Alcom-FT Technical Report Series
ALCOMFT-TR-02-21

AVERAGE CASE ANALYSIS OF THREE COMBINATORIAL ALGORITHMS UNDER LIMITED RANDOMNESS

CYRIL BANDERIER, KURT MEHLHORN, AND RENE BEIER

ABSTRACT. We perform an average case case analysis of three classical algorithms under limited randomness: left-to-right maxima counting, quicksort and shortest paths. This kind of analysis have been introduced by Spielman & al. (under the name "smoothed analysis") for continuous problems. The idea is to average not over the complete space of instances but only over the instances which are obtained from a fixed input by a perturbation.

1. INTRODUCTION

Recently, D. Spielman & al. [12] introduced a new kind of analysis, called smoothed analysis, which is intermediate between average case analysis and worst case analysis. It consists in studying the performance of a given algorithm on a perturbation of the worst case instance(s). For worst case analysis, one is interested in the quantity $\max C(x)$, the highest cost of the algorithm, for inputs x in a given set of data. For average case analysis, one is interested in the quantity $\mathrm{E}[C(x)]$, the average cost of the algorithm over all the inputs x in a given set of data. For the so-called smoothed analysis, the quantity of interest is

$$\max_x \mathrm{E}_\epsilon[C(x+\epsilon)]\,,$$

when $x+\epsilon$ is an "ϵ-perturbation of x".

For most algorithms, there is a discrepancy between the worst case and the average case behavior. Both quantities convey very useful information. The main aim of smoothed analysis is to know if worst instances are rather isolated. For example, D. Spielman & al. [12] proved that this is the case for the simplex algorithm as they showed it has a polynomial "smoothed complexity".

In this paper, we introduce the concept of "discrete smoothed analysis" for algorithms running on discrete data structures. As there is no canonical definition for the notion of perturbation of the instances in this case, we consider here two natural (practically speaking) models

---

Date: April 24, 2002.

The three authors are supported by the Future and Emerging Technologies programme of the EU under contract number IST-1999-14186 (ALCOM-FT).

of limited randomness: the first one corresponds to a *partial permutation* (of the instances) and the second one corresponds to a *partial randomization* (of some bits of the instances).

Partial permutations.

The first probabilistic model (partial permutations) is parameterized by p ($0 \leq p \leq 1$) and is defined as follows: Consider the sequence $(1, \dots, n)$. Select each element (independently) with probability p. Let m be the number of selected elements (in average $m = pn$) then take one of the $m!$ permutations of $\mathfrak{S}_m$ (uniformly at random) and let it act on the selected elements. E.g., for $p = 1/2$ and $n = 7$, one can get $(1, \overline{2}, 3, \overline{4}, 5, 6, \overline{7})$. One has marked $m = 3$ elements (namely, 2, 4, and 7). Choose a random permutation from $\mathfrak{S}_3$, for example (312). One then gets the following sequence $(1, \overline{7}, 3, \overline{2}, 5, 6, \overline{4})$. One had a probability $p^3(1-p)^4/3!$ to get this sequence.

Partial bit randomization.

The second probabilistic model (partial randomization) is parameterized by an integer k ($k \geq 0$) and is designed for weighted inputs (for example, graphs with some integer weights on the edges or on the vertices). Here, one starts with the worst instance x, and the last k bits of all the weights are randomly modified.

In Section 2, we study left-to-right maxima under the partial permutations model. In Section 3, we study the complexity of quicksort for partial permutations. In Section 4, we study the shortest path in a weighted graph with a partial bit randomization.

2. Left-to-right maxima in partial permutations

2.1. **Partial Permutations.** What is the average number of left-to-right maxima in the sequence where only the selected elements have been permuted according to the randomly chosen permutation (while the nonselected elements stay at the same place)?

If $p = 0$, none of the elements is selected and there are n left-to-right maxima. If $p = 1$, all the elements are permuted and there is $H_n = 1 + 1/2 + \cdots + 1/n$ left-to-right maxima. In combinatorics, left-to-right maxima are often called "records"; a bijection due to Foata links them to the number of cycles in the permutation (for which a generating function approach is easy, leading to $(1-z)^{-u}$ and thus to the Stirling numbers).

Theorem 1. *The average number of left-to-right maxima in a partial permutation is*

$$\mathrm{LR}(n) = \log(pn) + \gamma + 2\,\frac{1-p}{p} + \left(\frac{1}{2} + \frac{2\,(1-p)}{p^2}\right)\frac{1}{n} + O\left(\frac{1}{n^2}\right),$$

where $\gamma \approx .5772$ is Euler's constant.

What should we expect? At both ends of the partial permutation, pieces of length $1/p$ (in average) are not selected and in the remaining part, one should see about $\log(pn)$ left-to-right maxima from the selected elements. Our first proof makes this argument rigorous. Our second proof, based on a generating function approach, gives the full asymptotics.

Proof via analytic approximations. Let X_i be the probability that the i-th position is not selected and is a maximum and let Y_i be the probability that the i-th position is selected and is a maximum.

Consider first a selected position i. A selected position contains a maximum iff it is a maximum among the selected elements. Assume that it is a maximum among the selected elements. Then its value is at least i and hence it is also a maximum when the elements not selected are considered. Thus $\sum_i Y_i$ is simply the number of maxima among the selected elements. The number of selected elements concentrates around pn and hence

$$\mathrm{E}[\sum_i Y_i] \approx \log(pn) \ .$$

Assume next that i is not selected. We start with the observation that X_i and X_{n+1-i} have the same distribution. We therefore concentrate on $i \leq n/2$.

If k_1 elements among the first $i-1$ and k_2 elements among the last $n-i$ are selected, the probability that i stays a maximum is

$$p(k_1, k_2) = \frac{k_1! \cdot k_2!}{(k_1+k_2)!} \ .$$

The expression for $p(k_1, k_2)$ is decreasing in both arguments. Namely,

$$\frac{p(k_1, k_2+1)}{p(k_1, k_2)} = \frac{k_1! \cdot (k_2+1)! \cdot (k_1+k_2)!}{(k_1+k_2+1)! \cdot k_1! \cdot k_2!} = \frac{k_1+1}{k_1+k_2+1} \leq 1 \ .$$

We want to compute $\mathrm{E}[\sum_{i \leq n/2} X_i]$. We split the sum into two parts: $i \leq c(1/p)\log n$ and $i \geq c(1/p)\log n$.

For the second part, we observe that we expect to select about $pi \geq c\log n$ elements less than i and about $p(n-i) \geq pn/2$ elements larger than i the probability that we select less than half the stated number in either part is less than $\exp(-dc\log n)$ for some constant d by Chernoff bounds. This is $O(n^{-2})$ for $c \geq 1/d$. If at least $c/2\log n$ elements smaller i are selected and at least $pn/4$ elements larger i are selected the probability that i is a maximum is less than $p(c/2\log n, pn/4) = O(n^{-2})$. Thus $\mathrm{prob}(X_i = 1) = O(n^{-2})$ for $i \geq c/p\log n$.

We turn to the i's with $i \leq c/p\log n$. If none of the first $i-1$ elements is selected i stays a maximum. If at least one for the first $i-1$ elements is chosen, the probability that i stays a maximum is at most $O(n^{-2}) + 4/pn$. The first term accounts for the fact that less $pn/4$ elements larger i are selected and the second term accounts for the fact that at least $pn/4$ elements larger i are selected and none of them is moved to a position before i. Thus

$$\mathrm{prob}(X_i = 1) \leq (1-p)((1-p)^{i-1} + O(n^{-2}) + 4/pn$$

and hence

$$\mathrm{E}[\sum_{i \leq c/p \log n} X_i] \leq (1-p)(\frac{1}{p} + O(\frac{\log n}{n^2}) + \frac{4 \log n}{p^2 n}) \ .$$

We conclude

$$\mathrm{E}[\sum_i (X_i + Y_i)] = \log(pn) + \frac{2(1-p)}{p} + o(1)$$

for constant p. In fact, constant p is not required. The argument works as long as $\log n/(p^2 n) = o(1)$, i.e., for $p \gg \sqrt{\log n/n}$.

□

Proof via generating functions. Note $\mathrm{LR_s}(n)$ (resp. $\mathrm{LR_{ns}}(n)$) the average number of left-to-right maxima arising from selected (resp. nonselected) elements. One now studies the asymptotic contribution of these two quantities to the sum $\mathrm{LR}(n) = \mathrm{LR_s}(n) + \mathrm{LR_{ns}}(n)$.

Contribution of selected left-to-right maxima:

$$\mathrm{LR_s}(n) = E[\sum_{i=1}^{n} Y_i] = \sum_{k=0}^{n} \binom{n}{k} p^k (1-p)^{n-k} H_k \,,$$

(with $H_0 = 0$). This formula is obtained by looking at all the configurations with k selected elements, each left-to-right maximum there is a left-to-right for the whole sequence, and each selected left-to-right maximum in the whole sequence is a left-to-right maximum in the subsequence made up of the selected elements, thus in average there is H_k such left-to-right maxima. Set $\alpha = \frac{p}{1-p}$ and $q := 1-p$, thus

$$\mathrm{LR_s}(n) = q^n \sum_{k=0}^{n} \binom{n}{k} \alpha^k H_k \,. \tag{1}$$

In this sum, one recognizes an Euler transform: the transformation of a sequence f_n into a sequence $\sum_{k=0}^{n} \binom{n}{k} f_k$. It is easy to check that if $F(z)$ is the generating function associated to f_n, then $\frac{F(\frac{z}{1-z})}{1-z}$ is the generating function associated to its Euler transform. In our case, the generating function of $\alpha^k H_k$ is $\frac{\log \frac{1}{1-\alpha z}}{1-\alpha z}$, and its Euler transform is

$$\sum_{n \geq 0} \left(\sum_{k=0}^{n} \binom{n}{k} \alpha^k \right) H_k \, z^n = \frac{\log \frac{1}{1-\alpha \frac{z}{1-z}}}{1-\alpha \frac{z}{1-z}} \frac{1}{1-z} = \frac{\log \frac{1-z}{1-(\alpha+1)z}}{1-(\alpha+1)z} \,. \tag{2}$$

Taking into account the multiplication by q^n in (1), one performs the substitution $z \mapsto qz$ in (2) and one gets

$$\sum_{n\geq 1} \mathrm{LR_s}(n) z^n = \frac{\log(\frac{1-qz}{1-z})}{1-z}\,. \tag{3}$$

The radius of convergence is $1 < 1/q$ and analysis of singularity gives $\mathrm{LR_s}(n) = H_n + \log(p) + o(C^n)$ (for $C<1$), then, using $H_n = \log(n) + \gamma + \frac{1}{2n} - \frac{1}{12n^2} + O(\frac{1}{n^3})$, one has

$$\mathrm{LR_s}(n) = \log(pn) + \gamma + \frac{1}{2n} + O(\frac{1}{n^2})\,.$$

Contribution of nonselected left-to-right maxima: The probability to have a non selected maximum at position i is given by

$$\mathrm{prob}(X_i = 1) = \sum_{k_1=0}^{i-1}\sum_{k_2=0}^{n-i} \binom{i-1}{k_1}\binom{n-i}{k_2} p^{k_1+k_2} q^{n-k_1-k_2} \frac{k_1!k_2!}{(k_1+k_2)!}\,,$$

where $q := 1-p$. This formula is obtained by considering the k_1 selected elements before i (as i is non selected, the value i is at position i and all the k_1 selected elements are $\leq i-1$ which is a left-to-right maximum) and k_2 selected elements after i ($\frac{1}{(k_1+k_2)!}$ is the "weight" of the chosen permutation). A nice (conjectural) observation is that $\mathrm{prob}(X_i \neq 1)$ is unimodular (for $i = 1, \ldots, n$) but we don't need this fact. One wants to find the asymptotics of

$$\begin{aligned}\mathrm{LR_{ns}}(n) &= \sum_{i=1}^{n} \mathrm{prob}(X_i = 1)\\ &= \sum_{i=1}^{n}\sum_{k_1=0}^{i-1}\sum_{k_2=0}^{n-i} \binom{i-1}{k_1}\binom{n-i}{k_2} p^{k_1+k_2} q^{n-k_1-k_2} \frac{k_1!k_2!}{(k_1+k_2)!}\,.\end{aligned}$$

In order to get the generating function associated to this triple sum, one can use some basic transformations, like the (inverse) Borel transform (multiplication or division of the sequence by $n!$), the Euler transform, the Hadamard product... All these transformations are closed in the space of D-finite functions (functions which satisfy a linear differential equations with polynomial coefficients), and are easily performed with a computer algebra package such as Gfun [11]. For example, the generating function (3) is given by $\mathcal{B}^{-1}\left(\exp(z)\ \mathcal{B}\left(\frac{\log(\frac{1}{1-\alpha z})}{1-\alpha z}\right)\right)$ where $\mathcal{B}$ stands for the Borel transform and $\mathcal{B}^{-1}$ for the inverse Borel transform. Similar manipulations for the $LR_{ns}(n)$'s lead to the generating function

$$\sum_{n\geq 1} \mathrm{LR_{ns}}(n) z^n = \frac{-q\log(1-z) + q\log(1-qz)}{2(1-z(1-\frac{p}{2}))^2} + \frac{q}{(1-z)(1-z(1-\frac{p}{2}))}\,.$$

As $1 < 1/(1-p/2) < 1/q$, the radius of convergence is 1 for the first log and the last summand, and $\frac{1}{1-p/2}$ for the second log, so singularity analysis gives (with $r := \frac{1-p}{2}$):

$$\mathrm{LR}_{\mathrm{ns}}(n) = r\big(\frac{4}{p^2 n}+O(\frac{1}{n^2})\big)+r\big(n\log(\frac{2}{p}-1)r^n+O(r^n)\big)+\frac{2q}{p}+\frac{q}{r}(\frac{p}{p-2})^n$$

Summing the contribution of $\mathrm{LR}_{\mathrm{ns}}(n)$ and $\mathrm{LR}_{\mathrm{s}}(n)$ gives the theorem. □

Theorem 1 "shows" that the contribution of the subsequent summands begins to be as important as the log part when $p \sim \frac{1}{\sqrt{n \log n}}$.

3. Quicksort

We now switch to an analysis of quicksort on partial permutations. One assumes that quicksort takes the first element of the list as the pivot and splits the input list with respect to the pivot into two parts: the elements smaller than the pivot and the elements larger than the pivot.

Theorem 2. *The expected running time of quicksort on a partial permutation of n elements is*

$$\begin{aligned} T_n &= an\log n + \big(\frac{a(\gamma-1)}{p}+b\big)\,n + c\log n \\ &+ \big(b+\frac{a}{2}+(a+c)\gamma+d\big)+\big(\frac{5a}{12}+\frac{c}{2}+e\big)\frac{1}{n}+o(\frac{1}{n}), \end{aligned}$$

where $\gamma \approx .5772$ is Euler's constant and where $a = 2/p$, $c = 2q/p^2$, $d = (pqb+3-p)/p^2$, where b and e are two constants (see details in the proof).

Again we give two proofs. The second proof gives the full asymptotics and the first proof gives only the leading term.

Proof. A first problem is that the content of the first position, even if it is selected, is *not* a random element of the sequence. It is more likely to be the smallest element than any other element. The other elements are equally likely. This unbalance results from the fact that if only one element is selected, the permutation of the selected elements has very little freedom.

Let p_1 be the probability that the first position is selected and filled by the smallest element by the permutation of the selected elements.

We have

$$
\begin{aligned}
p_1 &= p \cdot \sum_{0 \le k \le n-1} \binom{n-1}{k} p^k (1-p)^{n-1-k} \frac{1}{k+1} \\
&= \frac{1}{n} \cdot \sum_{0 \le k \le n-1} \binom{n}{k+1} p^{k+1} (1-p)^{n-1-k} \\
&= \frac{1}{n} \cdot \sum_{1 \le k \le n} \binom{n}{k} p^k (1-p)^{n-k} \\
&= \frac{1}{n} \cdot (1 - (1-p)^n)
\end{aligned}
$$

The final formula appeals for an application which, to any partial permutation with exactly k selected elements (including the first position, which is selected and filled by the smallest element), associates $n-1$ partial permutations with k selected elements (but whose first position which is either not selected, either not filled by the smallest element).

Similarly, we can compute p_i, the probability that the first element is selected and filled with the i-th smallest element, $i \ge 2$. By symmetry $p_2 = p_3 = \ldots = p_i = \ldots = p_n$ and hence $p_i = (p - p_1)/(n-1)$. We can also compute p_i directly as

$$
\begin{aligned}
\sum_{2 \le i \le n} p_i &= p \cdot \sum_{0 \le k \le n-1} \binom{n-1}{k} p^k (1-p)^{n-1-k} \frac{k}{k+1} \\
&= p - \frac{1}{n} \cdot (1 - (1-p)^n)
\end{aligned}
$$

We now try an argument similar to the one used in the book by Motwani and Raghavan [9]. Consider a fixed element i and a call in which it participates. Call the call *good* if the subproblem containing the element has less than $3/4$ its original size. If the element is chosen as the pivot the call is also good. We ask ourselves how many calls are needed until the element is contained in a subproblem of size d/p for some constant d still to be chosen. Clearly the number of good calls is logarithmically bounded. More precisely, it is bounded by $\log_{4/3} n$.

What is the probability that a call is good? A call is good is the pivot element is among the elements with rank $n/4$ to $3n/4$ and hence a call is good with probability

$$
p_{n/4} + \ldots + p_{3n/4} \ge \frac{n}{2} \cdot p_2 \ge \frac{n}{2} \cdot \frac{p - 1/n}{n-1} \ge \frac{p - p/d}{2} .
$$

We need $\log_{4/3} n$ good calls and the probability of a call being good is $\pi := \frac{p-p/d}{2}$. Thus the expected number of calls required is $\frac{1}{\pi} \log_{4/3} n$ and hence running time is bounded by $\frac{1}{\pi} n \log_{4/3} n + dn/p$ where the second term accounts for the total cost of the small calls. $\square$

We come to the second proof.

Proof. Let T_n be the expected running time of quicksort on a partial permutation of n elements. Then

$$T_n = (n-1) + (1-p)T_{n-1} + p_1(n)T_{n-1} + \sum_{2\le i\le n} p_i(n)(T_{i-1} + T_{n-i})$$

since the sublists resulting from splitting are again partial permutations (as each element is selected independently and all the permutations of the remaining selected elements are equally likely). We simplify and obtain

$$T_n = (n-1) + (1 - p + p_1(n) - p_2(n))T_{n-1} + 2p_2(n) \sum_{1\le i\le n-1} T_i \ . \tag{4}$$

Here is used the fact that the $p_i(n)$ are identical, and this last sum is actually $(T_1+T_{n-2})+\ldots+(T_{n-1}+T_0)$. Making explicit the recurrence, one has:

$$T_n = (n-1) + (q + \frac{q-q^n}{n-1})\, T_{n-1} + 2\frac{p - \frac{1-q^n}{n}}{n-1} \sum_{1\le i\le n-1} T_i \ . \tag{5}$$

When $p=1$, one recognizes the classical recurrence for quicksort (under the uniform randomness model).

Consider now the following generating function (with $T_0 = 0$):

$$T(z) = \sum_{n\ge 0} T_n z^n \ .$$

Multiplying (5) by $(n-1)nz^n$ and summing for $n \ge 1$, one has

$$\begin{aligned}\sum_{n\ge1}(n-1)nT_n z^n &= \sum_{n\ge1} n(n-1)^2 z^n + \sum_{n\ge1} n(nq-q^n)\, T_{n-1} z^n \\ &+ 2\sum_{n\ge1}(np-1+q^n) \sum_{1\le i\le n-1} T_i z^n \ .\end{aligned}$$

This gives

$$\begin{aligned} z^2T''(z) &= \frac{2z^2(2z+1)}{(1-z)^4} + qz\big(z^2T''(z) + 3zT'(z) + T(z)\big) - qz\big(qzT'(qz) + T(qz)\big) \\ &+ 2pz\big(z\frac{T'(z)}{1-z} + z\frac{T(z)}{(1-z)^2} + \frac{T(z)}{1-z}\big) - 2z\frac{T(z)}{1-z} + 2qz\frac{T(qz)}{1-qz} \ .\end{aligned}$$

This leads to the following difference-differential equation

$$\begin{aligned}&(z^2 - qz^3)T''(z) + \frac{z^2(3qz-q-2)}{1-z}T'(z) + q^2z^2T'(qz) \\ &\qquad -\frac{z(qz^2+2pz-q)}{(1-z)^2}T(z) - qz\frac{1+qz}{1-qz}T(qz) \quad = \quad \frac{2z^2(2z+1)}{(1-z)^4} \ .\end{aligned}$$

The asymptotic behavior of the coefficients T_n of $T(z)$ depends on the behavior of $T(z)$ around its dominating singularity. The complexity of Quicksort being between $n\log n$ and n^2, the dominating singularity is $z = 1$ (the other singularities would imply asymptotics of the kind ρ^{-n}, with $|\rho| \neq 1$). It implies that $z = q$ is not a singularity. One sets $\tau_1 := T'(q)$ and $\tau_2 := T(q)$. Happily enough, the two terms which involve τ_1 and τ_2 are also regular in $z = 1$ so they won't contribute to the leading terms of the asymptotics. Considering the above functional equation for $z \sim 1$ gives a classical differential equation:

$$\begin{aligned} & z(1-qz)T''(z) + \frac{z(3qz-q-2)}{1-z}T'(z) - \frac{qz^2+2pz-q}{(1-z)^2}T(z) \\ = \; & \frac{2z(2z+1)}{(1-z)^4} - q^2 z\tau_1 + q\frac{1+qz}{1-qz}\tau_2 \; . \end{aligned}$$

Injecting the behavior $(\rho - z)^\alpha$ in the homogeneous differential equation associated to the above equation gives the indicial equation (see [7] and its section 9.6 about Frobenius method for further explanations): $-6(q-1)^3(\alpha-1)(\alpha+2)^2 = 0$. Thus $\alpha = -2$ is a characteristic exponent, but this root is not simple and as there are roots which differ by an integer, one has to look for solutions of the kind $\sum_{j,k} F_{jk}(z-\rho)^{\mu_j}\log^k(z-\rho)$.

In our case, for $z \sim 1$, one has, for some constants a, b, c, d, e:

$$T(z) \sim a\frac{\log\frac{1}{1-z}}{(1-z)^2} + b\frac{1}{(1-z)^2} + c\frac{\log\frac{1}{1-z}}{(1-z)} + d\frac{1}{1-z} + e\log\frac{1}{1-z} + O(1)\, .$$

Identifying the constants gives $a = 2/p$, $c = 2\frac{q}{p^2}$, $d = \frac{pqb+3-p}{p^2}$. Singularity analysis then gives

$$\begin{aligned} T_n \;=\; & a\left(n\log n + (\gamma-1)n + \log n + (\frac{1}{2}+\gamma) + \frac{5}{12n}\right) + b(n+1) \\ + \; & c\left(\log n + \gamma + \frac{1}{2n}\right) + d + e\frac{1}{n} + o(\frac{1}{n}) \end{aligned}$$

This leads to the theorem. □

Note that we don't have any closed form for b and e. This a well-known open problem for asymptotics of D-finite functions when the roots of the indicial polynomial of the differential equation differ by an integer.

4. Single Source Shortest Path Problems

We consider the single source shortest path problem with nonnegative integer edge weights. We assume our edge weights to be in $[0, 2^K - 1]$, i.e., edge weights are K bit integers. Meyer [8] has shown that the average complexity of the problem is linear $O(n + m)$. He assumes that the edge weights are random K bit integers and that a certain set of primitive operations on such integers can be performed in constant time (addition, finding the first bit where two integers differ, ...). The graph is arbitrary. His construction was later refined by

Goldberg [5] who showed that the radix heap implementation of the algorithm of Denardo and Fox [3, 1] runs in linear time for random edge weights. The worst case complexity of this algorithm is $O(m + nK)$. Algorithms with better worst case behavior are known [2, 10, 6]. We analyze the radix heap implementation of the algorithm of Denardo and Fox under limited randomness.

Theorem 3 (Shortest Paths under Limited Randomness)**.** *Let G be an arbitrary graph, let $c : E \mapsto [0, \ldots, 2^K - 1]$ be an arbitrary cost function, and let k be such that $0 \le k \le K$. Let $\overline{c}$ be obtained from c by making the last k bits of each edge cost random. Then the single source shortest path problem can be solved in expected time $O(m + n(K - k))$.*

With full randomness the expected running time is $O(m + n)$, with no randomness the running time is $O(m + nK)$. Limited randomness interpolates linearly between the extremes.

We sketch the proof. For a node v, let $min\_in\_cost(v)$ be the minimum cost of an incoming edge. Goldberg has shown that the running time of the radix heap implementation of the algorithm of Denardo and Fox is

$$O(n + m + \sum_v (K - \log min\_in\_cost(v) + 1) \ .$$

Next observe that $min\_in\_cost(v)$ is the minimum of $indeg(v)$ numbers of which the last k bits are random; here $indeg(v)$ is the indegree of v. For an edge e let $r(e)$ be the number of leading zeroes in the random part of e. Then $\mathrm{E}[r(e)] = 2$ and

$$\begin{aligned} K - \log min\_in\_cost(v) &\le K - k + \max\{r(e)\,;\, e \in inedges(v)\} \\ &\le K - k + \sum\{r(e)\,;\, e \in inedges(v)\} \end{aligned}$$

Thus

$$\mathrm{E}[K - \log min\_in\_cost(v)] \le K - k + O(indeg(v))$$

and the time bound follows.

5. Conclusion

We think the three analysis that we have performed here are quite typical of the techniques which should be used for "discrete smoothed complexity" analysis and it is very likely that most of the classical algorithms can be revisited with the tools that we used in this article.

Higher moments (e.g., the variance) and the limit laws seems to be quite hard to get for yet, even with the help of computer algebra systems. It involves a kind of functional equations, sparsely encountered in analysis of algorithms until now, which seems however to have some deep combinatorial properties.

It is not excluded that these approaches allow in few years to answer to questions such as "How far is randomization helpful?", e.g., for NP-complete problems.

Acknowledgements. Cyril Banderier is partially supported by the INRIA postdoctoral programme and by the Max-Planck-Institut. We thank Philippe Flajolet for helpful comments on a preliminary version of this article, and for a discussion about the Frobenius method, which he used for an analysis on quadtrees [4].

References

[1] Ravindra K. Ahuja, Kurt Mehlhorn, James B. Orlin, and Robert E. Tarjan. Faster algorithms for the shortest path problem. *J. Assoc. Comput. Mach.*, 37(2):213–223, 1990.

[2] Boris V. Cherkassky, Andrew V. Goldberg, and Craig Silverstein. Buckets, heaps, lists, and monotone priority queues. *SIAM J. Comput.*, 28(4):1326–1346 (electronic), 1999.

[3] Eric V. Denardo and Bennett L. Fox. Shortest-route methods. I. Reaching, pruning, and buckets. *Oper. Res.*, 27(1):161–186, 1979.

[4] Philippe Flajolet, Gaston Gonnet, Claude Puech, and J. M. Robson. Analytic variations on quadtrees. *Algorithmica*, 10(6):473–500, 1993.

[5] Andrew V. Goldberg. A simple shortest path algorithm with linear average time. In *Proceedings of the 9th European Symposium on Algorithms (ESA '01)*, pages 230–241. Springer Lecture Notes in Computer Science LNCS 2161, 2001.

[6] Torben Hagerup. Improved shortest paths on the word RAM. In *Automata, languages and programming (Geneva, 2000)*, pages 61–72. Springer, Berlin, 2000.

[7] Peter Henrici. *Applied and computational complex analysis. Vol. 2.* John Wiley & Sons Inc., New York, 1991. Reprint of the 1977 original.

[8] Ulrich Meyer. Shortest-Paths on arbitrary directed graphs in linear Average-Case time. In *Proceedings of the Twelfth Annual ACM-SIAM Symposium on Discrete Algorithms (SODA-01)*, pages 797–806. ACM Press, 2001.

[9] Rajeev Motwani and Prabhakar Raghavan. *Randomized algorithms.* Cambridge University Press, Cambridge, 1995.

[10] Raman. Recent results on the single-source shortest paths problem. In *SIGACTN: SIGACT News (ACM Special Interest Group on Automata and Computability Theory)*, volume 28, pages 61–72. Springer, Berlin, 1997.

[11] Bruno Salvy and Paul Zimmermann. Gfun: a Maple package for the manipulation of generating and holonomic functions in one variable. *ACM Transactions on Mathematical Software*, 20(2):163–177, 1994.

[12] Daniel A. Spielman and Shang-Hua Teng. Smoothed analysis of algorithms: Why the simplex algorithm usually takes polynomial time. In *proceedings of the The Thirty-Third Annual ACM Symposium on Theory of Computing*, pages 296–3051, 2001.

Max-Planck-Institut für Informatik
Algorithms and Complexity Group (AG1)
Stuhlsatzenhausweg 85, 66123 Saarbrücken (Germany)
E-mail address: `banderie, rbeier, mehlhorn@mpi-sb.mpg.de`
`http://www.mpi-sb.mpg.de/units/ag1/people.html`