Online Random Weight Minimal Spanning Trees and a Stochastic Coalescent

David Aldous, Omer Angel, and Nathanael Berestycki

Draft version — June 16, 2008

Abstract

Let $\mathbf{cost}(n)$ be the minimal mean total cost of constructing a spanning tree of the complete n-vertex graph using edges with i.i.d. $U(0,1)$ weights, when decisions on acceptance or rejection of edges must be made online. We give asymptotic upper and lower bounds of 4 and 1.368 on $\mathbf{cost}(n)$. The key idea is that for each n we can calculate $\mathbf{cost}(n)$ via a recursion over partitions Π of $\{1,\ldots,n\}$ for $F(\Pi) :=$ mean cost if starting with a forest with components Π. Though we cannot solve the recursion analytically, a related inequality allows us to get explicit upper bounds. The optimal algorithm is analogous to a certain time-inhomogeneous stochastic coalescent. We conjecture that $\lim_n \mathbf{cost}(n)$ exists and is determined by the solution of a certain p.d.e. on the infinite simplex. Numerics suggest $\lim_n \mathbf{cost}(n) \approx 1.9$. We give lower bounds by different methods, variants on classical optimal online selection results.

1 Introduction

[[Note(david): Have reverted to traditional $U(0,1)$ law for edge-weights]]
[[Note(david): We are using "costs" or "weights" interchangeably]]

To edges e of the complete n-vertex graph K_n assign i.i.d. weights W_e distributed uniformly on $[0,1]$. A celebrated result of Frieze [5] asserts that the weight X_n^{offline} of the minimal spanning tree (MST) satisfies

$$\mathbb{E}X_n^{\text{offline}} \to \zeta(3) = 1.20...... \qquad (1)$$

We study the corresponding *online* problem in which edges are presented one by one and we must decide then whether to accept the edge for our spanning tree (see precise model description below). In recent related work, [4] study the two-stage process where one can choose some edges from one realization (W'_e) and then must choose remaining edges from an independent realization (W''_e): they show the limit mean weight is bounded above $\zeta(3)/2$. And [8] study an adversarial version in which an adversary can choose the order of edges (without knowing their weights); they show the mean weight is still $O(1)$.

Here are the two models we study.

The combinatorial model. The $\binom{n}{2}$ edges e are presented in uniform random order at times $1, 2, \ldots, \binom{n}{2}$ with i.i.d. $U(0,1)$ weights W_e.

The Poisson model. Here we use the continuous time interval $0 \le t \le 1$ and the infinite "weight" interval $\mathbb{R}^+$. Take a Poisson point process with density $\binom{n}{2}$ on $[0,1] \times \mathbb{R}^+$. For each point (t, x) of the process we are offered at time t a randomly chosen edge, assigned weight x.

[[Note(david): Note I'm using "Poisson model" for the $\mathbb{R}^+$ case.]]

The combinatorial model is intuitively more natural, but the Poisson model is technically simpler, because it has two properties that the combinatorial model lacks (see section 2.1): a scaling property and a Markovian property.

Why should we expect similar behavior in these two models? The optimal spanning tree will use edges of typical weight $O(1/n)$. Indeed, for the weight V_n^{offline} of a random edge of the offline MST, there is an explicit limit distribution for $n \times V_n^{\text{offline}}$: see e.g. [2] sec. 4.5. So truncating the weight space in the Poisson model from $\mathbb{R}^+$ to $(0,1)$ should have no asymptotic effect. In this truncated Poisson model, each edge e is presented a random (Poisson, mean 1) number of times with i.i.d. weights. This still may appear different from the combinatorial model, in which each e is presented exactly once. But if we consider only $O(1/n)$ weights, then in each model these are assigned to essentially the same number $O(n)$ of random edges in random order, and these are the edges which will be used.

[[Note(david): We don't yet have a complete proof that the costs are asymptotically the same in the two models; section 4 gives one side of proof.]]

Under either model let $\mathbf{cost}(n)$ be the minimal expected total cost, over all online algorithms, of the spanning tree constructed by the algorithm. Our fundamental result, Theorem 3, is that under the Poisson model one can derive $\mathbf{cost}(n)$ from equations for the expected cost $F(\Pi)$ of the optimal online algorithm as a function of the starting components Π. Though we cannot get an useful explicit expression for $\mathbf{cost}(n)$, by finding solutions F to an associated system of *inequalities* we can get explicit upper bounds, of which the simplest (Corollary 4) is $\mathbf{cost}(n) \le 4(n-1)/n$, and so in particular

Theorem 1. *In the Poisson model,* $\limsup \mathbf{cost}(n) \le 4$.

We conjecture that $\lim_n \mathbf{cost}(n)$ exists and is determined by the solution of a certain partial differential equation on the infinite simplex: see section 2.3. Numerics suggest $\lim_n \mathbf{cost}(n) \approx 1.9$. Our best lower bound is proved in section 3 via different methods, variants on classical optimal online selection results.

Theorem 2. *In either model,* $\liminf \mathbf{cost}(n) \ge 1.368\ldots$

Let us briefly describe an intriguing connection with stochastic coalescence. See [1] for general background and [7] for the following kind of technical results on stochastic coalescence. Given a kernel $K(x, y)$, $x, y = 1, 2, 3, \ldots$, we can for each $n \ge 2$ define the *Marcus-Lushnikov process* on partitions of $\{1, \ldots, n\}$ as the continuous-time Markov chain in which each pair of size x and size y components may merge at rate $n^{-1}K(x,y)$. Under technical conditions, the random process $(Y_i^n(t), i \geq 1)$ where $Y_i^n(t) = n^{-1}\times$(number of size-i components at time t) converges as $n \to \infty$ to a deteministic limit $(y_i(t), i \geq 1)$ which is the solution of the *Smoluchowski coagulation equation* (SCE). Now our optimal online algorithm is (Theorem 3(b)) for each n conceptually similar to some time-inhomogeneous Marcus-Lushnikov process. So the conjecture that $\lim_n \mathbf{cost}(n)$ is determined by the solution of the PDE (9) arises as a analog of convergence of Marcus-Lushnikov processes to SCE. However, in the latter setting we use the same, given $K(x,y)$ for each n, whereas in our setting the coalescence rates ($\theta_{i,j,t}$ at (5)) are defined in parallel with the algorithm in terms of the solutions of a set of equations, separately for each n, making it much harder to get started on a convergence proof.

2 Exact formulas and upper bounds

2.1 The recursive equation for optimal cost

The key insight is Theorem 3, which gives equations for the mean cost of the optimal online algorithm as a function of the starting configuration. We work with the Poisson model, so the time interval is $0 \leq t \leq 1$, and it is convenient to reverse direction of time so that "time t" means there is time t remaining. In this section, n is fixed. The proof below assumes that $\mathbf{cost}(n)$ is finite, which is easy to prove directly but is also a corollary of known (less easy) results: see section 2.2.

The Poisson model has the following two features that make it technically simpler than the combinatorial model. The **scaling property** is the fact that the map $(t,x) \to (at, x/a)$ from $[0,1]\times\mathbb{R}^+$ to $[0,a]\times\mathbb{R}^+$ preserves the distribution of the Poisson point process. In words, working on a time interval of length a, has the same effect as dividing all weights by a, hence the optimal algorithm's cost will be divided by a. The **Markov partition property** is that the current state of the optimal algorithm can be represented by the partition of $\{1,\ldots,n\}$ into tree-components; we do not need to keep track of which edges were rejected previously.

To set up some notation, write Π for a partition of $\{1,\ldots,n\}$ into $\#\Pi$ components of sizes $X_1(\Pi),\ldots,X_{\#\Pi}(\Pi)$. For $1 \leq i < j \leq \#\Pi$, let $\Pi^{i,j}$ denote the partition obtained from Π by merging the i'th and j'th components of Π. Write $[n]$ for the partition into the single component $\{1,2,\ldots,n\}$ and write $[11\ldots11]$ for the partition into n components of size 1. Consider a non-negative function F of partitions satisfying

$$F(\Pi) > F(\Pi^{i,j}) \text{ for all } \Pi \text{ and all } 1 \leq i < j \leq \#\Pi; \quad F([n]) = 0. \qquad (2)$$

For any such F define

$$F_t(\Pi) = \frac{1}{t}F(\Pi),\ 1 \geq t > 0. \qquad (3)$$

Theorem 3. *(a) Recursively on $\#\Pi \geq 2$ define $F(\Pi)$ to be the minimal solution satisfying (2) of the quadratic equation*

$$F(\Pi) = \sum_{i<j} X_i(\Pi) X_j(\Pi) \frac{(F(\Pi) - F(\Pi^{i,j}))^2}{2}. \tag{4}$$

Then $F(\Pi)$ equals the optimal cost of constructing an online spanning tree starting with a forest with components Π. In particular, $\mathbf{cost}(n) = F([11\ldots 11])$.
(b) The unique algorithm attaining $\mathbf{cost}(n)$ *is of the form: at time t, accept an edge connecting components i and j of the current partition Π_t if its weight is smaller than*

$$\theta_{i,j,t} := F_t(\Pi_t) - F_t(\Pi_t^{i,j}). \tag{5}$$

(c) Let $F(\Pi)$ be any function on partitions satisfying (2) and

$$F(\Pi) \leq \sum_{i<j} X_i(\Pi) X_j(\Pi) \frac{(F(\Pi) - F(\Pi^{i,j}))^2}{2} \quad \textit{for all } \Pi. \tag{6}$$

Then $\mathbf{cost}(n) \leq F([11\ldots 11])$.

[[Note(david): Previous version asserted "maximal solution" in (a) but I don't see an argument. Of course the solution may be unique under (2) but again I don't see an argument. The logical layout of the "minimality" proof below is slightly subtle.]]

Proof. Define $\hat{F}(\Pi)$ to be the optimal cost of constructing an online spanning tree starting with a forest with components Π. Clearly $\hat{F}$ satisfies (2). The scaling property implies that for $1 \geq t > 0$, the optimal cost starting at time t with components Π equals $\hat{F}_t(\Pi) := \frac{1}{t}\hat{F}(\Pi)$.

Fix Π and consider an optimal algorithm starting from Π at time 1. Let W be the total cost and Π_t be the partition at time t. Let $\mathcal{G}_t$ denote the filtration generated by the edges and weights seen during time $[1, t]$. Then $M_t = \mathbb{E}(W|\mathcal{G}_t)$ is a martingale. Note we can write $M_t = S_t + \hat{F}_t(\Pi_t)$ where S_t is the sum of all payments during time $[1, t]$. Note also that an optimal algorithm must be of the form

> at time t, accept an edge connecting components i and j of the current partition Π_t if its weight is smaller than
>
> $$\theta_{i,j,t} := \hat{F}_t(\Pi_t) - \hat{F}_t(\Pi_t^{i,j}).$$

So for small ε,

$$\begin{aligned} \mathbb{E}(M_{1-\varepsilon} - M_1) &= \mathbb{E}S_{1-\varepsilon} + \sum_{i<j} X_i(\Pi) X_j(\Pi) \theta_{i,j,1} \varepsilon [\hat{F}(\Pi^{i,j}) - \hat{F}(\Pi)] \quad (7) \\ &\quad + (\hat{F}_{1-\varepsilon}(\Pi) - \hat{F}(\Pi))(1 - O(\varepsilon)) + o(\varepsilon). \end{aligned}$$

Now

$$\mathbb{E}S_{1-\varepsilon} = \sum_{i<j} X_i(\Pi) X_j(\Pi) \frac{\theta^2_{i,j,1}}{2} \varepsilon + o(\varepsilon)$$

and using the definition of $\theta_{i,j,1}$ we see that the sum of the two terms on the right of (7) equals $-\sum_{i<j} X_i(\Pi)X_j(\Pi)\frac{\theta^2_{i,j,1}}{2}\varepsilon$. The scaling property implies $(F_{1-\varepsilon}(\Pi) - F(\Pi)) = F(\Pi)\varepsilon + o(\varepsilon)$. So we have shown

$$0 = \mathbb{E}(M_{1-\varepsilon} - M_1) = -\sum_{i<j} X_i(\Pi) X_j(\Pi) \frac{\theta^2_{i,j,1}}{2} \varepsilon + \hat{F}(\Pi)\varepsilon + o(\varepsilon), \qquad (8)$$

implying $\hat{F}(\Pi) = \sum_{i<j} X_i(\Pi)X_j(\Pi)\frac{\theta^2_{i,j,1}}{2}$. In other words, $\hat{F}$ satisfies equation (4). The minimality assertion in (a) will follow from part (c), as explained later, and then assertion (b) will follow from the construction.

To prove (c) we re-use the same calculations but intepret them slightly differently. Suppose a function $F(\Pi)$ satisfies (6) and (2). Define F_t by the scaling (3) and consider the algorithm specified by the threshold $\theta_{i,j,t}$ defined by (5). Write Π_t for the partition at time t, and define the accumulated cost S_t as before. Define a process

$$M_t := S_t + F_t(\Pi_t) \ , 1 \geq t > 0.$$

Repeating the calculations that gave (8) shows that, for the algorithm started with partition Π,

$$\mathbb{E}(M_{1-\varepsilon} - M_1) = -\sum_{i<j} X_i(\Pi) X_j(\Pi) \frac{\theta^2_{i,j,1}}{2} \varepsilon + F(\Pi)\varepsilon + o(\varepsilon).$$

By inequality (6) and scaling, this implies that $\mathbb{E}M_t$ is non-increasing as t decreases over $1 \geq t > 0$, and so

$$F(\Pi) = \mathbb{E}M_1 \geq \lim_{t\downarrow 0} \mathbb{E}M_t = \mathbb{E}S_0 + \lim_{t \downarrow 0} \mathbb{E}F_t(\Pi_t) \geq \mathbb{E}S_0.$$

[[**Note(david): Curiously, the argument so far has not proved or assumed that the algorithms do produce spanning trees! All we have assumed is that some algorithm has finite mean cost.**]] From the strict inequality in the side conditions (2) it is easy to check that the algorithm does produce a spanning tree (not just a forest), because the threshold grows as $1/t$ as $t \downarrow 0$. So $\mathbb{E}S_0$ is the mean total cost of this algorithm starting with Π. So we have proved, in setting (c), that

$$F(\Pi) \geq \hat{F}(\Pi) := \text{ cost of optimal algorithm starting with } \Pi.$$

Applying this recursively to a F satisfying the *equality* (4) proves the minimality assertion in (a). Applying instead to a F satisfying the *inequality* (6) and to $\Pi = [11\ldots11]$ proves (c). □

[[**Note(david): Proving $\lim_{t\downarrow 0} \mathbb{E}F_t(\Pi_t) = 0$ would not help with uniqueness. It would just show that, if F satisfies the equations (4), then running the algorithm associated with F has cost $F(\Pi)$.**]]

Remarks. (a) Computing $F([11\ldots11])$ for $n \leq 73$ suggests that $\mathbf{cost}(n)$ is decreasing after $n = 4$ and converges slowly to roughly 1.9.
(b) In the case $n = 2$, (4) becomes the single equation $f = f^2/2$ and so $\mathbf{cost}(2) = 2$. This is a classical "optimal online selection" result, the continuous version of Proposition 6 with $n = 1$.
(c) Theorem 3(b) shows there is a unique algorithm attaining the minimum cost, and so it makes sense to talk about the random variable X_n^{online} giving the total cost in a realization. Of course $\mathbf{cost}(n) = \mathbb{E}X_n^{\text{online}}$. There is also a random variable V_n^{online} giving the weight of a random edge of the constructed spanning tree, and it is natural to conjecture that (as in the offline case) there is a limit distribution for $n \times V_n^{\text{online}}$.

2.2 Upper bounds

The fact that $\sup_n \mathbf{cost}(n) < \infty$ can be deduced (in the combinatorial model) from the "adversaries" result in [8], because it's just the special case where the adversary picks edges in random order. But our Theorem 3(c) can be used to give more explicit bounds (in the Poisson model).

Corollary 4. $\mathbf{cost}(n) \leq 4(n-1)/n$.

Proof. Consider component sizes $x_1, x_2, \ldots, x_{b+1}$ in a partition of $\{1, \ldots, n\}$ into $b+1$ components. The maximum possible value of $\sum_i x_i^2$ occurs when the sizes are $n-b, 1, 1, \ldots, 1$, so we always have $\sum_i x_i^2 \leq (n-b)^2 + b$. So

$$\begin{aligned}\sum_{i<j} x_i x_j &= \tfrac{1}{2}(n^2 - \sum_i x_i^2) \\ &\geq \tfrac{1}{2}(n^2 - (n-b)^2 - b) \\ &= b(n - \tfrac{1}{2}(b+1)) \\ &\geq \tfrac{bn}{2}\end{aligned}$$

the final equality because $b + 1 \leq n$. We want to check that the inequalities (6) hold for

$$F(\Pi) := c(\#\Pi - 1)$$

and this reduces to checking

$$cb \leq \tfrac{c^2}{2}\ \tfrac{bn}{2} \text{ for all } b.$$

This is true for $c = 4/n$, and so by Theorem 3(c)

$$\mathbf{cost}(n) \leq F([11\ldots11]) = 4(n-1)/n.$$

□

In fact, we can take any $F(\Pi)$ satisfying (2), then by considering the smallest c such that cF satisfies (6), Theorem 3(c) implies

Corollary 5.

$$\mathbf{cost}(n) \le F([11\ldots 11]) \times \max_{\Pi} \frac{F(\Pi)}{\sum_{i<j} X_i(\Pi) X_j(\Pi) \frac{(F(\Pi)-F(\Pi^{i,j}))^2}{2}}.$$

[[**Note(david): I assume we can do better with another choice of F, such as $\sum_i 1/X_i(\Pi)$.**]]

2.3 A limit PDE

In section 2.1 n was fixed, and we represented a partition Π of $\{1,\ldots,n\}$ via its component sizes $(X_1,\ldots,X_{\#\Pi})$. To study the optimal algorithm as n varies we use a different representation. Associate a partition Π with the vector $\mathbf{y} = (y_1, y_2, \ldots)$ where

$$y_i = \tfrac{i}{n}\#\{j : \#X_j = i\} = \text{ proportion of elements in size-}i\text{ components.}$$

Note that the set $\mathcal{D}_n$ of possible such vectors $\mathbf{y}$ is a subset of the infinite dimensional simplex $\mathcal{D}_\infty := \{\mathbf{y} :\ y_i \ge 0\ \forall i,\ \sum_i y_i \le 1\}$. Translating the definition of $\Pi^{i,j}$, we define $\mathbf{y}_n^{i,j}$ as the vector $\mathbf{y} + n^{-1}((i+j)\delta_{i+j} - i\delta_i - j\delta_j)$. Theorem 3 translates to

$$F_n(\mathbf{y}) = n^2 \sum_{i<j} y_i y_j \frac{(F_n(\mathbf{y}) - F_n(\mathbf{y}^{i,j}))^2}{2} + \sum_i \binom{ny_i}{2} \frac{(F_n(\mathbf{y}) - F_n(\mathbf{y}^{i,i}))^2}{2}; \quad \mathbf{y} \in \mathcal{D}_n.$$

This suggests formulating the intuitively corresponding $n \to \infty$ limit equation for a function G from $\mathcal{D}_\infty$ to $\mathbb{R}^+$:

$$G(\mathbf{y}) = \sum_i \sum_j \tfrac{1}{4} ij y_i y_j G_{ij}^2(\mathbf{y}),\ \mathbf{y} \in \mathcal{D}_\infty \tag{9}$$

where G_{ij} denotes partial derivative. If we impose boundary conditions analogous to (2):

$$(i+j)G_{i+j} - iG_i - jG_j < 0; \quad G(0,0,0,\ldots) = 0$$

then it is natural to conjecture that (9) has a unique solution $G(\mathbf{y})$, that $\lim_n F_n(\mathbf{y}) = G(\mathbf{y})$ and therefore $\lim_n \mathbf{cost}(n) = G(1,0,0,\ldots)$.

[[**Note(david): A first step in proving the above would be to prove the following claim. At first I thought some simple coupling argument would work, but it now seems more difficult . . .**]]

Claim: Let $\mathbf{y}^n, \mathbf{z}^n \in \mathcal{D}_n$ be such that

$$\lim_n y_i^n = \lim_n z_i^n\ \forall i.$$

Then $\lim_n (F_n(\mathbf{z}^n) - F_n(\mathbf{y}^n)) = 0$.

3 Lower bounds

3.1 A classical selection problem

The discrete version of Proposition 6 is due to Moser [6]: see [3] for history and connection to the better known *secretary problem*. The continuous version is (at least) folklore [[**Note(david): which means I'm too lazy to look for references . . .**]]

Proposition 6. **[Discrete version].** *Over all online algorithms for selecting exactly one item* I *from* n *items with i.i.d.* $U(0,1)$ *weights* W_i*, let* $C_0(n)$ *be the minimum value of the expected weight* $\mathbb{E}W_I$*. Then* $C_0(n) \sim 2/n$ *as* $n \to \infty$*.*
[Continuous version]. *Consider a Poisson process of points* $(t, x(t))$ *with density* n *on* $[0,1] \times \mathbb{R}^+$*. Over all online algorithms for selecting exactly one point* $(T, x(T))$*, let* $C_0^*(n)$ *be the minimum value of the expected weight* $\mathbb{E}x(T)$*. Then* $C_0^*(n) = 2/n$*.*

Proof. [Discrete version]. The key idea is that it is easy to describe the optimal algorithm in terms of its expected cost. If we reject the first offer we are faced with the same problem with one less random variable, so the expected cost on rejection is $C_0(n-1)$. If we accept the first offer we have no further cost afterwards, so the threshold for accepting the first offer is $C_0(n-1)$. This gives a recursion

$$C_0(n) = \frac{C_0(n-1)^2}{2} + (1 - C_0(n-1))C_0(n-1) = C_0(n-1) - \frac{C_0(n-1)^2}{2},$$

which determines the sequence. One can transform it into the equivalent recursion $\frac{1}{C_0(n)} = \frac{1}{C_0(n-1)} + \frac{1}{2 - C_0(n-1)}$, and it easily follows that $C_0(n) \sim 2/n$.

[Continuous version]. By the scaling property for this Poisson process, $C_0^*(n) = n^{-1}C_0^*(1)$, so we may take take $n = 1$. Define $h(t)$ to be the minimum expected value in the problem restricted to time $[t, 1]$. As in the discrete case, the optimal algorithm is of the form

select the point (t, x) with smallest t subject to $x \leq h(t)$.

The time T of selection has

$$G(t) := \mathbb{P}(T > t) = \exp(-\int_0^t h(s)\ ds)$$

and so from the definition of $h(t)$

$$h(t) = \int_t^1 \tfrac{1}{2}h(s) \times \mathbb{P}(T \in ds | T > t) = \int_t^1 \tfrac{1}{2}h(s) \times \tfrac{h(s)G(s)}{G(t)}\ ds.$$

Since $G'/G = -h$ we find

$$-\tfrac{G'(t)}{G(t)} = h(t) = \int_t^1 \tfrac{1}{2}(\tfrac{G'(s)}{G(s)})^2 \times \tfrac{G(s)}{G(t)}\ ds.$$

This reduces to $G'(t) = -\frac{1}{2}\int_t^1 \frac{(G'(s))^2}{G(s)}\, ds$ and then to $G'' = \frac{(G')^2}{2G}$. The boundary conditions $G(0) = 1,\ G(1) = 0$ determine the solution $G(t) = (1-t)^2$. This gives $h(t) = 2/(1-t)$ and then

$$C_0^*(1) = h(0) = \int_0^1 \tfrac{1}{2}h^2(s)G(s)\ ds = \int_0^1 \tfrac{1}{2}\tfrac{4}{(1-s)^2}\ (1-s)^2\ ds = 2.$$

□

[[Note(david): The continuous proof above can be shortened by noting that, by scaling, h must be of the form $\gamma/(1-t)$. But one still needs a calculation to find γ.]]

Let us also record the much simpler result where the constraint is on the mean number of items selected. For the rest of section 3 we treat the combinatorial model; the Poisson model is similar but simpler.

Lemma 7. *Fix n and a real $a \in [0, n]$. Consider algorithms for selecting a (possibly empty) subset S from n items with i.i.d. $U(0,1)$ weights W_i, with constraint $\mathbb{E}|S| \ge a$. Let $C_1(a,n)$ be the minimal mean cost over all (offline) such algorithms. Then $C_1(a,n) = \frac{a^2}{2n}$, and this is attained by the online algorithm "choose all items with $W_i < \frac{a}{n}$".*

Proof. **[[Note(david): Simpler than Cauchy-Schartz?]]** Clearly the stated online algorithm satisfies the constraint and has mean cost $\frac{a^2}{2n}$. Consider some offline algorithm, and let $\nu(\cdot) = \sum_i \mathbb{P}(i \in S, W_i \in \cdot)$. Then ν has total mass $\ge a$, and $\nu(\cdot) \le n\Lambda(\cdot)$ for the $U(0,1)$ law Λ; subject to these constraints on a measure ν, the associated mean cost $\int x\nu(dx)$ is clearly minimized when $\nu(\cdot) = n\Lambda(\cdot \cap [0, a/n])$, as for the online algorithm. □

3.2 A quick lower bound

This is a digression to give a simple argument showing that the online case is different from the offline case.

Proposition 8.

$$\liminf_{n\to\infty} \mathbf{cost}(n) \ge \frac{5}{4} > \zeta(3) = 1.202\ldots$$

Proof. We divide each edge as two half-edges each of which is attached to one of the endpoints. Each half-edge has the same weight as the full edge. The only properties of spanning trees we use are that a spanning tree contains at least one half-edge attached to each vertex, and $2n-2$ half-edges altogether.

Consider an online algorithm to generate a spanning tree T on K_n. We separate the half-edges of the tree into sets A, B. The first half-edge attached to a vertex is put into A, and all later ones are put into B. Note that the first half-edge attached to a vertex is selected with no knowledge of unseen edges attached to the vertex, so by Proposition 6 the expected weight of the half-edge in A attached to each i is at least $C_0(n-1) = (2+o(1))/n$. Thus $W(A)$ (:= the sum of weights of half-edges in A) satisfies $\mathbb{E}W(A) \geq 2 + o(1)$.

For the half-edges in B we only use the fact that $|B| = n-2$, so $W(B) \geq W(B')$, where B' contains the $n-2$ lightest half-edges in the graph. Now, B' contains both halves of the lightest $\lfloor (n-2)/2 \rfloor$ edges in the graph (and another half-edge if n is odd). Since the k'th lightest edge out of $N = \binom{n}{2}$ has expected weight $k/(1+N) \approx 2k/n^2$, we find that $\mathbb{E}W(B') \geq 1/2 + o(1)$ (remembering to count both halves of each edge).

Now, $W(T) = \frac{1}{2}(W(A) + W(B))$, since each edge of T is counted twice in A, B, so $\mathbb{E}W(T) > 5/4 + o(1)$. □

3.3 Proof of Theorem 2

Theorem 2, the best numerical lower bound we know, is obtained by combining the three lemmas stated next.

Definition. Consider algorithms for selecting a subset S from n items with i.i.d. $U(0,1)$ weights W_i, with constraints

$$|S| \geq 1; \quad \mathbb{E}|S| = 2. \tag{10}$$

Let $C_2(n)$ be the minimal mean cost $\mathbb{E}\sum_{i \in S} W_i$ over all online such algorithms.

Lemma 9. $\liminf_n \mathbf{cost}(n) \geq \frac{1}{2} \liminf_n nC_2(n)$.

Lemma 10. *$2C_2(n)$ equals the minimum, over $1 = q_0 \geq q_1 \geq \ldots \geq q_{n-1} \geq q_n = 0$, of*

$$\sum_{k=1}^{n} \frac{(q_{k-1} - q_k)^2}{q_{k-1}} \; + \; \frac{1}{n - \sum_{k=0}^{n} q_k}.$$

Lemma 11. *Define γ_* as the minimum of*

$$\int_0^1 \frac{(G'(t))^2}{G(t)} \, dt + \frac{1}{1 - \int_0^1 G(t) \, dt} \tag{11}$$

over functions G constrained by

$$G(0) = 1; \quad G(1) = 0; \quad G \textit{ decreasing, absolutely continuous.} \tag{12}$$

Then $\gamma_* = 5.47494...$ is determined by equation (18). Moreover

$$\lim_n nC_2(n) = \gamma_*/2. \tag{13}$$

So Theorem 2 holds with limit lower bound $\gamma_*/4$. The key idea is that, instead of trying to analyze directly the finite-n discrete optimization problem in Lemma 10, we analyze the $n \to \infty$ limit continuous optimization problem in Lemma 11.

Proof of Lemma 9. For any spanning tree on K_{n+1}, the set S of edges at a uniform random vertex satisfies

$$|S| \geq 1; \quad \mathbb{E}|S| = \tfrac{2n}{n+1}. \tag{14}$$

By considering how an online spanning tree algorithm on K_{n+1} selects edges at a uniform random vertex, we see that

$$\mathbf{cost}(n+1) \geq \tfrac{1}{2}(n+1)C_2^*(n)$$

where $C_2^*(n)$ is defined as $C_2(n)$ but with constraint (10) replaced by constraint (14). So

$$\liminf_n \mathbb{E}X_n \geq \tfrac{1}{2} \liminf_n nC_2^*(n).$$

To relate $C_2^*(n)$ to $C_2(n)$, consider $1 \leq m < n$ and set $a = 2 - \frac{2m}{m+1} = \frac{2}{m+1}$. Apply to $(W_i, 1 \leq i \leq m)$ the selection procedure attaining $C_2^*(m)$, and then use Lemma 7 to choose from $(W_i, m+1 \leq i \leq n)$ a minimum-weight set of items with mean number of items $= a$. This construction shows

$$C_2(n) \leq C_2^*(m) + \tfrac{a^2}{2(n-m)} = C_2^*(m) + \tfrac{2}{(m+1)^2(n-m)}$$

and then choosing $m \approx n - n^{1/2}$, say, shows

$$\liminf_n nC_2(n) \leq \liminf_n nC_2^*(n).$$

□

Proof of Lemma 10. Consider an online algorithm. Let T be the index of the first selected item and let $q_k = \mathbb{P}(T > k)$ be the probability that the first k offers are all rejected. This sequence is decreasing from $q_0 = 1$ to $q_n = 0$. The threshold $\theta_k = \theta_k(W_1, \ldots, W_{k-1})$ for accepting the k'th offer will have $\mathbb{E}\theta_k 1(T \geq k) = \frac{q_{k-1}-q_k}{q_{k-1}}$. Conditional on $\{T \geq k\}$, the expected cost of accepting item k equals $\mathbb{E}\theta_k^2/2$, so in an optimal algorithm θ_k must be non-random, which we now assume. [[**Note(david): Argument above is correct but hard to write convincingly ......**]]

On the event $\{T = k\}$ let $a_k = a_k(W_1, \ldots, W_k)$ be the conditional expected number of subsequent offers accepted . By Lemma 7 the conditional expected cost of the subsequent offers is at least $a_k^2/2(n-k)$. Thus

$$\mathbb{E}(\text{total cost }) \geq \sum_{k=1}^{n} (q_{k-1} - q_k) \left(\frac{q_{k-1} - q_k}{2q_{k-1}} + \frac{\mathbb{E}(a_k^2 | T = k)}{2(n-k)} \right)$$

and by taking a_k non-random we make this an equality.

Any choice of

$$1 = q_0 \geq q_1 \geq \ldots \geq q_n = 0; \quad 0 \leq a_k \leq n - k, \ 1 \leq k \leq n$$

is feasible, and the constraint $\mathbb{E}|S| = 2$ becomes the constraint

$$\sum_{k=1}^{n} (q_{k-1} - q_k) a_k = 1. \tag{15}$$

Thus

$$C_2(n) = \min \sum_{k=1}^{n} (q_{k-1} - q_k) \left(\frac{q_{k-1} - q_k}{2q_{k-1}} + \frac{a_k^2}{2(n-k)} \right)$$

minimized over (q_k) and (a_k) satisfying the constraints above.

Fixing (q_k) and optimizing over the a_k's we find $a_k = \beta(n-k)$ for some β. Note we can then use (15) to calculate

$$\beta^{-1} = \sum_{k=1}^{n} (q_{k-1} - q_k)(n-k) = n - \sum_{k=0}^{n-1} q_k. \tag{16}$$

We have now shown

$$2C_2(n) = \min_{1=q_0 \geq q_1 \geq \cdots \geq q_n = 0} \sum_{k=1}^{n} (q_{k-1} - q_k) \left(\frac{q_{k-1} - q_k}{q_{k-1}} + (n-k)\beta^2 \right)$$

for $\beta = \beta(q_0, \ldots, q_n)$ defined by (16). Note that the sum can be rewritten, using the first equality in (16), as

$$\beta + \sum_{k=1}^{n} \frac{(q_{k-1} - q_k)^2}{q_{k-1}}$$

and then the second equality in (16) gives the form stated in Lemma 10. □

[[**Note(david): To be precise we should check $\beta \leq 1$, needed to make a_k feasible.**]]

3.4 Proof of Lemma 11

[[**Note(david): This is presented just as an analysis problem, without the interpretation of T as a random selection time in a continuous model**]]

We treat the optimization problem as a "minimization under constraint" problem with an equality constraint on $\int_0^1 G(t)\ dt$. This will lead to a one-parameter family of solutions, and we will ultimately pick the best one. Thus we introduce a Lagrange multiplier λ^2 and consider the problem: minimize

$$\Lambda(G) := \int_0^1 \frac{(G')^2}{G}\ dt + \lambda^2 \int_0^1 G(t)\ dt$$

subject to (12). Recall the basic idea of the calculus of variations: introduce a test function $D(t)$ and note that the minimizing function G must have the property

$$\tfrac{d}{d\varepsilon} \Lambda(G + \varepsilon D) = 0 \text{ at } \varepsilon = 0.$$

In our case this becomes

$$\int_0^1 \left(\frac{2G'D'}{G} - \frac{(G')^2 D}{G^2} + \lambda^2 D \right) \; dt = 0.$$

Integrating the first term by parts:

$$\int_0^1 \left(-\left(\frac{2G'}{G}\right)' - \frac{(G')^2}{G^2} + \lambda^2 \right) \; D \; dt = 0.$$

Because this must hold for all test functions D, we get the differential equation

$$-\left(\frac{2G'}{G}\right)' - \frac{(G')^2}{G^2} + \lambda^2 = 0.$$

Introducing

$$h := -G'/G$$

we have the differential equation

$$2h' = h^2 - \lambda^2. \tag{17}$$

The boundary conditions (12) imply

$$G_\lambda(t) = \exp\left(-\int_0^t h_\lambda(s) \; ds \right) \qquad \text{where } h_\lambda(t) \to \infty \text{ as } t \uparrow 1.$$

The latter constraint on $h(t) \to \infty$ fixes the particular solution of (17)

$$h_\lambda(t) = \lambda \left(\frac{2}{1 - e^{\lambda(t-1)}} \; - 1 \right) = \frac{\lambda}{\tanh(\lambda(t-1))}.$$

Returning to the statement of Lemma 11, the function G attaining γ_* must minimize $\int_0^1 \frac{(G')^2}{G} \; dt$ for its given value of $\int_0^1 G(t) \; dt$, and so by the theory of Lagrange multipliers must be G_λ for some λ. That is,

$$\gamma_* = \min_{0<\lambda<\infty} \left(\int_0^1 \frac{(G_\lambda'(t))^2}{G_\lambda(t)} \; dt + \frac{1}{1 - \int_0^1 G_\lambda(t) \; dt} \right).$$

Dropping the subscript λ, observe that because

$$\tfrac{d}{dt} \log(e^{-\lambda(t-1)} - 1) = \frac{-\lambda e^{-\lambda(t-1)}}{e^{-\lambda(t-1)} - 1} = \frac{-\lambda}{1 - e^{\lambda(t-1)}}$$

we can integrate h to get

$$\int_0^t h(u) \; du = -2 \log\left(\frac{e^{-\lambda(t-1)} - 1}{e^{\lambda} - 1} \right) \; - \; \lambda t.$$

So

$$G(t) = \exp\left(- \int_0^t h(s)\ ds \right) = \left(\frac{e^{-\lambda(t-1)} - 1}{e^\lambda - 1} \right)^2\ e^{\lambda t}.$$

Because $G'/G = -h$ and $G' = -hG$ we have

$$(G')^2/G = h^2 G.$$

Writing

$$h = \lambda\ \frac{1 + e^{\lambda(t-1)}}{1 - e^{\lambda(t-1)}} = \lambda\ \frac{e^{-\lambda(t-1)} + 1}{e^{-\lambda(t-1)} - 1}$$

we find

$$h^2 G = \frac{\lambda^2}{(e^\lambda - 1)^2}\ e^{\lambda t}(e^{-\lambda(t-1)} + 1)^2.$$

So it is elementary to calculate

$$\begin{aligned} \int_0^1 h^2(t)G(t)\ dt &= \frac{\lambda^2}{(e^\lambda - 1)^2} \int_0^1 e^{\lambda t}(e^{-\lambda(t-1)} + 1)^2\ dt \\ &= \frac{\lambda^2}{(e^\lambda - 1)^2} \left(\frac{e^{2\lambda} - 1}{\lambda}\ +\ 2e^\lambda \right) \\ &= a(\lambda),\ \text{say}. \end{aligned}$$

Similarly

$$\begin{aligned} \int_0^1 G(t)\ dt &= \frac{1}{(e^\lambda - 1)^2} \int_0^1 e^{\lambda t}(e^{-\lambda(t-1)} - 1)^2\ dt \\ &= \frac{1}{(e^\lambda - 1)^2} \left(\frac{e^{2\lambda} - 1}{\lambda}\ -\ 2e^\lambda \right) \\ &= b(\lambda),\ \text{say}. \end{aligned}$$

Thus

$$\gamma_* = \min_\lambda Q(\lambda) \text{ where } Q(\lambda) := a(\lambda) + 1/(1 - b(\lambda)). \tag{18}$$

Numerically, $Q(\lambda)$ is remarkably close to constant over $0 < \lambda < 2$. It turns out [[**Note(david): can one of you supply the algebra?**]] that $\frac{d}{d\lambda}Q(\lambda)$ factors nicely and Q takes its minimal value at $\lambda_* = 1.45\ldots$ defined as the positive solution of $(\lambda - 2)e^{2\lambda} + 2e^\lambda + \lambda = 0$. Then $\gamma_* = Q(\lambda_*) = 5.47494$.

Finally, let us just outline the convergence result (13). We can rephrase Lemma 10 as saying that $2nC_2(n)$ is the minimum of

$$\frac{1}{n} \sum_{k=1}^{n} \frac{(n(q_{k-1} - q_k))^2}{q_{k-1}}\ +\ \frac{1}{1 - \frac{1}{n}\sum_{k=0}^{n} q_k}. \tag{19}$$

Given G_* attaining γ_*, set $q_k = G_*(k/n)$ and check that the quantity in (19) converges to γ_*: this shows

$$\liminf_n 2nC_2(n) \le \gamma_*.$$

Conversely, take G_n to be the function that linearly interpolates the points $(\frac{k}{n}, q_k)$ for the (q_k) attaining the minimum in (19), and pass to a subsequence of n through which $\limsup_n 2nC_2(n)$ is attained and $G_n(t)$ converges pointwise to some limit $G(t)$. Check that

$$\limsup_n 2nC_2(n) \geq \int_0^1 \frac{(G'(t))^2}{G(t)}\ dt + \frac{1}{1 - \int_0^1 G(t)\ dt} \geq \gamma_*.$$

3.5 xxx

[[Note(david): This is all comments for co-authors!]]

Our methodology suggests an alternative approach to the result of [4]. I haven't tried to do the calculation but one of you might like to try!

Setup; We are given one realization $0 < \xi^1_1 < \xi^1_2 < \xi^1_3, \ldots$ of a rate-1 Poisson process of weights of items; we have to choose some number $N_1 \geq 0$ of these items. We are then given another realization $0 < \xi^2_1 < \xi^2_2 < \xi^2_3, \ldots$ of a rate-1 Poisson process of weights of items; we have to choose some number $N_2 \geq 0$ of these items. Our constraint is that $N := N_1 + N_2$ must satisfy

$$N \geq 1; \quad \mathbb{E}N = 2.$$

Let C_4 be the minimal expectation of sum of weights of items picked. Then $C_4/2$ is a lower bound (different from theirs) for the problem in [4].

Question: is $C_4 > \zeta(3)$?

4 Equivalence of the two models

Write $\mathbf{cost}_{\text{Pois}}(n)$ and $\mathbf{cost}_{\text{comb}}(n)$ for the optimal mean costs under the two models. We would like to show (as a separate issue from existence of a limit) that $\mathbf{cost}_{\text{Pois}}(n) - \mathbf{cost}_{\text{comb}}(n) \to 0$. Below we give an argument (xxx a few details missing) for one half:

Proposition 12. $\mathbf{cost}_{comb}(n) \leq \mathbf{cost}_{Pois}(n) + o(1)$.

[[Note(david): Not so easy to prove as one might think. I haven't tried to think about the opposite half.]]

In the first part of the argument (Proposition 13) we show that in the combinatorial model we can mimic the optimal Poisson-model algorithm in choosing the first $(1-\delta)n$ edges. In the second part (section 4.3)) we analyze a simple algorithm to complete the spanning tree.

[[Note(david): In the Poisson model, time decreases from 1 to 0. In the combinatorial model, items are presented in usual order $1, 2, \ldots, \binom{n}{2}$. This is a bit awkward to remember ......]]

xxx notation "cost" below.

Proposition 13. *Fix t and take $m(n) \sim (1-t)\binom{n}{2}$. There is an online algorithm which produces, from the first $m(n)$ edges offered in the combinatorial model, a forest with edge-set $\mathcal{F}_{m(n)}$ such that*

$$\begin{aligned} \mathbb{E}cost(\mathcal{F}_{m(n)}) &\leq (1-2t)^{-1}\mathbf{cost}_{Pois}(n) && (20) \\ \mathbb{E}\#\mathcal{F}_{m(n)} &\geq (1-o(1))n. && (21) \end{aligned}$$

[[Note(david): The proof doesn't use the explicit structure of the optimal algorithm in the Poisson model.]]

4.1 A construction scheme

In order to apply results for the Poisson model to the combinatorial model we need a scheme by which, given the combinatorial model as input, one can synthesize the Poisson model. Here is the scheme we will use. Fix n and (small) $b = b(n) > 0$. Take $(U_i, 1 \leq i \leq \binom{n}{2})$ i.i.d. $U(0,1)$ and take $(e'_i, 1 \leq i \leq \binom{n}{2})$ as a uniform random ordering of the edges of K_n. Independently take $(e''_i, 1 \leq i \leq \binom{n}{2})$ to be i.i.d. uniform on the edges of K_n. Define $(e_i, 1 \leq i \leq \binom{n}{2})$ as follows. We use an accompanying process $\mathcal{E}_i$ of edge-sets, with $\mathcal{E}_0$ empty.

if $U_i > b$ then set $e_i = e''_i$ and $\mathcal{E}_i = \mathcal{E}_{i-1}$

if $U_i \leq b$ then with probability $1 - \#\mathcal{E}_{i-1}/\binom{n}{2}$ set $e_i = e'_i$ and $\mathcal{E}_i = \mathcal{E}_{i-1} \cup \{e_i\}$, and for each $e^* \in \mathcal{E}_{i-1}$ with probability $1/\binom{n}{2}$ set $e_i = e^*$ and $\mathcal{E}_i = \mathcal{E}_{i-1}$.

This construction makes (U_i, e_i) be i.i.d. uniform on $(0,1) \times \{\text{edges of } K_n\}$, and

$$\mathcal{E}_i = \{e_j : \ 1 \leq j \leq i, \ e_j = e'_j \text{ and } U_j \leq b\}.$$

We record some other properties needed later.

Lemma 14. *(a) If $nb(n) \uparrow \infty$ sufficiently slowly then*

$$\mathbb{E}\#\{1 \leq j \leq \binom{n}{2} : U_j \leq b(n) \text{ and } j \notin \mathcal{E}_{\binom{n}{2}}\} = o(n).$$

(b) Fix $1 \leq m < \binom{n}{2}$. Conditionally on $((e_i, U_i), 1 \leq i \leq m)$, the distribution of $(e'_j, m < j \leq \binom{n}{2})$ is as sampling without replacement from $\{$edges of $K_n\} \setminus \mathcal{E}_m$.

Proof. An elementary calculation shows $\mathbb{E}\#\mathcal{E}_m = \binom{n}{2}\left(1 - (1 - b/\binom{n}{2})^m\right)$ and so

$$\text{if } b(n) \to 0 \text{ then } \mathbb{E}\#\mathcal{E}_{\binom{n}{2}} \sim b(n)\binom{n}{2}.$$

But $\mathbb{E}\#\{1 \leq j \leq \binom{n}{2} : U_j \leq b(n)\} = b(n)\binom{n}{2}$ and so

$$\text{if } b(n) \to 0 \text{ then } \mathbb{E}\#\{1 \leq j \leq \binom{n}{2} : U_j \leq b(n) \text{ and } j \notin \mathcal{E}_{\binom{n}{2}}\} = o\left(b(n)\binom{n}{2}\right)$$

and (a) follows (xxx by general abstract nonsense).

xxx and (b) is intuitively obvious, but needs explanation because it's key to later analysis.

□

Now introduce a rate -$\binom{n}{2}$ Poisson process of times $1 > \tau_1 > \tau_2 > \tau_3 \ldots > \tau_{\binom{n}{2}} \approx 0$. Then the process (formally, a weighted point process) which at time τ_j offers edge e_j with weight U_j can be modified to become the process in the Poisson model, as follows:

if $\tau_{\binom{n}{2}} < 0$ delete the points with $\tau_j < 0$;
if $\tau_{\binom{n}{2}} > 0$ add extra points following the Poisson model with time-weight density $\binom{n}{2}$ on $[\tau_{\binom{n}{2}}, 0] \times (0, 1]$;
add further points according to the Poisson model with time-weight density $\binom{n}{2}$ on $[1, 0] \times (1, \infty)$.

Given an online spanning tree algorithm A for the Poisson model, we can now define an algorithm A' for the combinatorial model $(e'_i, U_i, 1 \leq i \leq \binom{n}{2})$ which at step m has produced a forest (represented by its edge-set) $\mathcal{F}_m$, as follows. Synthesize the Poisson model as above, apply algorithm A, and say that A' accepts edge e_m if A accepts edge e_m and $e_m \in \mathcal{E}_m$. In this case $e_m = e'_m$ and so $\mathcal{F}_m \subseteq \{e'_1, \ldots, e'_m\}$.

Lemma 15. *Fix $t > 0$. Let A be an online spanning tree algorithm for the Poisson model, with mean cost $c(n)$ such that $\sup_n c(n) < \infty$, and such that A only accepts points in the time interval $[1, 1 - 2t]$. Take $nb(n) \uparrow \infty$ sufficiently slowly and let A' be the algorithm for the combinatorial model described above and $\mathcal{F}_m$ the forest of accepted edges. Then for $m(n) \sim (1 - t)\binom{n}{2}$,*

$$\mathbb{E}cost(\mathcal{F}_{m(n)}) \leq c(n) \quad (22)$$

$$\mathbb{E}\#\mathcal{F}_{m(n)} \geq n - o(n). \quad (23)$$

Proof. Every edge accepted by A' is accepted by A, with the same weight, so (22) is immediate.

Ignoring an event of exponentially small probability (which makes negligible contribution to (23)) we may assume the time $\tau_{m(n)}$ is between $2t$ and 0. The difference $q(n, t_0) - \mathbb{E}\#\mathcal{F}_{m(n)}$ is upper bounded by the mean number of j such that either
(a) e_j is accepted by A and $U_j > b(n)$; or
(b) $U_j \leq b(n)$ and $j \notin \mathcal{E}_j$.

Now Lemma 14 says that the mean number in (b) is $o(n)$, and Markov's inequality says the mean number in (a) is $\leq c(n)/b(n) = o(n)$. □

Proof of Proposition 13. By Poisson scaling, the optimal algorithm in the Poisson model can be used to define an algorithm A which only accepts points in the time interval $[1, 1-2t]$ and which has mean cost $(1-2t)^{-1}\mathbf{cost}(n)$. Apply Lemma 15 to this A. □

4.2 Technical lemmas

Before continuing to the proof of Proposition 12 we give two technical lemmas which will be needed.

Lemma 16. *Suppose there exists an algorithm A in the combinatorial model that produces a random forest $\mathcal{F}$ such that, for $\Omega_n := \{\mathcal{F}$ is a spanning tree $\}$, we have $1 - \mathbb{P}(\Omega_n) = o(n^{-1})$. Then $\mathbf{cost}_{Pois}(n) \leq \mathbb{E}cost(\mathcal{F})1(\Omega_n) + o(1)$.*

Proof. On event Ω_n^c there is some first time m such that the current offered edge would connect two components but no remaining unseen edges connect these components, and such that A refuses this edge. Modify A by accepting this edge and then greedily accept every future edge connecting two components. The cost (conditional on Ω_n^c) of this modification is $O(n)$, and the modified algorithm produces a spanning tree. □

Next is a variation of the classical selection problem (Proposition 6) where a *random* number of items will be offered. In this context we just need the correct order of magnitude, not the optimal constant.

Lemma 17. *In each of $r(n)$ steps $1 \leq i \leq r(n)$ we are offered, with probability $\geq p(n)$, an item at cost U_i, for i.i.d. $U(0,1)$ costs. Consider the algorithm: choose the first offered item i, if any, with $1 \leq i \leq r(n) - \lfloor n^{4/3} \rfloor$ and $U_i \leq \gamma(i) := \frac{4}{p(n)(r(n)-i)}$. Suppose $r(n) \sim \varepsilon n^2$ and $p(n) \sim 1/n$. Then with probability $o(n^{-2})$ the algorithm selects no item and otherwise selects exactly one item at some time T such that $\mathbb{E}\gamma(T) \leq n^{-1}(\frac{6}{\varepsilon} + o(1))$.*

Proof. w.l.o.g. the probability $= p(n)$. Check (xxx details omitted) that

$$\mathbb{P}(T > j) \approx \left(\frac{r(n)-j}{r(n)}\right)^4$$

and calculate $\mathbb{E}\gamma(T)$. □

4.3 Proof of Proposition 12

[[**Note(david): This is still rather sketchy, due to lack of time**]]

Recall the conclusion of Proposition 13. We have $m(n) \sim (1-t)\binom{n}{2}$, and we have an edge-set $\mathcal{F}_{m(n)}$ which defines a partition $\Pi_{m(n)}$ with $o(n)$ components. We condition on $\Pi_{m(n)} = \Pi^*$ with $k(n) = o(n)$ components. By Lemma 14(b) the remaining edges $(e'_j, m(n) < j \le \binom{n}{2})$ which will be offered to the algorithm are distributed as $\binom{n}{2} - m(n)$ samples without replacement from $\{\text{edges of } K_n\} \setminus \mathcal{F}_{m(n)}$. [[**Note(david): Please check you believe this!**]]

To prove Proposition 12 we will use an algorithm, over time $m(n) < j \le \binom{n}{2}$ producing a partition Π_j and filtration $\mathcal{G}_j$, of the "threshold" format. There are constants γ_j. The edge e_{j+1} offered at time $j+1$ with weight U_{j+1} is accepted if $U_{j+1} \le \gamma_{j+1}$ and e_{j+1} connects two components of Π_j.

Lemma 18. *For any such threshold algorithm, and any edge e connecting two components of Π_j,*

$$\mathbb{P}(e \textit{ offered at step } j+1 | \mathcal{G}_j) \ge 1/\binom{n}{2}.$$

Proof. (xxx outline). Imagine first seeing U_{j+1}; if $U_{j+1} > \gamma_{j+1}$ we move on without looking at e_{j+1}; otherwise we look at e_{j+1} and either accept it or add it to a stack $\mathcal{H}_{j+1}$ of "seen but unused" edges. Then the conditional probability in question equals 1/ (number of unseen edges) and

$$(\text{number of unseen edges}) = \binom{n}{2} - \#\mathcal{F}_j - \#\mathcal{H}_j \le \binom{n}{2}.$$

□

The algorithm we use over time $m(n) < j \le \binom{n}{2}$ has two stages. The first stage lasts as long as the maximal component size is $\le n/2$, which implies that the number of edges between components is at least $n^2/4$. Apply the algorithm in the format above with threshold $\gamma_j = 1/n$. Then by Lemma 18, the conditional probability of some edge linking two components being offered is at least 1/2, so the chance of such an edge being offered and accepted is at least $1/(2n)$. Since there are $k(n) = o(n)$ components to connect, this stage takes $o(n^2)$ steps, outside an exponentially rare event. The latter event is taken care of by Lemma 16. The cost of this stage is $o(1)$. So we may assume that at some time $m'(n) \sim (1-t)\binom{n}{2}$ the algorithm has produced a forest with $o(n)$ components, with one "giant component" having size $> n/2$, at total cost $\le (1-2t)^{-1}\mathbf{cost}_{\mathrm{Pois}}(n) + o(1)$.

[[**Note(david): Heuristically, for the second stage we just connect small components to the giant component. But there's a technical problem. To bound conditional probabilities at each step we need something like Lemma 18. But Lemma 18 just isn't true for a general online algorithm (it could code non-acceptence of some offered edge in some weird way). No doubt one could prove a version of Lemma 18**

for the heuristic scheme. Instead I give an indirect way of using the existing Lemma 18.]]

For the second stage, set $\gamma(j) = \frac{4}{\frac{1}{n}(\binom{n}{2}-j)}$ for $m'(n) < j \leq \binom{n}{2}$ and consider the associated threshold algorithm. Let (T_u) be the times of accepted edges; note the cost is at most $\sum_u \gamma(T_u)$. We need to specify a labeling scheme. At "original" time $m'(n)$ label components $u = 1, 2, \ldots$ in (say) increasing order of size. Then we can define T_u as the first time that the component containing the original component u is merged with a component containing some higher-labeled original component. This scheme gives each T_u a different label u.

A time T_u might be a time when two small components merge, but we can compare as follows with times involving the giant component. From each original component u pick a vertex v_u. Define T_u^* as the first time j, if any, that an edge e_j between v_u and the original giant component is offered with $U_j \leq \gamma(j)$. Clearly $T_u \leq T_u^*$.

Because we are using a threshold algorithm, the conclusion of Lemma 18 holds, and now we can use Lemma 17 to bound $\mathbb{E}T_u^*$. That is, the hypotheses of Lemma 17 hold for "edges between v_u and the original giant component" with $p(n) = 1/n$ and $r(n) = \binom{n}{2} - m'(n)$ giving $\varepsilon = t/2$; and our γ_j corresponds (xxx say better) to γ_i. The conclusion of Lemma 17 is that T_u exists outside an event of probability $o(n^{-2})$ and that $\mathbb{E}\gamma(T_u^*) \leq n^{-1}(\frac{12}{t} + o(1))$. Because the number of components at time $m'(n)$ is $o(n)$, we see that the threshold algorithm does produce a spanning tree (outside an event of probability $o(n^{-1})$) and that the mean cost of the second stage is $o(1)$. So the total cost of the algorithm is still $\leq (1-2t)^{-1}\mathbf{cost}_{\mathrm{Pois}}(n) + o(1)$. Now Proposition 12 follows from Lemma 16.

References

[1] D.J. Aldous. Deterministic and stochastic models for coalescence (aggregation and coagulation): a review of the mean-field theory for probabilists. *Bernoulli*, 5:3–48, 1999.

[2] D.J. Aldous and J.M. Steele. The objective method: Probabilistic combinatorial optimization and local weak convergence. In H. Kesten, editor, *Probability on Discrete Structures*, volume 110 of *Encyclopaedia of Mathematical Sciences*, pages 1–72. Springer-Verlag, 2003.

[3] T. S. Ferguson. Who solved the secretary problem? *Statist. Sci.*, 4(3):282–296, 1989.

[4] A. D. Flaxman, A. Frieze, and M. Krivelevich. On the random 2-stage minimum spanning tree. *Random Structures Algorithms*, 28(1):24–36, 2006.

[5] A.M. Frieze. On the value of a random minimum spanning tree problem. *Discrete Appl. Math.*, 10:47–56, 1985.

[6] L. Moser. On a problem of Cayley. *Scripta Math.*, 22:289–292, 1956.

[7] J.R. Norris. Smoluchowski's coagulation equation: Uniqueness, non-uniqueness and a hydrodynamic limit for the stochastic coalescent. *Ann. Appl. Probab.*, 9:78–109, 1999.

[8] J. Remy, A. Souza, and A. Steger. On an online spanning tree problem in randomly weighted graphs. *Combin. Probab. Comput.*, 16(1):127–144, 2007.