k \times n$ matrix with vertices of S forming the row vectors. We may think of M as being generated by independently assigning a 0 or 1 to each entry with probability $\frac{1}{2}$. Now, fix an integer r , with $q \leq r \leq 2q - 2$ and an $r \times n$ submatrix A of M . For each $j \in [n]$, let \mathbf{a}_j denote the j th column of A . Fix a vector $\mathbf{c} \in \{0, 1\}^r$. For each $j \in [n]$, let $X_j^{\mathbf{c}} = 1$ if $\mathbf{a}_j = \mathbf{c}$, and let $X_j^{\mathbf{c}} = 0$ otherwise. Then $N_{\mathbf{c}}(A) = \sum_{j=1}^n X_j^{\mathbf{c}}$. Now $X_1^{\mathbf{c}}, \dots, X_n^{\mathbf{c}}$ are independent, identically distributed random variables, where $X_j^{\mathbf{c}} = 1$ with probability $\frac{1}{2^r}$ and $X_j^{\mathbf{c}} = 0$ with probability

$1 - \frac{1}{2^r}$. Applying the Chernoff inequality with $p = \frac{1}{2^r}$ and $t = \epsilon \frac{n}{2^r}$, we have

$$(1) \quad \mathbb{P}\left(\left|N_{\mathbf{c}}(A) - \frac{n}{2^r}\right| > \epsilon \frac{n}{2^r}\right) \leq 2e^{-\frac{\epsilon^2 n}{3 \cdot 2^r}}.$$

Let Z_A denote the event that there exists some vector $\mathbf{c} \in \{0, 1\}^r$ in A such that $|N_{\mathbf{c}}(A) - \frac{n}{2^r}| > \epsilon \frac{n}{2^r}$. Then since there are 2^r choices for \mathbf{c} , by (1) we have $\mathbb{P}(Z_A) \leq 2^r \cdot 2e^{-\frac{\epsilon^2 n}{3 \cdot 2^r}}$. Let $Z = \bigcup_A Z_A$, where the union is taken over all $r \times n$ submatrices A of M , $q \leq r \leq 2q - 2$. Note that \bar{Z} is exactly the event that M is (q, ϵ) -uniform. Since for each $r \in [q, 2q - 2]$, there are $\binom{k}{r}$ many $r \times n$ submatrices, we have

$$(2) \quad \mathbb{P}(Z) \leq \sum_{r=q}^{2q-2} \binom{k}{r} 2^{r+1} e^{-\frac{\epsilon^2 n}{3 \cdot 2^r}} < q \binom{k}{2q} 2^{2q} e^{-\frac{\epsilon^2 n}{3 \cdot 2^{2q-2}}} < q \left(\frac{ke}{q}\right)^{2q} e^{-\frac{\epsilon^2 n}{3 \cdot 2^{2q-2}}}.$$

If $k < \alpha B^n = \frac{1}{\epsilon} q^{1 - \frac{1}{2q}} e^{\frac{2\epsilon^2 n}{3q \cdot 2^{2q-1}}}$, then the last quantity in (2) is less than 1 and

$$\mathbb{P}(M \text{ is } (q, \epsilon)\text{-uniform}) = \text{Prob}(\bar{Z}) > 0.$$

If $k/(\alpha B^n) \rightarrow 0$ as $n \rightarrow \infty$, then

$$\mathbb{P}(M \text{ is } (q, \epsilon)\text{-uniform}) = \text{Prob}(\bar{Z}) \rightarrow 1$$

as $n \rightarrow \infty$. In the former case, we can conclude that there exists at least one family S of k vertices that is (q, ϵ) -uniform. In the latter case, we conclude that almost always a family S of k vertices is (q, ϵ) -uniform. \square

LEMMA 4.4. *Consider an integer $q \geq 2$, a tree T , and $S \subseteq V(T)$. Then there exists a collection of edge-disjoint subtrees T_0, T_1, \dots, T_c of T such that $S \subseteq \bigcup_{i=0}^c V(T_i)$, $|V(T_0) \cap S| \leq q - 1$, and $q \leq |V(T_i) \cap S| \leq 2q - 2$ for each $i \in [c]$.*

Proof. The claim holds trivially if $|S| \leq q - 1$. So we assume that $|S| \geq q$. By Lemma 2.4, T has a subtree T_1 such that $q \leq |T_1 \cap S| \leq 2q - 2$ and that $T - E(T_1)$ has a component T' that contains all of $S - (T_1 \cap S)$. We repeat the argument with T being replaced by T' and S being replaced by $S - (V(T_1) \cap S)$ to find a subtree T_2 . Continue the process until the remaining nontrivial component has fewer than q vertices of S . Denote this final remaining nontrivial component by T_0 . \square

For each $I \subseteq [k]$, let $\mathcal{S}(I, k)$ denote the set of all $k \times 1$ $(0, 1, *)$ -matrices M , where the i th row M_i of M satisfies $M_i \in \{0, 1\}$ for all $i \in I$ and $M_i = *$ for all $i \in [k] \setminus I$. Note that $|\mathcal{S}(I, k)| = 2^{|I|}$. Let $\{d_k\}_{k=0}^\infty$ be a sequence defined by letting $d_0 = d_1 = 0$ and letting $d_k = \frac{1}{2}d_{k-1} + \frac{1}{2}(1 + d_{k-2})$ for all $k \geq 2$. (Note that the sequences $\{d_k\}$ and $\{e_k\}$ obey the same recurrence but have different initial conditions.) Using the particular solution $\frac{k}{3}$ to the recurrence, and the characteristic roots 1 and $-\frac{1}{2}$ to the corresponding homogeneous recurrence $d_k = \frac{1}{2}d_{k-1} + \frac{1}{2}d_{k-2}$, one can show that $d_k = \frac{k}{3} + \frac{2}{9}[-1 + (-\frac{1}{2})^k]$. The following lemma shows that the average merging cost of a fixed k -merge pattern π relative to M over all $k \times 1$ $(0, 1, *)$ -matrices $M \in \mathcal{S}(I, k)$ depends only on $|I|$ and is in fact equal to $d_{|I|}$.

LEMMA 4.5. *Let $k \geq 2$ be an integer. Let π be a k -merging pattern. Let $I \subseteq [k]$, and let $m = |I|$. We have $\frac{1}{2^m} \sum_{M \in \mathcal{S}(I, k)} \text{cost}(M, \pi) = d_m = \frac{m}{3} + \frac{2}{9}[-1 + (-\frac{1}{2})^m]$.*

Proof. We use induction on k . For the basis step, let $k = 2$. If $m = 0$ or 1, then each $M \in \mathcal{S}(I, 2)$ has only at most one entry that is not a $*$. We have $\text{cost}(M, \pi) = 0$

for each $M \in \mathcal{S}(I, 2)$. Thus their average is $0 = d_m$. If $m = 2$, then $\mathcal{S}(I, 2)$ consists of all 2×1 $(0, 1)$ -matrices, and the average of $\text{cost}(M, \pi)$ over all such M is $\frac{1}{2} = d_2$. This completes the basis step.

For the induction step, let $k \geq 3$ and suppose the claim has been verified for smaller values of k . Suppose $\pi = (a_k, b_k), (a_{k-1}, b_{k-1}), \dots, (a_2, b_2)$. Let $\pi' = (a_{k-1}, b_{k-1}), \dots, (a_2, b_2)$. We consider two cases.

Case 1. $a_k, b_k \in I$.

In this case, for 2^{m-1} of the M 's, we have $M_{a_k} = M_{b_k}$, and the resulting set of a_k, b_k -merges of these M 's, each denoted by M' , is exactly the set $\mathcal{S}(I', k - 1)$, where $I' = I \setminus \{b_k\}$. The average of $\text{cost}(M', \pi')$ over all such M' is d_{m-1} by the induction hypothesis. Since the a_k, b_k -merge itself has 0 cost in this case, the average of $\text{cost}(M, \pi)$ over such M is d_{m-1} .

For the other 2^{m-1} of the M 's, $\{M_{a_k}, M_{b_k}\} = \{0, 1\}$, and the resulting a_k, b_k -merges M' of these M constitute the set $\mathcal{S}(I'', k - 1)$, where $I'' = I \setminus \{a_k, b_k\}$. By the induction hypothesis, the average of $\text{cost}(M', \pi')$ over all these M' is d_{m-2} . Since the cost of first step of π is 1, the average of $\text{cost}(M, \pi)$ over all such M is $1 + d_{m-2}$.

Combining considerations of these two types of members described above, we see that $\sum_{M \in \mathcal{S}(I, k)} \text{cost}(M, \pi) = \frac{1}{2}d_{m-1} + \frac{1}{2}(1 + d_{m-2}) = d_m$.

Case 2. Either $a_k \notin I$ or $b_k \notin I$.

In this case, the merging cost of the first step of π is 0. Furthermore, the resulting a_k, b_k -merge M' of M over all $M \in \mathcal{S}(I, k)$ forms the set $\mathcal{S}(I', k - 1)$, where $I' = I$ if $b_k \notin I$ and $I' = (I \setminus \{b_k\}) \cup \{a_k\}$ if $a_k \notin I, b_k \in I$. By the induction hypothesis, the average of $\text{cost}(M', \pi')$ over all such M' is d_m , and so the average of $\text{cost}(M, \pi)$ over all such M is d_m .

This completes the induction step and the proof. □

Applying Lemma 4.5 to $\mathcal{A}_r = \mathcal{S}([r], r)$, the set of all $r \times 1$ $(0, 1)$ -matrices, we get the following corollary.

COROLLARY 4.6. *Let $r \geq 2$ be an integer. Let π be an r -merging pattern. Let \mathcal{A}_r be the family of all $r \times 1$ $(0, 1)$ -matrices. Then $\frac{1}{2^r} \sum_{A \in \mathcal{A}_r} \text{cost}(A, \pi) = d_r = \frac{r}{3} + \frac{2}{9}[-1 + (-\frac{1}{2})^r]$.*

We now use the sequence d_k to show that every ϵ -good family of vertices has high cost.

LEMMA 4.7. *Let ϵ be small and positive. Let r, n be positive integers where $r \geq \frac{1}{2\epsilon} - \frac{1}{2}$. Let S be an ϵ -good family of r distinct vertices in Q_n . Then $\text{cost}(S) \geq (\frac{1}{3} - \epsilon)nr$.*

Proof. Viewing S as an $r \times n$ matrix A , by Lemma 2.3 there exists an r -merging pattern π such that $\text{cost}(S) = \text{cost}(A, \pi)$. Again let \mathcal{A}_r denote the set of all $r \times 1$ $(0, 1)$ -matrices. For each $\mathbf{c} \in \mathcal{A}_r$, recall that $N_{\mathbf{c}}(A)$ is the number of columns of A that match \mathbf{c} . Since S is ϵ -good, we have $N_{\mathbf{c}}(A) \geq (1 - \epsilon)\frac{n}{2^r}$ for each of these 2^r vectors \mathbf{c} . Applying Corollary 4.6 we have

$$\begin{aligned} \text{cost}(A, \pi) &= \sum_{\mathbf{c} \in \mathcal{A}_r} N_{\mathbf{c}}(A) \text{cost}(\mathbf{c}, \pi) \geq \sum_{\mathbf{c} \in \mathcal{A}_r} \frac{n}{2^r} (1 - \epsilon) \text{cost}(\mathbf{c}, \pi) \\ &= \frac{n}{2^r} (1 - \epsilon) \sum_{\mathbf{c} \in \mathcal{A}_r} \text{cost}(\mathbf{c}, \pi) = n(1 - \epsilon)d_r. \end{aligned}$$

Applying the formula for d_r , it suffices to show that under our assumptions on r and ϵ , we have

$$(1 - \epsilon) \left(1 - \frac{2}{3r} + \frac{2}{3r} \left(-\frac{1}{2} \right)^r \right) \geq 1 - 3\epsilon.$$

Some manipulation shows that this is equivalent to $\frac{2}{3^r}(1 - (-\frac{1}{2})^r) \leq \frac{2\epsilon}{1-\epsilon}$. Since $1 - (-\frac{1}{2})^r \leq \frac{3}{2}$ for positive integers r , it then suffices to take $r \geq \frac{1}{2\epsilon} - \frac{1}{2}$. \square

THEOREM 4.8. *Let ϵ be small and positive. Let $q = \lceil \frac{1}{\epsilon} \rceil + 1$. Let constants α and B (depending on ϵ and q) be as in Lemma 4.3. For all k satisfying $q^2 < k < \alpha B^n$ there is a family S of k vertices in Q_n with $\text{cost}(S) \geq (\frac{1}{3} - \epsilon)nk$. Furthermore, if $k = o(\alpha B^n)$ (as a function of n), then as $n \rightarrow \infty$ almost always a family S of k vertices in Q_n randomly and independently selected with repetition allowed satisfies $\text{cost}(S) \geq (\frac{1}{3} - \epsilon)nk$.*

Proof. By Lemma 4.3 there exists a family S of k vertices in Q_n that is (q, ϵ) -uniform. Furthermore, if $k = o(\alpha B^n)$, then as $n \rightarrow \infty$, almost always a random family S of k vertices in Q_n randomly and independently selected with repetition allowed is (q, ϵ) -uniform.

Let T be a Steiner tree for such a set S of vertices in Q_n , so that $|E(T)| = \text{cost}(S)$. By Lemma 4.4 we can find edge disjoint subtrees T_0, T_1, \dots, T_c of T such that $S \subseteq \bigcup_{i=0}^c V(T_i)$, $|V(T_0) \cap S| \leq q - 1$, and $q \leq |V(T_i) \cap S| \leq 2q - 2$ for $1 \leq i \leq c$. Let $S_i = V(T_i) \cap S$. Since S is (q, ϵ) -uniform, S_i is ϵ -good. Since $|S_i| \geq q > \frac{1}{\epsilon}$, by Lemma 4.7, we have $\text{cost}(S_i) \geq (\frac{1}{3} - \frac{\epsilon}{2})n|S_i|$ for $1 \leq i \leq c$. For each i , since T_i connects S_i , $|E(T_i)| \geq \text{cost}(S_i)$. Hence, letting $r_0 = |V(T_0) \cap S|$, we have

$$\text{cost}(S) = |E(T)| \geq \sum_{i=0}^c |E(T_i)| \geq \sum_{i=1}^c \text{cost}(S_i) \geq \left(\frac{1}{3} - \frac{\epsilon}{2}\right)(k - r_0)n.$$

So it remains to show that $(\frac{1}{3} - \frac{\epsilon}{2})(k - r_0)n \geq (\frac{1}{3} - \epsilon)kn$, and this is equivalent to $\frac{\epsilon}{2}k \geq (\frac{1}{3} - \frac{\epsilon}{2})r_0$. So it suffices to prove $\frac{\epsilon}{2}k \geq \frac{1}{3}r_0$. But since $q > \frac{1}{\epsilon}$ and $k > q^2$ we have $\frac{\epsilon}{2}k \geq \frac{1}{2}q \geq \frac{1}{3}r_0$, where the last inequality follows from $r_0 \leq q - 1 < \frac{3}{2}q$. \square

To close this section, we note that when k is a fixed constant, we have the following even tighter estimate.

THEOREM 4.9. *Let n, k be positive integers. When k is fixed and $n \rightarrow \infty$, almost always a family S of k vertices in Q_n randomly and independently selected with repetition allowed satisfies*

$$\left[\frac{k}{3} + \frac{2}{9} \left(-1 + \left(-\frac{1}{2} \right)^k \right) \right] n - \sqrt{n \ln n} \leq \text{cost}(S) \leq \left[\frac{k}{3} + \frac{2}{9} \left(-1 + \left(-\frac{1}{2} \right)^k \right) \right] n + \sqrt{n \ln n}.$$

Proof. Randomly and independently select a set S of k points from Q_n with repetition allowed. As discussed earlier in the section, the corresponding $k \times n$ $(0, 1)$ -matrix whose rows are the strings representing the points of S can be viewed as generated by taking a random $k \times n$ $(0, 1)$ -matrix A , where each entry is assigned a 0 or a 1 independently with probability $\frac{1}{2}$. For each $\mathbf{c} \in \{0, 1\}^k$, let $N_{\mathbf{c}}(A)$ denote the number of columns of A that match \mathbf{c} . As in the proof of Theorem 4.3, by the Chernoff inequality, for each \mathbf{c} we have

$$\mathbb{P} \left(\left| N_{\mathbf{c}}(A) - \frac{n}{2^k} \right| > \sqrt{n} \cdot (\ln n)^{1/3} \right) \leq 2e^{-\frac{2^k}{3}(\ln n)^{2/3}}.$$

So, the probability that there exist $\mathbf{c} \in \{0, 1\}^k$, $|N_{\mathbf{c}}(A) - \frac{n}{2^k}| > \sqrt{n} \cdot (\ln n)^{1/3}$ is at most $2^k \cdot 2e^{-\frac{2^k}{3}(\ln n)^{2/3}}$, which tends to 0 as $n \rightarrow \infty$ for fixed k . So, almost always the matrix A associated with S satisfies that $|N_{\mathbf{c}}(A) - \frac{n}{2^k}| \leq \sqrt{n}(\ln n)^{1/3}$ for each $\mathbf{c} \in \{0, 1\}^k$. Consider any such S . By Lemma 2.3 there exists a k -merging pattern π such that $\text{cost}(S) = \text{cost}(A, \pi)$. Let A_1, \dots, A_n denote the columns of A . We have

$cost(A, \pi) = \sum_{i=1}^n cost(A_i, \pi)$. Since, for each $\mathbf{c} \in \{0, 1\}^k$, the number of columns of A that match \mathbf{c} is between $\frac{n}{2^k} - \sqrt{n}(\ln n)^{1/3}$ and $\frac{n}{2^k} + \sqrt{n}(\ln n)^{1/3}$, we have

$$\begin{aligned} & \sum_{\mathbf{c} \in \{0,1\}^k} cost(\mathbf{c}, \pi) \left(\frac{n}{2^k} - \sqrt{n}(\ln n)^{1/3} \right) \\ & \leq cost(A, \pi) \leq \sum_{\mathbf{c} \in \{0,1\}^k} cost(\mathbf{c}, \pi) \left(\frac{n}{2^k} + \sqrt{n}(\ln n)^{1/3} \right). \end{aligned}$$

By Corollary 4.6, $\sum_{\mathbf{c} \in \{0,1\}^k} (cost(\mathbf{c}, \pi) \cdot \frac{n}{2^k}) = d_k n = \left[\frac{k}{3} + \frac{2}{9}(-1 + (-\frac{1}{2})^k) \right] n$. Also, note that $\sum_{\mathbf{c} \in \{0,1\}^k} cost(\mathbf{c}, \pi) \leq k2^k$ and hence $\sum_{\mathbf{c} \in \{0,1\}^k} (cost(\mathbf{c}, \pi) \cdot \sqrt{n}(\ln n)^{1/3}) \leq k2^k \cdot \sqrt{n}(\ln n)^{1/3} \leq \sqrt{n \ln n}$ for large n . This yields

$$\begin{aligned} & \left[\frac{k}{3} + \frac{2}{9} \left(-1 + \left(-\frac{1}{2} \right)^k \right) \right] n - \sqrt{n \ln n} \\ & \leq cost(S) \leq \left[\frac{k}{3} + \frac{2}{9} \left(-1 + \left(-\frac{1}{2} \right)^k \right) \right] n + \sqrt{n \ln n}. \quad \square \end{aligned}$$

Remark 4.10. In Theorem 4.9, we let k be a constant. Naturally, at the cost of a larger error term, we may relax the condition on k as far as the Chernoff inequality allows, as we did earlier in the section. Theorem 4.9 is interesting in its own right, since it tells us that for fixed k and $n \rightarrow \infty$, the value of $cost(S)$ is almost surely $\left[\frac{k}{3} + \frac{2}{9}(-1 + (-\frac{1}{2})^k) \right] n + o(n)$. For example, if one randomly selects a set S of 3, 4, or 5 vertices in Q_n , then $cost(S)$ is almost surely $\frac{3}{4}n + o(n)$, $\frac{9}{8}n + o(n)$, or $\frac{23}{16}n + o(n)$, respectively. This can be compared to the exact values of the maximum of $cost(S)$ (see [20]), which for $k = 3, 4, \text{ or } 5$ vertices are $n, \lfloor \frac{5}{3}n \rfloor$, or $2n - \lceil \frac{n}{10} \rceil - \lceil \frac{n-4}{10} \rceil$, respectively. While there is a gap in the coefficient of n between the almost sure value and the exact maximum for these small values of k , this gap disappears as $k \rightarrow \infty$.

5. A high concentration result on merging patterns. In this section we prove that the cost function relative to a random merging pattern is highly concentrated around its expected value. Recall that for each $k \geq 2$, we let Ω_k be the set of all $\prod_{i=2}^k \binom{t}{2}$ many k -merging patterns $\pi = (a_k, b_k), \dots, (a_2, b_2)$. We define \mathbb{P}_k to be the uniform probability distribution on Ω_k . Given a fixed $k \times n$ matrix M with entries from $\{0, 1, *\}$, as in section 3, we continue to let $\mathbb{E}_k(M)$ denote the expected value of $cost(M, \pi)$ over all π in (Ω_k, \mathbb{P}_k) , and for an event A in (Ω_k, \mathbb{P}_k) , we continue with the notation $\mathbb{E}_k(M|A)$ for the conditional expectation $\mathbb{E}(cost(M, \pi)|A)$. The following lemma is related to Lemma 3.2.

LEMMA 5.1. *Let $k \geq 2$ be an integer. Let M be a $k \times 1$ matrix with entries from $\{0, 1, *\}$. Let M' be the $(k - 1) \times 1$ matrix obtained from M by deleting some entry from M . Then $\mathbb{E}_{k-1}(M') \leq \mathbb{E}_k(M) \leq \mathbb{E}_{k-1}(M') + 1$.*

Proof. We use induction on k . The claim is trivial when $k = 2$. Let $k \geq 3$. If M contains a $*$, then we can use Lemma 3.2 to eliminate a $*$ from each of M and M' and apply the induction hypothesis to the shortened M and M' . Hence we may assume that M and M' contain no $*$.

By Proposition 3.1, we may assume that the entry w of row k is deleted. Let $\pi = (a_k, b_k), (a_{k-1}, b_{k-1}), \dots, (a_2, b_2)$ be a random k -merging pattern. Let A denote the event that the first step (a_k, b_k) does not involve row k (i.e., $k \neq b_k$), so \bar{A} is the event that (a_k, b_k) involves row k (i.e., $k = b_k$). First, consider $\mathbb{E}_k(M|A)$. For each

$1 \leq i < j \leq k - 1$, let $M^{(ij)}$ be the (i, j) -merge of M , and let $(M^{(ij)})'$ be obtained from $M^{(ij)}$ by deleting its last entry w in row k . For each $i \in [k]$, let M_i (resp., M'_i) denote the i th row of M (resp., M'). We have

$$\begin{aligned} \mathbb{E}_k(M|A) &= \sum_{1 \leq i < j \leq k-1} \frac{1}{\binom{k-1}{2}} \cdot (\text{dist}(M_i, M_j) + \mathbb{E}_{k-1}(M^{(ij)})). \\ \mathbb{E}_{k-1}(M') &= \sum_{1 \leq i < j \leq k-1} \frac{1}{\binom{k-1}{2}} \cdot (\text{dist}(M'_i, M'_j) + \mathbb{E}_{k-2}[(M^{(ij)})']). \end{aligned}$$

For all $1 \leq i < j \leq k - 1$, we have $\text{dist}(M_i, M_j) = \text{dist}(M'_i, M'_j)$, and by the induction hypothesis, $\mathbb{E}_{k-2}[(M^{(ij)})'] \leq \mathbb{E}_{k-1}(M^{(ij)}) \leq \mathbb{E}_{k-2}[(M^{(ij)})'] + 1$. Thus, comparing the two equations above yields $\mathbb{E}_{k-1}(M') \leq \mathbb{E}_k(M|A) \leq \mathbb{E}_{k-1}(M') + 1$.

Next, consider $\mathbb{E}_k(M|\bar{A})$. We have

$$(3) \quad \mathbb{E}_k(M|\bar{A}) = \sum_{i=1}^{k-1} \frac{1}{k-1} (\text{dist}(M_i, w) + \mathbb{E}_{k-1}(M^{(ik)})).$$

Let $i \in [k - 1]$. If $M_i = w$, then $\text{dist}(M_i, w) = 0$ and $M^{(ik)} = M'$, and the i th summand above is $\frac{1}{k-1} \mathbb{E}_{k-1}(M')$. If M_i and w have opposite values, then $\text{dist}(M_i, w) = 1$ and $M_i \wedge w = *$. Let M'' be obtained from $M^{(ik)}$ by deleting this resulting $*$. Then M'' is a $(k - 2) \times 1$ matrix obtained from M' by deleting its row i . By Lemma 3.2, $\mathbb{E}_{k-1}(M^{(ik)}) = \mathbb{E}_{k-2}(M'')$. By our induction hypothesis, $\mathbb{E}_{k-1}(M') - 1 \leq \mathbb{E}_{k-2}(M'') \leq \mathbb{E}_{k-1}(M')$. Hence adding $1 = \text{dist}(M_i, w)$ in both of the preceding inequalities, we see that the i th summand in (3) is between $\frac{1}{k-1} \mathbb{E}_{k-1}(M')$ and $\frac{1}{k-1} (\mathbb{E}_{k-1}(M') + 1)$. By our arguments so far, $\mathbb{E}_{k-1}(M') \leq \mathbb{E}_k(M|\bar{A}) \leq \mathbb{E}_{k-1}(M') + 1$.

Combining our discussions above, we have $\mathbb{E}_{k-1}(M') \leq \mathbb{E}_k(M) \leq \mathbb{E}_{k-1}(M') + 1$, completing the proof. \square

Given a k -merging pattern $\pi = (a_k, b_k), (a_{k-1}, b_{k-1}), \dots, (a_2, b_2)$, for $j = 2, \dots, k - 1$, we let $\pi(j)$ denote the merging pair (a_j, b_j) of π , so that $\pi = \pi(k), \pi(k-1), \dots, \pi(2)$. Now, fixing a $k \times n$ matrix M with entries from $\{0, 1, *\}$, we define a random variable X on Ω_k by letting $X(\pi) = \text{cost}(M, \pi)$ for each $\pi \in \Omega_k$. We show that X is highly concentrated about its expected value $\mathbb{E}(X) = \mathbb{E}_k(M)$. We do this by using a martingale X_0, X_1, \dots, X_{k-1} associated with X , defined as follows. For each $\pi \in \Omega_k$, let $X_0(\pi) = \mathbb{E}(X)$, so X_0 is a constant random variable on Ω_k . For each $i \in [k - 1]$ and each $\pi \in \Omega_k$, define

$$X_i(\pi) = \mathbb{E}(X(\tilde{\pi}) | \tilde{\pi}(k) = \pi(k), \tilde{\pi}(k-1) = \pi(k-1), \dots, \tilde{\pi}(k-i+1) = \pi(k-i+1)).$$

In other words, $X_i(\pi)$ is the expected value of $X(\tilde{\pi})$ over all $\tilde{\pi} \in \Omega_k$ that agree with π in the first i merging steps. In particular, we have $X_0(\pi) = \mathbb{E}(X)$ and $X_{k-1}(\pi) = X(\pi)$. It is straightforward to verify that X_0, X_1, \dots, X_{k-1} is a martingale (see [2, 7, 18]). We show that this martingale satisfies the Lipschitz condition described below. First, we show it for $k \times 1$ matrices.

LEMMA 5.2. *Fix a $k \times 1$ matrix M with entries from $\{0, 1, *\}$. Let X_0, X_1, \dots, X_{k-1} be the martingale defined as above for M . Then $|X_{i+1}(\pi) - X_i(\pi)| \leq 1$ for each $i = 0, 1, \dots, k - 2$ and $\pi \in \Omega_k$.*

Proof. For each $i = 0, \dots, k - 1$, let $\Omega(i)$ denote the set of members of Ω_k that agree with the given π in the first i merging steps. In other words, $\Omega(i)$ is the event that our random member $\tilde{\pi} \in \Omega_k$ agrees with π in the first i merging steps. First note

that for each $i = 0, \dots, k - 2$, $\Omega(i + 1) \subseteq \Omega(i)$. For each $\tilde{\pi} \in \mathcal{S}_i$, when we perform the merging of M relative to $\tilde{\pi}$, the first i steps match those of the merging of M relative to π . Let $M' = M(k - i)$ denote the $(k - i) \times 1$ matrix obtained from M after these i merges. Suppose the total merging cost to this point is s . Let $M'' = M(k - i - 1)$ be the (a_{k-i}, b_{k-i}) -merge of M' . We have

$$(4) \quad \begin{aligned} X_i(\pi) &= \mathbb{E}_k(M|\Omega(i)) = s + \mathbb{E}_{k-i}(M'), \\ X_{i+1}(\pi) &= \mathbb{E}_k(M|\Omega(i + 1)) = s + \text{cost}_M(k - i) + \mathbb{E}_{k-i-1}(M''), \end{aligned}$$

where we recall that $\text{cost}_M(k - i) = 0$ or 1 is the cost of merging row a_{k-i} and row b_{k-i} of M' . Suppose first that $\text{cost}_M(k - i) = 0$. Then row a_{k-i} and b_{k-i} of M' do not have opposite values, and we may view M'' as being obtained from M' by deleting a row and then permuting the remaining rows if necessary. By Proposition 3.1 and Lemma 5.1, $\mathbb{E}_{k-i-1}(M'') \leq \mathbb{E}_{k-i}(M') \leq \mathbb{E}_{k-i-1}(M'') + 1$. By (4), this yields $X_{i+1}(\pi) \leq X_i(\pi) \leq X_{i+1}(\pi) + 1$.

Next, suppose $\text{cost}_M(k - i) = 1$. Then row a_{k-i} and row b_{k-i} of M' have opposite values, and row a_{k-i} of the resulting matrix M'' has a $*$. Let M''' be obtained from M'' by deleting this $*$ -entry/row. By Lemma 3.2, $\mathbb{E}_{k-i-1}(M'') = \mathbb{E}_{k-i-2}(M''')$. Also, we may view M''' as being obtained from M' by deleting two of its rows. By applying Lemma 5.1 twice, we get $\mathbb{E}_{k-i-2}(M''') \leq \mathbb{E}_{k-i}(M') \leq \mathbb{E}_{k-i-2}(M''') + 2$. For convenience, let $t = \mathbb{E}_{k-i-2}(M''')$. By (4) and our discussion above, we have $s + t \leq X_i(\pi) \leq s + t + 2$, while $X_{i+1}(\pi) = s + 1 + t$. Hence $|X_i(\pi) - X_{i+1}(\pi)| \leq 1$. This completes our proof. \square

For a general $k \times n$ matrix M with entries from $\{0, 1, *\}$, we can apply Lemma 5.2 to each column of M and use linearity of expectation (noting that the merging cost is the sum of merging costs of the columns) to get the following.

LEMMA 5.3. *Let k, n be positive integers and M a fixed $k \times n$ matrix with entries from $\{0, 1, *\}$. Let X_0, X_1, \dots, X_{k-1} be the martingale defined above for M . For each $i = 0, 1, \dots, k - 2$, we have $|X_{i+1}(\pi) - X_i(\pi)| \leq n$ for each $\pi \in \Omega_k$.*

Now, recall Azuma’s well-known inequality (see [2, 7, 18] for details).

LEMMA 5.4 (Azuma’s inequality). *Let X be a random variable, and let X_0, \dots, X_t be a martingale obtained from X , where $X_0 = \mathbb{E}(X)$ and $X_t = X$. Suppose there are constants c_1, \dots, c_t such that $|X_i - X_{i-1}| \leq c_i$ for each $i \in [t]$. Then*

$$\mathbb{P}(|X - E(X)| \geq \lambda) \leq 2e^{-\frac{\lambda^2}{2 \sum_{i=1}^t c_i^2}}.$$

Applying Lemmas 5.3 and 5.4 with $t = k - 1$ and $c_i = n$ for each i , we get the following theorem.

THEOREM 5.5. *Let k, n be positive integers. Let \mathcal{F}_k be a family of k subcubes in Q_n . Let Ω_k be the probability space of all k -merging patterns where each pattern is equally likely. For each $\pi \in \Omega_k$, let $X(\pi) = \text{cost}(\mathcal{F}_k, \pi)$. Then for all $\lambda > 0$,*

$$\mathbb{P}(|X - E(X)| \geq \lambda) \leq 2e^{-\frac{\lambda^2}{2kn^2}}.$$

Theorems 5.5, 3.5, and 4.8 yield the following corollary. It says that a randomly chosen merging pattern will with high probability yield (by Algorithm 2.2) a connecting subgraph for a family \mathcal{F}_k whose number of edges is near optimal for almost all such families.

COROLLARY 5.6. *Let ϵ be small and positive. There exist positive constants K_1, K_2, n_0, b , where $1 < b < 2$, for which the following hold. Let k, n be positive*

integers satisfying $n \geq n_0$ and $K_1 \leq k \leq 2^n$. Let \mathcal{F}_k be a family of k subcubes in Q_n . Let π be a random k -merging pattern chosen from Ω_k . Then, as $k, n \rightarrow \infty$, almost always

$$\text{cost}(\mathcal{F}_k, \pi) \leq \left(\frac{1}{3} + \epsilon\right)kn.$$

Furthermore, if $K_2 \leq k \leq b^n$, then for almost all families \mathcal{F}_k of k vertices in Q_n ,

$$\text{cost}(\mathcal{F}_k) \geq \left(\frac{1}{3} - \epsilon\right)kn.$$

Proof. Let $X(\pi) = \text{cost}(\mathcal{F}_k, \pi)$. By Theorem 3.5, $\mathbb{E}(X) \leq (\frac{1}{3}k + 1 + \frac{1}{2} \ln k)n$. Since $k \geq K_1$, by taking K_1 sufficiently large, we can ensure $\mathbb{E}(X) \leq (\frac{1}{3} + \frac{\epsilon}{2})kn$. Now set $\lambda = \frac{1}{2}\epsilon kn$ and apply Lemma 5.4. We get

$$\mathbb{P}\left(X > \left(\frac{1}{3} + \epsilon\right)kn\right) \leq \mathbb{P}\left(|X - \mathbb{E}(X)| > \frac{1}{2}\epsilon kn\right) \leq 2e^{-\frac{\epsilon^2}{8}k} \rightarrow 0 \quad \text{as } k \rightarrow \infty.$$

This proves the first part of the statement.

For the second part we appeal to Theorem 4.8. We need only observe that for ϵ small enough the constant B in that theorem satisfies $1 < B < 2$. So we can take b to be any constant satisfying $1 < b < B$. Then $b^n = o(\alpha B^n)$ as n grows. Thus $\text{cost}(\mathcal{F}_k) \geq (\frac{1}{3} - \epsilon)kn$ for almost all subcube families \mathcal{F}_k of size k satisfying $K_2 \leq k \leq b^n$, where we can take $K_2 = q^2$ in the statement of that theorem. \square

This corollary yields an algorithm as follows, where we continue with the notation of the corollary. In applying a k -merging pattern π to a set $S \subseteq V(Q_n)$ of size k (presented as a $k \times n$ binary matrix), each merging step of two rows of length n takes running time $O(n)$. Since there are $k - 1$ such steps, the total running time of Algorithm 2.2 for any given input set S and k -merging pattern π is $O(kn)$. Thus we get a probabilistic algorithm, consisting of a random choice of k -merging pattern π followed by an execution of Algorithm 2.2 with inputs π and S , which produces a subgraph H_π that connects S . The running time is $O(kn)$, and provided that $k \geq K_1$, we have $|E(H_\pi)| \leq (\frac{1}{3} + \epsilon)kn$ with probability approaching 1 (over the space Ω_k) as $k, n \rightarrow \infty$. Further, for almost any input set S satisfying $K_2 \leq k \leq b^n$ for n large enough, the output graph H_π is near optimal in the sense that

$$\left(\frac{1}{3} - \epsilon\right)kn \leq \text{cost}(S) \leq |E(H_\pi)| \leq \left(\frac{1}{3} + \epsilon\right)kn.$$

REFERENCES

- [1] R. AGARWALA AND D. FERNÁNDEZ-BACA, *A polynomial-time algorithm for the perfect phylogeny problem when the number of character states is fixed*, SIAM J. Comput., 23 (1994), pp. 1216–1224.
- [2] N. ALON AND J. SPENCER, *The Probabilistic Method*, 2nd ed., John Wiley & Sons, New York, 2001.
- [3] E. ALTHAUS AND R. NAUJOKS, *Computing Steiner minimum trees in Hamming metric*, in Proceedings of the Seventeenth Annual ACM-SIAM Symposium on Discrete Algorithms, 2006, pp. 172–181.
- [4] A. BORCHERS AND D.-Z. DU, *The k -Steiner ratio in graphs*, SIAM J. Comput., 26 (1997), pp. 857–869.
- [5] T. BRUEN AND D. BRYANT, *A subdivision approach to maximum parsimony*, Ann. Comb., 12 (2008), pp. 45–51.

- [6] C. CHEKURI, G. EVEN, AND G. KORTSARZ, *A combinatorial approximation algorithm for the group Steiner problem*, Discrete Appl. Math., 154 (2006), pp. 15–34.
- [7] F. CHUNG AND L. LU, *Concentration inequalities and martingale inequalities: A survey*, Internet Math., 3 (2006), pp. 79–127.
- [8] W. DAY, D. JOHNSON, AND D. SANKOFF, *The computational complexity of inferring rooted phylogenies by parsimony*, Math. Biosci., 81 (1986), pp. 33–42.
- [9] D. FERNANDEZ-BACA AND J. LAGERGREN, *On the approximability of the Steiner tree problem in phylogeny*, Discrete Appl. Math., 88 (1998), pp. 129–145.
- [10] L. R. FOULDS AND R. L. GRAHAM, *The Steiner problem in phylogeny is NP-complete*, Adv. Appl. Math., 3 (1982), pp. 43–49.
- [11] P. FRANKL AND V. RÖDL, *Lower bounds for Turan’s problem*, Graphs Combin., 1 (1985), pp. 213–216.
- [12] M. R. GAREY, R. L. GRAHAM, AND D. S. JOHNSON, *The complexity of computing Steiner minimal trees*, SIAM J. Appl. Math., 32 (1977), pp. 835–859.
- [13] M. R. GAREY AND D. S. JOHNSON, *The rectilinear Steiner tree problem is NP-complete*, SIAM J. Appl. Math., 32 (1977), pp. 826–834.
- [14] D. GUSFIELD, *Algorithms on Strings, Trees, and Sequences*, Cambridge University Press, Cambridge, UK, 1997.
- [15] D. GUSFIELD, *Efficient algorithms for inferring evolutionary trees*, Networks, 21 (1991), pp. 19–28.
- [16] D. GUSFIELD, *The Steiner Tree Problem in Phylogeny*, Technical report 334, Computer Science Department, Yale University, New Haven, CT, 1984.
- [17] F. HWANG, D. RICHARDS, AND P. WINTER, *The Steiner Tree Problem*, Ann. Discrete Math. 53, North-Holland, Amsterdam, 1992.
- [18] S. JANSON, T. ŁUCZAK, AND A. RUCIŃSKI, *Random Graphs*, Wiley-Interscience, New York, 2000.
- [19] T. JIANG, E. LAWLER, AND L. WANG, *Aligning sequences via an evolutionary tree: Complexity and approximation*, in Proceedings of the Twenty-Sixth Annual ACM Symposium on Theory of Computing (STOC ’94), Montreal, Canada, 1994, pp. 760–769 (preliminary version of [29]).
- [20] Z. MILLER AND M. PERKEL, *The Steiner problem in the hypercube*, Networks, 22 (1992), pp. 1–19.
- [21] Z. MILLER AND D. PRITIKIN, *Applying a result of Frankl and Rödl to the construction of Steiner trees in the hypercube*, Discrete Math., 131 (1994), pp. 183–194.
- [22] M. MOLLOY AND B. REED, *Graph Colouring and the Probabilistic Method*, Springer-Verlag, Berlin, 2002.
- [23] C. H. PAPADIMITRIOU AND M. YANNAKAKIS, *Optimization, approximation, and complexity classes*, J. Comput. System Sci., 43 (1991), pp. 425–440.
- [24] P. PEVZNER, *Computational Molecular Biology: An Algorithmic Approach*, MIT Press, Cambridge, MA, 2000.
- [25] H. J. PRÖMEL AND A. STEGER, *The Steiner Tree Problem: A Tour through Graphs, Algorithms, and Complexity*, Vieweg, Braunschweig, Germany, 2002.
- [26] G. REICH AND P. WIDMAYER, *Beyond Steiner’s problem: A VLSI oriented generalization*, in Graph-Theoretic Concepts in Computer Science, Lecture Notes in Comput. Sci. 411, Springer-Verlag, Berlin, 1990, pp. 196–210.
- [27] G. ROBINS AND A. ZELIKOVSKY, *Tighter bounds for graph Steiner tree approximation*, SIAM J. Discrete Math., 19 (2005), pp. 122–134.
- [28] L. TREVISAN, *When Hamming meets Euclid: The approximability of geometric TSP and Steiner tree*, SIAM J. Comput., 30 (2000), pp. 475–485.
- [29] L. WANG, T. JIANG, AND E. LAWLER, *Approximation algorithms for tree alignment with a given phylogeny*, Algorithmica, 16 (1996), pp. 302–315.
- [30] H. T. WAREHAM, *A simplified proof of the NP- and MAX SNP-hardness of multiple sequence tree alignment*, J. Comput. Biol., 2 (1995), pp. 509–514.