Algorithmica (2010) 56: 141–159
DOI 10.1007/s00453-008-9164-4

Note on the Structure of Kruskal's Algorithm

Nicolas Broutin · Luc Devroye · Erin McLeish

Received: 18 October 2005 / Accepted: 3 January 2008 / Published online: 24 January 2008
© Springer Science+Business Media, LLC 2008

Abstract We study the merging process when Kruskal's algorithm is run with random graphs as inputs. Our aim is to analyze this process when the underlying graph is the complete graph on n vertices lying in $[0, 1]^d$, and edge set weighted with the Euclidean distance. The height of the binary tree explaining the merging process is proved to be $\Theta(n)$ on average. On the way to the proof, we obtain similar results for the complete graph and the d-dimensional square lattice with i.i.d. edge weights.

Keywords Random trees · Minimum spanning tree · Kruskal · Height · Random graphs · Percolation

1 Introduction

The minimum spanning tree (MST) problem is one of the most studied in combinatorial optimization. In a connected graph $G = (V, E)$, each edge e is given a weight $w(e)$. Let the weight of a graph be the sum of the weights of its edges. The MST problem consists then in computing a minimal weight tree, whose edges cover the entire set of vertices V. Fast greedy algorithms are known to solve this problem, namely the algorithms of Prim [10, 15, 25], Kruskal [18] and Borůvka [8]. Minimum spanning trees with random edge weights have been analyzed in several papers. Examples of

Research of the authors was supported by NSERC Grant A3456 and by a James McGill Fellowship.

N. Broutin · L. Devroye (✉) · E. McLeish
School of Computer Science, McGill University, Montreal H3A2K6, Canada
e-mail: luc@cs.mcgill.ca

N. Broutin
e-mail: nbrout@cs.mcgill.ca

E. McLeish
e-mail: mcleish@cs.mcgill.ca

such random graphs include the complete graph or d-dimensional hypercube with random independent edge weights [22], and the complete graph over n points in $\mathbb{R}^d$ with the Euclidean distances as weights [26, 28]. The properties studied include the overall weight [12], the degree sequence [2], and the maximum weight [21, 23].

In this paper, instead of studying the MST itself, we rather study the process that builds it. Such a point of view has already been taken by McDiarmid, Johnson and Stone [19] who studied the first few steps of Prim's algorithm when the underlying graph is complete and the edge weights are independent identically distributed (i.i.d.) random variables. We give results for Kruskal's forest growing method, and more particularly about the structure of the merging process. With this in mind, we want to define a *tree explaining* Kruskal's algorithm, and we proceed as follows.

We grow two graph processes side by side. Assume that G is the underlying weighted graph for which we are computing the MST. Let $\{M_i(G), 1 \le i \le n\}$ be the random graph process for Kruskal's algorithm on a graph G of n vertices, where $M_i(G)$ is the forest after exactly $i-1$ edges have been added, and $M_n(G)$ is the resulting MST. The second process is denoted by $\{F_i(G), i \le i \le n\}$, where $F_i(G)$ is a binary forest over $n+i-1$ vertices. We use M_i and F_i for short, when the underlying graph G is already specified. The F_i's model the merging process between components during the execution of Kruskal's algorithm. Each tree in the forest F_i is rooted, and each tree in F_i corresponds to a connected component in M_i. One can think of the leaves of F_i as the vertices in M_i and the internal nodes as being the connecting edges. When $i = n$, the process finishes with the tree explaining Kruskal's algorithm: $T_n \stackrel{\text{def}}{=} F_n$. Also, let $\{\mathcal{F}_i, 1 \le i \le n\}$ be the natural filtration adapted to the random processes. The two processes grow from $i = 1$ to n in the following way: Initially, M_1 consists of n isolated vertices $v_1, v_2, \ldots, v_n$ and no edges, and F_1 consists of n vertices $u_1, u_2, \ldots, u_n$, which are merely copies of the vertices in the graph G. Kruskal's algorithm adds the edges of G by increasing weight, as long as they do not create a cycle. Suppose we have already built M_i and F_i, and that e is the next edge to be added by Kruskal's algorithm. We grow the forest M_{i+1} by simply adding the edge e: $M_{i+1} = M_i \cup \{e\}$. Next, suppose the endpoints of e are vertices u and v. Then the edge e connects two components of M_i: one containing vertex u, and the other, v. In the forest F_i, a component is uniquely represented by the root of the tree it corresponds to. Let r_u and r_v be the roots of the trees in F_i containing u and v respectively. Grow the forest F_{i+1} from F_i by using two new edges and a new root to join the roots r_u and r_v, creating one tree in F_{i+1} out of the two trees in F_i. Eventually, the process leads to a random binary tree $T_n = F_n$ that models the way the components were merged during the execution of Kruskal's algorithm. The main parameter of interest for us is the height $H_n(G)$ of T_n, which is the height of the tree T_n when the underlying weighted graph is G.

The purpose of the paper is to give asymptotic results for H_n when the underlying graph is the complete graph over n independent uniform points lying in $[0,1]^d$, and the edges are weighted with the Euclidean distance between their endpoints.

Theorem 1 *Let $X_1, X_2, \ldots, X_n$ be n independent uniform points in $[0,1]^d$. Let E_n be the complete graph with vertex set $\{X_1, X_2, \ldots, X_n\}$. For each $1 \le i, j \le n$, let the weight of the edge X_iX_j be the Euclidean distance between X_i and X_j. Let $H_n(E_n)$*

be the height of the tree $T_n(E_n)$ explaining Kruskal's algorithm when the input is E_n. Then there exists a constant $c \in (0,1)$ such that

$$\lim_{n\to\infty} \mathbf{P}\{H_n(E_n) \geq cn\} = 1.$$

In particular, $\mathbf{E}H_n(E_n) = \Theta(n)$, as n goes to infinity.

Hence, in some sense, Kruskal's algorithm does not proceed very differently from Prim's algorithm, which grows a tree of height exactly $n-1$. Our proof uses ideas arising from two simpler examples. The Euclidean case has some dependence between the distances between points, and one can naturally consider first the complete graph K_n with i.i.d. edge weights.

Theorem 2 *$H_n(K_n)$ is the height of the tree $T_n(K_n)$ explaining Kruskal's algorithm when the input is the complete graph K_n with edges weighted by independent $[0,1]$-uniform random variables. There exists a constant $c \in (0,1)$ such that*

$$\lim_{n\to\infty} \mathbf{P}\{H_n(K_n) \geq cn\} = 1.$$

As a consequence, $\mathbf{E}H_n = \Theta(n)$.

This simpler case is related to the analysis of union-find algorithms, and quickfind in particular [1]. In quickfind, the cost of merging two sets is proportional to the size of the one of them, the one that is "merged into" the other. A result of Knuth and Schönhage [17] about the average-case behavior in a model of Yao [27] shows that the cost of $n-1$ random unions is $\Theta(n^2)$. It immediately follows that some set of linear size must be involved in a linear number of union operations. Further information about quickfind and related algorithms may be found in the analysis of Bollobás and Simon [5, 6], which, as our proof of Theorem 2, relies on the theory of random graphs.

Further, we add a geometrical aspect to the previous example by considering a lattice. Let $\mathbb{L}^d$ be the d-dimensional square lattice: the infinite graph with vertex set $\mathbb{Z}^d$, and the edges between vertices at distance 1 from each other. Consider the graph D_n, which is the restriction of $\mathbb{L}^d$ to a d-dimensional cubic box consisting of n vertices, where $n = k^d$ for some integer $k \geq 0$. Assign i.i.d. random weights to the edges in D_n.

Theorem 3 *Let $H_n(D_n)$ be the height of the tree $T_n(D_n)$ explaining Kruskal's algorithm when the input is D_n with edges weighted by independent $[0,1]$-uniform random variables. There exists a constant $c \in (0,1)$ such that*

$$\lim_{n\to\infty} \mathbf{P}\{H_n(D_n) \geq cn\} = 1,$$

as $n \to \infty$. Hence, in particular, $\mathbf{E}H_n(D_n) = \Theta(n)$.

After some remarks valid for all three cases (Sect. 2), we start by proving Theorems 3 and 2. Theorem 1 is proved in Sect. 5.

2 Preliminaries

The height H_n of T_n cannot exceed n, and it suffices to focus on a lower bound. The present section explains a generic strategy to obtain the required lower bound. Step i of Kruskal's algorithm is the step when the ith edge is added, i.e., M_i becomes M_{i+1}. Observe that when a new edge is added to M_i, two components are joined together by the edge, and the two corresponding trees in F_i are merged, using a new root and two new edges. After this merge, the height of the new tree in F_{i+1} is at least one more than the height of either of the two trees before the merging. Suppose we track a particular component during the execution of Kruskal's algorithm. Let $C_i(x)$ be the connected component in M_i containing vertex x. Each time this component is involved in a merge, the height of the corresponding tree in F_i increases by at least 1. As a consequence, the number of merges that $C_i(x)$ is involved in, as i increases to n, is a lower bound on the height of the final tree T_n.

Let ξ_i be the indicator random variable that $C_i(x) \neq C_{i+1}(x)$, in other words, that component $C_i(x)$ is merged to another component to create $C_{i+1}(x)$. Fix any k, $1 \leq k \leq n-1$. Then the number of merges that the component $C_k(x)$ is involved in starting at time k is exactly $\sum_{i=k}^{n-1} \xi_i$. The next lemma formalizes this as a lower bound for the height $H_n(G)$ for any graph G.

Lemma 1 *Let $H_n(G)$ be the height of the tree explaining Kruskal's algorithm when run on the input graph G with vertex set V. Fix some vertex $x \in V$. Define the indicator random variables ξ_i, $1 \leq i \leq n-1$ to be 1 when the edge added at the ith step of Kruskal's algorithm is incident to the component containing vertex x. Then,*

$$H_n(G) \geq \sum_{j=k}^{n-1} \xi_i \tag{1}$$

for $1 \leq k \leq n-1$.

Lemma 1 lies at the heart of the approach we use in Sects. 3, 4 and 5 to find lower bounds on the height H_n for the examples we study. In the following, we write $|\cdot|$ for the cardinality function. Unless specified otherwise, $|\cdot|$ counts vertices for components of graphs. Also, in Euclidean settings, we write $\|\cdot\|$ to denote the volume of a subset of $[0,1]^d$.

3 Proof of Theorem 2

In this section, we deal with the case of the complete graph weighted with i.i.d. $[0,1]$-uniform random variables. This version of the problem is tightly connected to the structure of Erdős–Rényi random graphs [11].

Observe first that the weights on the edges induce a random permutation of the edges. This permutation is a uniformly random permutation. In particular, the m first edges of the permutation define a random graph $G_{n,m}$ consisting of m random edges

(see Janson et al. [14] or Bollobás [4]). Processing the elements of this random permutation in order, Kruskal's algorithm adds an edge to the spanning forest unless it creates a cycle. Accordingly, the connected components of $M_i(K_n)$ are the same (vertex-wise), as the connected components of the Erdős–Rényi random graph $G_{n,m}$ for some $m \geq i$. In other words, studying the structure of the merges in Kruskal's algorithm reduces to a similar study in a $G_{n,m}$ model. In their seminal paper [11], Erdős and Rényi showed that a phase transition occurs in the component structure of $G_{n,m}$ random graphs when $m = \lfloor \beta n/2 \rfloor$ and $\beta = 1$. Indeed, for $\beta < 1$, the largest component, L_1 of $G_{n,\lfloor \beta n/2 \rfloor}$ satisfies almost surely (a.s.) $|L_1| = \Theta(\log n)$, while for $\beta > 1$ it has size $\Theta(n)$.

For our purposes, properties of $H_n(K_n)$ rely deeply on the component structure of $G_{n,m}$: once a component is large enough, it tends to absorb more and more small components. Let $1 < \beta < 2$ and consider the random graph $G_{n,m}$ where $m = \lfloor \beta n/2 \rfloor$. The connected components of this graph correspond exactly to those in the forest $F_{m'}$ for some $m' \leq m$. Note that m' is a random variable. We consider the evolution of Kruskal's algorithm from this point on. Let $C_{m'}$ be the largest component at time m', and let C_i denote the component containing $C_{m'}$ at a later step $i \in \{m', \ldots, n\}$ of the algorithm. Notice that the C_i's are increasing in size, that is, $|C_i| \leq |C_{i+1}|$ for any $i \in \{m', \ldots, n-1\}$. As in the previous section, let ξ_i be the indicator random variable for the event that C_i is merged with another component at time i. Then by Lemma 1

$$H_n(K_n) \geq \sum_{i=m'}^{n-1} \xi_i.$$

Now, for $\beta > 1$, there is $\theta \in (0, 1)$ such that the largest component of $G_{n,\lfloor \beta n/2 \rfloor}$ has at least θn vertices with probability going to 1. The constant θ is independent of n. Let Z be the indicator random variable for the event that $|C_{m'}| \geq \theta n$. Note that $\mathbf{P}\{Z = 1\} \to 1$, as $n \to \infty$. Then certainly

$$H_n(K_n) \geq \sum_{i=m'}^{n-1} \xi_i \cdot Z.$$

Consider the forest M_i, and let A_i be the set of edges in K_n between C_i and the rest of the graph. The edges in A_i are the edges that may hook C_i to an other component. Let B_i be the remaining edges *not* incident to C_i and which do not create cycles. Then

$$|A_i| \geq (n - |C_i|) \cdot |C_i|,$$

and

$$|B_i| \leq \frac{(n - |C_i|) \cdot (n - |C_i| - 1)}{2}.$$

The edge added at step i is uniformly random among those not creating a cycle, and

$$\mathbf{P}\{\xi_i = 1 \mid |C_i|\} \geq \frac{|A_i|}{|A_i| + |B_i|} \geq \frac{2|C_i|}{|C_i| + n}.$$

The events $\xi_i = 1$, $i \in \{m', \ldots, n-1\}$ are not independent, since the more often C_i merges, the bigger it gets and the more likely it is to merge again. We define new random variables, completely independent of the graph process to go around this. Let S_i be a Bernoulli random variable with parameter θ. For $i \in \{m', \ldots, n-1\}$, $\xi_i \cdot Z$ stochastically dominates $S_i \cdot Z$:

$$\begin{aligned}\mathbf{P}\{\xi_i \cdot Z = 1\} &= \mathbf{P}\{\xi_i = 1 \mid Z\} \cdot \mathbf{P}\{Z = 1\} \\ &= \mathbf{P}\{\xi_i = 1 \mid Z, |C_i| \geq \theta n\} \cdot \mathbf{P}\{Z = 1\} \\ &\geq \frac{2\theta n}{\theta n + n} \cdot \mathbf{P}\{Z = 1\} \\ &\geq \theta \cdot \mathbf{P}\{Z = 1\} = \mathbf{P}\{S_i \cdot Z = 1\},\end{aligned}$$

where the last step follows from the independence of S_i and Z. Thus, for every t,

$$\mathbf{P}\{H_n(K_n) \geq t\} \geq \mathbf{P}\left\{\sum_{m'}^{n-1} S_i \cdot Z \geq t\right\} = \mathbf{P}\left\{\sum_{m'}^{n-1} S_i \geq t\right\} \cdot \mathbf{P}\{Z = 1\}.$$

Recall that $m' \leq n\beta/2$. By the law of large numbers,

$$\mathbf{P}\left\{\sum_{m'}^{n-1} S_i \geq n\left(1 - \frac{\beta}{2}\right) \cdot \frac{\theta}{2}\right\} \geq \mathbf{P}\left\{\sum_{m'}^{n-1} S_i \geq (n - m') \cdot \frac{\theta}{2}\right\} \xrightarrow[n\to\infty]{} 1.$$

Finally, $Z \to 1$ almost surely, and

$$\mathbf{P}\left\{H_n(K_n) \geq n \cdot \frac{\theta}{2} \cdot \left(1 - \frac{\beta}{2}\right)\right\} \xrightarrow[n\to\infty]{} 1.$$

Since, $\theta(1 - \beta/2) > 0$ by our definitions of β and θ, this completes the proof of Theorem 2.

4 Proof of Theorem 3

We now deal with the case of the finite box D_n in the cubic lattice $\mathbb{L}^d$. As in the proof of Theorem 2, we show that there is a large component at some intermediate stage of Kruskal's algorithm, which will be merged to a linear number of other components as the algorithm goes on. By Lemma 1, this gives a lower bound on $H_n(D_n)$. To ensure a linear number of merges, we look for a large component with a specified structure involving "traps" for other components. Our approach is constructive and goes in two phases. First, a connected component C_{p_0} made of a large number of traps is built using only edges of weight less than a fixed constant $p_0 \in (0, 1)$. The proof of its existence relies on concepts from percolation theory [13, 16]. Then, the minimal spanning tree process is completed by adding the edges with weight more than p_0. In this second phase, the traps prevent small components from merging together before they hook up to C_{p_0}, hence the large number of merges to C_{p_0}.

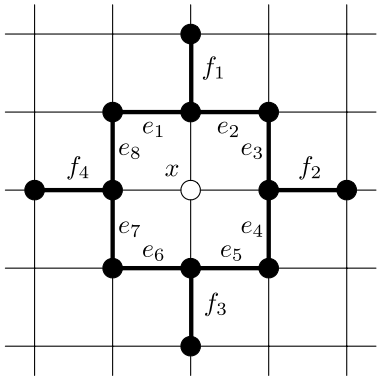

Fig. 1 A portion of the square lattice $\mathbb{L}^2$ is shown, and the event E_x. Edges labelled $e_1, \ldots, e_8$ are in $N_1(x)$, and edges labelled $f_1, \ldots, f_4$ are in $N_2(x)$

We begin by defining a subset of $V(D_n)$, denoted I_n, for the sites of a percolation process. Suppose that the left bottom corner vertex of D_n is at the origin, and using this coordinate system, let

$$I_n = \{v : \forall i, v_i \mod 4 = 1\},$$

where v_j is the jth coordinate of vertex v. Each $x \in I_n$ is a *site* in a (site) percolation process. Two sites in $u, v \in I_n$ are neighbors if they are at distance 4 from each other along any one coordinate, i.e., u and v are neighbors if $|u - v|_1 = 4$ and $|u - v|_\infty = 4$. For any $x \in I_n$, define a structural event E_x which is depicted in Figure 1, and defined in terms of the following sets:

$$U(x) = \{v \in V(D_n) : |v - x|_\infty = 1\},$$

where $|\cdot|_\infty$ defines the ℓ_∞-norm. Next, we define the edges on the unit box around x:

$$N_1(x) = \{e = (u, v) : u, v \in U(x), |u - v|_1 = 1\}.$$

Define the sets

$$F_1(x) = \{v \in V(D_n) : |v - x|_1 = 1\},$$

and

$$F_2(x) = \{v \in V(D_n) : |v - x|_1 = 2 \text{ and } |v - x|_\infty = 2\},$$

where $|\cdot|_1$ defines the ℓ_1-norm. Finally, the outward edges in Fig. 1 are defined as:

$$N_2(x) = \{e = (u, v) : u \in F_1(x) \text{ and } v \in F_2(x)\}.$$

The event E_x holds when all edges in $N_1(x)$ and $N_2(x)$ have weight less than p_0. Figure 1 depicts the event E_x in $d = 2$. Note that there is no restriction on the weights of the other edges (those not in $N_1(x)$ or $N_2(x)$) in the event E_x.

A site $x \in I_n$ is declared *open* if the event E_x is true. This defines precisely a site percolation process since the probability that sites are open are independent from site to site. Figure 2 shows an instance of this site percolation process. A *cluster* in this

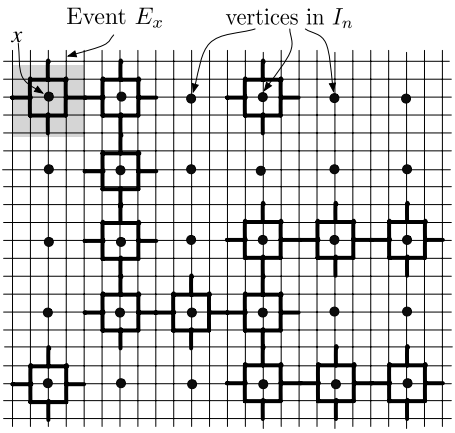

Fig. 2 An instance of site percolation on I_n, where the probability of an open site x is exactly the probability that event E_x occurs

model is defined in the usual way: the cluster of a site x is the set of all open sites that can be reached along paths of open sites from x. For example, Fig. 2 shows a cluster of size 13. Each site is a "trap" for its center vertex, and the goal is now to show the existence of a large open cluster in the site percolation process for some choice of $p_0 \in (0, 1)$.

It has been known for quite some time that the usual site percolation process on the infinite lattice $\mathbb{L}^d$ exhibits a phase transition phenomenon analogous to that in the random graph process: there exists a constant $p_c \in (0, 1)$ such that for $p < p_c$, there is almost surely no infinite cluster, whereas for $p > p_c$ such an infinite cluster a.s. exists [13, 16]. In our model, we are dealing with site percolation on a finite lattice, which is in fact a subgraph of the infinite lattice $\mathbb{L}^d$. The phase transition on the finite lattice has been analyzed by Borgs, Chayes, Kesten, and Spencer [7], who obtained an exact analog of the phase transition in random graphs for bond percolation in the finite hypercube $\Lambda_n = [0, n)^d$. In particular, they showed:

Lemma 2 *Let Λ_n denote a hypercube lattice of size n. Consider bond percolation, where each edge in the lattice is open with probability p, and let $L_1^{bond}(p)$ be the size of the largest open cluster in this hypercube. Let $|C_p(0)|$ denote the size of the open cluster at the origin on the infinite lattice $\mathbb{L}^d$, and let $\theta^{bond}(p) = \mathbf{P}\{|C_p(0)| = \infty\}$. Finally, let p_c^{bond} be the critical probability for an infinite cluster in bond percolation on the infinite lattice $\mathbb{L}^d$. Then for all $p > p_c^{bond}$,*

$$\frac{|L_1^{bond}(p)|}{|\Lambda_n| \cdot \theta^{bond}(p)} \xrightarrow[n\to\infty]{} 1 \quad \textit{in probability.}$$

A simple extension of this lemma is the following corollary, which gives a lower bound on the height of the largest open cluster in *site* percolation. The proof of the lemma is included at the end of this section.

Lemma 3 *Consider site percolation on the finite hypercube lattice Λ_n. Let $L_1^{site}(p)$ be the largest open cluster. There exist constants $c_1 > 0$ and $p_s \in (0, 1)$ such that for*

$p > p_s$:

$$\mathbf{P}\left\{|L_1^{site}(p)| \geq c_1 n^d\right\} \xrightarrow[n\to\infty]{} 1.$$

The constants c_1 and p_s depend only on the dimension d.

In our site percolation model, the probability that a site is open is exactly $\mathbf{P}\{E_x\}$, which depends on our choice of p_0. We are now ready to nail down the constant p_0. With the previous definitions of $N_1(x)$ and $N_2(x)$, $\mathbf{P}\{E_x\} = p_0^{|N_1(x)|+|N_2(x)|}$. The sizes of the sets $N_1(x)$ and $N_2(x)$ are increasing in d, but constant in terms of n. Therefore, we can choose $p_0 \in (0, 1)$ large enough that $p_s < \mathbf{P}\{E_x\} < 1$, where $\mathbf{P}\{E_x\}$ is a function of d only. Consider now the largest open cluster in the site percolation process on I_n and let $L_1(I_n)$ be its vertex set. Note that by definition, the lattice I_n has size $|I_n| \geq n/4^d$. By Lemma 3,

$$\mathbf{P}\left\{|L_1(I_n)| \geq \frac{c_1 n}{4^d}\right\} \xrightarrow[n\to\infty]{} 1. \tag{2}$$

Next, we isolate the vertices of the connected component of D_n associated with the largest open cluster, $L_1(I_n)$:

$$C_{p_0} = \{v \in V(D_n) : \exists x \in L_1(I_n), v \in U(x) \text{ or } v \in F_2(x)\}.$$

Recall that $\{M_i, 1 \leq i \leq n\}$ is the graph process associated with Kruskal's algorithm, and let $i^\star$ be the largest index for which all the edges of $M_{i^\star}$ have weight at most p_0. Then the vertices of C_{p_0} belong to a connected component in $M_{i^\star}$. Thus it remains only to provide a lower bound on the number of merges to this component as Kruskal's algorithm completes.

For each $x \in I_n$, we define an event J_x which ensures that the "trap" at x is successful and that a small component joins C_{p_0} at some stage $p > p_0$ of Kruskal's algorithm. Define the set of edges

$$N_3(x) = \{e = (x, v) : |v - x|_1 = 1\}.$$

Then the event J_x is the event that all edges in $N_3(x)$ have weight *more* than p_0. Thus if J_x holds, the vertex x is *not* connected to any vertex in C_{p_0} in the graph $M_{i^\star}$, but it will be connected at some later stage of Kruskal's algorithm. Notice that J_x, $x \in I_n$, are defined using edges disjoint from $N_1(y)$ and $N_2(y)$, for $y \in I_n$, so that $\{J_x, x \in I_n\}$ is independent of $\{E_y, y \in I_n\}$. Moreover, the events J_x, $x \in I_n$, are independent. The proof of the theorem is now straightforward: we show that J_x is true for a constant proportion of the sites in $L_1(I_n)$, implying that a constant proportion of sites in $L_1(I_n)$ will be responsible for merges to that component as Kruskal completes. In other words, by Lemma 1,

$$H_n(D_n) \geq \sum_{x \in L_1(I_n)} \mathbf{1}[J_x].$$

But the events J_x and $\{x \in L_1(I_n)\}$ are independent, and therefore by (2),

$$\mathbf{P}\left\{\sum_{x \in L_1(I_n)} \mathbf{1}[J_x] \geq cn\right\} \xrightarrow[n\to\infty]{} 1$$

where $c = c_1(1-p)^{2d}/(2 \cdot 4^d)$. It follows that $\mathbf{P}\{H_n(D_n) \geq cn\} \to 1$ as $n \to \infty$, which completes the proof of Theorem 3.

Proof of Lemma 3 Consider the lattice Λ_n where each edge $e = (x, y)$ is represented by two *coloured* edges, one red and one green. Each coloured edge is opened independently with probability p. Now consider a bond percolation process on Λ_n where a bond is considered open if *both* the red and green edges between two vertices are open. Thus the bond probability is exactly p^2. Let $L_1^{bond}(p^2)$ be the largest open cluster in this bond percolation process on Λ_n.

Next, consider a site percolation process, also on Λ_n. Begin by duplicating the edges as described above: each edge is replaced by one red and one green edge, and each coloured edge is opened with probability p. Next, alternate colouring the vertices of Λ_n red and green, so a red vertex is surrounded by green vertices, and vice versa. A red site is declared open if at least one of its incident red edges is open. A green site is open if at least one of its incident green edges is open. Because of the alternating colours, sites are opened independently, with probability $1-(1-p)^{2d}$. An open cluster is defined in the usual way to be a cluster of open sites, regardless of their colour. Let $L_1^{site}(p')$ be the size of the largest cluster in this site percolation process on Λ_n, where $p' = 1-(1-p)^{2d}$.

Now we couple the two models. Consider and instance of Λ_n with duplicated edges, where we open the coloured edges with probability p. Notice that if a bond is open in the bond percolation model, then both the red and green edges are open. So the red vertex incident to the red edge is open in the site model, and the green vertex incident to the green edge is also open in the site model. Thus the two neighboring sites are both open in the site model. This implies that the vertices in the cluster $L^{bond}(p^2)$ are *also* in a cluster in the site percolation model. Thus $|L^{bond}(p^2)| \leq |L^{site}(p')|$. By Lemma 2, if $p^2 > p_c^{bond}$, then $\mathbf{P}\{|L_1^{bond}(p^2)| \geq c_1 n^d\} \to 1$ for a constant $c_1 > 0$. Thus

$$\mathbf{P}\left\{|L_1^{site}(1-(1-p)^{2d})| \geq c_1 n^d\right\} \xrightarrow[n\to\infty]{} 1$$

for $p^2 > p_c^{bond}$. So in general, for any $p > 1-(1-\sqrt{p_c^{bond}})^{2d}$,

$$\mathbf{P}\left\{|L_1^{site}(p) \geq c_1 n^d\right\} \xrightarrow[n\to\infty]{} 1.$$

Setting $p_s = 1-(1-\sqrt{p_c^{bond}})^{2d} \in (0,1)$ completes the proof. □

5 Random Euclidean Edge Lengths

Random Euclidean minimum spanning trees are related to random geometric graphs [24] and continuum percolation [20]. One can define an analog of Erdős–Renyi ran-

dom graphs [4, 14] where the vertex set consists of points lying in the cube $[0,1]^d$. The distance between pairs of vertices is defined to be the Euclidean distance. The classical coupling used to build Erdős–Rényi $G_{n,p}$, assigns to each edge of the complete graph K_n an independent copy of a $[0,1]$-uniform random variable, and lets $G_{n,p}$ be the graph containing edges shorter than p. Similarly, the random geometric graph $G(\mathcal{X}_n, r)$ is defined to be the graph with vertex set $\mathcal{X}_n$, which consists of n independent uniform points in $[0,1]^d$, and edges shorter than r. For technical reasons, it is usually easier to consider random graphs $G(\mathcal{P}_n, r)$ that are defined similarly, but with vertex set given by a Poisson point process $\mathcal{P}_n$ with rate n in $[0,1]^d$. It is convenient to see random geometric graphs as a vertex set $\mathcal{X}$ together with closed balls $B(x, r/2)$ of radius $r/2$ surrounding every $x \in \mathcal{X}$. A pair of points $x, y \in \mathcal{X}$ share an edge in $G(\mathcal{X}, r)$ if and only if $B(x, r/2)$ and $B(y, r/2)$ intersect. Equivalently, the points $x, y \in \mathcal{X}$ share an edge if the Euclidean distance from x to y is at most r.

The graphs $G(\mathcal{X}_n, r_n)$ and $G(\mathcal{P}_n, r_n)$ with radius

$$r_n = \left(\frac{\lambda}{n}\right)^{1/d}$$

are known to admit a threshold for the size of the largest component for some critical $\lambda_c > 0$ [20]. In other words, if $L_1(n)$ is the largest component in the random graph $G(\mathcal{P}_n, r_n)$ in $[0,1]^d$, then there exists a λ_c such that if $\lambda > \lambda_c$,

$$n^{-d}|L_1(n)| \xrightarrow[n\to\infty]{} \lambda p_\infty(\lambda),$$

in probability [24, Theorem 10.9]. So, as in the $G_{n,p}$ model, the largest component has size linear in the volume. The constant $p_\infty(\lambda)$ is defined in terms of a Poisson process $\mathcal{P}$ with rate 1 in $\mathbb{R}^d$: it is the probability that the origin lies in the set of balls of an infinite component of the geometric graph $G(\mathcal{P}, \lambda^{1/d})$. Random geometric graphs differ from the Erdős–Rényi model when we consider the second largest component. For the Erdős–Rényi random graph with $p = cn/2$ and $c > 1$, the order of the second largest component grows as $\log n$. For the random geometric graph, it grows not faster than a larger power of the logarithm [24, Theorem 10.18]. Let $L_2(n)$ be the second largest component in $G(\mathcal{P}_n, r_n)$ and $\lambda > \lambda_c$, then for $d \geq 2$, there exists a constant c_1 such that

$$\mathbf{P}\left\{|L_2(n)| < c_1 \cdot (\log n)^{d/(d-1)}\right\} \xrightarrow[n\to\infty]{} 1.$$

With these two facts in hand, a lower bound on the height H_n immediately follows. We can grow the minimum spanning forest using increasing radii $r_n = (\lambda/n)^{1/d}$, until $\lambda > \lambda_c$. Then, use the number of components that hook up to $L_1(n)$ from this point on as a lower bound on H_n. The size of any component merging with $L_1(n)$ is at most of order $(\log n)^{d/(d-1)}$ and hence

$$H_n = \Omega\left(\frac{n}{(\log n)^{d/(d-1)}}\right)$$

in probability. We now strengthen this bound. We begin by proving the following Poissonized version of Theorem 1. Theorem 1 is proved for a fixed number of points in Sect. 6.

Theorem 4 *Let N be a Poisson random variable with mean n. Let $X_1, X_2, \ldots, X_N$ be N independent uniform points in $[0,1]^d$. Let E_N be the complete graph with vertex set $\{X_1, \ldots, X_N\}$ and edge weights given by the Euclidean distance between X_i and X_j. Then, the height of the tree explaining Kruskal's algorithm with input E_N satisfies*

$$\lim_{n\to\infty} \mathbf{P}\{H_N(E_N) \geq cn\} = 1,$$

for some $c \in (0,1)$. Also, $\mathbf{E}H_N(E_N) = \Theta(n)$, as n goes to infinity.

The proof of the above theorem relies on techniques similar to the ones we developed in the proof of Theorem 3. Consider the forest $M_{i^\star}$ produced by Kruskal's algorithm at step $i^\star \in \{1, \ldots, n-1\}$, the largest i such that all edges of M_i have weights less than $r_n = (\lambda_0/n)^{1/d}$, for a fixed constant $\lambda_0 > 0$ to be chosen later. We prove the existence of a large component consisting of small "traps" with certain good properties, and count the number of new components that join the large one.

The analysis becomes easier if we rescale to a box of volume n. Instead of a Poisson process with rate n, consider a Poisson point process with rate 1 in the cubic box $[0, n^{1/d}]^d$ of volume n. Note that for this rescaled process, the radius of the balls is $\lambda^{1/d}$ instead of $(\lambda/n)^{1/d}$. We refer to this graph process as $G(\mathcal{P}, r)$, where now $\mathcal{P}$ is the set of Poisson points in $[0, n^{1/d}]^d$, and $r = \lambda^{1/d}$.

Define

$$t = \left\lfloor \frac{n^{1/d}}{\alpha \lambda_0^{1/d}} \right\rfloor,$$

for a constant $\alpha > 0$, and partition $[0, n^{1/d}]^d$ into exactly t^d smaller cubic boxes each of side length $\alpha\lambda_0^{1/d}(1 + O(n^{-1/d}))$. Label these smaller boxes $Q_1, Q_2, \ldots, Q_{t^d}$. The partition is shown in Fig. 3.

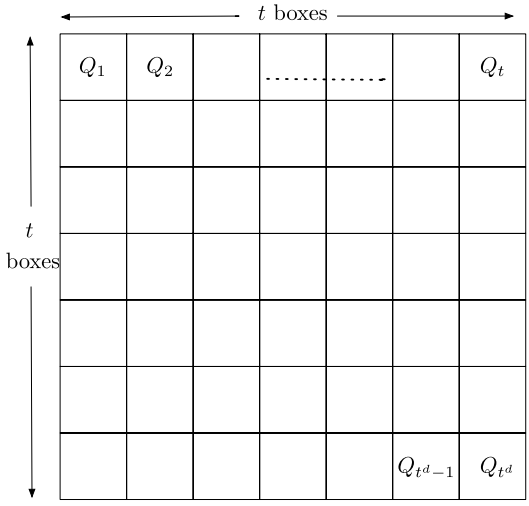

Fig. 3 The region $[0, n^{1/d}]^d$ and its division into t^d boxes, labeled $Q_1, \ldots, Q_{t^d}$

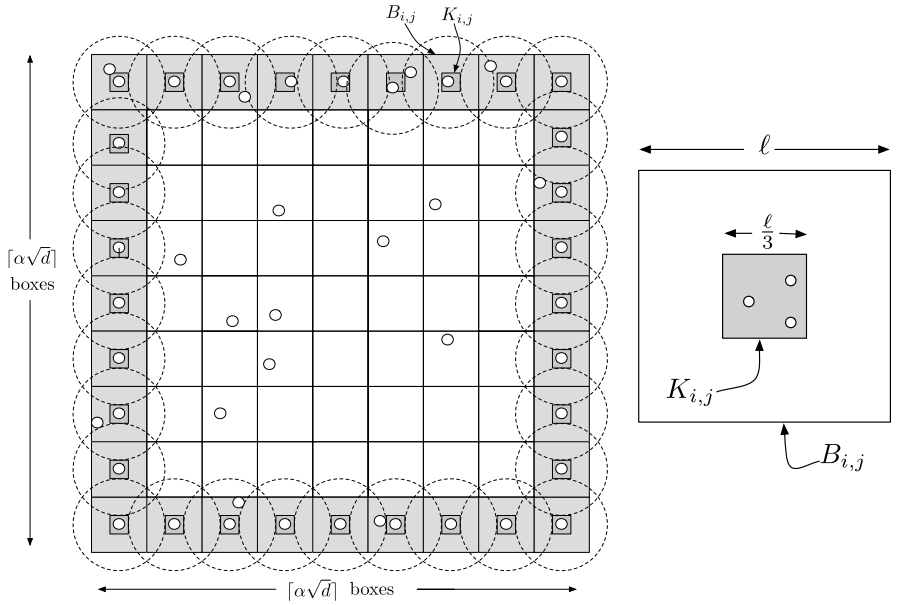

Fig. 4 The box Q_i when the event E_i is true. The *shaded region* represents the boundary region S_i of Q_i. The kernel boxes $K_{i,j}$ are shown, and the ball of radius $\lambda_0^{1/d}$ centered at a point in $K_{i,j}$. The figure also shows a close-up of a small box $B_{i,j}$, and its kernel box, $K_{i,j}$

We start by describing particular events E_i, $1 \le i \le t^d$, where each E_i depends only on the points lying in Q_i. In essence, E_i is the event that the boundary of Q_i is "covered" by balls of radius λ_0 centered at the Poisson points. This is illustrated in Fig. 3. Each box Q_i is further split into even smaller boxes, $B_{i,j}$, $1 \le j \le \lceil \alpha\sqrt{d} \rceil^d$ (see Fig. 4). Each box $B_{i,j}$ has side length ℓ such that

$$(1 - n^{-1/d})\lambda_0^{1/d}/\sqrt{d} \le \ell \le \lambda_0^{1/d}/\sqrt{d}. \tag{3}$$

Let S_i be the set of indices j for which $B_{i,j}$ touches the boundary of Q_i. Each box $B_{i,j}$, $j \in S_i$ has a kernel $K_{i,j}$, which is a box located at the center of $B_{i,j}$. The kernel boxes $K_{i,j}$ are chosen with side length $\ell/3$. The event E_i is the event that there is at least one point in each $K_{i,j}$, $j \in S_i$:

$$E_i = \bigcap_{j \in S_i} \{\mathcal{P} \cap K_{i,j} \ne \emptyset\}.$$

From now on, we refer to the *boundary* of Q_i as the region $\bigcup_{j \in S_i} B_{i,j}$.

Let us first verify that the boundary boxes $B_{i,j}$, $j \in S_i$ are connected in Q_i. Suppose $x \in K_{i,j}$, $j \in S_i$. The distance from x to a corner point of the box is at most:

$$(2/3)\ell\sqrt{d} \le (2/3)\lambda_0^{1/d},$$

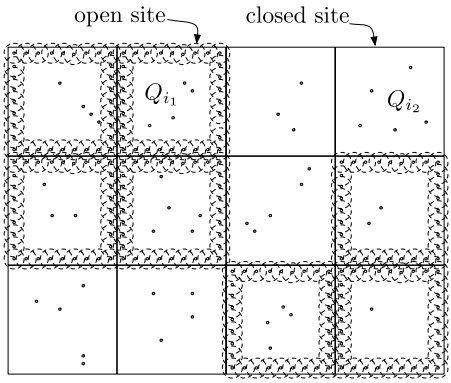

Fig. 5 A portion of the site percolation process on sites $Q_i, i \in \{1, \ldots, t^d\}$. Notice that when two open sites are neighbours, then the vertices in the boundaries of each site are in the same connected component in $G(\mathcal{P}, \lambda_0)$

where we upper bound ℓ using (3). But the radius of the ball centered at x is $r = \lambda_0^{1/d}$, implying that $B(x, r)$ will contain all corners of $B_{i,j}$. Therefore if $B_{i,j}$ and $B_{i,k}$ are neighbors and $x \in K_{i,j}$, $y \in K_{i,k}$, then the balls $B(x, r)$ and $B(y, r)$ intersect. Hence if E_i occurs, all points in the $B_{i,j}$, $j \in S_i$ are part of the same connected component in $G(\mathcal{P}, r)$. Notice also that if boxes Q_{i_1} and Q_{i_2} are neighbors in the partition of Fig. 3 and if E_{i_1} and E_{i_2} both hold, then all the points in the boundaries of each box lie in the same connected component of $G(\mathcal{P}, r)$.

Now let each box Q_i, $1 \leq i \leq t^d$, be a *site* in a (site) percolation process. A site is declared open if the event E_i holds in the corresponding box Q_i. Since the events E_i are independent and occur with the same probability, this defines a site percolation process on the finite square lattice of t^d sites. Figure 5 shows a portion of the site percolation process. An open site creates the "trap" we are looking for, since the points lying in the inside have to merge with the boundary.

The probability $\mathbf{P}\{E_i\}$ that a specified site is open is a constant depending only on λ_0 and d. More specifically, the probability that there is at least one point in a $K_{i,j}$ is at least $1 - \exp(-(\ell/3)^d)$. Thus

$$\mathbf{P}\{E_i\} \geq (1 - \exp(-(\ell/3)^d))^{|S_i|},$$

where $|S_i| \leq 2d(\alpha\sqrt{d} + 1)$. We are again interested in the existence of a large component of open sites. By Lemma 3, there exists a $p_s > 0$ such that if the probability of an open site is p, and $p > p_s$, then $\mathbf{P}\{|L_1^{site}(p)| \geq c_1 t^d\} \to 1$, as $t \to \infty$. Thus in order to guarantee that $\mathbf{P}\{E_i\} > p_s$, we require that

$$-\log(1 - p_s^{1/|S_i|}) \;<\; (\ell/3)^d. \tag{4}$$

By definition, $\ell \geq (1 - n^{-1/d})\lambda_0^{1/d}/\sqrt{d}$, and so $\ell \geq \lambda_0^{1/d}/(2\sqrt{d})$ for $n > 2$. Notice the left side of (4) is a constant, thus we can choose $\lambda_0 > 0$ large enough that satisfies (4). Let $L_1(I_n) \subseteq \{1, \ldots, t^d\}$ denote the index set of the sites in the largest open component. Then by Lemma 3,

$$\mathbf{P}\left\{|L_1(I_n)| \geq c_1 t^d\right\} \xrightarrow[n\to\infty]{} 1.$$

Next, we establish the existence of a large connected component in the random geometric graph. Let

$$C_{\lambda_0} = \left\{ x \in \mathcal{P} : x \in \bigcup_{j \in S_i} B_{i,j}, \text{for some } i \in L_1(I_n) \right\},$$

then the vertices in C_{λ_0} belong to a connected component in $G(\mathcal{P}, \lambda_0)$. It remains only to provide a lower bound on the number of merges to that connected component as Kruskal's algorithm completes (i.e., as λ increases above λ_0).

Define a new event J_i depending only on the points in Q_i in the following way: suppose $B_{i,0}$ is the mini-box at the very center of Q_i. Let $K_{i,0}$ be the kernel box associated with this $B_{i,0}$. The event J_i occurs when there is at least one point in $K_{i,0}$ *and* the rest of Q_i is free of points, except for the boundary region. In other words,

$$J_i = \left[\mathcal{P} \cap \{ B_{i_0} \setminus K_{i_0} \} = \emptyset \right] \cap \left[\mathcal{P} \cap K_{i_0} \neq \emptyset \right] \cap \left[\bigcap_{j \notin S_i, j \neq 0} \{ \mathcal{P} \cap B_{i,j} = \emptyset \} \right].$$

The event J_i is illustrated in Fig. 6, and ensures that the "trap" created by E_i is successful. As with the event E_i, $\mathbf{P}\{J_i\} > 0$ is a constant depending only on λ_0 and d. Clearly, the events J_i are independent from each other and from all the events E_i.

Let us first show that at $r = \lambda_0^{1/d}$ any points in $K_{i,0}$ are not connected to the boundary points. Set $\alpha = 6$, let $x \in K_{i,0}$ and suppose J_i is true. Observe that the distance from x to any other Poisson point in Q_i (but outside of $K_{i,0}$) is at least

$$(\alpha\sqrt{d}/2 - 2)\ell = (3 - 2/\sqrt{d})(1 - n^{-1/d})\lambda_0^{1/d} \geq 3\lambda_0^{1/d}(1 - n^{-1/d}) > 2\lambda_0^{1/d} = 2r,$$

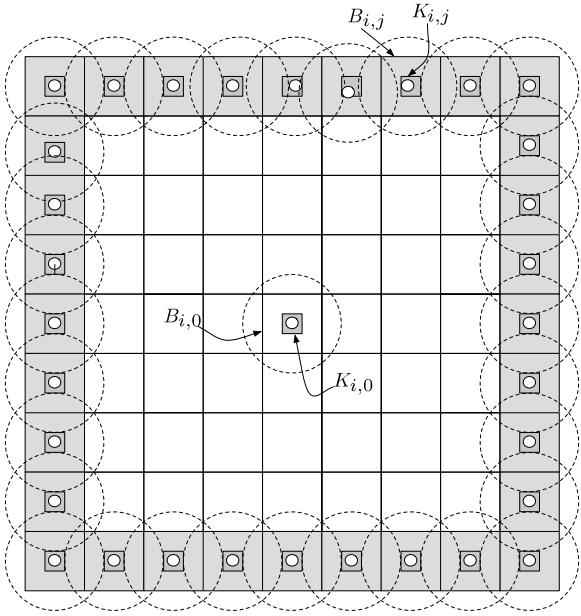

Fig. 6 Event J_i for the box Q_i. The only Poisson points are in $K_{i,0}$ and $K_{i,j}$ for $j \in S_i$

for $n^{1/d} > 3$. Therefore, the ball centered at x does not intersect any other ball in Q_i, if $n^{1/d} > 3$. In other words, $x \in K_{i,0}$ is *not* connected to the vertices of C_{λ_0} in the graph $M_{i^\star}$, but it will be at some later stage of Kruskal's algorithm.

Finally, let $M(n)$ be the number of merges to the vertices of C_{λ_0} for $i > i^\star$. Each event J_i accounts for one connection to C_{λ_0}, thus

$$M(n) \geq \sum_{i \in L_1(I_n)} J_i.$$

Using the fact that $\mathbf{P}\{|L_1(I_n)| \geq c_1 t^d\} \to 1$ and that $\mathbf{P}\{J_i\} > 0$ is constant, where J_i is independent of the event $\{i \in L_1(I_n)\}$, we conclude that

$$\mathbf{P}\left\{M(n) \geq (c_1/2)\mathbf{P}\{J_i\}\, t^d\right\} \xrightarrow[n\to\infty]{} 1.$$

Now use the definition of t and:

$$\mathbf{P}\{M(n) \geq cn\} \xrightarrow[n\to\infty]{} 1,$$

where $c = c_1 \mathbf{P}\{J_i\}/(3\alpha^d \lambda_0)$ is a constant independent of n. This finishes the proof of Theorem 4.

6 Proof of Theorem 1

In this section, we extend the result of the previous section for a fixed number of points uniformly distributed in $[0,1]^d$. In other words, we depoissonize Theorem 4 to prove Theorem 1. From now on, we use H_n and T_n instead of $H_n(E_n)$ and $T_n(E_n)$, since we are always referring to the graph E_n (defined in Sect. 1), constructed using a point set $\mathcal{X}_n = \{x_1, \ldots, x_n\}$ where each $x_i \in [0,1]^d$.

Lemma 4 (Stability of the Euclidean MST) *Let $x_1, x_2, \ldots, x_{n+1} \in [0,1]^d$. Let MST_n and MST_{n+1} be the minimum spanning trees on $\mathcal{X}_n$ and $\mathcal{X}_{n+1}$, respectively, where Euclidean distance is used for the edge weights. Assume both MST_n and MST_{n+1} are unique. Then MST_n and MST_{n+1} differ by at most $2\deg(x_{n+1}) = O(1)$ edges, where $\deg(\cdot)$ denotes the degree in MST_{n+1}.*

Proof Let N be the set of neighbours of x_{n+1} in MST_{n+1}. We can build MST_{n+1} from MST_n as follows: Add vertex x_{n+1} and add all edges from x_{n+1} to its neighbors $y \in N$. Next, we use the well known characterization of edges of the minimum spanning tree: an edge is part of the minimum spanning tree of a weighted graph G if and only if there is no cycle in G for which it is the longest edge. By adding edges from x_{n+1} to N, we created at most $|N|$ new cycles. So for each newly created cycle, delete the longest weight edge in the cycle. Doing this, we delete at most $|N|$ edges. After this process, the remaining graph is exactly MST_{n+1}, due to the characterization given above. Therefore, the number of edges that were added or deleted is bounded above by $2\deg(x_{n+1})$. In any dimension, two edges incident to the same point in the

Euclidean minimum spanning tree must define an angle of at least 60 degrees, (see, e.g., Lemma 4 of Aldous and Steele [3]). Thus the value $\deg(x_{n+1})$ is bounded above by a constant in any dimension d. □

We can now establish an almost increasing property of the height of the tree explaining Kruskal:

Lemma 5 *Let $x_1, x_2, \ldots, x_{n+1}$ be any points (not necessarily random) in $[0,1]^d$. Let H_n and H_{n+1} be the heights of the tree explaining Kruskal's algorithm for the point sets $\mathcal{X}_n = \{x_1, \ldots, x_n\}$ and $\mathcal{X}_{n+1} = \{x_1, \ldots, x_{n+1}\}$, respectively. Then, there exists some non-negative constant $a \geq 0$ independent of $\mathcal{X}_{n+1}$ such that*

$$H_{n+1} \geq H_n - a.$$

Proof Let MST_n be the minimum spanning tree on the point set $\mathcal{X}_n$ and let T_n be the tree explaining Kruskal's algorithm for this point set, and as usual H_n its height. Let $h = H_n$ for short, and let π be a path of length h in the tree T_n. Each non-leaf vertex in the path π corresponds to an *edge* in MST_n. Label these edges $e_1, \ldots, e_h$ starting from the bottom of the tree. Now consider MST_{n+1}. Not all of the edges $\{e_i, 1 \leq i \leq h\}$ are also in MST_{n+1}. Let $K \subseteq [0,h]$ be the index set for those edges also in MST_{n+1}: $K = \{k \in [1,h] : e_k \in MST_{n+1}\}$. The goal is to show that $\{e_k : k \in K\}$ all lie on a single path in the tree explaining Kruskal for MST_{n+1}. From Lemma 4, $|K| \geq h - 2\deg(x_{n+1})$, and therefore if the edges are indeed on the same path in T_{n+1}, then the height of this tree is at least $H_n - a$ for $a > 0$ a positive constant.

Let $w(e_k)$ be the weight of the edge e_k and for any $k \in K$ define C_{e_k} to be the connected component in the graph $G(\mathcal{X}_{n+1}, w(e_k))$ which contains the edge e_k. We show the following fact: The component C_{e_k} contains all "previous" edges, $\{e_j : j \in K, j \leq k\}$. It is easy to see that this fact implies that the edges $\{e_k : k \in K\}$ lie on the same path in the tree T_{n+1}, since they all grow out from the same component. And the proof of the fact is also fairly simple: By definition of the edges $\{e_i, 1 \leq i \leq h\}$, there is a connected component in $G(\mathcal{X}_n, w(e_k))$ containing all edges $\{e_j : j \leq k\}$. When we add a point, x_{n+1}, the vertices that are connected by edges of weight less than $w(e_k)$ are *still* connected—but perhaps by a different path. Therefore, all edges $\{e_j : j \in K, j \leq k\}$ are in the component C_{e_k}. □

Finally, we are ready to prove our main theorem:

Proof of Theorem 1 Let $\epsilon \in (0,1)$. Let H_N be the height of the tree explaining Kruskal, for a Poisson point process on $[0,1]^d$ with rate $n(1-\epsilon)$. Then by Theorem 4, there exists a constant $c > 0$ such that $\mathbf{P}\{H_N \geq cn(1-\epsilon)\} \to 1$, as $n \to \infty$. Either N is close enough to its mean, and then we can apply Lemma 5, or it is far from the mean, but this happens only with low probability. More precisely, let $A(n,\epsilon)$ be the event that $n - 2\epsilon n \leq N \leq n$ and let $A(n,\epsilon)^c$ be its complement.

$$\mathbf{P}\{H_N \geq cn(1-\epsilon)\} \leq \mathbf{P}\left\{A(n,\epsilon)^c\right\} + \mathbf{P}\{H_N \geq cn(1-\epsilon),\ A(n,\epsilon)\}.$$

Now, if $A(n,\epsilon)$ holds then $H_n \geq H_N - a(n-N)$, by Lemma 5. Therefore,

$$\mathbf{P}\{H_N \geq cn(1-\epsilon),\ A(n,\epsilon)\} \leq \mathbf{P}\{H_n \geq cn(1-\epsilon) - a(n-N),\ A(n,\epsilon)\}.$$

Pick $\epsilon \in (0, \frac{c}{2(c+2a)})$. This guarantees that

$$\mathbf{P}\{H_n \geq cn(1-\epsilon) - 2an\epsilon\} \leq \mathbf{P}\{H_n \geq cn/2\}.$$

Finally, we need a bound on the probability that N is far from its mean. By Chernoff's bound [9],

$$\mathbf{P}\left\{A(n,\epsilon)^c\right\} \leq \mathbf{P}\{|N - \mathbf{E}N| \geq \epsilon n\} \leq 2e^{-\gamma_\epsilon n},$$

where $\gamma_\epsilon = \epsilon - \ln(1-\epsilon) > 0$. Putting everything together, for this range of ϵ,

$$\mathbf{P}\{H_N \geq cn(1-\epsilon)\} \leq \mathbf{P}\left\{A(n,\epsilon)^c\right\} + \mathbf{P}\{H_n \geq cn/2\}.$$

And thus,

$$\mathbf{P}\{H_n \geq cn/2\} \ \geq\ \mathbf{P}\{H_N \geq cn(1-\epsilon)\} - 2e^{-\gamma_\epsilon n}.$$

Letting $n \to \infty$ proves that $\mathbf{P}\{H_n \geq cn/2\} \to 1$. □

Acknowledgement We thank Michael Drmota for directing us towards this problem, and two anonymous referees for their helpful comments.

References

1. Aho, A.V., Hopcroft, J.E., Ullman, J.D.: The Design and Analysis of Computer Algorithms. Addison-Wesley, Boston (1974)
2. Aldous, D.: A random tree model associated with random graphs. Random Struct. Algorithms **4**, 383–402 (1990)
3. Aldous, D., Steele, J.M.: Asymptotics for euclidean minimum spanning trees on random points. Probab. Theory Relat. Fields **92**, 247–258 (1992)
4. Bollobás, B.: Random Graphs, 2nd edn. Cambridge Studies in Advanced Mathematics. Cambridge University Press, Cambridge (2001)
5. Bollobás, B., Simon, I.: On the expected behavior of disjoint set union algorithms. In: STOC '85: Proceedings of the Seventeenth Annual ACM Symposium on Theory of Computing, pp. 224–231. ACM, New York (1985)
6. Bollobás, B., Simon, I.: Probabilistic analysis of disjoint set union algorithms. SIAM J. Comput. **22**, 153–1074 (1993)
7. Borgs, C., Chayes, J.T., Kesten, H., Spencer, J.: The birth of the infinite cluster: Finite-size scaling in percolation. Commun. Math. Phys. **224**, 153–204 (2001)
8. Borůvka, O.: O jistém problému minimálním. Práce Mor. Přírodověd. Spol. v Brně (Acta Soc. Sci. Natur. Moravicae) **3**, 37–58 (1926)
9. Chernoff, H.: A measure of asymptotic efficiency for tests of a hypothesis based on the sum of observations. Ann. Math. Stat. **23**, 493–507 (1952)
10. Dijkstra, E.: A note on two problems in connection with graphs. Numer. Math. **1**, 269–271 (1959)
11. Erdős, P., Rényi, A.: On the evolution of random graphs. Publ. Math. Inst. Hung. Acad. Sci. **5**, 17–61 (1960)
12. Frieze, A.M.: On the value of a random minimum spanning tree problem. Discrete Appl. Math. **10**, 47–56 (1985)
13. Grimmett, G.R.: Percolation, 2nd edn. A Series of Comprehensive Studies in Mathematics, vol. 321. Springer, New York (1999)

14. Janson, S., Łuczak, T., Ruciński, A.: Random Graphs. Wiley, New York (2000)
15. Jarńik, V.: O jistém problému minimálnim. Práce Mor. Přírodověd. Spol. v Brně (Acta Soc. Sci. Natur. Moravicae) **6**, 57–63 (1930)
16. Kesten, H.: Percolation Theory for Mathematicians. Birkhäuser, Boston (1980)
17. Knuth, D.E., Schönhage, A.: The expected linearity of a simple equivalence algorithm. Theor. Comput. Sci. **6**, 281–315 (1978)
18. Kruskal, J.B.: On the shortest spanning subtree of a graph and the traveling salesman problem. Proc. Am. Math. Soc. **2**, 48–50 (1956)
19. McDiarmid, C., Johnson, T., Stone, H.S.: On finding a minimum spanning tree in a network with random weights. Random Struct. Algorithms **10**(1–2), 187–204 (1997)
20. Meester, R., Roy, R.: Continuum Percolation. Cambridge Tracts in Mathematics. Cambridge University Press, Cambridge (1996)
21. Penrose, M.: The longest edge of a random minimal spanning tree. Ann. Appl. Probab. **7**(2), 340–361 (1997)
22. Penrose, M.: Random minimal spanning tree and percolation on the n-cube. Random Struct. Algorithms **12**, 63–82 (1998)
23. Penrose, M.: A strong law for the longest edge of the minimal spanning tree. Ann. Probab. **27**(1), 246–260 (1999)
24. Penrose, M.: Random Geometric Graphs. Oxford Studies in Probability. Oxford University Press, Oxford (2003)
25. Prim, R.C.: Shortest connection networks and some generalizations. Bell Syst. Tech. J. **36**, 1389–1401 (1957)
26. Steele, J.M.: Probability Theory and Combinatorial Optimization. CBMS-NSF Regional Conference Series in Applied Mathematics. SIAM, Philadelphia (1997)
27. Yao, A.C.: On the average behavior of set merging algorithms. In: STOC '76: Proceedings of the 8th Annual ACM Symposium on Theory of Computing, pp. 192–195 (1976)
28. Yukich, J.E.: Probability Theory of Classical Euclidean Optimization Problems. Lecture Notes in Mathematics, vol. 1675. Springer, New York (1998)