Science Journal of Applied Mathematics and Statistics
2018; 6(6): 148-160
http://www.sciencepublishinggroup.com/j/sjams
doi: 10.11648/j.sjams.20180606.11
ISSN: 2376-9491 (Print); ISSN: 2376-9513 (Online)

Performance Analysis of Sorting Process with Different Sampling Strategies

Mahmoud Ragab

Mathematics Department, Faculty of Science, Al-Azhar University, Cairo, Egypt

Email address:
mahmoud.ragab@bue.edu.eg, meragab@yahoo.com

To cite this article:
Mahmoud Ragab. Performance Analysis of Sorting Process with Different Sampling Strategies. *Science Journal of Applied Mathematics and Statistics.* Vol. 6, No. 6, 2018, pp. 148-160. doi: 10.11648/j.sjams.20180606.11

Received: November 14, 2018; **Accepted**: December 13, 2018; **Published**: January 16, 2019

Abstract: Sorting data is one of the most important problems that play an important rule in many applications in operations research, computer science and many other applications. Many sorting algorithms are well studied but the problem is not to find a way or algorithm to sort elements, but to find an efficiently way to sort elements and do the job. The output is a stream of data in time and it is a sorted data array. We are interested in this flow of data to estaplish a smart technique to sort elements as well as efficient complexity. For the performance of such algorithms, there has been little research on their stochastic behavior and mathematical properties such existance and convergence properties. In this paper we study the mathematical behavior of some different versions sorting algorithms in the case when the size of the input is very large. This work also discuss the corresponding running time using some different strategies in terms of number of comparisons and swaps. Here, we use a nice approach to show the existence of partial sorting process via the weighted branching process. This approach was inspired by the methods used for the analysis of Quickselect and Quichsort in the standard cases, where fixed point equations on the Cadlag space were considered for the first time.

Keywords: Sorting, Algoritms, Divide and Conquer, Performance, Stochastic Process, Convergence Cadlag Functions, Asymptotics, Skorodhod Metric

1. Introduction

Classical analyses of sorting algorithms make the efficiency is the general purpose. The efficiency of the good algorithm depends on many various factors such as, software and hardware use. More recently researchers have recognized that on modern computers the cost of access- ing memory can vary dramatically depending on whether the data can be found in the first-level cache, or must be fetched from a lower level of cache or even main memory. The efficiency of an algorithm using a poor code, is differ than using a good code, in addition the operating system itself. The running time of executing sub-arrays is particularly important for sorting because the inner-loops of most common sorting algorithms consist of comparisons of items to be sorted. Thus, the expected of these comparisons to do the job is critical to the performance of sorting algorithms. Throughout this paper ignores all aspects of these factors unless the the behavior the sorting scheme whose outcome depends on a number of comparisons needed to the sorting algorithm in its input.

Selecting a good pivot is important. A poor choice of a pivot could give a running time quadratic proportional to the number of elements squared. Therefore selecting a good pivot greatly improves the speed of the Quicksort algorithm. Many people just use the first element in the list as the pivot, however this causes the sort to perform very badly if the data is already sorted. There are several methods to avoid the worst case in practical solutions. Unix uses the median of the first the last and the element in the middle. Therefore, it is important to be careful when choosing the pivot in each sorting step. For example, when Quicksort is used in web services, it is possible for an attacker to intentionally exploit the worst case performance and choose data which will cause a slow running time or maximize the chance of running out of stack space. The choice of a good pivot greatly improves the speed of the Quicksort algorithm.

The simplest way is to choose an arbitrary element say the first for example as pivot, this does not avoid the worst case. Instead of using the first element, a much better method is

called median of three Quicksort. In that method choose the pivot of each recursive stage as the median of a sample of three elements. Other method is to take tree samples, each sample contain 3 elements, take the median for each sample and choose the median of three medians as a pivot,this method called pseudomedian of 9 Quicksort.

To make sure to avoid any kind of presorting it is better to use the median element of the first, middle, and the last element as a pivot. To optimize the algorithm, for an array smaller than 7, the pivot is chosen as the middle key or sort with the standard Quicksort, for mid-sized arrays(for an array of size between 8 and 39) the pivot is chosen using the median-of-three Quicksort, and finally for larger arrays use the pseudomedian of 9 Quicksort. This helps some but unfortunately simple anomalies happens [1].

Some recent papers considered a version of Quicksort algorithgm called the random median Quicksort. For a random variable *k*, the pivot element has been selected to be the median of *2k+1* elements. In each step recursively recall of the algorithm. Later, many researchers has received the interest of the visualization of multi-pivot Quicksort in accordance with Yaroslavskiy proposed the duality pivot process which outperforms standard Quicksort by Java JVM. After that, this algorithm has been explained in terms of comparisons and swaps to discuss the efficiency [2]. Not long ago, a new version from Dual-pivot Quicksort algorithm has some other number k of pivots. Hence, this work discusses the idea of picking k pivots, $k = 1,2,\dots,n$ by random way and splitting the list simultaneously according to these. The modified version generalizes these results for multi process. The paper shows that the average number of swaps done by Multi-pivot Quicksort process and a special case. Moreover, Ragab et al. obtained a relationship between the average number of swaps of Multi-pivot Quicksort and Stirling numbers of the first kind [3].

Let X_n be the number of comparisons used by Quicksort to sort a list of size n. The random variable X_n is basically proportional to the running time of Quicksort which depends (a little bit) on the implementation and computer hardware. The average number $E(X_n)$ of comparisons is $E(X_n) \approx n \ln n$ [1]. The first complete running time analysis for a random divide and conquer was for Quicksort [4]. The random variable

$$Y_n = \frac{X_n - E(X_n)}{n}$$

converges in distribution to a random variable Y, which distribution is characterized as the unique solution of the stochastic fixed point equation

$$Y \overset{D}{=} UY^1 + (1-U)Y^2 + C(U)$$

with expectation 0 and finite variance. Here U is uniformly distributed on [0,1] and U, Y^1, Y^2 are independent. Y^1 and Y^2 have the same distribution and C is given by

$$C(x) := 2x \ln x + 2(1-x)\ln(1-x) + 1, \qquad x \in [0,1].$$

Quickselect or FIND, introduced by Hoare [2] in 1961 is a search algorithm widely used for finding the l-th smallest element out of n distinct numbers. Most of the mathematical results on the complexity of Quickselect are about expectations or distributions for the number of comparisons needed to complete its task by the algorithm [5, 6]. A pivot is uniformly chosen at random from the available n elements, and compares the $n-1$ remaining elements against it.

Let $X_n(k)$ be the number of comparisons needed to find the l-th smallest out of n. The running time of this algorithm is always a random variable either by random input or internal randomness. The expectation of $X_n(k)$ is explicitly known [1]

$$E(X_n(k)) = 2(n+3+(n+1)H_n - (k+2)H_k - (n+3-k)H_{n+1-k})$$

for $1 \le k \le n$, and H_k denotes the k-th harmonic number. An asymptotic approximation as $n \to \infty$ is

$$\frac{E(X_n(k))}{n} \approx 2 - 2t\ln t - 2(1-t)\ln(1-t)$$

for $0 \le t = \frac{k}{n} \le 1$. The asymptotic variance $Var(X_n(k))$ was derived using combinatorial and generating function methods [5]. Furthermore, Roesler [7] studied the limiting distribution of $X_n(k)$ or X_n as a process. A major tool are fixed point equation and the contraction method for operator K like

$$K(\mu) \overset{D}{=} \sum_{i \in \mathbb{N}} A_i Y_i + C \tag{1}$$

The random variables $(A,B,C), Y_i, i \in \mathbb{N}$ are independent and the random variables Y_i have the same distribution μ. In a more general form of (1), Knof and Roesler [3] considered general recurrence

$$Y^n \overset{D}{=} \sum_{i \in \mathbb{N}} A_i^n Y_i^{I_n} \circ B_i^n + C^n \tag{2}$$

on the set D of cadlag functions on the unit interval $[0,1]$. Here $((A_i^n, B_i^n, I_i^n)_i, C^n), Y_k^j$, $i,j,k \in \mathbb{N}$ are all independent. Y_i^j, A_i^n, C^n have values in the set D. The random variables B_i^n take values in $D_\uparrow$, the set of all maps from the unit interval to itself and piecewise increasing.

Under some assumptions they showed the existence of solutions of (2) via the weighted branching process, and Y^n converges in distribution to Y satisfying

$$Y \overset{D}{=} \sum_{i \in \mathbb{N}} A_i Y_i \circ B_i + C \tag{3}$$

The contraction method invented for the analysis of Quicksort, proved to be very successful for other algorithms [7, 8]. The contraction method is a general method to derive convergence in distribution of recursive structures. This method was pioneered by Roesler [7] and later by Neininger [9]. This method was explained in the context of several divide and conquer algorithms [8]. Knof [10] studied the finite dimensional distributions of D-valued processes Y^n by the contraction method. He introduced a suitable complete metric space and showed convergence of all finite dimensional distributions. His results include Quicksort. A nice version of the Quickselect processes is used to and show the convergence in other topology to a limiting process Y which is a fixed point of the map K [11]. In our work, we use a smart approach to show the existence of partial sorting process via the weighted branching process. Our approach was inspired by the methods used for the analysis of Quickselect [11], where fixed point equations on D were considered for the first time.

The equation (2) implies the distributional equality and $Y = Y^{\varnothing}$ satisfies the fixed point equation (12). The family Y^v is explicitly given in more details [5, 8].

This result is a probabilistic result and obtained via the Weighted Branching Process [7] and an explicitly given nice family of processes Y_n^v indexed by $n \in \mathbb{N}$ and the binary tree. Basically by the splitting U - rvs for the Y process also for the Y_n process.

2. The Recursive Equation of Comparisons

Like in the clasical Quicksort [2], choose with a uniform distribution a pivot, split the set S of numbers into the set $S_<$ of strictly smaller ones than the pivot, the pivot and the set $S_>$ of strictly larger ones in this order. Then continue recalling Quicksort always for the left list $S_<$. If $S_<$ is empty continue with the next leftmost sublist recalling the algorithm. If $S_<$ consists only of one element, output this number immediately and continue with the next leftmost list. Now define the random variable, $X(S,l)$ of comparisons required to sort l smallest elements in an array of distinct $|S|$ numbers. The random variable X satisfies the recursive formula

$$\begin{aligned}\big(X(S,l)\big)_l = \Big(&|S|-1+\mathbf{1}_{l\le I}X^1(S_<,l)\\ &+\mathbf{1}_{l>I}\left(X^1(S_<,I-1)+X^2(S_>,l-I)\right)\Big)_l\end{aligned} \tag{4}$$

for $l = 1,2,\ldots,|S|$.

Here $I = I(S) = |S_<| + 1$ denotes the rank of the pivot after comparisons. I means the index of the position it occupies in the sorted sequence and has values in $\{1,2,\ldots,n\}$ with a uniform distribution. The rv I is independent of all X random variables in equation (17). The random variable $X(S_<,\cdot)$ satisfies a similar recursion, where $I(S_<)$ is independent of everything before. Continuing this way we find the distribution of $X(S,|S|)$ as the Quicksort distribution sorting S by standard Quicksort. In the next proposition we will show that the distribution of $X(S,l)$ depends on S only via the size n of S.

Proposition 2.1 Let $S, \bar{S}$ be two sets of n different reals, and let $l = 1,2,\cdots n$. Then

$$\mathcal{L}\big(X(S,l)\big) = \mathcal{L}\big(X(\bar{S},l)\big).$$

Proof.

By induction on n. It is true for $n = 1$ and we are done. Assume it is true for $k \le n$, so we use the notation

$$\mathcal{L}(X(S,l))_l = \mathcal{L}((X(|S|,l))_l \quad for \quad |S| \le k. \tag{5}$$

The random variable $I(S) = I(|S|)$ is uniformly distributed on $\{1,2,\cdots k\}$. Let $(X^1(k,l))_{l\in\{1,2,\cdot,k\}}$, $(X^2(k,l))_{l\in\{1,2,\cdot,k\}}$ be independent random variables independent of I and with the distribution given in (5). Since $|S_<|, |S_>|, |\bar{S}_<|, |\bar{S}_>| \le n$ then

$$\mathcal{L}(X^1(S_<,l))_l = \mathcal{L}((X(\bar{S}_<,l))_l$$

and

$$\mathcal{L}(X^1(S_>,l))_l = \mathcal{L}((X(\bar{S}_>,l))_l$$

Now let $|S| = |\bar{S}| = n+1$. Then

$$\begin{aligned}\mathcal{L}(X(S,l))_l &= \sum_{i=1}^{k+1} P(I=i)\mathcal{L}(k-1+\mathbf{1}_{l\le i}X^1(S_<,l)\\ &+{}_{l>i}(X^1(S_<,i-1)+X^2(S_>,l-i)))_l\\ &= \sum_{i=1}^{k+1} P(I=i)\mathcal{L}(k-1+\mathbf{1}_{l\le i}X^1(i-1,l)\\ &+{}_{l>i}(X^1(i-1,i-1)+X^2(n-i,l-i)))_l\\ &= \sum_{i=1}^{k+1} P(I=i)\mathcal{L}(k-1+\mathbf{1}_{l\le i}X^1(\bar{S}_<,l)\\ &+{}_{l>i}(X^1(\bar{S}_<,i-1)+X^2(\bar{S}_>,l-i)))_l\\ &= \mathcal{L}(X(\bar{S},l))_l.\end{aligned}$$

$$\mathcal{L}(X^1(S_<,I-1)) = \mathcal{L}(X^1(|S_<|,I-1)),$$

and

$$\mathsf{L}(X^2(S_>, l-I)) = \mathsf{L}(X^2(|S_>|, l-I)).$$

Then

$$\mathsf{L}(X(S,l)) = \mathsf{L}(X(k+1,l)).$$

Then the statement is true for $n = k+1$ and therefore true for all $n \in \mathbb{N}$.

Remark:

The above Proposition is true states that the distribution of $X(S,l)$ depends only on $|S|$ and l. The equation (17) determines the distribution of $X(S,l)$ via the distribution for smaller sets and notice $I(S) = I(|S|)$

$$X(n,\cdot) \overset{D}{=} (n-1+1_{l<I_n} X^1(I_n-1,l) + 1_{l\geq I_n}(X^1(I_n-1, I_n-1) + X^2(n-I_n, l-I_n)))_l \quad (6)$$

For the distribution of $X(n,\cdot) = (X(n,l))_{l=1,2,\ldots,n}$, $n \in \mathbb{N}$. The rvs I_n, $(X^i(j,k))_{k=0}^{j-1}$, $i = 1,2$, $j < n$ are independent. The rv I_n has values on $\{1,2,\ldots,n\}$, $X^i(j,\cdot)$ has the same distribution as $X(j,\cdot)$ by recursion. We put for notational reasons (boundary conditions) $X^i(\cdot,0) \equiv 0 \equiv X^i(0,\cdot)$ for $i = 1,2$. Notice $X(n,n) = X(n,n-1)$.

In our version of Quicksort we use internal randomness by picking the pivot by random with a uniform distribution. Like in standard Quicksort, we could instead of internal randomness also use external randomness. Choose as input a uniform distribution on all permutations π of order n and pick as pivot any, for example always the first in the list. Now $X(\pi,\cdot)$ is a deterministic function depending on the input π. Seen as a rv with random input π we face the same distribution as with internal randomness. The main advantage using internal randomness is the same distribution of X for every input of the same seize. Alternatively we could start with iid uniform random variables on [0,1] and choose as pivot always the first element of the list. The algorithm itself would be deterministic, the time spend ($= X$) is a rv via the input of an iid sequence.

From equation (6) we obtain a recursion for the expectation $a(n,l) = E(X(n,l))$

$$a(n,l) = n-1+\frac{1}{n}\sum_{j=1}^{l}(a(j-1,j-1)) + a(n-j,l-j)) + \frac{1}{n}\sum_{j=l+1}^{n} a(j-1,l)$$

The term $a(n,n)$ is the expectation of sorting n numbers by Quicksort. All $a(n,l)$ are uniquely defined by the above equations and the starting conditions. Martinéz [12] obtained the explicit formula

$$a(n,l) = 2n + 2(n+1)H_n - 2(n+3-l)H_{n+1-l} - 6l + 6 \quad (7)$$

$1 \leq l \leq n \in \mathbb{N}$. (Notice not for $n = 0$ and $l = 0$.) H_j denotes the j-th harmonic number $H_j = \sum_{i=1}^{j} \frac{1}{i}$. For Quicksort we obtain the well known formula $a(n,n) = 2(n+1)H_n - 4n$.

Martinéz argued with *Partial Quicksort* $PQ(n,l)$, which for fixed n,l sorts the l smallest elements of a list. For more results and versions of it, optimality and one-dimensional distributions for Partial Quicksort [12]. The Quicksort process is an extension of Partial Quicksort in the sense of taking l as a time variable. We find first the $l-1$-smallest elements, then continue this search for the l-th smallest, then $l+1$-th smallest and so on. Now to the distribution of the process $X(n,\cdot)$ in the limit. This is the question, how much $X(n,l)$ differs from the average $a(n,l)$. We suggest a normalization of $X(n,l)$ to obtain a non degenerate limit distribution. Often this is the random variables of the form

$$Y_n(\frac{l-1}{n}) = \frac{X(n,l)-a(n,l)}{n} \quad (8)$$

for $l = 1,\ldots,n$ and $Y_n(1) = Y_n(\frac{n-1}{n})$ satisfy the recursion, $n \geq 1$

$$\left(Y_n(\frac{l}{n})\right)_{0\leq l\leq n} \overset{D}{=} (C(n,l,I_n) + 1_{l+1\leq I_n}\frac{I_n-1}{n} Y^1_{I_n-1}(\frac{l}{I_n-1}) + 1_{l+1>I_n}(\frac{I_n-1}{n} Y^1_{I_n-1}(1) + (1-\frac{I_n}{n}) Y^2_{n-I_n}(\frac{l-I_n}{n-I_n})))_{0\leq l\leq n}$$

$$C(n,l,i) = \frac{1}{n}(n-1+1_{l+1<i}(a(i-1,l+1)) + 1_{l+1=i} a(i-1,i-1) + 1_{l+1>i}(a(i-1,i-1) + a(n-i,l+1-i))$$

Notice Y_n is well defined and there are no boundary conditions besides $Y_0 \equiv 0 \equiv Y_1$.

We extend the process Y_n nicely to a process on the unit interval [0,1] with values in the space $D = D[0,1]$ of cadlag functions (right continuous functions with existing left limits) on the unit interval. This can be done by linear interpolation or a piece wise constant function. We shall use the extension

$$Y_n(t) := Y_n(\frac{\lfloor nt \rfloor}{n}) \quad (9)$$

The process Y_n is continuous at 1 and satisfies the recursion, we use $U_n = \frac{I_n}{n}$

$$Y_n \overset{D}{=} (C(n,\lfloor nt \rfloor, I_n) + 1_{t<U_n}\frac{I_n-1}{n} Y^1_{I_n-1}(\frac{nt}{I_n-1} \wedge 1)$$

$$+ {}_{t\geq U_n}\left(\frac{I_n-1}{n}Y^1_{I_n-1}(1)+\left(1-\frac{I_n}{n}\right)Y^2_{n-I_n}\left(\frac{t-U_n}{1-U_n}\right)\right)_t$$

for $n\in\mathbb{N}$. In short notation

$$Y_n \overset{D}{=} \varphi_n(U_n,(Y^1_k)_{k<n},(Y^2_k)_{k<n}) \tag{10}$$

for a suitable function φ_n.

If $n\to\infty$ then U_n converges in distribution to a rv U with a uniform distribution. We might expect that the process Y_n converges in some sense to a limiting process $Y=(Y(t))_{t\in[0,1]}$ with values in D satisfying something like the stochastic fixed point equation

$$Y \overset{D}{=} \left({}_{t<U}UY^1\left(\frac{t}{U}\right)+{}_{t\geq U}(UY^1(1)+(1-U)Y^2\left(\frac{t-U}{1-U}\right)\right) + C(U,t))_t \tag{11}$$

$$Y \overset{D}{=} \varphi(U,Y^1,Y^2) \tag{12}$$

for a suitable function φ. The rvs Y^1,Y^2,U are independent. Y^1 and Y^2 have the same distribution as Y and U is uniformly distributed on the unit interval [0,1]. The cost function $C=C(U,\cdot)$ is given by

$$C(u,t)=C(u)+2\,{}_{t<u}((1-t)\ln(1-t)-(1-u)\ln(1-u)$$
$$-(u-t)\ln(u-t)-(1-u)) \tag{13}$$

and is the limit of $C(n,l,i)$ with $\frac{l}{n}\to_n t,\frac{i}{n}\to_n u$, [8].

3. Binary Ulam-Harris Trees

Consider the infinite Ulam-Harris tree

$$\mathsf{V}:=\bigcup_{n\in\mathsf{N}_0}\mathsf{N}^n$$

be the infinite tree rooted at $\{\phi\}$ where $\mathbb{N}=\{1,2,\ldots\}$ denotes the set of positive integers and by convention $\mathbb{N}^0:=\{\phi\}$ contains the null sequence ϕ. Each $v=(v_1,v_2,\ldots,v_n)\in\mathbb{V}$ is called a node or vertex which we also write as $v_1v_2\ldots v_n$. The vertex v is uniquely connected to the root ϕ by the path

$$\varphi\to v_1\to v_1v_2\to\ldots\to v_1\ldots v_n.$$

The length of v is denoted by $|v|$, thus $|v_1\cdots v_n|=n$ and in particularly we use $|\phi|:=0$. For all $v\in\mathbb{V}$ and for every $k\in\mathbb{N}$ define

$$v_{|k}:=\begin{cases}\phi & k=0\\ v_1\cdots v_k & k<|v|\\ v & k\geq|v|\end{cases}$$

Further we use the notations for $w=w_1\ldots w_m$

$$vw:=v_1v_2\ldots v_nw_1w_2\ldots w_m,$$

$$\phi v:=v \quad and \quad v\phi:=v$$

In our work we suppress if possible the root ϕ.

Definition 3.1 For all $v,w\in\mathbb{V}$, the prefix order $\preceq$ on $\mathbb{V}$ is given by $v\preceq w:\Leftrightarrow\exists u\in\mathbb{V}:vu=w.$

The node v is called an ancestor or progenitor of w and conversely w is called a descendant of v. If $u\in\mathbb{N}$, then v is also called a mother of w and, conversely, w is called a child or offspring of v.

We further define $v\prec w:\Leftrightarrow v\preceq w\wedge v\neq w.$

In the context of branching tree, the relation $v<w$ may be interpreted as, the node v is strictly older than w in terms of generations. We extend this definition to subsets of $\mathbb{V}$. For all $L\subseteq\mathbb{V}$ and $v\in\mathbb{V}$, Define

$$v\circ L:\Leftrightarrow\exists w\in\mathsf{V}:vw\in L,$$

$$v\prec L:\Leftrightarrow\exists w\in\mathsf{V}\setminus\{\varphi\}:vw\in L$$

and

$$L\preceq v:\Leftrightarrow\exists w\in L \quad and \quad w\preceq v,$$

$$L\prec v:\Leftrightarrow\exists w\in L \quad and \quad w\prec v.$$

We use the m-ary tree $\{1,\ldots m\}^*:=\bigcup_{n\in\mathbb{N}_0}\{1,\ldots m\}^n$. In the case that $\mathsf{V}:=\bigcup_{n\in\mathsf{N}_0}\{1,2\}^n$, the tree is called binary tree.

Let $(\Omega,\mathcal{A},P)$ be a probability space, rich enough to carry all occurring random variables in our work. Let $(G,*)$ be a measurable semigroup ($*:G\times G\to G,(g,h)=g*h$ associative and measurable) with a grave $\Delta,(\forall g\in\Delta:\Delta*g=\Delta=g*\Delta)$ and a neutral element e ($\forall g\in G:e*g=g=g*e$).

The semigroup $(G,*)$ operates transitive and measurable on the measurable space H via $\otimes:G\times H\to H$ and $G\times H$ is endowed with the product σ-field. Let $T:\Omega\to G^{\mathbb{N}}$ be a random variable relative to the product space.

We use the notation $T=(T_1,T_2,\cdots)$, where $T_i:\Omega\to G$ is the i-th projection of T. Let $C:\Omega\to H$ be a random variable with values in a measurable semigroup H.

Let $(T^v,C^v),v\in\mathsf{V}$, be independent copies of (T,C) on the same probability space $(\Omega,\mathcal{A},P)$. We call T_i^v, the weight attached to the edge (v,vi) connecting v and vi. C^v

is called the weight of the vertex v. The interpretation of C^v is as a cost function on a vertex $v \in \mathbb{V}$. A tuple $(\mathsf{V},(T,C),(G,*),(H,\otimes))$ as above is called a weighted branching process (WBP).

For a weighted branching process without costs we write also $(\mathsf{V},T,(G,*))$. We shall use freely other trees such as m-ary trees $\{1,2,\cdots m\}^*$ of all sequences in an appropriate sense. The interpretation of G is as maps from H to H. If H has additional structure then we might enlarge G to have the induced structure.

For example if H is a vector space or an ordered set, we may extended G to a vector space of maps or ordered sets via the natural extension.

Definition 3.2 *Define recursively a family* $L := (L_v)_{v\in\mathbb{V}}$ *of random variables* $L_v : \Omega \to G$ *by*

$$L_\varphi := e, L_{vi} := L_v * T_i^v \quad \text{for all } v \in \mathsf{V}, i \in \mathsf{N}.$$

We call L_v the path weight from the root ϕ to the node v.

Similarly, we define recursively for all $v \in \mathbb{V}$, the family of path weights from v to vw $L^v := (L_w^v)_{w\in\mathbb{V}}$ by

$$L_\phi^v := e, \quad L_{wi}^v := L_w^v * T_i^{vw} \quad \text{for all} \quad w \in \mathbb{V}, i \in \mathbb{N}. \tag{14}$$

The path weight L_v has the following product representation

$$L_v = T_{v_1}^{\varphi} * T_{v_2}^{v_1} * T_{v_3}^{v_1v_2} * \ldots * T_{v_n}^{v_1v_2\cdots v_{n-1}}$$

$$=: \prod_{k=0}^{n-1} T_{v_{k+1}}^{v|k} \tag{15}$$

for $v = v_1v_2\cdots v_n$. Hence L_v is just the accumulated multiplicative weight along the path connecting the root ϕ with the node v. L_v forms the total weight of the branch starting from the root ϕ to node v accumulated under operation * of the edge weights.

An individual or a node v is called alive, if $L_v \neq \Delta$, otherwise the node is called dead. In particular all nodes with weight Δ are skipped in pictures. Define the total weight (cost) regarded up to the n-th generation by

$$R_n := \sum_{|v|<n} L_v \otimes C^v, \quad n \in \mathbb{N} \tag{16}$$

Because we deal only with positive values, everything will be well defined in our examples. We explain the forward and backward view via a weighted branching process on $\mathbb{R}^+$ for simplicity. The same argument will hold later for the the weighted branching process $(\mathbb{V},(T,C),(D,*),(D_\uparrow,\otimes))$. Consider a weighted branching process $(\mathbb{V},(T,C),(\mathbb{R}^+,\cdot),(\mathbb{R},\cdot))$.

For the next sections we need the following two examples. Although they provide known results the novelty is the line of arguments, which can be generalized and which are the key for the Quicksort process.

Example 3.1 *Here we show mainly the existence of the Quicksort distribution. Consider the weighted branching process* $(\{1,2\}^*,(\mathbb{R},\cdot,\mathbb{R},\cdot),((U,1-U),C(U)))$ *with* U *has a uniform distribution and*

$$C(x) := 1 + 2x\ln x + 2(1-x)\ln(1-x). \tag{17}$$

G is the multiplicative semi group $\mathbb{R}$ with the neutral element $e = 1$ and the grave $\Delta = 0$. G operates transitive on $H = \mathbb{R}$ by multiplication. Let $U^v, v \in V$ be independent rvs with a uniform distribution on $[0,1]$. Put

$$T_1^v = U^v, \quad T_2^v = 1 - U^v, \quad C^v = C(U^v)$$

Since H is an ordered vector space, we extend G with the interpretation of maps to the ordered vector space generated by the maps.

The total weighted cost $R_m := \sum_{v\in V_{<m}} L_v C^v$ up to the $m-1$ generation is an L_2-martingale and converges in L_2 and a.e. to a rv Q. The distribution of Q is called the Quicksort distribution. The distribution is uniquely characterized [13] as the solution of the stochastic fixed point equation

$$Q \overset{D}{=} UQ_1 + (1-U)Q_2 + C(U) \tag{18}$$

with expectation 0 and finite variance. Here $\overset{D}{=}$ denotes equality in distribution. The random variables U, Q_1, Q_2 are independent, U is uniformly distributed and Q_1, Q_2 have the same distribution as Q.

By the a.s. convergence of $R_m^v = \sum_{w\in V_{<m}} L_w^v C^{vw}$ the rvs

$$Q^v := \sum_{w\in V} L_w^v C^{vw} \tag{19}$$

exist and satisfy a.e.

$$Q^v = U^v Q^{v1} + (1-U^v)Q^{v2} + C(U^v) \tag{20}$$

for every $v \in V$. Of course the distribution of Q^v is a solution of (18).

Example 3.2 Convergence of the discrete Quicksort distributions [5]. The original problem concerns the number X_n of comparisons to sort n distinct reals. We use internal randomness. Then for $n \in \mathbb{N}$

$$X_n \overset{D}{=} n-1+X^1_{I_n-1}+X^2_{n-I_n}$$

with I_n, X^1, X^2 are independent, I_n has a uniform distribution on $1,\ldots,n$ and X^1_i, X^2_i have the same distribution as X_i. The boundary conditions are X_0 and X_1 are identical 0. The expectation of X_n is $a_n = a(n,n)$. The normalized rvs $Y_n = \frac{X_n - a_n}{n}$ satisfy the recursion

$$Y_n \overset{D}{=} \frac{I_n - 1}{n} Y^1_{I_n-1} + \frac{n-I_n}{n} Y^2_{n-I_n} + C_n(I_n)$$

where

$$C_n(i) := \frac{n-1-a_n+a_{i-1}+a_{n-i}}{n}$$

Now the abstract embedding into a WBP with an additional parameter $n \in \mathbb{N}$. Let H be the set of functions $h: \mathbb{N}_0 \to \mathbb{R}$ and G the set $H \times G_2$ where G_2 are the functions $g: \mathbb{N}_0 \to \mathbb{N}_0$ satisfying $g(0) = 0$ and $g(n) < n$ for all $n \in \mathbb{N}$. The semi group structure is given by

$$(f_1, g_1) * (f_2, g_2) = (f_1 f_2 \circ g_1, g_2 \circ g_1)$$

and the operation on H via

$$(f, g) \odot h = fh \circ g$$

The interpretation of $(f, g) \in G$ is as a map on H with f is a multiplicative factor and g an index transformation. The operation * corresponds to the convolution of maps on H. Since H is a vector space we may enlarge G naturally to a vector space.

Consider the binary tree V and let $U^v, v \in V$ be independent rvs with a uniform distribution. Let $I^v_n = \lceil nU^v \rceil$ (upper Gauss bracket) and define the transformations on the edges $(v, v1), (v, v2)$ by

$$J^v_1(n) = I^v_n - 1 \quad J^v_2(n) = n - I^v_n$$

$$T^v_1(n) = \left(\frac{J^v_1(n)}{n}, J^v_1(n)\right) \quad T^v_2(n) = \left(\frac{J^v_2(n)}{n}, J^v_2(n)\right)$$

and the vertex weight

$$C^v(n) = C_n(I^v_n)$$

The random variables

$$R^v_m := \sum_{w \in V_{<m}} L^v_w \odot C^{vw}$$

converge as $m \to 0$ a.e. and in L_2 to a limit R^v_∞ and satisfy

$$R^v_m = \sum_i T^v_i \odot R^{vi}_{m-1} + C^v$$

for $m \in \overline{N} = \mathbb{N} \cup \{\infty\}$.

Notice the connection to the previous description,

$$Y_n \overset{D}{=} R_\infty(n)$$

$R^v_\infty(n)$ converge for every $v \in V$ in L_2 to the rv Q^v from the Quicksort example [14]. We shall use $Q^v_n = R^v_\infty(n)$ in the sequel.

It is worth while to put the two examples together. Use $\overline{\mathbb{N}_0} = \mathbb{N}_0 \cup \{\infty\}$ instead of $\mathbb{N}_0$ in the second example and incorporate the first example via the value ∞.

For later purpose we establish a general Lemma. Let $(V = \{1,2\}^*, T = (U, 1-U), (\mathbb{R}_+, \cdot))$ be a WBP without costs. Let U have a uniform distribution. Let $0 \le C^v_n, n \in \mathbb{N}_0, v \in V$ be a sequence of positive real rvs. Let $X^v_0, v \in V$ be rvs and define $X^v_n, v \in V, n \in \mathbb{N}$ by

$$X^v_n = (U^v X^{v1}_{n-1}) \vee ((1-U^v) X^{v2}_{n-1}) + C^v_n$$

Lemma 3.1 Assume for $p > \ln 2$ $\infty > \sup_{v \in V} \left\| C^v_n \right\|_p \to_n 0$. Then X^v_n converges as $n \to \infty$ a.e. to 0 for all $v \in V$.

Proof. Let $M^v_n = \sup_{w \in V_n} L^v_w$. We intend to use a result of $\frac{\ln M^v_n}{n}$ converges as $n \to \infty$ to some $a \in \mathbb{R}$ where a is uniquely determined by $I(a) = 0$ [5]. The function

$$I(x) = \inf_{\alpha > 0} (\ln m(\alpha) - x\alpha)$$

is the Fenchel-Legendre Transform of the the convex function $\mathbb{R}_+ \ni \alpha \mapsto \ln m(\alpha)$ where $m(\alpha) = E \sum_{i \in \mathbb{N}} (T_i)^\alpha$. In our case $m(\alpha) = \frac{1}{\alpha + 1}$. The minimum of the function $\mathbb{R}_+ \ni \alpha \mapsto (\ln m(\alpha) - x\alpha)$ is attained at $\alpha = \alpha(x)$ with $\frac{-1}{\alpha + 1} - x = 0$. We obtain

$$I(x) = \ln(-x) - x\alpha(x) = \ln x + x - 1$$

Therefore $I(a) = 0 \leftrightarrow 0 = e^{1+a} + a$. The solution $a = -1$ is uniquely determined. The assumptions of Biggins result are satisfied.

For given $0 < \varepsilon < p - \ln 2$ let $n^v_0 = n^v_0(\omega)$ be such that for

all $n \geq n_0^v \;\; M_n^v \leq e^{(a+\varepsilon)n}$.

By induction we obtain

$$X_n^v \leq C_n^{\varnothing} + \vee_i T_i^v C_{n-1}^v \leq \sum_{j=0}^{n-1} \vee_{w \in V_j} L_w^v C_{n-j}^{vw}$$

We shall use for some n_1 sufficiently large

$$X_n^v \leq C_n^{\varnothing} + \vee_i T_i^v C_{n-1}^v \leq \sum_{j=0}^{n_0^v \vee n_1 - 1} L_w^v C_{n-j}^{vw} + \sum_{j=n_0^v \vee n_1}^{n-1} L_w^v C_{n-j}^{vw}.$$

The first sum converges a.e. to 0 for every n_1 and $v \in V$.

For the second let $0 < \varepsilon_n$ satisfy $\sum_n \varepsilon < \infty$. We intend to use Borel-Cantelli for the sequence $D_j^v = \{M_j^v \leq e^{(a+\varepsilon)j}, \vee_{w \in V_j} L_w^v C_{n-j}^{vw} \geq \varepsilon_j\}$

$$P(D_j^v) \leq \sum_{w \in V_j} \frac{E(C_{n-j}^{vw})^p}{e^{-(a+\varepsilon)pj} \varepsilon_j^p}$$

$$\leq e^{(\ln 2 + ap + \varepsilon p)j} \frac{c}{\varepsilon_j^p}$$

where c is the finite constant $\sup_n \sup_{v \in V} \left\| C_n^v \right\|_p \to_n 0$. Since $\sum_j P(D_j^v) < \infty$ the events D_j^v appear only finitely often for every $v \in V$. We conclude

$$\sum_{j=n_0^v \vee n_1}^{n-1} \vee_{w \in V_j} L_w^v C_{n-j}^{vw} \leq \sum_{j=n_0^v \vee n_1}^{n-1} \varepsilon_j + \sum_{j=n_0^v \vee n_1}^{n-1} {}_{D_j^v} \vee_{w \in V_j} L_w^v C_{n-j}^{vw} < \infty.$$

The first term on the right side is arbitrary small for sufficiently large n_1. The second term converges to 0 a.e. if n_1 converges to ∞. This concludes the statement.

4. The Quicksort Process

In this section we specify the Partial Quicksort process by using the weighted branching process. For the number of comparisons obtained by the Partial Quicksort $X(n,l)$, we suggest the normalization

$$Y^n(\frac{l}{n}) := \frac{X(n,l) - a(n,l)}{n} \tag{21}$$

Assume for simplicity that the rank of the pivot $I(n,l)$ is uniformly distributed random variable. The distribution depending only on l and on the size of the list n, we obtain a recursion given by the following Lemma [5].

Lemma 4.1

$$Y^n(\frac{1}{n}) = 1_{I \leq l} \left(1_{I \leq l} \frac{I-1}{n} Y_1^{I-1}(1) + \frac{n-I}{n} Y_2^{n-I}(\frac{l-I}{n-I}) \right) + 1_{I>l} \frac{I-1}{n} Y_1^{I-1}(\frac{l}{I-1}) + C^n(\frac{l}{n}) \tag{22}$$

where

$$C^n(\frac{l}{n}) = \frac{n-1}{n} - \frac{a(n,l)}{n} + 1_{I \leq l} \left(\frac{a(I-1,I-1)}{n} + \frac{a(n-I,l-I)}{n} \right) + 1_{I>l} \frac{a(I-1,l)}{n}. \tag{23}$$

In the next lemma we will give an explicit formula of the cost function $C(n,l,i)$. For the proof we need the well known properties of the n-th harmonic number [1].

Lemma 4.2 The function C as defined in (13) has the explicit representation

$$C(n,l,i) = 2\frac{i}{n+1}(H_i - H_{n+1}) + 2(1 - \frac{i}{n+1})(H_{n+1-i} - H_{n+1}) + 2_{l \leq i}(-\frac{i+2-l}{n+1}(H_{i+2-l} - H_{n+1}) - (1 - \frac{i}{n+1})(H_n + 1 - i) - H_{n+1} + (n+3-l)(H_{n+1-l} - H_{n+1}) - n + i - 2 + \frac{1}{(n+1)(i+1+l)} - \frac{1}{(n+1)(n+2-l)}).$$

Proof. From the equation above and using the properties of the n-th harmonic number we have

$$(n+1)C(n,l,i) = n - 1 - a(n,l) + 1_{l<i} a(i-1,l) + 1_{l=i} a(i-1,i-1) + {}_{l>i}(a(i-1,i-1) + a(n-i,l-i))$$

$$= -n - 1 - 2(n+1)H_n + 2(n+3-l)H_{n+1-l} + 6l - 6 + {}_{l<i}(2(i-1) + 2iH_{i-1} - 2(i+2-l)H_{i-1} - 6l + 6) + {}_{l=i}(2iH_{i-1} - 4(i-1)) + {}_{l<i}(2(n-i) + 2(n-i+1)H_{n-i}) - 2(n+3-l)H_{n+1-i} - 6(l-i) + 6$$

$$= -n - 1 - 2(n+1)H_{n+1} + 2(n+3-l)H_{n+1-l} + 2iH_i + 2_{l<i}(-(i+2-l)H_{i-1} + i - 1) + 2_{l=i}(l-1) + 2_{l>i}((n+1-i)H_{n+1-i} - 2(n+3-l)H_{n+1-l} + n + 1)$$

$$= 2iH_i + 2(n+1-i)H_{n+1-i} - 2(n+l)H_{n+1} + n + 1 + 2_{l \leq i}(-n + i - 2 - (i+2-l)H_{i-1} - (n+1-i)H_{n+1-i}$$

$$+(n+3-l)H_{n+1-l})$$

$$=2iH_i+2(n+1-i)H_{n+1-i}-2(n+l)H_{n+1}+n+1$$

$$+2\,{}_{l\le i}(-n+i-2-(i+2-l)H_{i+2-l}-(n+1-i)H_{n+1-i}$$

$$+(n+3-l)H_{n+3-l}-n+i-2+\frac{1}{i+1-l}-\frac{1}{n+2-l}).$$

Let D be the vector space of cadlag functions $f:[0,1]\to\mathbb{R}$ (right continuous with existing left limits). D is endowed with the Skorodhod topology induced by the Skorodhod J_1-metric

$$d(f,g)=\inf\{\varepsilon>0\,|\,\exists\lambda\in\Lambda:\|f-g\circ\lambda\|_\infty<\varepsilon,\ \|\lambda-id\|_\infty<\varepsilon\} \tag{24}$$

where Λ is the set of all bijective increasing functions $\lambda:[0,1]\to[0,1]$. We use the supremum norm $\|f\|_\infty=\sup_t|f(t)|$. The space (D,d) is a separable, non complete metric space, but a polish space [5]. The σ-field $\sigma(D)$ is the Borel-σ-field via the Skorodhod metric. The σ-field is isomorphic to the product σ-field $\mathsf{R}^A\cap D$ where A is a dense subset of $[0,1]$ containing the 1.

Let $\mathsf{F}(D)$ be the space of all measurable functions X with values in D. For $1\le p<\infty$ let $\mathsf{F}_p(D)$ be the subspace such that

$$\|X\|_{\infty,p}:=\big\|\|X\|_\infty\big\|_p<\infty \tag{25}$$

is finite. Here $\|\cdot\|_p, p<\infty$, denotes the usual L_p-norm for rvs. The map $\|\cdot\|_{\infty,p}$ is a pseudo metric on $\mathsf{F}_p(D)$. Let ~ be the common equivalence relation

$$X\sim Y\Leftrightarrow P(X\neq Y)=0$$

and $F_p(D)$ be the set of equivalence classes $[X]=\{Y\in\mathsf{F}(D)\,|\,X:Y\}$ intersected with $\mathsf{F}_p(D)$. Then it is well known

Proposition 4.1 *For* $1\le p<\infty$ *is* $(F_p(D),\|\cdot\|_{\infty,p})$ *a Banach space with the usual addition and multiplication*

$$[f]+[g]=[f+g],\quad c[f]=[cf],$$
$$\|[f]\|[f]\|_{\infty,p}=\|f\|_{\infty,p}.$$

Let $D_\uparrow$ be the subset of all functions $f:[0,1]\to[0,1]$ in D such that there exists $0=t_0<t_1<\cdots<t_r=1$ satisfying f is increasing on the interval $[t_{i-1},t_i)$ for $i=1,2,\cdots,r$. In the following we consider the composition of random variables and give some results needed in our work.

Lemma 4.3 Let X be a random variable with values in D and let B be a random variable with values in $D_\uparrow$. Then $X\circ B$ is a random variable with values in D.

Proof. Let $X:\Omega\to D$, and $B:\Omega\to D_\uparrow$. Since for all $\omega\in\Omega$, $B(\omega)\in D_\uparrow$, then the function $(X\circ B)(\omega)\in D$. It is sufficient to show $X\circ B(t):\Omega\to\mathbb{R}$ is measurable for all $t\in[0,1]$.

For all $t\in[0,1]$ define $B_j:\Omega\to\mathbb{R}, j\in\mathbb{N}$ by

$$B_j(\omega):=\frac{\lceil B(\omega)(t)\cdot j\rceil}{j}$$

we will approximate the $X\circ B$ by the random variables $X\circ B_j$ using the discretization on the values $0,\frac{1}{j},\frac{2}{j}\cdots,1-\frac{1}{j}$. $B_j(t)$ is a measurable for all $\omega\in\Omega$ since

$$\big(B_j(t)\big)^{-1}([a,1))=\big(B(t)\big)^{-1}\left([\frac{\lceil a\cdot j\rceil}{j},1)\right)\in\mathcal{A},$$

for all $j\in\mathbb{N}$ and $t\in[0,1]$. By the definition of B_j, $B_j(\omega)\ge B(\omega)(t)$ and $\lim_j B_j(\omega)=B(\omega)(t)$. Furthermore

$$X\circ B(\omega)(t)=X(\lim_j B_j(\omega)(t))=\lim_j X(B_j(\omega)(t))=\lim_j(X\circ B_j(\omega)(t))$$

Then $\lim_{j\to\infty}(X\circ B_j(t))$ is measurable since $X\circ B_j(t)$ is measurable. This implies that $X\circ B(t)$ is measurable for all $t\in[0,1)$ with respect to $\mathcal{B}(\mathbb{R})$.

5. Model Description

Let $G:=D\times D_\uparrow$ and let $H:=D$. For all $f,f_1,f_2\in D$ and $g,g_1,g_2\in D_\uparrow$ define the operation $*:G\times G\to G$ by

$$*((f_1,g_1),(f_2,g_2)):=(f_1,g_1)*(f_2,g_2)=(f_1\cdot f_2\circ g_1,g_2\circ g_1). \tag{26}$$

where $\circ$ denotes the convolution and $\cdot$ is the pointwise multiplication in D.

For all $f,f_1,h\in D$ and $g,g_1\in D_\uparrow$, define the operation $\otimes:G\times H\to H$ by

$$\otimes((f,g),h):=(f,g)\otimes h=f\cdot h\circ g.$$

For all $f,f_1,f_2,f',f_1',f_2'\in D$,the operation $*$ is bilinear in the first coordinate on $D\times D_\uparrow$,

$$(f_1+f_2,g)*(f',g')=(f_1,g)*(f',g')+(f_2,g)*(f',g')$$

$$(f,g)*((f_1'+f_2',g')=(f,g)*(f_1',g')+(f,g)*(f_2',g').$$

And for all $f,f_1',f_2',h\in D$ $g,\in D_{\uparrow}$

$$(f_1+f_2,g)\otimes h=(f_1,g)\otimes h+(f_2,g)\otimes h,$$

$$(f,g)\otimes(h_1+h_2)=(f,g)\otimes h_1+(f,g)\otimes h_2.$$

The tuple $(f,g)\in G$ has the interpretation of a map $M_{f,g}:H\to H$ acting as

$$(M_{f,g}(h))(t,n)=f(t,n)h(g(t,n)). \tag{27}$$

The first coordinate f is a space transformation and the second coordinate g is a time and index transformation. The semigroup structure $*$ is the composition of the corresponding maps. Since H is a vector space and $\mathbb{R}$ is a lattice, we will embed G to maps H^H and use freely the induced structures $+,\cdot$ and $\vee$.

$$(M_{f,g}+M_{f_1,g_1})(h)=M_{f,g}(h)+M_{f_1,g_1}(h),$$

$$a\cdot(M_{f,g})(h)=(a\cdot M_{f,g})(h),$$

and

$$(M_{f,g}\vee M_{f_1,g_1})(h)=((M_{f,g})(h))\vee(M_{f_1,g_1}(h)).$$

It is easy to see the equation (26) as follow

$$M_{f,g}\circ M_{f_1,g_1}(h)(t,n)=M_{f,g}(f_1(t,n),h(g_1(t,n)))$$

$$=f(t,n)\cdot(f_1,h(g_1))(g(t,n))$$

$$=f(t,n)f_1(g(t,n)h(g_1(g(t,n)))$$

$$=M_{f\cdot f_1\circ g,g_1\circ g}(h)(t,n).$$

We notice here, the sorting partitioning strategy suggests a binary tree. Consider the binary tree $\mathsf{V}=\{1,2\}^{\mathsf{N}}$. For $v=v_1v_2\cdots v_n\in\mathbb{V},n\in\mathbb{N}$ and for $m\in\mathbb{N}$, $m\le n$, $v\,|\,m=v_1v_2\cdots v_m$ denote to the m-th coordinate of v. Let $U^v:\Omega\to[0,1],v\in\mathsf{V}$ be independent and identically uniformly distributed random variables on the unit interval $[0,1]$. Let $Q^v,v\in\mathsf{V}$ be the random variable has a limiting Quicksort distribution.

Define a map $T_i^v:\Omega\to G$, the weights on the edges $(v,vi),v\in\mathbb{V},i\in\{1,2\}$ by

$$T_i^v:=\left(A_i^v,B_i^v\right)\quad for\ all\quad v\in\mathsf{V}$$

and

$$T^v=\left(T_1^v,T_2^v,0,\cdots\right)$$

For all $v\in\mathsf{V}$, define the following parameters

$$A_1^v(t):=_{U^v>t}U^v,$$

$$A_2^v(t):=_{U^v\le t}(1-U^v),$$

$$A_3^v(t):=0=A_4^v(t)=\cdots$$

$$B_1^v(t):=\left(\frac{t}{U^v}\wedge 1\right)_{t\in[0,1]}$$

$$B_2^v(t):=\left(\frac{t-U}{1-U}\vee 0\right)_{t\in[0,1]}$$

$$B_3^v(t):=0=B_4^v(t)=\cdots. \tag{28}$$

Define a map $C^v:\Omega\to H$, the vertex weight by

$$C^v:=C(U^v,.)+_{U^v\le t}U^vQ^{v1}, \tag{29}$$

where C is given in (23).

Here $\left(A^v,B^v,C^v\right),v\in\mathsf{V}$ in terms of U^v are iid copies of (A,B,C). The edge weight T_i^v attached to the edge (v,vi) is given by $\left(A_i^v,B_i^v\right)$. The tuple $\left(\mathsf{V},(T^v,C^v)_{v\in\mathsf{V}},(G,*,H,\otimes)\right)$ is a weighted branching process. Consider the weighted branching process as given above and for all $v\in\mathbb{V}$, define the sequence

$$R_n^v:=\sum_{w\in\mathbb{V}_{<n}}L_w^v\otimes C^{vw},\quad R_0^v=0, \tag{30}$$

where the family of path weights $L^v:=(L_w^v)_{w\in\mathsf{V}}$ from the node v is given recursively by (14).

Lemma 5.1 Let U be a uniformly distributed random variable on $[0,1]$ and C defined as in (29). Then for all $t\in[0,1)$ $E(C)=0$ and $Var(C)<\infty$.

Proof. $E(C)=2\int_0^1 u\ln u du+E\int_0^t(1+uQ+2(1-u)\ln(1-u))du$

$$+\int_t^1(2u-1+2(1-t)\ln(1-t)-2(u-t)\ln(u-t))du.$$

It follows

$$\int_0^1 u\ln u du=\int_0^1 u^2(\ln u)^2 du=-\frac{1}{4},$$

$$\int_0^1(1-u)^2(\ln(1-u))^2 du=\frac{2}{27},$$

$$\int_0^t (1-u)\ln(1-u)du = \frac{1}{2}\left(-(1-t)^2\ln(1-t)+\frac{t^2}{2}-t\right),$$

$$\int_t^1 (u-t)\ln(u-t)du = \frac{1}{2}\left((1-t)^2\ln(1-t)-\frac{1}{2}(1-t)^2\right),$$

$$\int_0^t (t-u)\ln(t-u)du = \frac{1}{2}\left(t^2\ln t-\frac{t^2}{2}\right),$$

and

$$\int_0^1 u(1-u)(\ln u)\ln(1-u)du = \frac{1}{108}\left(37-3\pi^2\right)$$
$$\approx 0,068437 < 1.$$

Therefore

$$E(C)) = 2\int_0^1 u(\ln u)du + \int_0^t (1+uQ+2(1-u)\ln(1-u))du = 0.$$

Moreover

$$E\left(C^2\right) \le 1+4\int_0^1 u\ln u du+4\int_0^1 u^2(\ln u)^2 du$$
$$+4\int_0^1 (1-u)\ln(1-u)du$$
$$+4\int_0^1 (1-u)^2(\ln(1-u))^2 du$$
$$+8\int_0^1 u(1-u)\ln u\ln(1-u)du$$
$$+\frac{1}{3}E(Q^2) < \infty.$$

Lemma 5.2 The random variables R_n^v defined as in (30) satisfies the backward recursion

$$R_m^v = \sum_{i=1}^{2} T_i^v \otimes R_{m-1}^{vi} + C^v$$

for all $v \in \mathbb{V}$ and $m \in \mathbb{N}$.

Proof. The sum R_m^v is well defined and by equations (30) and (??), we have

$$R_m^v = \sum_{|w|<m} (T_{w_1}^v * T_{w_2}^{vw_1} * \cdots T_{w_k}^{vw_1\cdot w_{k-1}}) \otimes C^{vw}$$

$$= \sum_{k=0}^{m}\sum_{w\in V_k} (T_{w_1}^v * T_{w_2}^{vw_1} * \cdots T_{w_k}^{vw_1\cdot w_{k-1}}) \otimes C^{vw}$$

$$= C^v + \sum_{k=1}^{m}\sum_{i=1}^{2}\sum_{x\in V_{k-1}} (T_i^v * T^{vi_{x_1}} * \cdots T^{vix_1\cdots x_{k-2}}) \otimes C^{vix}$$

$$= C^v + \sum_{i=1}^{2} T_i^v * (\sum_{k=1}^{m}\sum_{x\in V_{k-1}} T^{vi_{x_1}} * \cdots T^{vix_1\cdots x_{k-2}}) \otimes C^{vix}$$

$$= C^v + \sum_{i=1}^{2} T_i^v * R_{m-1}^{vi}.$$

The connection between the operator K and the weighted branching process R_n, in the case of the Partial Quicksort is given by the following Corollary.

Corollary 5.1 Let the starting measure μ_0 be the point measure on the function $\underline{0} \in D$ identical 0. Then the random variables R_n defined as in (30) satisfies

$$R_{n+1} = C^\phi + \sum_{i\in\mathbb{N}} A_i^\phi \cdot R_n^i \circ B_i^\phi, \tag{31}$$

where R_n^i denotes the the random variable R_n for the tree with root i. The distribution of R_n is $K^n(\mu_0)$.

Proof. In the positive case all R_n and $n-$ fold iterates $K^n(\mu_0)$ are well defined [3]. By Lemma 5.2, the sequence R_n satisfies the backward recursion

$$R_{n+1} = C_\phi + \sum_{i\in\mathbb{N}} T_i^\phi \otimes R_n^i.$$

If $T_i^v := \left(A_i^v, B_i^v\right)$ *for all* $v \in \mathsf{V}, i \in \mathsf{N}$ and $T^v = (T_1^v, T_2^v)$, then by the settings in Section 5 we have

$$R_n = C^\phi + \sum_{i\in\mathbb{N}} \left(A_i^\phi, B_i^\phi\right) \otimes R_n^i$$

$$= C^\phi + \sum_{i\in\mathbb{N}} A_i^\phi \cdot R_n^i \circ B_i^\phi.$$

To prove the distributional result on R_n we use the mathematical induction on n. The induction base case when $n=1$ is true since $R_1 = C^\phi$ has the distribution $K(\mu_0)$. For the inductive step n to $n+1$ argue by the backward recursion

$$R_{n+1} = C^\phi + \sum_{i\in\mathbb{N}} A_i^\phi \cdot R_n^i \circ B_i^\phi$$

$$= K(\mathcal{L}(R_n)) = K(K^n(\mu_0)) = K^{n+1}(\mu_0).$$

Here the random variables $\left(A_i^\phi, B_i^\phi, C^\phi\right), R_n^i, i \in \mathbb{N}$ are

independent.

For all $v \in \mathsf{V}, n \in \mathsf{N}$ define $S_n^v : \Omega \to D$ by

$$S_n^v := \sum_{|w|=n} L_w^v \otimes C^{vw}, \quad S_0^v = C^v. \tag{32}$$

The equation (32) defines the total weight in the n-th generation. Our interest concentrates on the total weight (cost) regarded up to the n-th generation. From equations (32) and (16), we have

$$S_n^v = R_{n+1}^v - R_n^v \tag{33}$$

Now, it is easy to show that for all $\omega \in \Omega$ and $S_n(\omega) \in D$,

$$S_n^v = \sum_{i \in \blacklozenge} T_i^{\varphi} * S_{n-1}^{vi} \tag{34}$$

where T_i and S_n^1 and S_n^2 are the S_n random variable for the tree with root i.

Using the equation (33) and (34), for all $t \in [0,1]$ we obtain

$$S_n^v(t) = \sum_{i \in \mathbb{N}} \left(A_i^v \cdot S_{n-1}^{vi} \circ B_i^v \right)(t)$$

$$= \left(A_1^v \cdot S_{n-1}^{v1} \circ B_1^v + A_2^v \cdot S_{n-1}^{v2} \circ B_2^v \right)(t)$$

and therefore

$$S_n^v(t) = {}_{U^v > t} U^v S_{n-1}^{v1}\left(\frac{t}{U^v}\right) + {}_{U^v \le t}(1-U^v) S_{n-1}^{v2}\left(\frac{t-U^v}{1-U^v}\right). \tag{35}$$

Lemma 5.3 Let $S_n^v, n \in \mathbb{N}, v \in \mathbb{V}$ be as above. Then for all $n \in \mathbb{N}$, $E(\|S_n\|_\infty) = 0$ and $Var(\|S_n\|_\infty)$ converges exponentially fast to 0, as $n \to \infty$.

Proof. Notice $(S_n^v)_n \overset{\mathsf{D}}{=} (S_n^{\phi})_n$. By equation (35)

$$\|S_n\|_{2,D}^2 = \|T_i S_{n-1}\|_{2,D}^2$$

$$= E\left(\sup_t \left| {}_{t<U} U S_{n-1}^1\left(\frac{t}{U}\right) + {}_{t \ge U}(1-U) S_{n-1}^2\left(\frac{t-U}{1-U}\right) \right|^2 \right)$$

$$= E \sup_t \left(\left({}_{t<U} U S_{n-1}^1\left(\frac{t}{U}\right) \right)^2 + \left({}_{t \ge U}(1-U) S_{n-1}^2\left(\frac{t-U}{1-U}\right) \right)^2 \right).$$

All mixed terms are zero. The first term is

$$E \sup_t \left({}_{t<U} U S_{n-1}^1\left(\frac{t}{U}\right) \right)^2 = E\left(U^2 \sup_{t \le u} \left(S_{n-1}^1\left(\frac{t}{U}\right)\right)^2 \right)$$

$$\le E\left(U^2 \left\| S_{n-1}^1 \right\|_\infty^2 \right)$$

$$\le E(U^2) E\left(\left\| S_{n-1}^1 \right\|_\infty^2 \right) \tag{36}$$

And the second term is

$$E \sup_t \left({}_{t \ge U}(1-U) S_{n-1}^2\left(\frac{t-U}{1-U}\right) \right)^2$$

$$= E\left((1-U)^2 \sup_{t \ge U} S_{n-1}^2\left(\frac{t-U}{1-U}\right)^2 \right)$$

$$\le E\left((1-u)^2 \left\| S_{n-1}^2 \right\|_\infty^2 \right)$$

$$\le E\left(1-u\right)^2 E\left(\left\| S_{n-1}^2 \right\|_\infty^2 \right) \tag{37}$$

Then we have

$$\|S_n\|_{2,D}^2 \le \left(EU^2 + E(1-U)^2 \right) E\left(\left\| S_{n-1}^2 \right\|_\infty^2 \right)$$

$$\le \frac{2}{3} E\left(\left\| S_{n-1}^2 \right\|_{2,D}^2 \right).$$

For $n \in \mathbb{N}$, let $b_n := \|S_{n-1}\|_{2,D}$. Notice b_n does not depend on the vertex v. Then

$$b_n^2 \le \frac{2}{3} b_{n-1}^2,$$

and therefore by iteration of the inequality

$$b_n^2 \le \left(\frac{2}{3}\right)^n b_0^2 < \infty.$$

Where $b_0 = \|S_0\|_{2,D} = \|C(U,.)\|_{2,D} < \infty$.

Let $A(t) := \sum_i |A_i(t)|$ for all $t \in [0,1]$ and note that $E\|A\|_\infty = E(U) + E(1-U) = 1$ and

$$\sum_{i=1,2} E \sup_{t \in [0,1]} |A_i(t)|^2 = EU^2 + E(1-U)^2 = 2E(U^2) = \frac{2}{3} < 1.$$

Lemma 5.4 For fixed $t \in [0,1]$ and $n \in \mathbb{N}$, $Var(R_n(t))$ converges exponentially fast to 0, as $n \to \infty$.

Proof. The random variables $S_j, \in \mathbb{N}$ are pointwise well defined and measurable. By Theorem 5.3 we have

$$E\left(R_n(t)^2\right) = E\left(\left(\sum_{j=0}^{n-1} S_j(t) \right)^2 \right) = E\left(\sum_i \sum_j S_j(t) S_j(t) \right)$$

$$= \sum_i \sum_j E\left(S_i(t) S_j(t) \right)$$

$$= \sum_{i=0}^{n-1} E\left((S_i(t))^2\right) + \sum_{i=0}^{n-1}\sum_{i\neq j} E\left(S_i(t)S_j(t)\right).$$

By the Cauchy-Schwarz inequality, $S_i(t)S_j(t)$ is integrable. For $i, j \in \mathbb{N}$ and $i \leq j$, define

$$\mathcal{B}_i := \sigma\left(\left(T^v, C^v\right)_{|v|\leq n}\right)$$

By conditional expectation we have for $i \leq j$

$$E\left(S_i(t)S_j(t)\right) = E\left[E\left(S_i(t)S_j(t) \mid \mathcal{B}_i\right)\right]$$

$$= E\left[S_i(t)E\left(S_j(t) \mid \mathcal{B}_i\right)\right] = 0$$

Therefore

$$E\left((R_n(t))^2\right) = \sum_{i=0}^{n-1} E\left((S_i(t))^2\right)$$

And therefor Lemma 5.3 finishes the proof of the following main result.

Theorem 5.1 Let $(V,((T_1,T_2),C),(G,*),(H,\otimes))$ be the weighted branching process defined as above. Then R_n^v converges uniformly as $n \to \infty$ almost everywhere in D to a random variable R^v for all $v \in \mathbb{V}$. The family $R^v, v \in \mathbb{V}$ satisfies

$$R^v = \sum_{i=1}^{2} T_i^v R^{vi} + C^v \tag{38}$$

almost everywhere. Moreover, for every $p > 1$ holds

$$\left\|\|R\|_\infty\right\|_p \leq \frac{8 + (\frac{1}{p+1})^{\frac{1}{p}} \|Q\|_p}{1 - k_p} \tag{39}$$

where $k_p = (\frac{2}{p+1})^{\frac{1}{p}}$ and Q is a random variable with the Quicksort distribution.

6. Conclusion

This work has successfully expressed a new variant of sorting algorithm mathematically in terms of stochastic processes model. The corresopnding mathematical model (2) implies the distributional equality and $Y = Y^{\varnothing}$ satisfies the fixed point equation (12). The family Y^v is explicitly given in [8]. This result is a probabilistic result and obtained via the Weighted Branching Process [12] and an explicitly given nice family of processes Y_n^v indexed by $n \in \mathbb{N}$ and the binary tree. Basically by the splitting U - rvs for the Y process also for the Y_n process. In particular, the full potential of the technique used in this paper is useful for further studies of a modified version of the sch kind of the sorting algorithm for developing the eefficiency. In the future, we can expect vigorous research progress along these stratigies. The weighted branching process represents a fundamental approach for analyzing algorithms efficiency that will remain a source of inspiration for researchers for the recent era.

References

[1] Daniel H. Greene and Donald E. Knuth. *Mathematics for the analysis of algorithms*. Modern Birkhäuser Classics. Birkhäuser Boston Inc., Boston, MA, 2008. Reprint of the third (1990) edition.

[2] C. A. R. Hoare. Quicksort. *Comput. J.*, 5:10–15, 1962.

[3] Mahmoud Ragab, Beih El-Sayed El-Desouky, and Nora Nader. Analysis of the multi-pivot quicksort process. Open Journal of Modelling and Simulation, 5(01):47–58, 2017.

[4] U. Rösler. On the analysis of stochastic divide and conquer algorithms. Algorithmica, 29(1-2):238–261, 2001. Average-case analysis of algorithms (Princeton, NJ, 1998).

[5] Mahmoud Ragab. Partial Quicksort and weighted branching processes. PhD thesis, Kiel, Christian-Albrechts-Universität, Diss., 2011, 2011.

[6] Mahmoud Ragab. Running time Analysis of Sorting Algorithm via an Asymptotic Distribution. International Refereed Journal of Engineering and Science (IRJES), 6(8):55–69, 2017.

[7] Uwe Rösler. The weighted branching process. Dynamics of complex and irregular systems (Bielefeld, 1991), pages 154–165, 1993.

[8] U. Roesler and L. Rueschendorf. The contraction method for recursive algorithms. Algorithmica, 29(1-2):3–33, 2001. Average-case analysis of algorithms (Princeton, NJ, 1998).

[9] Ralph Neininger and Ludger Rüschendorf. Analysis of algorithms by the contraction method: additive and max-recursive sequences. In Interacting stochastic systems, pages 435–450. Springer, Berlin, 2005.

[10] Diether Knof and Uwe Roesler. The analysis of find or perpetuities on cadlag functions. *Discrete Mathematics and Theoretical Computer Science*, 2010.

[11] Mahmoud Ragab and Uwe Roesler. The quicksort process. Stochastic processes and their Applications, 124(2):1036–1054, 2014.

[12] Conrado Martnez and Uwe Roesler. Partial quicksort and quickpartitionsort. *DMTCS Proceedings*, (01):505–512, 2010.

[13] Mahmoud Ragab. On the quicksort algorithm and its related process. Journal of Mathematical Modeling and Operations Research, 01(01):13–30, 2015.

[14] Mahmoud Ragab, Beih El-Sayed El-Desouky, and Nora Nader. On the convergence of the dual-pivot quicksort process. Open Journal of Modelling and Simulation, 4(01):1–15, 2016.