Cutting and Destroying Graphs using \( k \)-cuts

Alfred Minh Lindström
Cutting and Destroying Graphs using $k$-cuts

Alfred Minh Lindström

June 20, 2019

Contents

1 Introduction ....................................... 1
   1.1 Cutting Algorithm ................................. 2
   1.2 $r$-records on Trees ............................... 2

2 A Lower and an Upper Bound .................. 3
   2.1 $k$-cuts on Paths ................................. 4
   2.2 $k$-cuts on Complete Graphs .................... 7

3 More Examples on Trees ...................... 8
   3.1 Complete Binary Trees ............................ 8
   3.2 Random Recursive Trees ........................... 9
   3.3 Conditional Galton-Watson Trees ............... 10

4 Simulations ........................................ 11
   4.1 Paths ................................................ 11
   4.2 Complete graphs .................................. 11

A Code in python .................................. 15
   A.1 Paths .............................................. 15
   A.2 Complete Graphs ................................. 16

1 Introduction

We start by introducing the most fundamental concepts. The only considered graphs are rooted graphs:

Definition 1.1. A rooted graph is a connected graph $G_n$, with $n \geq 2$ vertices where one, and only one vertex is labelled as the root $r$.

Note that the only restriction we thus put on $G_n$ is that it is connected; other than that, in theory our rooted graphs could be anything from paths to complete graphs, or lack any obvious structure, e.g., a randomly generated graph. In practice, however, some specific structure is often given, which will be made explicit.
1.1 Cutting Algorithm

Let $k$ be a fixed positive number. To cut a graph $G_n$, we have the following cutting algorithm:

1. Choose a vertex $v$ uniformly at random from the component that contains the root $r$. Cut $v$.
2. If $v$ has been cut $k$ times, remove $v$ from $G_n$ along with its adjacent edges.
3. If the root has been removed, stop. Otherwise, repeat the steps.

We can now define the $k$-cut number.

**Definition 1.2.** Let $k \in \mathbb{N}$. The $k$-cut number of a rooted graph $G_n$, denoted $K^k(G_n)$, is the total number of cuts needed to run and finish the cutting algorithm. Note that $K^k(G_n)$ is random.

Traditionally, cutting has been done with $k = 1$, and on edges instead of vertices. For this we use the same algorithm, but choose an edge $e$ uniformly at random during step 1 instead. We denote the edge version of the $k$-cut number as $K^e(G_n)$.

For trees, there is a natural correspondence between the edge and vertex versions. If $T$ is the rooted tree, let $T'$ be the rooted forest obtained by deleting the root of $T$. Then $K^e(T') = K^v(T')$ if we run the vertex version of the algorithm on all rooted trees in the forest $T'$, and take the sum of each of the $k$-cut number.

Conversely, let $\tilde{T}$ be the tree obtained by adding a new root to $T$, with the old root as its only child. Then $K^v(T) = K^e(\tilde{T})$.

1.2 $r$-records on Trees

When cutting on trees (i.e., a connected graph with no circles), we have an equivalent way to define the $k$-cut number. The equivalence was first studied by Janson [5] on 1-cuts of trees, and is based on order statistics.

Let $\mathbb{T}_n$ be a rooted tree with $n$ vertices. The idea for 1-cuts is to give each vertex $v \in \mathbb{T}_n$ a timestamp $T_v$, drawn independently from the same continuous distribution. We call $T_v$ a record if it is the smallest value in the path from the root $r$ to $v$ (including $T_r$). Then the sum of records equals $K^v(\mathbb{T}_n)$. To see this, generate the values of $T_v$, $\forall v \in \mathbb{T}_n$. Then cut the tree, with each iteration choosing the vertex with the smallest $T_v$ (in other the words the fastest, to continue with the time analogy), among the remaining ones. This is equivalent to the former cutting algorithm, and a vertex $v$ is cut if and only if its value $T_v$ is a record, i.e., the number of records equals the number of cuts.

Note that the distribution of $T_v$ is arbitrary (except for the assumption that the distribution is continuous to avoid ties), since only the order relations is of importance. To make this point clear, consider the fact that the order of $T_v$ essentially is just a random permutations of $1, ..., |\mathbb{T}_n|$.

We now generalize this idea for arbitrary $k$. 

For a vertex $v$ in a tree $T_n$, let the timestamps $T_{1,v}, ..., T_{k,v}$ be independent and identically distributed $exp(1)$ random variables. Think of each timestamp $(T_{i,v})_{1 \leq i \leq k,v \in T_n}$ as the time it takes for $v$ to complete its $i$th lap in some arbitrary competition. Thus, the total time for $v$ to complete its second lap is $T_{1,v} + T_{2,v}$ etc. Every time a lap is completed, we consider cutting the vertex, if it is still in the tree. Note that by the memoryless property of the exponential random variables, the vertex being cut is chosen uniformly at random.

We now need the following well known result (Theorem 2.1.12. in [3]):

**Lemma 1.** If $X_1, ..., X_n$ are independent exponential random variables with parameter $\lambda$ denoted $Exp(\lambda)$, then $X_1 + ... + X_n$ has a $\Gamma(n, \lambda)$ distribution.

Let $1 \leq r \leq k$ and define $G_{r,v} := \sum_{i=1}^{r} T_{i,v}$, i.e., $G_{r,v}$ is the total time it takes for $v$ to complete its $r$th lap. By the previous lemma, $G_{r,v}$ has a gamma distribution with parameters $(r, 1)$, denoted $Gam(r)$. By generating $G_{r,v}$ for all $v \in T_n$, $1 \leq r \leq k$, we get an ordering (of $G_{r,v}$), which allows us to define records, analogous to the 1-cut case. However, since a vertex can be cut up to $k$-times before we remove it, it makes sense that a vertex can be record up to $k$-times. Thus, when vertex $v$ has completed its $r$th lap for $r \leq k$, and no vertex above $v$ has completed its $k$th lap, we say that $v$ is an $r$-record. Note that if $v$ is an $r$-record, then it must also be an $i$-record for $i \in \{1, ..., r-1\}$ . We formalize this:

**Definition 1.3.** For $v \in T_n$ and $r \leq k$, let

$$I_{r,v} := \left[ G_{r,v} < \min\{G_{k,u} : u \in T_n, u \text{ is an ancestor of } v\} \right]$$

where $[\cdot]$ is the Iverson bracket, i.e., $[S] = 1$ if statement $S$ is true and $[S] = 0$ otherwise. We define $v$ as being an $r$-record if $I_{r,v} = 1$.

**Definition 1.4.** Let $K_r(T_n) := \sum_{v \in T_n} I_{r,v}$ denote the number of $r$-records. Moreover, we define the $k$-cut number as $K(T_n) := \sum_{r=1}^{k} K_r(T_n)$, i.e., the total number of $r$-records.

## 2 A Lower and an Upper Bound

We will now examine $k$-cuts $K(G_n)$, for the cases when $G_n$ is a path $P_n$, and $G_n$ is a complete graph $K_n$. Intuitively, graphs with fewer or no circles, along with fewer leafs, will be easier to cut, and vice versa. With $P_n$ being the tree with the fewest leafs (exactly one), and $K_n$ being the graph with the most circles, examining these structures naturally gives us a stochastic lower and upper bound for $K(G_n)$, respectively.

**Proposition 1.** Let $k \in \mathbb{N}$. For all rooted graphs $G_n$,

$$K(P_n) \leq K(G_n) \leq K(K_n)$$
2.1 $k$-cuts on Paths

We start with the lower bound by looking at $P_n$. Since a path is a tree structure, we can use $r$-records. Label all vertices in $P_n$ with 1,...,n, where 1 is the root of the graph, and n is the leaf. For brevity, we denote $I_{r,v}, G_{r,v}$, and $T_{r,v}$ as $I_{r,i}, G_{r,i}$, and $T_{r,i}$ where $i$ is the depth of vertex $v$.

By our definitions, vertex $i$ is an $r$-record if and only if $G_{r,i} < \min\{G_{k,1}, ..., G_{k,i-1}\}$.

Since $G_{r,i}$ has distribution $\text{Gam}(r)$ and each $(G_{k,j})_{0 \leq j \leq i-1}$ has distribution $\text{Gam}(k)$, we can use the following equality,

$$P[\text{Gam}(k) > x] = P\left[\sum_{j=0}^{k-1} E_j > x\right] = P[\text{Poi}(x) < k]$$  \hspace{1cm} (1)

where $E_1, E_2$ are i.i.d. $\text{Exp}(1)$ random variables. The equality follows from the relationship of the Gamma, Exponential and Poisson distributions. Note that we can do this since $k$ is an integer.

Thus, by taking the Poisson distributions cumulative function of the we get:

**Lemma 2.** Let $\alpha = \frac{1}{2} \left(\frac{1}{r} + \frac{1}{r+1}\right)$ and $x_0 = m^{-\alpha}$. Then uniformly for all $x \in [0, x_0]$,

$$P[\text{Gam}(k) > x]^m = \left(\sum_{l=0}^{k-1} \frac{e^{-x} x^l}{l!}\right)^m = \left(1 + O\left(m^{-\frac{1}{k}}\right)\right) \exp\left(-\frac{m x^k}{k!}\right)$$

For the proof see Lemma 16 in [1].

From this, we can find the expected value of vertex $i+1$ being an $r$-record, since

$$\mathbb{E}[I_{r,i+1}] = \int_0^\infty \frac{e^{-x} x^{r-1}}{\Gamma(r)} P[\text{Gam}(k) > x]^i dx.$$

**Lemma 3.** Uniformly for all $i \geq 1$ and $r \in \{1, ..., k\}$,

$$\mathbb{E}[I_{r,i+1}] = \left(1 + O\left(i^{-\frac{1}{k}}\right)\right) \frac{(k)^{\frac{1}{r}} \Gamma\left(\frac{k}{r}\right)}{\Gamma(r)} i^{-\frac{r}{k}}$$

where $\Gamma(z)$ is the gamma-function.

For the proof, see Lemma 1 in [1].

By fixing $r$ and taking the sum of the expected values for all vertices $i$ being $r$-records, we get the expected value of the number of $r$-records for our path:

**Theorem 1.** For all fixed $k \in \mathbb{N}$,

$$\mathbb{E}[\mathcal{K}_r(P_n)] = \sum_{i=0}^{n-1} \mathbb{E}[I_{r,i+1}] \sim \begin{cases} \eta_{k,r} n^{1-\frac{1}{k}} & (1 \leq r < k) \\ \log n & (r = k) \end{cases}$$

4
where
\[ \eta_{k,r} = \frac{(k)!}{k - r} \frac{\Gamma(k)}{\Gamma(r)} \]

This implies that the number of 1-records dominates the number of higher records. Therefore \( E[K(P_n)] \sim E[K_1(P_n)] \).

To see that \( E[K_r(P_n)] \sim \log n \) for \( r = k \), we use the fact that \( G_{k,1}, \ldots, G_{k,n} \) are i.i.d. random variables with distribution \( \text{Gam}(k) \), so its ordering induces a uniform random permutation \((\sigma_1, \sigma_2, \ldots, \sigma_n)\) of \( \{1, \ldots, n\} \). Thus, the probability of vertex \( i \) being a \( k \)-record, i.e., \( G_{k,i} < \min\{G_{k,j}\} \) for \( j < i \), is \( 1/i \), since this is the probability that \( \sigma_i \) is the smallest among \((\sigma_1, \sigma_2, \ldots, \sigma_i)\). It follows that \( E[I_{k,i}] = 1/i \), and moreover \( E[K_k(P_n)] = n \sum_{i=1}^{n} E[I_{k,i}] \sim \log n \).

One-records dominates other records since for \( r \geq 2 \),
\[ \frac{E[K_r(P_n)]}{n^{1 - \frac{1}{k}}} \to 0 \]
so after rescaling by \( n^{1 - \frac{1}{k}} \), the \( r \)-records vanish and do not contribute to the limit.

For the variance, we have the following theorem:

**Theorem 2.** For all fixed \( k \in \mathbb{N} \)
\[ E[K_1(P_n)(K_1(P_n) - 1)] \sim E[(K_1(P_n))^2] \sim \gamma_k n^{2 - \frac{2}{k}} \]
where
\[ \gamma_k = \frac{\Gamma(\frac{k}{2})(k!)^{\frac{3}{2}}}{k - 1} + 2\lambda_k \]

and
\[ \lambda_k = \begin{cases} \frac{\pi}{2} \cot(\frac{\pi}{k})^{\frac{3}{2}} & (k > 2), \\ \frac{\pi^2}{4} & (k = 2) \end{cases} \]

Therefore
\[ \text{Var}(K_1(P_n)) \sim (\gamma_k - \eta_{k,1}^2) n^{2 - \frac{2}{k}} \]

For the complete proof, see [1]. The idea is to let \( E_{i,j} \) denote the event that vertex \( i + 1 \) and \( j + 1 \) are one-records, \( A_{x,y} \) denote the event that \( [G_{i,i+1} = x] \cap [G_{i,j+1} = y] \). Then conditioning on \( A_{x,y} \),
\[ E_{i,j} = \left[ \bigcap_{1 \leq s \leq i} G_{k,s} > \max\{x,y\} \right] \cap [G_{k,i+1} > y] \cap \left[ \bigcap_{i+2 \leq s \leq j} G_{k,s} > y \right] \]
By conditioning on $A_{x,y}$, we have that $G_{k,i+1} = \text{Gam}(k-1) + x$ and $G_{k,s} = \text{Gam}(k)$ for $s \notin \{i+1, j+1\}$. Since all these all random variables are independent, we get that

$$P[E_{i,j} \mid A_{x,y}] = P[G_{k-1,1} + x > y]P[G_{k,1} > \max\{x, y\}]P[G_{k,1} > y]^{j-i-1}$$

Note that $G_{1,i+1} = G_{1,j+1} = \text{Exp}(1)$, and thus

$$P[E_{i,j}] = \int_0^\infty \int_0^y e^{-x-y}P[E_{i,j} \mid A_{x,y}]dxdy + \int_y^\infty \int_0^y e^{-x-y}P[E_{i,j} \mid A_{x,y}]dxdy$$

The rest of the proof, which is quite technical, is to estimate these integrals, since we have that

$$\mathbb{E}[\mathcal{K}_1(P_n)(\mathcal{K}_1(P_n) - 1)] = 2 \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} P[E_{i,j}]$$

For the limit distribution when $k \geq 2$, Cai et al. [1] used the idea to condition on the positions and values of the $k$-records, to then study the distribution of the number of one-records between two consecutive $k$-records. Therefore, let $P_{n,1}, P_{n,2}, \ldots$ be the positions of the $k$-records (starting from the end of the path), and $R_{n,1}, R_{n,2}, \ldots$ be the time when they have been cut $k$ times. To formalize this, let $R_{n,0} := 0$ and $P_{n,0} := n + 1$. For $q \geq 1$, if $P_{n,q-1} > 1$ let

$$R_{n,q} := \min\{G_{k,j} : 1 \leq j < P_{n,q-1}\},$$

$$P_{n,q} := \text{argmin}\{G_{k,j} : 1 \leq j < P_{n,q-1}\}$$

i.e., $P_{n,q}$ is the index which satisfies $G_{k,P_{n,k}} < G_{k,i}$ for all $1 \leq i < P_{n,q-1}$. Otherwise, let $P_{n,q} = 1$ and $R_{n,q} = \infty$.

Conditioned on $P_{n,p-1}$ and $P_{n,p}$ the number of one-records between $P_{n,p-1}$ and $P_{n,p}$ is roughly

$$B_{n,p} := \text{Bin}(P_{n,p-1} - P_{n,p}, P[\text{Exp}(1) < R_{n,q}])$$

Figure 1: An example of $(B_{n,q})_{q \geq 1}$ for $n = 12$
The rest of the proof is then to show that the sum $\sum_{1 \leq j \leq n} \frac{B_{n,j}}{n^{1-k}}$ converges to the random variable $B_k$, defined as follows:

$$U_j := \text{Unif}[0, 1], E_j := \text{Exp}(1)$$

$$S_p := \left( k! \sum_{1 \leq s \leq p} \left( \prod_{s \leq j < p} U_j \right) E_s \right)^{\frac{1}{k}},$$

$$B_p := \left( 1 - U_p \right) \left( \prod_{1 \leq j < p} U_j \right)^{1 - \frac{1}{k}} S_p,$$

$$B_k := \sum_{1 \leq p} B_p,$$

where $B_p$ is the limit of $B_{n,p}$ after rescaling and shifting.

**Theorem 3.** Let $k \in \{2, 3, \ldots\}$. Let $\mathcal{L}(B_k)$ denote the distribution of $B_k$. Then

$$\mathcal{L}\left( \frac{K_n(P_n)}{n^{1-k}} \right) \xrightarrow{d} \mathcal{L}(B_k).$$

What the density function of $B_k$ is analytically is unknown. However, it is proven to exist for all $k \geq 2$, and through simulation it seems like the density is very close to a normal distribution.

### 2.2 $k$-cuts on Complete Graphs

We now take a look at the complete graph $K_n$ to get an upper bound. We cannot use $r$-records directly, since complete graphs are not tree structures. However, consider the fact that when we remove a vertex in a complete graph other than the root, we cannot disconnect it. Thus, let $S_n$ be tree of $n$ vertices with 1 root and $n - 1$ leaves. Cutting on $S_n$ is then equivalent as cutting on $K_n$. We formalize this in the proof of the following theorem:

**Theorem 4.** Let $k \in \mathbb{N}, \mathcal{Y} = \text{Gam}(k)$. Then

$$\frac{K(K_n)}{n} \xrightarrow{d} \mathbb{E}[\min\{\text{Poi}(\mathcal{Y}), k\} \mid \mathcal{Y}]$$

Therefore,

$$\mathbb{E}[K_n] \sim k \left( 1 - \frac{1}{2^{2k}} \binom{2k}{k} \right) n$$

**Proof.** Let $S_n$ be defined as above. Then $K(K_n) = K(S_n)$ Let $\mathcal{Y}$ be the time when the root of $S_n$ is removed. Let $(W_j)_{j=1}^{n-1}$ be i.i.d. min$\{\text{Poi}(\mathcal{Y}), k\}$ random variables, i.e., the number of cuts each of the $n - 1$ leaves receives by time $\mathcal{Y}$. We do this since the in our Poisson point process, the number of points distributed in the time interval $[0, \mathcal{Y}]$, has Poisson distribution with parameter $\mathcal{Y}$ (see the
equivalence (1)), and each vertex cannot be cut more than \(k\) times. Condition on \(Y = y\), then by the law of large numbers,

\[
\frac{\mathcal{K}(S_n)}{n} = \frac{k + \sum_{i=1}^{n-1} W_i}{n} \xrightarrow{a.s.} \mathbb{E}[\min\{\text{Poi}(Y), k\} | Y]
\]

from which (2) and (3) follows. \(\square\)

3 More Examples on Trees

We now look at some random and deterministic trees.

3.1 Complete Binary Trees

Let \(T_{bi}^{n}\) be a complete binary tree with \(n = 2^m + 1\) vertices, i.e., the tree will have height \(m\) with \(2^m\) leaves, and each level \(i < m\) will have \(2^i\) vertices. Therefore, for each vertex \(v\) with depth \(i\), we only need to consider the path from the root to \(v\) (which is of length \(i\)), since the probability of \(v\) being an \(r\)-record only depends on whether or not one of its ancestors have been cut \(k\) times. Thus it follows from Lemma 3, that

\[
\mathbb{E}(K_r(T_{bi}^{n})) = m \sum_{i=0}^{m} 2^i \mathbb{E}(I_{r,i+1}) \sim \frac{(k)! \Gamma\left(\frac{1}{2}\right) 2^{m+1}}{k m^{\frac{1}{2}}} \Gamma\left(\frac{1}{k}\right)
\]

From this, we can see, as in the case of \(P_n\), that one-records dominate, thus

\[
\mathbb{E}(K(T_{bi}^{n})) \sim \mathbb{E}(K_1(T_{bi}^{n})) \sim \frac{(k)! \Gamma\left(\frac{1}{2}\right) 2^{m+1}}{k m^{\frac{1}{2}}} \Gamma\left(\frac{1}{k}\right) \frac{n}{(\log_2 n)^{\frac{1}{2}}}
\]

For the limit distribution, we have the following result by Cai and Holmgren [2]:

**Theorem 5.** Assume that \([\log n - \log \log n] \to \gamma \in [0, 1]\) as \(n \to \infty\). Then

\[
\frac{\log(n)^{\frac{1}{2}} + 1}{C_2(r)n} \mathcal{K}(T_{bi}^{n}) - \mu_{r,n} \xrightarrow{d} 1 - C_3(r)W_{r,k,\gamma},
\]

where

\[
\mu_{r,n} = k \frac{1}{r} \log(n) + \sum_{i=1}^{k} C_1(r,i) \log(n)^{1-\frac{i}{r}} + \log \log(n).
\]

The \(C_i(\cdot)\) are constants that do not depend on \(n\) implicitly defined in the proof.

The random variable \(W_{r,k,\gamma}\) has an infinitely divisible distribution with the characteristic function

\[
\mathbb{E}[\exp(itW_{r,k,\gamma})] = \exp\left(\frac{1}{2} \int_0^\infty (e^{itx} - 1 - itx \cdot 1[x < 1])d\nu_{r,k,\gamma}(x)\right),
\]

where \(\nu_{r,k,\gamma}\) are defined as infinite sums depending on \(r, k, \gamma\).
Theorem 6. Assume the same conditions as in Theorem 5. Then

\[
\frac{\log(n)^{\frac{1}{k}+1}}{C_2(1)n} \left( K(T_{n}^d) - \sum_{r=1}^{k} \frac{C_2(r)n}{\log(n)^{\frac{1}{k}+1} \mu_{r,n}} \right) \xrightarrow{d} 1 - C_3(1)W_{1,k,\gamma}.
\]

The same also holds for \( K^e(T_{n}^d) \).

3.2 Random Recursive Trees

A random recursive tree \( T_{n}^{rr} \) of \( n \) vertices is defined as follows: start with the root as an isolated vertex. Add the second vertex to the root with an edge, then add a third to one of the two vertices chosen uniformly at random. Repeat this \( n \) times, where for the \( k \)th time, we choose which of the \( k-1 \) vertices already in the tree to add uniformly at random.

This model was first studied by Meir and Moon [6] for 1-cuts on edges, where they showed that \( E[K^e(T_{n}^{rr})] \sim n \log n \). Iksanov and Möhle [4] later provided the following weak convergence result:

Theorem 7. Let \( k = 1, K^e(T_{n}^{rr}) = X_n = X_{n-D_n} + 1, n = 2, 3, \ldots \), and \( X_1 = 0 \), where \( D_n \) is a random variable with

\[
P(D_n = l) = \frac{1}{l(l+1)} \frac{n}{n-1}, \quad l \in \{1, 2, ..., n-1\},
\]

that is independent of \( (X_2, X_3, ..., X_{n-1}) \). In other words, \( D_n \) is the random variable given by the probability that after removing a random edge from a random recursive tree of size \( n \), the new tree has size \( l \).

Then, the sequence

\[
Y_n = \frac{\log^2 n}{n} n X_n - \log n - \log \log n
\]

converges weakly to a stable random variable \( Y \) with characteristic function

\[
\varphi_Y(\lambda) = e^{i\lambda \log |\lambda| - \frac{\pi}{2} |\lambda|}.
\]

For \( k \geq 2 \) we have the expectation by Cai, Holmgren and Skerman [1]:

Lemma 4. Let \( k \in \mathbb{N} \), then

\[
E[K^e(T_{n}^{rr})] \sim \frac{(k!)^{\frac{1}{k}}}{k} \frac{\Gamma(r)}{\Gamma(r) (\log n)^{\frac{1}{k}}}, \quad (1 \leq r \leq k)
\]

\[
E[K(T_{n}^{rr})] \sim \frac{(k!)^{\frac{1}{k}}}{k} \frac{\Gamma(r)}{\Gamma(r) (\log n)^{\frac{1}{k}}}
\]

The idea behind the expectation is to use the fact that almost all vertices in \( T_{n}^{rr} \) are at depth \( \log n \). Let \( v \) be good if it has depth \( d(v) \) where \( |d(v) - \log n| \leq (\log n)^{0.9} \), and otherwise bad. By Lemma 13 in [1], the expectation of the number of bad vertices are bounded by \( \frac{n}{(\log n)^{r}} \) and can therefore be neglected. Lemma 3 is then applied on the remaining good vertices.
3.3 Conditional Galton-Watson Trees

A Galton-Watson tree $T_{gw}$ is a random tree defined as follows: start with the root and recursively attach a random number of children to each vertex in the tree. The number of children for each vertex are drawn independently from the same distribution $\xi$, where $\xi$ is a non-negative integer valued random variable.

A Conditional Galton-Watson tree $T_{gw}^n$ is a Galton-Watson tree restricted to size $n$.

Janson [5] studied this model for 1-cuts and found the limit distribution and higher moments.

**Theorem 8.** Let $k=1$ and $T_{gw}^n$ be a conditioned Galton-Watson tree of size $n$, defined by offspring distribution $\xi$ satisfying

$$\mathbb{E}[\xi] = 1$$

and

$$0 < \sigma^2 = \text{Var}(\xi) < \infty$$

Then,

$$\frac{K(T_{gw}^n)}{\sigma \sqrt{n}} \xrightarrow{d} Z,$$

where $Z$ has Rayleigh distribution with density $xe^{x^2/2}, x > 0$. Moreover, if $\mathbb{E}[\xi^m] < \infty$ for every $m > 0$, then all moments converge in (4), and thus for every $l > 0$,

$$\mathbb{E}[K(T_{gw}^n)^l] \sim \sigma^l n^{l/2} \mathbb{E}[Z^l] = 2^{l/2} \sigma^l \Gamma\left(\frac{l}{2} + 1\right) n^{l/2}$$

For $k > 1$, Cai and Holmgren [1] found the expectation for the number of $r$-records by using the same method as Janson in [5].

**Lemma 5.** Let $k \in \mathbb{N}$ and assume that $\mathbb{E}[\xi^3] < \infty$. Then for $r \in \{1, ..., k\}$,

$$\frac{\mathbb{E}[K_r(T_{gw}^n)]}{n^{1-\frac{r}{2k}}} \rightarrow \frac{(k!)^{\frac{r}{k}}}{k} \frac{\Gamma\left(\frac{r}{k}\right)}{\Gamma(r)} \mathbb{E}\left[\int_0^1 \left(\frac{2e(t)}{\sigma}\right)^{-\frac{r}{k}} \frac{\sigma}{\sqrt{2}} \right]$$

Then,

$$\mathbb{E}[K(T_{gw}^n)] \sim \mathbb{E}[K_1(T_{gw}^n)] \sim \frac{(k!)^{\frac{r}{k}}}{k} \Gamma\left(\frac{1}{k}\right) \Gamma\left(1 - \frac{1}{2k}\right) \left(\frac{\sigma}{\sqrt{2}}\right)^{\frac{r}{k}} n^{1-\frac{r}{2k}}.$$
4 Simulations

We now present the data from our simulations. The simulations were done in Python, were $r$-records was implemented. All code is found in the appendix. 5000 iterations were done for each run.

4.1 Paths

<table>
<thead>
<tr>
<th>$k = 1$</th>
<th>$n$</th>
<th>Theoretical Mean</th>
<th>Simulation Average</th>
<th>Relative Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>$10^4$</td>
<td>9.2103</td>
<td>9.7678</td>
<td>0.0571</td>
<td></td>
</tr>
<tr>
<td>$10^5$</td>
<td>11.5129</td>
<td>12.0818</td>
<td>0.0471</td>
<td></td>
</tr>
<tr>
<td>$10^6$</td>
<td>13.8155</td>
<td>14.4604</td>
<td>0.0446</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>$k = 2$</th>
<th>$n$</th>
<th>Theoretical Mean</th>
<th>Simulation Average</th>
<th>Relative Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>$10^4$</td>
<td>250.66283</td>
<td>255.28726</td>
<td>0.018115</td>
<td></td>
</tr>
<tr>
<td>$10^5$</td>
<td>792.66546</td>
<td>796.79936</td>
<td>0.005188</td>
<td></td>
</tr>
<tr>
<td>$10^6$</td>
<td>2512.3905</td>
<td>2506.6283</td>
<td>0.002294</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>$k = 5$</th>
<th>$n$</th>
<th>Theoretical Mean</th>
<th>Simulation Average</th>
<th>Relative Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>$10^4$</td>
<td>4738.8042</td>
<td>5238.3565</td>
<td>0.0954</td>
<td></td>
</tr>
<tr>
<td>$10^5$</td>
<td>29899.8332</td>
<td>31580.8372</td>
<td>0.0615</td>
<td></td>
</tr>
<tr>
<td>$10^6$</td>
<td>188655.1938</td>
<td>195049.6191</td>
<td>0.0328</td>
<td></td>
</tr>
</tbody>
</table>

4.2 Complete graphs

<table>
<thead>
<tr>
<th>$k = 1$</th>
<th>$n$</th>
<th>Theoretical Mean</th>
<th>Simulation Average</th>
<th>Relative Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>$10^4$</td>
<td>5000</td>
<td>4984.5622</td>
<td>0.00310</td>
<td></td>
</tr>
<tr>
<td>$10^5$</td>
<td>50000</td>
<td>50013.3928</td>
<td>0.00026778</td>
<td></td>
</tr>
<tr>
<td>$10^6$</td>
<td>500000</td>
<td>502874.0474</td>
<td>0.0057152</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>$k = 2$</th>
<th>$n$</th>
<th>Theoretical Mean</th>
<th>Simulation Average</th>
<th>Relative Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>$10^4$</td>
<td>12500.0</td>
<td>12445.8116</td>
<td>0.00435395</td>
<td></td>
</tr>
<tr>
<td>$10^5$</td>
<td>125000.0</td>
<td>125448.4516</td>
<td>0.003574788</td>
<td></td>
</tr>
<tr>
<td>$10^6$</td>
<td>1250000.0</td>
<td>1247600.0552</td>
<td>0.001923649</td>
<td></td>
</tr>
</tbody>
</table>
Figure 2: 5000 runs of cutting paths of size $10^6$
Figure 3: 5000 runs of cutting complete graphs of size $10^6$
References


A Code in python

A.1 Paths

```python
import numpy as np
from numba import njit

@njit
def simulation(k, graph_size):
    # When the root dies
    root_dies = np.random.gamma(k)

    # stores the lowest record on the path
    low_record = root_dies

    # Store the number of records in this array
    records = np.ones(k, dtype=np.int32)

    for i in range(2, graph_size + 1):
        t = 0.0
        for r in range(k):
            t += np.random.exponential(1)
            if t < low_record:
                records[r] += 1
                if r == k - 1:
                    low_record = t

    return records

@njit
def print_simulation(k, graph_size, sample_number):
    for i in range(sample_number):
        # this prints out result for one simulation
        print(simulation(k, graph_size))

k = 2
graph_size = 10000
sample_number = 5000
print_simulation(k, graph_size, sample_number)
```
A.2 Complete Graphs

```python
import numpy as np
from numba import njit

@njit
def simulation(k, graph_size):
    # When the root dies
    root_dies = np.random.gamma(k)

    # Store the number of records in this array
    records = np.ones(k, dtype=np.int32)

    for i in range(2, graph_size + 1):
        t = 0.0
        for r in range(k):
            t += np.random.exponential(1)
            if t < root_dies:
                records[r] += 1

    return records

@njit
def print_simulation(k, graph_size, sample_number):
    for i in range(sample_number):
        # this prints out result for one simulation
        print(simulation(k, graph_size))

k = 2
graph_size = 1000000
sample_number = 5000

print_simulation(k, graph_size, sample_number)
```