Smoothed Analysis of Three Combinatorial Problems ⋆

Cyril Banderier[1], René Beier[2], and Kurt Mehlhorn[2]

[1] Laboratoire d'Informatique de Paris Nord Institut Galile, Universit Paris 13
99, avenue Jean-Baptiste Clment 93 430 Villetaneuse (France)
Cyril.Banderier@inria.fr
[2] Max-Planck-Institut für Informatik
Stuhlsatzenhausweg 85, 66123 Saarbrücken (Germany)
{rbeier,mehlhorn}@mpi-sb.mpg.de

Abstract. Smoothed analysis combines elements over worst-case and average case analysis. For an instance x, the smoothed complexity is the average complexity of an instance obtained from x by a perturbation. The smoothed complexity of a problem is the worst smoothed complexity of any instance. Spielman and Teng introduced this notion for continuous problems. We apply the concept to combinatorial problems and study the smoothed complexity of three classical discrete problems: quicksort, left-to-right maxima counting, and shortest paths.

1 Introduction

Recently, Spielman and Teng [13] introduced smoothed analysis, which is intermediate between average case analysis and worst case analysis. The smoothed complexity of an algorithm is

$$\max_x \mathrm{E}_{y \in U_\epsilon(x)} \; C(y)\,,$$

where x ranges over all inputs, y is a random instance in a neighborhood of x (whose size depends on the smoothing parameter ϵ), E denotes expectation, and $C(y)$ is the cost of the algorithm on input y. In other words, worst-case complexity is smoothed by considering the expected running time in a neighborhood of an instance instead of the running time at the instance. If $U_\epsilon(x)$ is the entire input space, smoothed analysis becomes average case analysis, and if $U_\epsilon(x) = \{x\}$ for all ϵ, smoothed analysis becomes worst case analysis. Smoothed analysis gives information whether instance space contains dense regions of hard instances, see Figure 1. The smoothed complexity of an algorithm is low if worst-case instances are "isolated events" in instance space. In other words, if the smoothed complexity is low, worst case instances are not robust under small changes; most small changes to the instance destroy the property of being worst-case; a small random perturbation destroys the property of being worst-case.

⋆ This work was partially supported by the Future and Emerging Technologies programme of the EU under contract number IST-1999-14186 (ALCOM-FT).

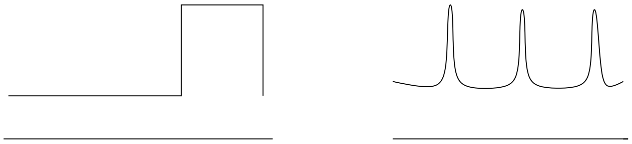

Fig. 1. The graphs show conceivable dependency of running time on input instance. We assume a one-dimensional input space; the running time on an instance is shown as the y-value of the graph. The neighborhood of an instance is simply an interval around the instance. In the situation on the left, the smoothed complexity will be equal to the worst case complexity (for all small enough ϵ), and in the situation on the right, the smoothed complexity decreases sharply as a function of ϵ.

In most sciences, one attempts to develop a theory that explains and predicts observed phenomena. Smoothed Analysis is one attempt to push the analysis of algorithms in this direction. To develop a mathematically rigorous theory of the behavior of algorithms, one must make mathematically rigorous assumptions about their inputs. Smoothed analysis makes the assumption that inputs are subject to noise, circumstance, or randomness. This is an assumption that is valid in most practical problem domains. While it is unreasonable to assume that one can develop a model of inputs subject to noise that precisely models practice, one can try to come close and then reason by analogy.

Spielman and Teng [13] showed that the smoothed complexity of the simplex algorithm (with the shadow-vertex pivot rule) for linear programming is polynomial. Linear programming is a continuous problem. The input is a sequence of real numbers (a cost vector, a constraint matrix, a right-hand side). The smoothing operation adds Gaussian noise with parameter σ to each number in the input. The expected running time of the simplex algorithm for such a perturbed instance is polynomial in $1/\sigma$ and the number of input variables. The other papers on smoothed analysis [3, 5] also discuss continuous problems.

Our paper is the first to apply the concept of smoothed analysis to discrete problems. We define natural models of perturbation for sequences and natural numbers and analyze the smoothed complexity of quicksort, left-to-right maxima and the shortest path problem.

Partial Permutations: Our first model applies to problems defined on sequences. It is parameterized by a real parameter p with $0 \le p \le 1$ and is defined as follows. Given a sequence $s_1, s_2, \ldots, s_n$, each element is selected (independently) with probability p. Let m be the number of selected elements (on average $m = pn$). Choose one of the $m!$ permutations of m elements (uniformly at random) and let it act on the selected elements. E.g., for $p = 1/2$ and $n = 7$, one might select $m = 3$ elements (namely, s_2, s_4, and s_7) out of an input sequence $(s_1, \overline{s_2}, s_3, \overline{s_4}, s_5, s_6, \overline{s_7})$. Applying the permutation (312) to the selected elements yields $(s_1, \overline{s_7}, s_3, \overline{s_2}, s_5, s_6, \overline{s_4})$. The probability to obtain this sequence in this way is $p^3(1-p)^4/3!$.

We show that the smoothed complexity of quicksort is $O((n/p)\ln p)$ (Section 2); however, it is not true that every fixed element takes part in $O((1/p)\ln n))$ comparisons on average. The maximum recursion depth is $\Omega(\sqrt{n/p})$ (Section 3).

Partial Bit Randomization: Our second model applies to problems involving natural numbers and is parameterized by an integer $k \geq 0$. For each integer of the input, the last k bits are randomly modified. This model is a discrete analogue of the model considered by Spielman and Teng. Notice, that in our model of perturbation the expectation of the resulting distribution is not necessarily equal to the unperturbed value. We show that the smoothed running time of the shortest path algorithm of Meyer [10] and Goldberg [6] on a graph with n nodes and m edges is $O(m + n(K-k))$ when the edge lengths are K bit integers, the last k of which are randomized. In [2] our model is used to analyze clairvoyant scheduling.

2 Quicksort

We analyze a deterministic variant of quicksort under partial permutations. We assume 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. We assume that the order of elements in the resulting two sublist is unchanged.

Theorem 1 (Quicksort under Limited Randomness). *The expected running time (i.e., number of comparisons) of quicksort on a partial permutation of n elements is $O((n/p)\ln n)$.*

Proof: We utilize a proof, based on randomized incremental constructions [11]. Let C denote the number of comparisons performed by the algorithm. Assume that we have a permutation of the numbers 1 to n. Let X_{ij} be the indicator variable which is 1 iff i and j are compared in a run of quicksort with i being the pivot. Clearly $C = \sum_{i,j} X_{ij}$. Then $X_{ij} = 1$ if and only if i occurs before all other elements with value between i and j in the sequence. Thus, for a random permutation, $\mathrm{prob}(X_{ij} = 1) = |1/(j-i+1)|$ and hence

$$\mathrm{E}[C] = \sum_{i \neq j} \frac{1}{|j-i+1|} \leq 2n \sum_{2 \leq k \leq n} \frac{1}{k} \leq 2n \ln n \ .$$

Next we estimate $\mathrm{prob}(X_{ij} = 1)$ for partial permutations. Let $s_1, \ldots, s_n$ be our initial permutation and let $L = (8/p)\ln n$. If i is among $s_1, \ldots, s_L$ or $|j-i| \leq L$, we estimate $\mathrm{prob}(X_{ij} = 1)$ for a total contribution of $O(\frac{n}{p}\ln n)$.

Next assume that there are at least L elements preceding i in the initial permutation and that $|i-j| > L$. We split our estimate for $\mathrm{prob}(X_{ij} = 1)$ into two parts. First assume that i is selected. Let $l = |i-j|$. The probability that at most $lp/2$ elements between i (exclusive) and j (inclusive) are selected is less than $\exp(-lp/8)$. If more than $lp/2$ elements are selected, $X_{ij} = 1$ implies that i is first in the permutation of the selected elements which happens with probability at most $2/(lp)$. Together we obtain $\mathrm{prob}(X_{ij} = 1) \leq \exp(-lp/8) + 2/(lp)$ and hence

$$\sum_{L<i\leq n} \sum_{l>L} \exp(-lp/8) + 2/(lp) = O\left(\frac{n \ln n}{p}\right) .$$

Assume next that i is not selected and let i be the k_i-th element in the initial sequence. The probability that less than $pk_i/2$ elements before i are chosen or less than $pl/2$ elements between i and j or more than $2pn$ elements are chosen altogether is less than $\exp(-pk_i/8)+\exp(-pl/8)+\exp(-pn/2)$. The contribution of these rare events to the $\mathrm{E}[C]$ is only $O(1/p)$ since for the first term we have

$$\sum_{k_i\geq L} \sum_{j} \exp(-pk_i/8) = n \sum_{m\geq 0} \exp(-pL/8) \exp(-p/8)^m = O(1/p) .$$

The same bound can be shown for the other two terms.

So assume that the required number of elements are chosen. If i occurs before all elements $i+1, \ldots, j$ in the partial permutation, it must be the case that none of the $pl/2$ selected elements between i and j is inserted before i which happens with probability at most

$$\left(\frac{2pn - k_i p/2}{2pn}\right)^{lp/2} \leq \exp(-k_i lp/(8n)).$$

Next observe that

$$\sum_{k=1}^{n} \sum_{l=1}^{n} \exp(-klp/(8n)) \leq \sum_{k=1}^{n} \frac{1}{1 - \exp(-kp/(8n))} \leq \sum_{k=1}^{n} \frac{16n}{kp} \leq \frac{16n \ln n}{p};$$

since $1 - e^{-x} \geq x/2$ for $0 \leq x \leq 1$ and hence $1/(1 - e^{-x}) \leq 2/x$. $\square$

Remark: When we consider partial permutations of a sorted sequence (the worst-case instance without permutations) we are able to get closed form formulae for the X_{ij}'s. We distinguish 10 subcases, most of them involving 7 nested sums. From these sums (involving binomials), it is possible to get the differential equation satisfied by their generating functions, and then the Frobenius method allows to get the full asymptotic scale which gives a $\frac{2}{p} n \ln n$ complexity. We refer the reader to the full paper for details.

Pitfalls: The expected running time of quicksort on random permutations can be analyzed in many different ways. Many of them rely on the fact that the subproblems generated by recursive calls are again random permutations. This is *not true* for partial permutations[3] as the following example demonstrates.

Consider an input $1, 2, 3, 4$ and define $q := 1 - p$. Assume that 2 is the pivot element and hence the second subproblem consists of the numbers $3, 4$. If 2 is the pivot (first element after permutation), at least the numbers 1 and 2 are selected. Conditioned on the fact that 1 and 2 are selected and 2 is made the first element we obtain subproblem $(3, 4)$

[3] In the first version of this paper, we fell into this pitfall.

- always, when neither 3 or 4 is selected (probability q^2),
- always[4], when 3 is selected, but 4 is not (probability pq)
- in one out of two cases[5], when 4 is selected but 3 is not (probability $pq/2$)
- in one out of two cases, when 3 and 4 are selected (probability $p^2/2$).

Thus $\text{prob}((3,4)) = q^2 + \frac{3}{2}pq + \frac{1}{2}p^2$. Consequently, $\text{prob}((4,3)) = \frac{1}{2}pq + \frac{1}{2}p^2$. On the other hand, applying partial permutations on input sequence 3, 4 gives $\text{prob}((3,4)) = q^2 + 2pq + \frac{1}{2}p^2$ and $\text{prob}((4,3)) = \frac{1}{2}p^2$.

We also point out 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 original 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.

The expected maximum recursion depth of quicksort on random permutations is $O(\ln n)$. For partial permutations the expected maximum recursion depth is $\Omega(\sqrt{n/p})$. This is a consequence of the result in the next section. We will show that the number of left-to-right-maxima in a partial permutation might be as large as $\Omega(\sqrt{n/p})$. The number of left-to-right-maxima is the number of times the largest element is compared to a pivot element. Thus some elements may take part in as many as $\Omega(\sqrt{n/p})$ recursive calls which disproves the conjecture that every element takes part in $O((1/p)\ln n)$ calls with high probability. In the full version of this paper we show that the worst-case instances (sorted sequence, increasing or decreasing) also exibit the highest smoothed complexity. The complexity landscape for quicksort has two dominant peaks with a rather sharp transition (cf. Figure 1). We will see in the next section that is not always the case.

3 Left-to-Right Maxima

The simplest strategy to determine the largest element in a sequence is to scan the sequence from left to right and to keep track of the largest element seen. The number of changes to the current maximum is called the number of *left-to-right maxima* in the sequence. The sequence $1,\ldots,n$ has n left-to-right maxima and the expected number of left-to-right maxima in a random permutation of n elements is $H_n = 1 + 1/2 + \cdots + 1/n$. Surprisingly, the worst case instance for the classical analysis is not the worst case under smoothing.

Theorem 2 (Left-to-Right Maxima). *Under the partial permutation model, the smoothed number of left-to-right maxima is* $\Omega(\sqrt{n/p})$ *and* $O(\sqrt{(n/p)\log n})$*, whereas the smoothed complexity for the sequence* $(1,\ldots,n)$ *is*

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

[4] The relevant partial permutations of the input are $(2,1,3,4)$ and $(2,3,1,4)$.

[5] The relevant partial permutations are $(2,1,3,4)$ and $(2,4,1,3)$. The former generates the subproblem $(3,4)$.

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

Proof: Due to space limitations we provide a proof only for the two first asymptotic terms of the smoothed complexity of the sorted sequence (see full version of the paper for a generating function proof which gives the full asymptotics).

The sequence $1, \ldots, n$ has n left-to-right maxima. Smoothing decreases the number to about $\ln(pn) + 2/p$ as we show next. 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. It contains a maximum if and only if it is a maximum among the selected elements. In this case its value is at least i and hence it is also a maximum considering all elements. 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. Consider $i < n/2$. Position i stays a maximum if non of the preceding $i-1$ elements move to a position larger than i. Analogously, position $n+1-i$ stays a maximum if non of the succeeding $i-1$ elements move to a position smaller than $i+1-i$. 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

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

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

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

We want to compute $\mathrm{E}[\sum_{i \leq n/2} X_i]$. Define $L = (16/p) \log n$. We split the sum into two parts: $i \leq L$ and $i > L$.

For the second part, $i > L$, we expect to select about $pi \geq pL$ 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 $\exp(-(16/8)\log n) = O(n^{-2})$ by Chernoff bounds. If at least $pL/2 = 8 \log n$ elements smaller than i are selected and at least $pn/4$ elements larger than i are selected the probability that i is a maximum is less than $f(8 \log n, pn/4) = O(n^{-2})$. Taking into account that i in not selected, we get $\mathrm{prob}(X_i = 1) = O((1-p)n^{-2})$ for $L \leq i \leq n/2$.

We turn to the i's with $i < L$. If none of the first $i-1$ elements is selected i stays a maximum. If at least one of the first $i-1$ elements is chosen, the probability that i stays a maximum is at most $e^{-pn/16} + 4/pn$. The first term accounts for the fact that less than $pn/4$ 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 for $i < L$

$$\mathrm{prob}(X_i = 1) \leq (1-p)\big((1-p)^{i-1} + e^{-pn/16} + 4/pn\big)$$

and hence

$$\mathrm{E}[\sum_{i=1}^{L-1} X_i] \leq \frac{1-p}{p} + (1-p)L(e^{-pn/16} + 4/pn) = \frac{1-p}{p}(1+o(1)),$$

for constant p. Taking into account all $i \geq n/2$, we conclude

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

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

We now come to the first assertion of the theorem: the complexity of the worst case among all perturbations. We show that, for $p < 1/2$, the smoothed number of left-to-right maxima in a permutation of n elements may be $\Omega(\sqrt{n/p})$. Consider the sequence

$$n-k, n-k+1, \ldots, n, 1, 2, \ldots, n-k-1 \qquad (\text{where } k = \sqrt{n/p})\,.$$

The first part of the sequence consists of the first k elements. Let $a \approx pk$ and $b \approx p(n-k)$ be the number of selected elements in the first and second part of the sequence respectively. For large n, the probability that $a > 2pk$ or $b < pn/2$ is exponentially small by Chernoff bounds. So assume $a \leq 2pk$ and $b \geq pn/2$. The probability that all elements selected in the first part are put into the second part by the random permutation of the selected elements is at least

$$q := \frac{b}{a+b} \cdot \frac{b-1}{a+b-1} \cdots \frac{b-a+1}{b+1} \ .$$

In this case all elements not selected in the first part are left-to-right maxima. We have

$$q \geq \left(\frac{b-a}{a+b}\right)^a = \left(1 - \frac{2a}{a+b}\right)^a = \exp\left(a \ln\left(1 - \frac{2a}{a+b}\right)\right) \geq \exp\left(-\frac{4a^2}{a+b}\right) \ .$$

since $\ln(1-x) \geq -2x$ for $0 \leq x \leq 3/4$. Using $a \leq 2pk$ and $b \geq pn/2$ we get

$$q \geq \exp\left(-\frac{4a^2}{a+b}\right) \geq \exp\left(-\frac{4(2p)^2 n/p}{pn/2}\right) \geq e^{-32} \ .$$

We conclude that with constant probability the number of left-to-right maxima in the perturbed sequence is at least $k - a \geq k(1-2p) = \Omega(\sqrt{n/p})$ for $p < 1/2$.

We next show an almost matching upper bound. Let $s_1, \ldots, s_n$ be an arbitrary permutation of the numbers 1 to n, let $k = \sqrt{8(n/p)\log n}$, and let I be the set of indices such that $i \geq k$ and $s_i \leq n-k$. Basically, I ignores the first k and the largest k elements of the permutation. We estimate how many s_i with $i \in I$ are left-to-right maxima in the perturbed sequence. Then the total number of maxima is at most $2k$ larger.

Consider a fixed s_i with $i \in I$. If s_i is selected and is a maximum in the partial permutation, it must be a maximum among the selected elements. The expected number of left-to-right maxima among the selected elements is $\ln pn$.

So assume that s_i is not selected. With high probability there are at least $kp/2$ elements preceding s_i among the selected elements, there are at least $kp/2$ elements larger than s_i among the selected elements, and there are at most $2np$ selected elements. Therefore, the probability that s_i is a maximum in the perturbed sequence is bounded by

$$\left(\frac{2np - kp/2}{2np}\right)^{kp/2} \leq \left(1 - \frac{k}{4n}\right)^{kp/2} \leq \exp(-k^2 p/(8n)) = \frac{1}{n}$$

and hence the expected number of left-to-right maxima in the perturbed sequence is $O(\sqrt{(n/p)\log n})$. $\square$

4 Single Source Shortest Path Problems

We consider the single source shortest path problem with nonnegative integer edge weights. As usual, let n and m denote the number of nodes and edges respectively. We assume our edge weights to be in $[0, 2^K - 1]$, i.e., edge weights are K bit integers. Meyer [10] has shown that the average complexity of the problem is linear $O(n+m)$. He assumes edge weights to be 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 algorithm can be used for arbitrary graphs. An alternative algorithm was later given by Goldberg [6] and his work is the starting point for this section. The worst case complexity of his algorithm is $O(m + nK)$. Algorithms with better worst case behavior are known [1, 4, 12, 8].

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 $0 \leq k \leq K$. 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.

Proof: Goldberg has shown that the running time of his algorithm is

$$O(n + m + \sum_v (K - \log \mathit{min\_in\_cost}(v) + 1),$$

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

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

Thus

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

and the time bound follows.

In our model of perturbation, the last k bits of each weight are set randomly. Alternatively, one might select bits with probability p and set selected bits to random values. With this definition, the smoothed complexity becomes $O(m/p)$. For an edge e, let $r(e)$ be the number of leading zeroes in the weight of e. Then $\mathrm{E}[r(e)] \leq 2/p$ and

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

Therefore, $\mathrm{E}[K - \log min\_in\_cost(v)] \leq O(indeg(v)/p)$ and the time bound follows. □

5 Conclusion

We analyzed the smoothed complexity of three combinatorial problems. Smoothed complexity gives additional information about the distribution of hard instances in instance space. We believe, that the analysis of further discrete problems is a worthwhile task. Beccetti et al. [2] have recently analyzed Non-Clairvoyant scheduling under the partial bit randomization model.

From a more theoretical viewpoint, it is natural to raise the question "Is there any relevant notion of smoothed complexity completeness?". Such a notion would have to extend the notion of average case completeness which was introduced by Levin [9, 7].

In conclusion, we believe that smoothed complexity is a key concept to get a better understanding of the behavior of algorithms in practice.

References

1. R. K. Ahuja, K. Mehlhorn, J. B. Orlin, and R. E. Tarjan. Faster algorithms for the shortest path problem. *J. Assoc. Comput. Mach.*, 37(2):213–223, 1990.
2. L. Becchetti, S. Leonardi, A. Marchetti-Spaccamela, G. Schäfer, and T. Vredeveld. Smoothening helps: A probabilistic analysis of the Multi-Level Feedback algorithm. submitted, 2003.

3. A. Blum and J.D. Dunagan. Smoothed analysis of the perceptron algorithm for linear programming. In *Proceedings of the 13th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA-02)*, pages 905–914. ACM Press, 2002.
4. 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.
5. J.D. Dunagan, D.A. Spielman, and S-H. Teng. Smoothed analysis of the renegar's condition number for linear programming. In *SIAM Conference on Optimization*, 2002.
6. 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.
7. Yuri Gurevich. Average case completeness. *Journal of Computer and System Sciences*, 42(3):346–398, 1991. Twenty-Eighth IEEE Symposium on Foundations of Computer Science (Los Angeles, CA, 1987).
8. Torben Hagerup. Improved shortest paths on the word RAM. In *Automata, languages and programming (ICALP 2000)*, pages 61–72. Springer, Berlin, 2000.
9. Leonid A. Levin. Average case complete problems. *SIAM J. Comput.*, 15(1):285–286, 1986.
10. 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.
11. Rajeev Motwani and Prabhakar Raghavan. *Randomized algorithms*. Cambridge University Press, Cambridge, 1995.
12. 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.
13. Daniel A. Spielman and Shang-Hua Teng. Smoothed analysis of algorithms: Why the simplex algorithm usually takes polynomial time. In *Proceedings of the Thirty-Third Annual ACM Symposium on Theory of Computing*, pages 296–3051, 2001.