

Algorithms, random tree models and combinatorial objects

Alois Panholzer

TU Wien

The author thanks the editors of IMN for their invitation to give an overview about his recent research topics. The following paper exemplifies by concrete examples my main research directions, which concern the analysis of algorithms and data structures, random tree models and combinatorial objects.¹

1 Average-case analysis of Union-Find algorithms

The concept of so-called “average case analysis of algorithms and data structures” has been introduced by Donald Knuth in his famous book series “The art of computer programming” [11] in the late sixties and early seventies of the last century. It has been popularized and developed further by well known mathematicians and computer scientists as, e.g., Philippe Flajolet, Rainer Kemp, Helmut Prodinger and Robert Sedgewick; together they founded an international conference series dedicated to that topic.

The average case analysis of algorithm deals with questions concerning the “average behaviour of cost measures” (as, e.g., running time, space requirement, recursion depth) of algorithms and data structures. Contrary to the “worst case analysis of algorithms”, where the most unfavourable situation for a particular algorithm is analyzed, one is here interested in the “typical” behaviour of it. In practice such an analysis is often of more interest, since it describes in a better way the actual “performance” when using many executions of the algorithm considered. As an example, the popular Quicksort sorting algorithm (basic implementation) for sorting a data array needs $\sim n^2/2$ comparisons between data elements to sort an array

¹*Anmerkung des Herausgebers:* Die Österreichische Mathematische Gesellschaft hat Alois Panholzer eingeladen, als Förderungspreisträger 2009 an dieser Stelle einen Überblick über seine Arbeit zu geben.

with n entries in the worst case; however, an average case analysis shows that the algorithm only uses $\sim 2n \log n$ comparisons on average (under the so-called “random permutation model”) and thus that the “typical” behaviour is much better.

Needless to say that the worst case analysis of an algorithm is also an important cost measure, since, informally speaking, it describes “what could happen” although, with some luck, “such bad situations will not occur often”. Moreover, unlike for a description of the worst case behaviour, it is for describing the average case behaviour of an algorithm important to introduce and apply appropriate probabilistic models for the distribution of the input data. Going back to the example of the Quicksort algorithm, e.g., if one already knows that the data array to be sorted consists of partially ordered lists then one cannot expect that an average case analysis of the non-randomized algorithm (using in every recursion step the first element of an array to compare it with each of the other elements) carried out for the “random permutation model” (where one assumes that all $n!$ permutations of $\{1, 2, \dots, n\}$ could be chosen as input array with the same probability) will describe the performance of the algorithm well in this situation.

In a mathematical setting an average case study of a cost measure of a particular algorithm corresponds to an analysis of the distributional behaviour of a sequence of random variables X_n , where n measures the size of the input data (sometimes it is appropriate to introduce random variables depending on further parameters, not only on the input size). Besides the description of the most basic quantity, namely the expected value $\mathbb{E}(X_n)$, one is often interested in a more detailed study of X_n leading to the variance $\mathbb{V}(X_n)$ (and thus to concentration results), results on the behaviour of higher moments, limiting distribution results, estimates on the occurrence of rare events (so-called “tail estimates”), etc.

Now we turn our attention to a particular problem, for which we will discuss such an average case analysis of an algorithm used in this context. The so-called “Union-Find problem” (see [1]) consists of maintaining a representation of equivalence classes or partitions of a finite set, such that the following two basic operations have to be supported, UNION: “merge two different equivalence classes s and t into a single equivalence class” and FIND: “find the equivalence class that contains a given element x ”. This problem arises naturally in several applications in computer science as, e.g., in minimum-cost spanning tree algorithms (amongst them the popular algorithm of Kruskal) and algorithms for detecting the equivalence of finite automata.

Following [1] the Union-Find problem for partitions $P(S)$ of a finite set S can be treated by introducing the following data structure:

For every element $x \in S$ we store in $R[x]$ the name of the equivalence class containing x . Furthermore for every equivalence class $s \in P(S)$ we store in $N[s]$ the number of elements of s and in $L[s]$ we store the elements of s in a linked list.

Yao [25] has described basic algorithms for implementing the operation UNION; amongst them the algorithm “Quick Find Weighted” is the most efficient and most popular one:

“Quick Find Weighted” (QFW) If we want to merge the different equivalence classes s and t then we update the class with less elements:

if $N[s] \leq N[t]$ then set $R[x] := t$ for all x in $L[s]$, append $L[s]$ to $L[t]$, set $N[t] := N[t] + N[s]$ and call the new equivalence class t , otherwise set $R[x] := s$ for all x in $L[t]$, append $L[t]$ to $L[s]$, set $N[s] := N[s] + N[t]$ and call the new equivalence class s .

The cost of the UNION operation when merging the equivalence classes s and t can be measured by the number of updated elements, i.e., the number of allocations $R[x] := s$ (or $R[x] := t$). For QFW the cost of one merging step is thus given by $\min(N[s], N[t])$, the minimum of the class sizes. When applying this algorithm the FIND operation for an element x , i.e., finding the equivalence class where x is contained, simply consists of evaluating $R[x]$ and can thus be carried out in bounded time; this explains the name “Quick Find”.

In order to measure the average behaviour of the QFW algorithm (and other merging algorithms) various models for sequences of UNION operations have been introduced. We focus here on the so-called *random spanning tree model*. We deal with a set S of size n , where at the beginning all elements $x \in S$ are forming an equivalence class $\{x\}$. These n equivalence classes will then be merged into larger and larger classes by carrying out UNION operations according to the following rules. In this model a spanning tree of the complete graph with vertex set S is chosen at random and then the edges of this spanning tree are randomly ordered, i.e., enumerated from 1 to $n - 1$. Let us assume this leads to a sequence of edges $e_1 = (x_1, y_1)$, $e_2 = (x_2, y_2)$, ..., $e_{n-1} = (x_{n-1}, y_{n-1})$, with $x_i, y_i \in S$. This gives then the following sequence of UNION operations: $\text{UNION}(R[x_1], R[y_1])$, $\text{UNION}(R[x_2], R[y_2])$, ..., $\text{UNION}(R[x_{n-1}], R[y_{n-1}])$. Thus in this model all $n^{n-2}(n-1)!$ possible sequence of UNION operations of that kind will occur equally likely.

The basic parameter of interest describing the average performance of the algorithm QFW is then the total cost, i.e., the sum of the costs of every merging step, when merging the elements of a set S of size n , where at the beginning all elements are lying in different equivalence classes, into one equivalence class (containing all elements of S) by carrying out a sequence of $n - 1$ UNION operations according to the rules given in the random spanning tree model. This parameter, which can be considered as a random variable depending only on the size n of the set S of elements, is denoted by X_n^{QFW} . The QFW algorithm under the random spanning tree model is illustrated by an example in Figure 1.

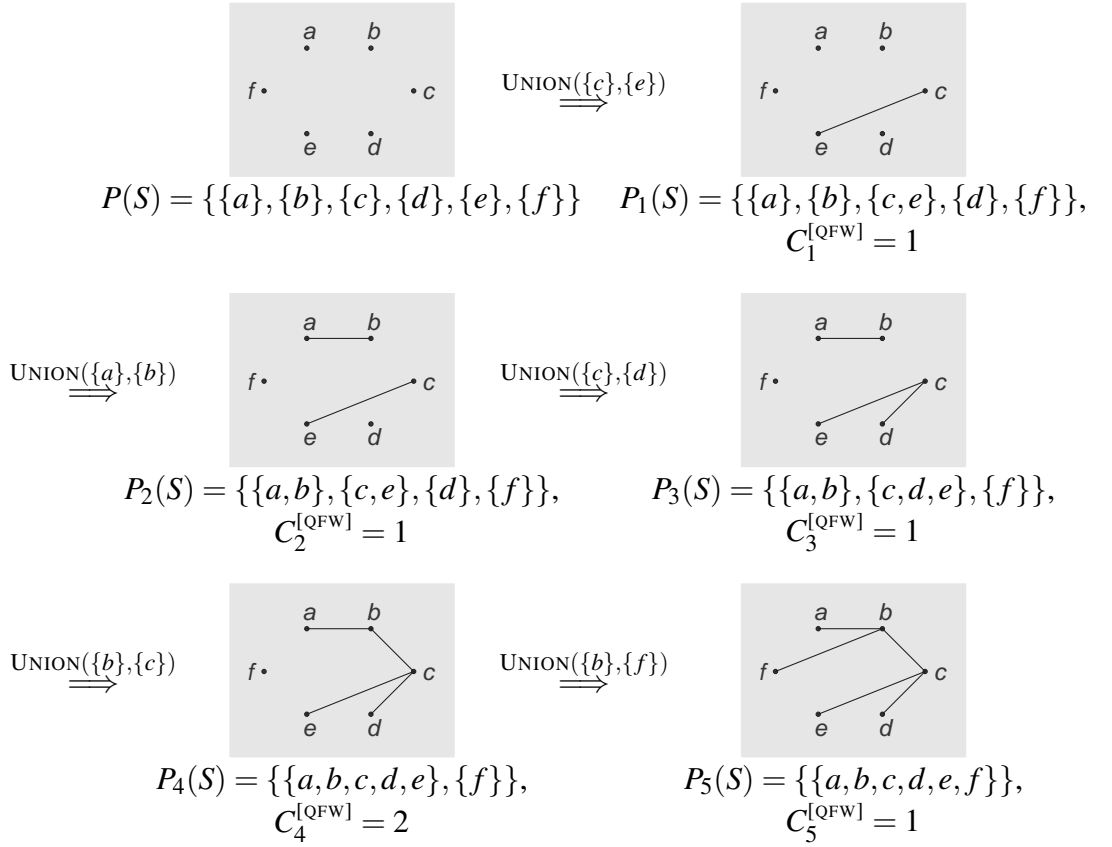
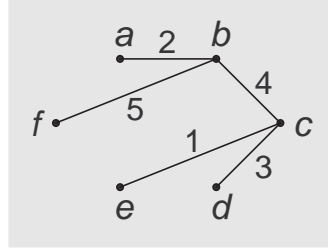


Figure 1: Choosing the particular spanning tree given in the example the QFW algorithm has total cost $X^{[QFW]} = \sum_{i=1}^5 C_i^{[QFW]} = 6$ to merge the elements $S = \{a, b, \dots, f\}$ starting with the partition $P(S) = \{\{a\}, \{b\}, \dots, \{f\}\}$. Here $C_i^{[QFW]}$ denotes the cost of the i -th merging step of the QFW algorithm.

Under the random spanning tree model the algorithm QFW has been analyzed first by [25] and [12]. Knuth and Schnhage [12] obtained the following asymptotic result for the expected total cost:

$$\mathbb{E}(X_n^{[\text{QFW}]}) = \frac{1}{\pi}n \log n + O(n).$$

Only recently in a collaboration with Markus Kuba [14] further progress in the analysis of $X_n^{[\text{QFW}]}$ has been made. First we showed a concentration result, namely that, after normalization, $X_n^{[\text{QFW}]}$ converges in the \mathcal{L}_2 metric to $\frac{1}{\pi}$:

$$\frac{X_n^{[\text{QFW}]} - \frac{1}{\pi}}{n \log n} \xrightarrow{\mathcal{L}_2} \frac{1}{\pi}.$$

However, the main contribution was a full characterization of the limiting distribution of $X_n^{[\text{QFW}]}$ by its sequence of positive integer moments. We remark that similar results have been obtained earlier by Hwang and Neininger [9] when characterizing the limiting distribution of the number of comparisons used in the Quicksort algorithm to sort a random permutation of length n .

Theorem 1 *Let $X_n^{[\text{QFW}]}$ denote the total cost of the algorithm “Quick Find Weighted” QFW to merge all elements of a finite set S of size n under the random spanning tree model. Then the expected value of $X_n^{[\text{QFW}]}$ has, for $n \rightarrow \infty$, the following asymptotic expansion:*

$$\mathbb{E}(X_n^{[\text{QFW}]}) = \frac{1}{\pi}n \log n + Cn + O(n^{\frac{3}{4}}),$$

with a certain constant $C \approx 0.6315$, which is given as follows:

$$C = \frac{\gamma + 2 \log 2}{\pi} + \sum_{n \geq 0} \frac{1}{n+1} \left[e^{-(n+1)} \left(R_{n+2} - R_{n+1} - \sum_{k=0}^n \frac{(k+1)^{k+1}}{(k+2)!} R_{n-k} \right) - \frac{1}{\pi} \right],$$

with

$$R_n = \sum_{k=1}^{n-1} \frac{k^k (n-k)^{n-k-1}}{k!(n-k)!} \min(k, n-k).$$

The suitably centered and normalized r.v. $X_n^{[\text{QFW}]}$ converges in distribution to a r.v. X , which can be characterized by its r -th integer moments:

$$\frac{X_n^{[\text{QFW}]} - \frac{1}{\pi}n \log n - Cn}{n} \xrightarrow{(d)} X, \quad \text{with} \quad \mathbb{E}(X^r) = m_r,$$

where m_r is given recursively as follows:

$$m_r = \frac{\Gamma(r-1)}{2\sqrt{\pi}\Gamma(r-\frac{1}{2})} \sum_{\substack{r_1+r_2+r_3=r, \\ r_2, r_3 < r}} \binom{r}{r_1, r_2, r_3} m_{r_2} m_{r_3} I_{r_1, r_2, r_3}, \quad \text{for } r \geq 2,$$

with initial values $m_0 = 1$ and $m_1 = 0$ and

$$I_{r_1, r_2, r_3} = \int_{[0,1]} \left(\frac{1}{\pi} (x \log x + (1-x) \log(1-x)) + \min(x, 1-x) \right)^{r_1} x^{r_2 - \frac{1}{2}} (1-x)^{r_3 - \frac{3}{2}} dx.$$

To show our results we used two main ingredients, namely, a suitable distributional recurrence for the random variable $X_n^{[\text{QFW}]}$ together with explicit solutions of the corresponding recurrences for the r -th integer moments in terms of lower order moments. To obtain the distributional recurrence for $X_n^{[\text{QFW}]}$, i.e., the starting point of our analysis, we consider the ‘‘inverse process’’: instead of merging equivalence classes by carrying out UNION operations and thus adding successively edges until one obtains a spanning tree, we start with a random spanning tree and remove successively edges until all nodes are isolated. The basis of the approach is the following simple fact. Let us assume we start with a random unrooted labelled tree of size n (this corresponds to the random spanning tree of the complete graph of a set S of size n) and remove one edge at random (this corresponds to the edge, which has been added in the final, i.e., the $(n-1)$ -st, merging step). Then it holds that both resulting subtrees, let us assume they are of sizes k and $n-k$, with $1 \leq k \leq n-1$, are itself *random* unrooted labelled trees of smaller sizes k and $n-k$, respectively. In the QFW algorithm the cost of this edge-removal step is given by $\min(k, n-k)$. This leads to the following distributional recurrence for the total cost $X_n := X_n^{[\text{QFW}]}$ of the algorithm QFW under the random spanning tree model, when merging the elements of a set of size $n \geq 2$ (with $X_1 = 0$):

$$X_n \stackrel{(d)}{=} X_{S_n} + X_{n-S_n}^* + t_{n, S_n}, \quad \text{for } n \geq 2, \quad (1)$$

where S_n is independent of $(X_j)_{j \geq 1}$ and $(X_j^*)_{j \geq 1}$, which are independent copies of each other. The toll function $t_{n,k}$ is for QFW given by

$$t_{n,k} := \min(k, n-k).$$

Furthermore, S_n is distributed as follows:

$$\mathbb{P}\{S_n = k\} = \binom{n}{k} \frac{k^k (n-k)^{n-k-1}}{(n-1)n^{n-1}}, \quad \text{for } 1 \leq k \leq n-1.$$

To treat the recurrences appearing for the r -th integer moments, which can be deduced easily from (1), we used a generating functions approach leading to explicitly solvable differential equations.

We remark that there is an interesting connection between merging models for Union-Find algorithms introduced in computer science and certain coagulation models for particles introduced in statistical physics. In particular, it is known [23]

that the random spanning tree model corresponds to the so-called additive Marcus-Lushnikov process; here the probability that in a merging step two particles of sizes x and y , respectively, will merge is proportional to the sum $x + y$ of their sizes. Of course, the approach presented for analyzing the cost of Union-Find algorithms could be applied also to analyze certain parameters for this coagulation model.

2 Growth models for random increasing trees

A study of random trees turns out to be of interest in various scientific branches as computer science, probability theory and combinatorics. Although trees can be considered as particular graphs the techniques introduced and applied to study important quantities for random trees are somewhat different from standard methods used in the study of random graphs; however, recently methods from analytic combinatorics (which are of great importance in the study of random trees) have been applied with success also to certain random graph models. A good source for methods and problems in connection with random trees is the recent book of Drmota [5], but also the general treatment on analytic combinatorics by Flajolet and Sedgewick [6].

We will consider here a particular class of tree models called “increasing trees”, which has been introduced independently in a combinatorial and a probabilistic context. The interest in these models comes from the fact that they are appropriate to describe the behaviour of a lot of quantities in various applications (see [20] for a survey). E.g., they are used as a model for the spread of epidemics, for pyramid schemes, for the family trees of preserved copies of ancient texts, and as a simplified growth model of the world wide web (there are relations to the so-called Barabási-Albert model for scale-free networks, see [3]).

Combinatorially increasing trees can be described as rooted ordered trees (the left-to-right order of the subtrees of a node is important), where the nodes are labelled by distinct integers of $\{1, 2, \dots, n\}$ (with n the size of the tree), in such a way that the label of a child node is always larger than the label of its parent node. Actually one considers weighted trees, where each node v in the tree gets a weight $\varphi_r > 0$ depending on the out-degree (i.e., the number of children) $d^+(v) = r$ of v ; the weight of a tree T is simply the product of the weights of all nodes $v \in T$. Given a degree-weight sequence $(\varphi_r)_{r \geq 0}$ this also leads to a natural definition of random increasing trees, where one simply assumes that each increasing tree of size n appears with a probability proportional to its weight. One can describe a combinatorial class \mathcal{T} of increasing trees also via a formal recursive equation (stated here somewhat informal avoiding the rigorous combinatorial constructions hidden behind, as the so-called partition product and the boxed product for labelled

combinatorial objects, see, e.g., [6]):

$$\mathcal{T} = \varphi_0 \cdot \textcircled{1} \dot{\cup} \varphi_1 \cdot \begin{array}{c} \textcircled{1} \\ | \\ \mathcal{T} \end{array} \dot{\cup} \varphi_2 \cdot \begin{array}{c} \textcircled{1} \\ / \quad \backslash \\ \mathcal{T} \quad \mathcal{T} \end{array} \dot{\cup} \varphi_3 \cdot \begin{array}{c} \textcircled{1} \\ | \quad / \quad \backslash \\ \mathcal{T} \quad \mathcal{T} \quad \mathcal{T} \end{array} \dot{\cup} \dots$$

This equation reflects the decomposition of a tree into the root node and its subtrees and thus often gives rise to a “top-down” approach for analyzing parameters in increasing tree families. The model of increasing trees has been introduced in its fully generality by Bergeron et al. [2], but important particular instances have been considered already earlier by Prodinger and Urbanek [24].

For applications those increasing tree models are of particular interest, which also allow a probabilistic description via a tree evolution process, i.e., where it holds that for every tree T' of size n with vertices v_1, \dots, v_n there exist probabilities $p_{T'}(v_1), \dots, p_{T'}(v_n)$, such that when starting with a *random tree* T' of size n of the tree family considered, choosing a vertex v_i in T' according to the probabilities $p_{T'}(v_i)$ and attaching node $n+1$ to it, we obtain again a *random tree* T of size $n+1$ of the tree family considered. There are several prominent instances of increasing tree models as “recursive tree” ($\varphi_r = \frac{1}{r!}$), “binary increasing trees” ($\varphi_r = \binom{2}{r}$) and “plane recursive trees” ($\varphi_r = 1$), which have been introduced in a probabilistic context via their simple tree evolution processes.

In a joint work with Helmut Prodinger [21] we fully answered the question, which increasing tree models also allow a description via tree evolution processes; it turns out that this is possible only for few instances, but for all these models the “insertion probabilities” $p_{T'}(v_i)$ are quite simple to describe, since they only depend on the size of the tree T' and on the out-degree of the node v_i . In the theorem stated below we could show that there are only three types of tree evolution models captured by increasing trees, but they are of particular importance: the probability that a new node is attached to an already existing one is (i) the same for all nodes (uniform attachment), (ii) proportional to a linear function of the out-degree of the node (preferential attachment, “success breeds success”), (iii) proportional to the difference between a maximal possible number d of children and the out-degree, i.e., the actual number of children (saturation model).

Theorem 2 *A family of increasing trees \mathcal{T} can be constructed via a tree evolution process if and only if there exist positive constants $a, b > 0$, such that the degree-weight generating function $\tilde{\varphi}(t) = \sum_{r \geq 0} \tilde{\varphi}_r t^r$ satisfies $\tilde{\varphi}(t) = a\varphi(bt)$, where $\varphi(t)$ is given by one of the following three formulae:*

- * Case A (recursive trees): $\varphi(t) = e^t$,
- * Case B (d -ary trees): $\varphi(t) = (1+t)^d$, for $d \in \{2, 3, 4, \dots\}$,

* *Case C (generalized plane recursive trees):* $\varphi(t) = \frac{1}{(1-t)^\alpha}$, for $\alpha > 0$.

The corresponding tree evolution processes, which generate random trees of arbitrary size n , can be described as follows:

- * *Step 1:* The process starts with the root labelled by 1.
- * *Step $i + 1$:* At step $i + 1$ the node with label $i + 1$ is attached to any previous node v (with out-degree $d^+(v)$) of the already grown tree of size i with probabilities $p(v)$ given as follows:

$$p(v) = \begin{cases} \frac{1}{i}, & \text{for Case A (uniform attachment model),} \\ \frac{d - d^+(v)}{(d-1)i + 1}, & \text{for Case B (saturation model),} \\ \frac{d^+(v) + \alpha}{(\alpha + 1)i - 1}, & \text{for Case C (preferential attachment model).} \end{cases}$$

The tree evolution process is illustrated in Figure 2 for random recursive trees, which are generated by a “uniform attachment model”, i.e., in the $(i + 1)$ -st step a random node in the tree already generated is chosen to attach the new node $i + 1$. Therefore, the increasing tree families characterized here have the advantage of leading to a combinatorial, but also to a probabilistic description. The description of the tree models by means of a tree evolution process often allows a “bottom-up” approach for analyzing tree parameters.

In a series of joint papers with Markus Kuba [13, 15, 16] we studied the distributional behaviour of important quantities in random increasing tree families, where the main focus has been given to a precise analysis of the behaviour of so-called label-dependent parameters such as the out-degree (i.e., the number of children) of the node labelled j , the number of descendants of the node labelled j , or the distance between the nodes labelled j_1 and j_2 , respectively, in a random tree of size n . The exact and asymptotic results lead to a precise description of the behaviour of “the j -th individual” during the tree evolution process according to the growth of $j = j(n)$ as $n \rightarrow \infty$. Figure 3 shall illustrate such important label-dependent parameters.

We state here just one particular result concerning the node-to-node distance (a fundamental quantity when analyzing network models) for the family of “plane recursive trees”.

Theorem 3 *Let $\Delta_{n;j_1,j_2}$ count the distance between the nodes with label j_1 and label j_2 in a random plane recursive tree of size n . Then it holds that the random variable*

$$\Delta_{n;j_1,j_2}^* := \frac{\Delta_{n;j_1,j_2} - \mu_{n;j_1,j_2}}{\sigma_{n;j_1,j_2}},$$

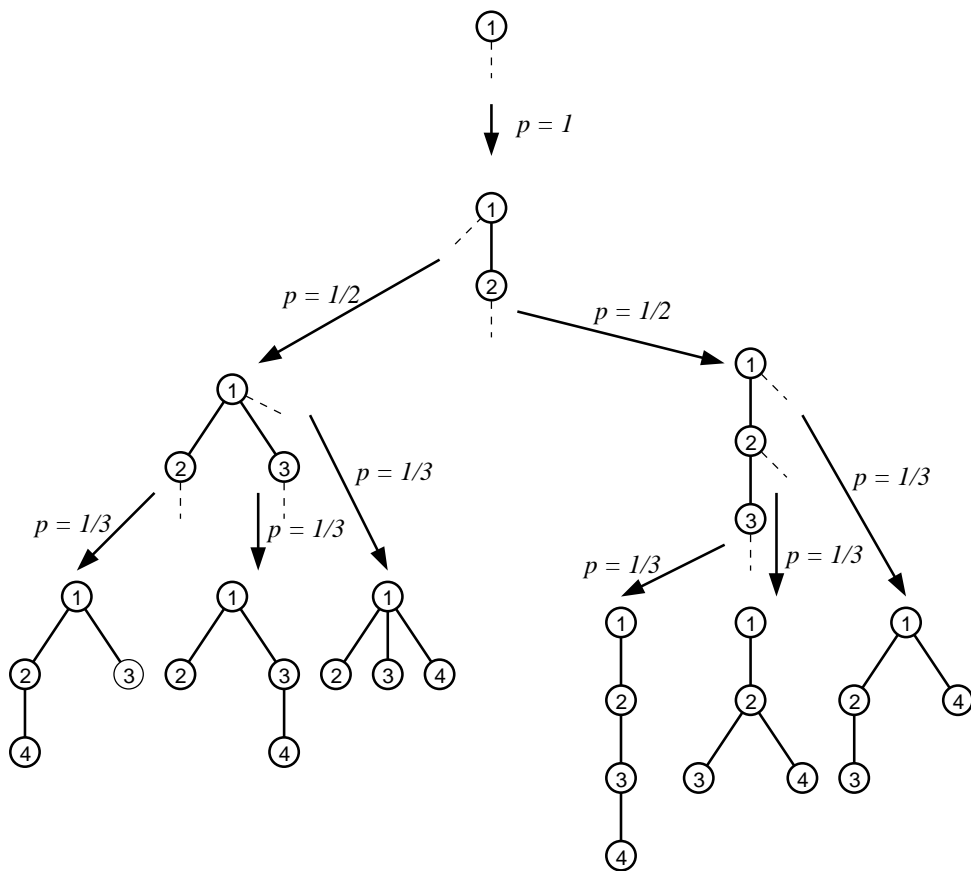


Figure 2: Generating random recursive trees via a tree evolution process.

with $\mu_{n;j_1,j_2} := \frac{1}{2}(\log j_1 + \log j_2)$ and $\sigma_{n;j_1,j_2}^2 := \frac{1}{2}(\log j_1 + \log j_2)$, is, for arbitrary sequences $(n, j_1(n), j_2(n))_{n \in \mathbb{N}}$, with $1 \leq j_1 = j_1(n), j_2 = j_2(n) \leq n$ and $j_1 \neq j_2$, provided that $\max(j_1, j_2) \rightarrow \infty$, asymptotically for $n \rightarrow \infty$ Gaussian distributed:

$$\Delta_{n;j_1,j_2}^* = \frac{\Delta_{n;j_1,j_2} - \mu_{n;j_1,j_2}}{\sigma_{n;j_1,j_2}} \xrightarrow{(d)} \mathcal{N}(0, 1).$$

We remark that interesting generalizations of the model of increasing trees have been introduced also. In one such generalization called “bucket increasing trees” introduced in a joint work with Markus Kuba [18] the nodes of a tree are buckets, which can contain up to a fixed integer amount of $b \geq 1$ elements (= labels). The bucketing effect is then related to the aging and fertility restrictions of generalized preferential attachment rules introduced in [4]. Another direction, which is pursued in a joint study with Georg Seitz [22], concerns the introduction and analysis

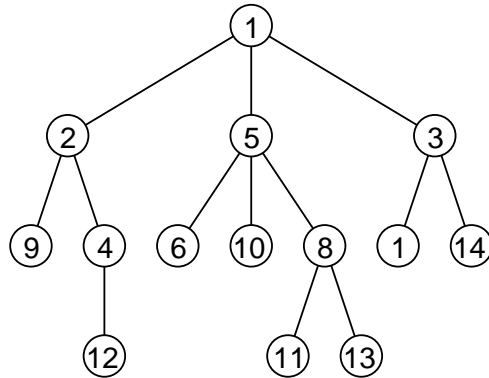


Figure 3: An increasing tree of size 14. The depth (i.e., the root-to-node distance) of node 5 is one, the distance between nodes 4 and 5 is three, the number of descendants of node 5 (including the node itself) is six and the out-degree of node 5 is three.

of evolution models for “ k -dimensional trees”, which are particular graph models that allow a “tree-like” combinatorial description. Only recently such models have been introduced as scale-free network models [7].

3 Analysis of diminishing urn models

Pólya-Eggenberger urn models are simple, useful mathematical tools for describing many evolutionary processes in diverse fields of application such as analysis of algorithms and data structures, statistics and genetics. Due to their importance in applications, there is a huge literature on the stochastic behaviour of urn models; see for example [10, 19].

In the simplest case of two types of colours for the balls Pólya-Eggenberger urn models can be described as follows. At the beginning, the urn contains m black and n white balls. At every step, we choose a ball at random from the urn, examine its colour and put it back into the urn and then add/remove balls according to its colour by the following rules. If the ball is white, then we put a white and b black balls into the urn, while if the ball is black, then c white balls and d black balls are put into the urn. The values $a, b, c, d \in \mathbb{Z}$ are fixed integer values and the urn model is specified by the transition matrix $M = \begin{pmatrix} a & b \\ c & d \end{pmatrix}$. Urn models with $r (\geq 2)$ types of colours can be described in an analogous way and are specified by an $r \times r$ transition matrix.

Most studies of urn models impose the so-called *tenability* condition on the transition matrix, so that the process can be continued *ad infinitum* (or no balls of a given colour being completely removed). However, in some applications (examples given below), there appear urn models with a very different nature, which we will

refer to as “diminishing urn models.” For simplicity of presentation, we describe them in the case of balls with two types of colours, black and white. We consider Pólya-Eggenberger urn models specified by a transition matrix $M = \begin{pmatrix} a & b \\ c & d \end{pmatrix}$, and in addition there is a set of absorbing states $\mathcal{S} \subseteq \mathbb{N} \times \mathbb{N}$. The urn contains m black balls and n white balls at the beginning and evolves by successive draws at discrete instances according to the transition matrix until an absorbing state $s = (j, k) \in \mathcal{S}$ is reached, i.e., when the urn contains exactly j black balls and k white balls. Then the urn process stops.

In contrast to “ordinary” urn models, where one is mainly interested in the exact (or limiting) distribution of the colours of the balls in the urn after a fixed amount of draws (or as the number of draws tends to infinity), for diminishing urns, the main question is different, namely: starting at state (m, n) , what is the probability of reaching the absorbing state $(j, k) \in \mathcal{S}$?, or (depending on the problem) what is the number of balls left in the urn when the process stops?.

We give a few motivating examples of diminishing urn models, which appear in the literature (see [8] for references).

The OK Corral problem. This corresponds to the urn $M = \begin{pmatrix} 0 & -1 \\ -1 & 0 \end{pmatrix}$ with two absorbing axes: $\mathcal{S} = \{(0, n) : n \geq 0\} \cup \{(m, 0) : m \geq 0\}$. An interpretation is as follows. Two groups of gunmen, group A and group B (with n and m gunmen, respectively), face each other. At every discrete time step, one gunman is chosen uniformly at random who then shoots and kills exactly one gunman of the other group. The bloody gunfight ends when one group gets completely “eliminated”. Two questions are of interest: (i) what is the probability that group A (group B) survives? and (ii) what is the probability that the gunfight ends with k survivors of group A (group B)?

This problem was introduced by Williams and McIlroy and studied recently by several authors (Kingman and Volkov; Flajolet, Dumas and Puyhaubert) using different approaches, leading to very interesting results. Also the urn corresponding to the OK corral problem can be viewed as a basic model in the mathematical theory of warfare and conflicts.

The cannibal urn. Introduced by Greene and analyzed in details by Pittel, this urn model is a slight modification of the diminishing urn with $M = \begin{pmatrix} 0 & -1 \\ 1 & -2 \end{pmatrix}$ and the vertical wall of absorbing states $\mathcal{S} = \{(0, n) : n \geq 0\} \cup \{(1, n) : n \geq 0\}$. In terms of weighted lattice paths, one starts at position (m, n) , the weight (and thus the probability) of a step to $(m - 1, n)$ is $\frac{n}{m-1+n}$ (not $\frac{n}{m+n}$), and the weight to $(m - 2, n + 1)$ is $\frac{m-1}{m-1+n}$.

Such an urn was introduced to model the behaviour of cannibals in biological populations. It can be described as follows. A population consists of cannibals and non-cannibals. At every time step, a non-cannibal is selected as victim and

removed; after that a member in the remaining population (cannibals and non-cannibals) is selected uniformly at random. If the selected individual is a cannibal it remains as a cannibal, but if the selected individual is a non-cannibal, it becomes then a cannibal. The question is, when starting with n cannibals and m non-cannibals, what is the number of resulting cannibals in the population at the moment when all non-cannibals are removed?

The pills problem. The transition matrix is given by $M = \begin{pmatrix} -1 & 0 \\ 1 & -1 \end{pmatrix}$ and the absorbing axis is $\mathcal{S} = \{(0, n) : n \geq 0\}$. An interpretation is as follows. An urn has two types of pills in it, which are single-unit and double-unit pills, respectively. At every step, we pick a pill uniformly at random. If a single-unit pill is chosen, then we eat it up, and if the pill is of double unit, we break it into two halves — one half is eaten up and the other half is now considered of single unit and thrown back into the urn. The question is then, when starting with n single-unit pills and m double-unit pills, what is the probability that k single-unit pills remain in the urn when all double-unit pills are drawn?

This problem has been stated by Knuth and McCarthy, where the authors asked for a formula for the expected number of remaining single-unit pills, when there are no double-unit pills in the urn. Interestingly the solution of the problem is given by a nice explicit formula.

Brennan and Prodinger proposed several generalizations of the problem. One natural generalization is to consider r types of pills, which are of i units, $i = 1, \dots, r$, respectively. At every time step, a pill is chosen uniformly at random; if the pill is of single unit, it is eaten up, and if the pill is of i units, $i \geq 2$, it is broken into two parts, one of single unit and the other of $(i - 1)$ units. The piece of single unit is eaten up and the remaining piece is thrown back into the urn. We stop if there are no more pills of the largest units r .

This problem corresponds to the diminishing urn model with the $r \times r$ transition matrix

$$M = \begin{pmatrix} -1 & & & & & \\ 1 & -1 & & & & \\ & & \ddots & \ddots & & \\ & & & & 1 & -1 \\ & & & & & 1 & -1 \end{pmatrix}$$

and the absorbing hyperplane $\mathcal{S} = \{(n_1, \dots, n_{r-1}, 0) : n_1, \dots, n_{r-1} \geq 0\}$. One might then be interested in finding the probability that k pills of single unit remain in the urn when there are no more pills of r units, the starting configuration being n_i pills of i units.

In the study of urn models it is helpful to describe the evolution of the urn by weighted lattice paths, which is described next in the case of urns with two types of balls. If the urn contains m black balls and n white balls and we select a white

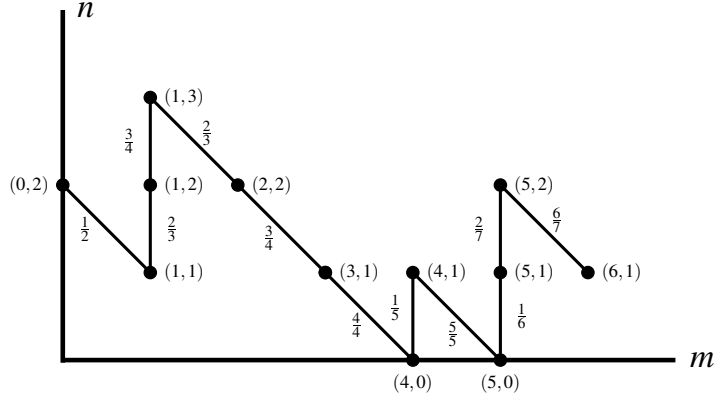


Figure 4: An example of a weighted path from $(6, 1)$ to the absorbing state $(0, 2)$ for the so-called pills problem with transition matrix $M = [-1, 0; 1, -1]$ and the vertical absorbing axis $\mathcal{S} = \{(0, n) : n \geq 0\}$. The illustrated path has weight $\frac{6}{7} \cdot \frac{2}{7} \cdot \frac{1}{6} \cdot \frac{5}{6} \cdot \frac{1}{5} \cdot \frac{4}{4} \cdot \frac{3}{4} \cdot \frac{2}{4} \cdot \frac{3}{4} \cdot \frac{2}{3} \cdot \frac{1}{2} = \frac{3}{3920}$.

ball (with probability $\frac{n}{m+n}$), then this corresponds to a step from (m, n) to $(m+a, n+b)$, to which the weight $\frac{n}{m+n}$ is associated; and if we select a black ball (with probability $\frac{m}{m+n}$), this corresponds to a step from (m, n) to $(m+c, n+d)$ (with weight $\frac{m}{m+n}$). The weight of a path after t successive draws consists of the product of the weights of every step. By this correspondence, the probability of starting at (m, n) and ending at (j, k) is equal to the sum of the weights of all possible paths starting at state (m, n) and ending at the absorbing state $(j, k) \in \mathcal{S}$ (which did not reach any absorbing state before). Unfortunately, the expressions so obtained for the probability are, although exact, less useful for large m or n . An example for the weighted path corresponding to the evolution of a diminishing urn is given in Figure 4.

As mentioned above, for diminishing urns one is interested in the position of the absorbing state. Probabilistically, we consider the pair of random variables $(X_{n,m}^{(1)}, X_{n,m}^{(2)})$, such that $\mathbb{P}\{(X_{n,m}^{(1)}, X_{n,m}^{(2)}) = (j, k)\}$ gives the probability that when starting at state (m, n) (with m black balls and n white balls), the urn process reaches the absorbing state (j, k) , namely, the process terminates with j black balls and k white balls. These probabilities can be encoded by the corresponding probability generating functions $h_{n,m}(v_1, v_2)$ defined via

$$h_{n,m}(v_1, v_2) := \sum_{j \geq 0} \sum_{k \geq 0} \mathbb{P}\{(X_{n,m}^{(1)}, X_{n,m}^{(2)}) = (j, k)\} v_1^j v_2^k.$$

According to the outcome of the first draw of the urn process, one obtains the

following recurrences for the probability generating functions:

$$h_{n,m}(v_1, v_2) = \frac{n}{m+n} h_{n+a, m+b}(v_1, v_2) + \frac{m}{m+n} h_{n+c, m+d}(v_1, v_2), \quad (2)$$

for $(m, n) \notin S$. The boundary values at the absorbing states $(m, n) \in S$ are given by $h_{n,m}(v_1, v_2) = v_1^m v_2^n$.

Together with Hsien-Kuei Hwang and Markus Kuba [8] we suggested a generating functions approach to treat the recurrences (2). For various diminishing urn models (containing, e.g., all the before mentioned urns) we could apply this approach with success, which leads to a study of first order linear partial differential equations for the corresponding generating functions (where difficulties as dealing with unknown boundary values will occur in some situations). As an example we state results leading to a precise description of the behaviour of the pills problem urn.

Theorem 4 *Starting with m double-unit pills and n single-unit pills, the probability generating function $h_{n,m}(v) := \sum_{k \geq 0} \mathbb{P}\{X_{n,m} = k\} v^k$ of the number $X_{n,m}$ of the remaining single-unit pills in the urn when all double-unit pills are already taken is given by*

$$h_{n,m}(v) = mv \int_0^1 (1 + (v-1)q)^n (1 - q - (v-1)q \log q)^{m-1} dq.$$

If $m \rightarrow \infty$, then the random variable $X_{n,m}$ converges, after suitable normalization, in distribution to an exponentially distributed random variable X with parameter $\lambda = 1$, namely

$$\frac{X_{n,m}}{\frac{n}{m} + \log m} \xrightarrow{(d)} X,$$

where X has density $f(x) = e^{-x}$ for $x \geq 0$.

If m is fixed and $n \rightarrow \infty$, then the random variable $X_{n,m}$ converges, after suitable normalization, in distribution to a Beta random variable B_m ; in symbol

$$\frac{X_{n,m}}{n} \xrightarrow{(d)} B_m \stackrel{(d)}{=} \text{Beta}(1, m),$$

where B_m has density $m(1-x)^{m-1}$, $0 \leq x \leq 1$.

Together with Markus Kuba we have pursued further problems in connection with urn models, e.g., describing the area under weighted lattice paths associated to triangular diminishing urns [17].

References

- [1] A. V. Aho, J. E. Hopcroft and J. D. Ullman, *The Design and Analysis of Computer Algorithms*, Addison-Wesley, Reading, 1974.
- [2] F. Bergeron, P. Flajolet and B. Salvy, Varieties of Increasing Trees, *Lecture Notes in Computer Science* 581, 24–48, 1992.
- [3] B. Bollobás and O. M. Riordan, Mathematical results on scale-free random graphs, in *Handbook of graphs and networks*, 1–34, Wiley-VCH, Weinheim, 2003.
- [4] C. Borgs, N. Berger, J. T. Chayes, R. D’Souza and R. D. Kleinberg, Degree distribution of competition-induced preferential attachment graphs, *Combinatorics, Probability and Computing* 14, 697–721, 2005.
- [5] M. Drmota, *Random Trees*, Springer, Wien, 2009.
- [6] P. Flajolet and R. Sedgewick, *Analytic Combinatorics*, Cambridge University Press, Cambridge, 2009.
- [7] Y. Gao, The degree distribution of random k -trees, *Theoretical Computer Science* 410, 688–695, 2009.
- [8] H.-K. Hwang, M. Kuba and A. Panholzer, Analysis of some exactly solvable diminishing urn models, in: “The 19th International Conference on Formal Power Series and Algebraic Combinatorics”, Nankai University, Tianjin, 2007.
- [9] H.-K. Hwang and R. Neininger, Phase change of limit laws in the quicksort recurrence under varying toll functions, *SIAM Journal on Computing* 31, 1687–1722, 2002.
- [10] N. L. Johnson and S. Kotz, *Urn models and their application. An approach to modern discrete probability theory*, John Wiley, New York, 1977.
- [11] D. E. Knuth, *The Art of Computer Programming*, Volume 1–3, Addison-Wesley, Reading, 1968 (Vol. 1), 1969 (Vol. 2), 1973 (Vol. 3).
- [12] D. E. Knuth and A. Schönhage, The expected linearity of a simple equivalence algorithm, *Theoretical Computer Science* 6, 281–315, 1978.
- [13] M. Kuba and A. Panholzer, Descendants in increasing trees, *Electronic Journal of Combinatorics* 13, research paper 8, 2006.
- [14] M. Kuba and A. Panholzer, Analysis of the total costs for variants of the Union-Find algorithm, *Discrete Mathematics and Theoretical Computer Science*, in: “2007 International Conference on the Analysis of Algorithms”, Proceedings AH, 259–268, 2007.
- [15] M. Kuba and A. Panholzer, On the degree distribution of the nodes in increasing trees, *Journal of Combinatorial Theory, Series A* 114, 597–618, 2007.
- [16] M. Kuba and A. Panholzer, On the distribution of distances between specified nodes in increasing trees, *Discrete Applied Mathematics* 158, 489–506, 2010.
- [17] M. Kuba and A. Panholzer, On the area under lattice paths associated with triangular diminishing urn models, *Advances in Applied Mathematics* 44, 329–358, 2010.

- [18] M. Kuba and A. Panholzer, A combinatorial approach to the analysis of bucket recursive trees, *Theoretical Computer Science*, to appear.
- [19] H. Mahmoud, *Pólya urn models*, CRC Press, Boca Raton, 2009.
- [20] H. Mahmoud and R. Smythe, A Survey of Recursive Trees, *Theoretical Probability and Mathematical Statistics* 51, 1–37, 1995.
- [21] A. Panholzer and H. Prodinger, Level of nodes in increasing trees revisited, *Random Structures & Algorithms* 31, 203–226, 2007.
- [22] A. Panholzer and G. Seitz, Ordered increasing k -trees: introduction and analysis of a preferential attachment network model, accepted for presentation and publication in the proceedings of “AofA’10”.
- [23] J. Pitman, Coalescent random forests, *Journal of Combinatorial Theory, Series A* 85, 165–193, 1999.
- [24] H. Prodinger and F. J. Urbanek, On monotone functions of tree structures, *Discrete Applied Mathematics* 5, 223–239, 1983.
- [25] A. C.-C. Yao, On the average behavior of set merging algorithms (Extended abstract), *Conference Record of the Eight Annual ACM Symposium on Theory of Computing*, 192–195, 1976.

Author’s address:

Alois Panholzer

Institut für Diskrete Mathematik und Geometrie, Technische Universität Wien.

Wiedner Hauptstr. 8–10/104, 1040 Wien.

e-mail alois.panholzer@tuwien.ac.at/<http://info.tuwien.ac.at/panholzer>