Smoothed Analysis of Binary Search Trees*

Bodo Manthey** and Rüdiger Reischuk

Universität zu Lübeck, Institut für Theoretische Informatik
Ratzeburger Allee 160, 23538 Lübeck, Germany
manthey/reischuk@tcs.uni-luebeck.de

Abstract. Binary search trees are one of the most fundamental data structures. While the height of such a tree may be linear in the worst case, the average height with respect to the uniform distribution is only logarithmic. The exact value is one of the best studied problems in average-case complexity.
We investigate what happens in between by analysing the smoothed height of binary search trees: Randomly perturb a given (adversarial) sequence and then take the expected height of the binary search tree generated by the resulting sequence. As perturbation models, we consider partial permutations, partial alterations, and partial deletions.
On the one hand, we prove tight lower and upper bounds of roughly $\Theta(\sqrt{n})$ for the expected height of binary search trees under partial permutations and partial alterations. This means that worst-case instances are rare and disappear under slight perturbations. On the other hand, we examine how much a perturbation can increase the height of a binary search tree, i.e. how much worse well balanced instances can become.

1 Introduction

To explain the discrepancy between average-case and worst-case behaviour of the simplex algorithm, Spielman and Teng introduced the notion of *smoothed analysis* [5]. Smoothed analysis interpolates between average-case and worst-case analysis: Instead of taking the worst-case instance or, as in average-case analysis, choosing an instance completely at random, we analyse the complexity of (worst-case) objects subject to slight random perturbations, i.e. the expected complexity in a small neighbourhood of (worst-case) instances. Smoothed analysis takes into account that a typical instance is not necessarily a random instance and that worst-case instances are often artificial and rarely occur in practice.

Let C be some complexity measure. The worst-case complexity is $\max_x C(x)$, and the average-case complexity is $\mathbb{E}_{x\sim\Delta} C(x)$, where $\mathbb{E}$ denotes the expectation with respect to some probability distribution Δ. The smoothed complexity is defined as $\max_x \mathbb{E}_{y\sim\Delta(x,p)} C(y)$. Here, x is chosen by an adversary and y is randomly chosen according to some probability distribution $\Delta(x,p)$ that depends

* A full version of this work with all proofs and experimental data is available as Report 05-063 of the Electronic Colloquium on Computational Complexity (ECCC).

** Supported by DFG research grant RE 672/3.

on x and a parameter p. The distribution $\Delta(x, p)$ should favour instances in the vicinity of x, i.e. $\Delta(x, p)$ should put almost all weight on the neighbourhood of x, where "neighbourhood" has to be defined appropriately depending on the problem considered. The smoothing parameter p denotes how strong x is perturbed, i.e. we can view it as a parameter for the size of the neighbourhood. Intuitively, for $p = 0$, smoothed complexity becomes worst-case complexity, while for large p, smoothed complexity becomes average-case complexity.

The smoothed complexity of continuous problems seems to be well understood. There are, however, only few results about smoothed analysis of discrete problems. For such problems, even the term "neighbourhood" is often not well defined. Thus, special care is needed when defining perturbation models for discrete problems. Perturbation models should reflect "natural" perturbations, and the probability distribution for an instance x should be concentrated around x, particularly for small values of the smoothing parameter p.

Here, we will conduct a smoothed analysis of an ordering problem, namely the *smoothed height of binary search trees*. Binary search trees are one of the most fundamental data structures and, as such, building blocks for many advanced data structures. The main criteria of the "quality" of a binary search tree is its height. Unfortunately, the height is equal to the number of elements in the worst case, i.e. for totally unbalanced trees generated by an ordered sequence of elements. On the other hand, if a binary search tree is chosen at random, then the expected height is only logarithmic in the number of elements. Thus, there is a huge discrepancy between the worst-case and the average-case behaviour of binary search trees.

We analyse what happens in between: An adversarial sequence will be perturbed randomly and then the height of the binary search tree generated by the sequence thus obtained is measured. Thus, our instances are neither adversarial nor completely random.

The height of a binary search tree obtained from a sequence of elements depends only on the ordering of the elements. Therefore, one should use a perturbation model that slightly perturbs the order of the elements of the sequence. We consider the perturbation models *partial permutations*, *partial alterations*, and *partial deletions*. For all three, we show tight lower and upper bounds. As a by-product, we obtain tight bounds for the smoothed number of left-to-right maxima, which is the number of new maxima seen when scanning a sequence from the left to the right. This improves a result by Banderier et al. [1].

In smoothed analysis one analyses how fragile worst-case instances are. We suggest examining also the dual property: Given a good instance, how much can the complexity increase by slightly perturbing the instance? In other words, how stable are best-case instances? We show that there are best-case instances that indeed are not stable, i.e. there are sequences that yield trees of logarithmic height, but slightly perturbing them yields trees of polynomial height.

Existing Results. Spielman and Teng introduced smoothed analysis as a hybrid of average-case and worst-case complexity [5]. Since then, smoothed analysis has been applied to a variety of fields [4].

Banderier, Beier, and Mehlhorn [1] applied smoothed analysis to ordering problems. In particular, they analysed the number of left-to-right maxima of a sequence. Here the worst case is the sequence $1, 2, \ldots, n$, which yields n left-to-right maxima. On average we expect $\sum_{i=1}^{n} 1/i \approx \ln n$ left-to-right maxima. Banderier et al. used the perturbation model of *partial permutations*, where each element of the sequence is independently selected with a probability of $p \in [0, 1]$ and then a random permutation on the selected elements is performed. They proved that the number of left-to-right maxima under partial permutations is $O(\sqrt{(n/p)\log n})$ in expectation for $0 < p < 1$. Furthermore, they showed a lower bound of $\Omega(\sqrt{n/p})$ for $0 < p \leq 1/2$.

Given a sequence $\sigma = (\sigma_1, \sigma_2, \ldots, \sigma_n)$ of n distinct elements from any ordered set, we obtain a binary search tree $T(\sigma)$ by iteratively inserting $\sigma_1, \sigma_2, \ldots, \sigma_n$ into the initially empty tree (this is formally described in Section 2). The study of binary search trees is one of the most fundamental problems in computer science since they are the building blocks for a large variety of data structures.

The worst-case height of a binary search tree is obviously n: just take the sequence $\sigma = (1, 2, \ldots, n)$. (We define the length of a path as the number of vertices.) The expected height of the binary search tree obtained from a random permutation (with all permutations being equally likely) has been the subject of a considerable amount of research in the past, culminating in Reed's result [3] that the expectation of the height is $\alpha \ln n + \beta \ln(\ln n) + O(1)$ with $\alpha \approx 4.31107$ being the larger root of $\alpha \ln(2e/\alpha) = 1$ and $\beta = \frac{3}{2\ln(\alpha/2)} \approx 1.953$. Drmota [2] and Reed [3] proved independently of each other that the variance of the height is $O(1)$.

Although the worst-case and average-case height of binary search trees are very well understood, nothing is known in between, i.e. when the sequences are not completely random, but the randomness is limited.

New Results. We consider the height of binary search trees subject to slight random perturbations, i.e. the expected height under limited randomness.

We consider three perturbation models, which will formally be defined in Section 3. *Partial permutations*, introduced by Banderier et al. [1], rearrange some elements, i.e. they randomly permute a small subset of the elements of the sequence. The other two perturbation models are new. *Partial alterations* do not move elements, but replace some elements by new elements chosen at random. Thus, they change the rank of the elements. *Partial deletions* remove some of the elements of the sequence without replacement, i.e. they shorten the input. This model turns out to be useful for analysing the other two models.

We prove matching lower and upper bounds for the expected height of binary search trees under all three perturbation models (Section 5). More precisely: For all $p \in (0, 1)$ and all sequences of length n, the expectation of the height of a binary search tree obtained via p-partial permutation is at most $6.7 \cdot (1-p) \cdot \sqrt{n/p}$ for sufficiently large n. On the other hand, the expected height of a binary search tree obtained from the sorted sequence via p-partial permutation is at least $0.8 \cdot (1-p) \cdot \sqrt{n/p}$. This lower bound matches the upper bound up to a constant factor.

For the number of left-to-right maxima under partial permutations, we are able to prove an even better upper bound of $3.6 \cdot (1-p) \cdot \sqrt{n/p}$ for all sufficiently large n and a lower bound of $0.4 \cdot (1-p) \cdot \sqrt{n/p}$ (Section 4).

All these bounds hold for partial alterations as well.

For partial deletions, we obtain $(1-p) \cdot n$ both as lower and upper bound.

In smoothed analysis one analyses how fragile worst case instances are. We suggest examining also the dual property: Given a good instance, how much can the complexity increase if the instance is perturbed slightly?

The main reason for considering partial deletions is that we can bound the expected height under partial alterations and permutations by the expected height under partial deletions (Section 6). The converse holds as well, we only have to blow up the sequences quadratically.

We exploit this when considering the stability of the perturbation models in Section 7: We prove that partial deletions and, thus, partial permutations and partial alterations as well can cause best-case instances to become much worse. More precisely: There are sequences of length n that yield trees of height $O(\log n)$, but the expected height of the tree obtained after smoothing the sequence is $n^{\Omega(1)}$.

2 Preliminaries

For any $n \in \mathbb{N}$, let $[n] = \{1, 2, \ldots, n\}$ and $[n - \frac{1}{2}] = \{\frac{1}{2}, \frac{3}{2}, \ldots, n - \frac{1}{2}\}$.

Let $\sigma = (\sigma_1, \ldots, \sigma_n) \in S^n$ for some ordered set S. We call σ a **sequence** of length n. Usually, we assume that all elements of σ are distinct. In most cases, σ will simply be a permutation of $[n]$. We denote the sorted sequence $(1, 2, \ldots, n)$ by σ^n_{sort}. When considering partial alterations, we define $\sigma^n_{\text{sort}} = (\frac{1}{2}, \frac{3}{2}, \ldots, n - \frac{1}{2})$ instead (this will be clear from the context).

Let $\sigma = (\sigma_1, \ldots, \sigma_n)$ be a sequence. We obtain a **binary search tree $T(\sigma)$** from σ by iteratively inserting the elements $\sigma_1, \sigma_2, \ldots, \sigma_n$ into the initially empty tree as follows: The root of $T(\sigma)$ is the first element σ_1 of σ. Let $\sigma_< = \sigma_{\{i|\sigma_i<\sigma_1\}}$ be σ restricted to elements smaller than σ_1. The left subtree of the root σ_1 of $T(\sigma)$ is obtained inductively from $\sigma_<$. Analogously, let $\sigma_> = \sigma_{\{i|\sigma_i>\sigma_1\}}$ be σ restricted to elements greater than σ_1. The right subtree of σ_1 of $T(\sigma)$ is obtained inductively from $\sigma_>$. Figure 1 shows an example. We denote the height of $T(\sigma)$, i.e. the number of nodes on the longest path from the root to a leaf, by **height(σ)**.

The element σ_i is called a **left-to-right maximum** of σ if $\sigma_i > \sigma_j$ for all $j \in [i-1]$. Let **ltr(σ)** denote the number of left-to-right maxima of σ. We have $\text{ltr}(\sigma) \leq \text{height}(\sigma)$ since the number of left-to-right maxima of a sequence is equal to the length of the right-most path in the tree $T(\sigma)$.

3 Perturbation Models for Permutations

Since we deal with ordering problems, we need perturbation models that slightly change a given permutation of elements. There seem to be two natural possi-

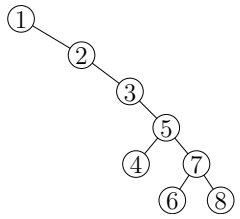

Fig. 1. The tree $T(\sigma)$ obtained from $\sigma = (1, 2, 3, 5, 7, 4, 6, 8)$. We have height$(\sigma) = 6$.

bilities: Either *change the positions* of some elements, or *change the elements* themselves.

Partial permutations implement the first option: A subset of the elements is randomly chosen, and then these elements are randomly permuted.

The second possibility is realised by partial alterations. Again, a subset of the elements is chosen randomly. These elements are then replaced by random elements.

The third model, partial deletions, also starts by randomly choosing a subset of the elements. These elements are then removed without replacement.

For all three models, we obtain the random subset as follows. Let σ be a sequence of length n and $p \in [0, 1]$ be a probability. Every element of σ is marked independently of the others with probability p.

By $\mathbf{height\text{-}perm}_p(\sigma)$, $\mathbf{height\text{-}alter}_p(\sigma)$, and $\mathbf{height\text{-}del}_p(\sigma)$ we denote the expected height of the binary search tree $T(\sigma')$, where σ' is the sequence obtained from σ by performing a p-partial permutation, alteration, and deletion, respectively (all three models will be defined formally in the following). Analogously, by $\mathbf{ltr\text{-}perm}_p(\sigma)$, $\mathbf{ltr\text{-}alter}_p(\sigma)$, and $\mathbf{ltr\text{-}del}_p(\sigma)$ we denote the expected number of left-to-right maxima of the sequence σ' obtained from σ via p-partial permutation, alteration, and deletion, respectively.

Partial Permutations. The notion of ***p*-partial permutations** was introduced by Banderier et al. [1]. Given a random subset M_p^n of $[n]$, the elements at positions in M_p^n are permuted according to a permutation drawn uniformly at random: Let $\sigma = (\sigma_1, \ldots, \sigma_n)$. Then the sequence $\sigma' = \Pi(\sigma, M_p^n)$ is a random variable with the following properties:

- Π chooses a permutation π of M_p^n uniformly at random and
- sets $\sigma'_{\pi(i)} = \sigma_i$ for all $i \in M_p^n$ and $\sigma'_i = \sigma_i$ for all $i \notin M_p^n$.

Figure 2 shows an example of a partial permutation.

By varying p, we can interpolate between the average and the worst case: for $p = 0$, no element is marked and $\sigma' = \sigma$, while for $p = 1$, σ' is a random permutation of the elements of σ with all permutations being equally likely.

Partial permutations are a suitable perturbation model since the distribution of $\Pi(\sigma, M_p^n)$ favours sequences close to σ. To show this, we have to introduce a metric on sequences. Let σ and τ be two sequences of length n. We assume that

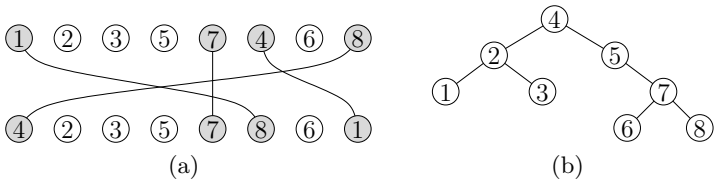

Fig. 2. A partial permutation. (a) Top: The sequence $\sigma = (1, 2, 3, 5, 7, 4, 6, 8)$; Figure 1 shows $T(\sigma)$. The first, fifth, sixth, and eighth element is (randomly) marked, thus $M_p^n = \{1, 5, 6, 8\}$. Bottom: The marked elements are randomly permuted. The result is the sequence $\sigma' = \Pi(\sigma, \mu)$, in this case $\sigma' = (4, 2, 3, 5, 7, 8, 6, 1)$. (b) $T(\sigma')$ with $\text{height}(\sigma') = 4$.

both are permutations of $[n]$ and define the distance $d(\sigma, \tau)$ between σ and τ as $d(\sigma, \tau) = |\{i \mid \sigma_i \neq \tau_i\}|$, thus d is a metric.

The distribution of $\Pi(\sigma, M_p^n)$ is symmetric around σ with respect to d. Furthermore, the probability that $\Pi(\sigma, M_p^n)$ equals some τ decreases exponentially with $d(\sigma, \tau)$. Thus, the distribution of $\Pi(\sigma, M_p^n)$ is highly concentrated around σ.

Partial Alterations. Let us now introduce ***p*-partial alterations**. For this perturbation model, we restrict the sequences of length n to be permutations of $[n - \frac{1}{2}] = \{\frac{1}{2}, \frac{3}{2}, \ldots, n - \frac{1}{2}\}$.

Every element at a position in M_p^n is replaced by a real number drawn uniformly and independently at random from $[0, n)$ to obtain a sequence σ'. All elements in σ' are distinct with probability one.

Instead of considering permutations of $[n - \frac{1}{2}]$, we could also consider permutations of $[n]$ and draw the random values from $[\frac{1}{2}, n + \frac{1}{2})$. This would not change the results. Another possibility would be to consider permutations of $[n]$ and draw the random values from $[0, n + 1)$. This would not change the results by much either. However, for technical reasons, we consider partial alterations as introduced above.

Like partial permutations, partial alterations interpolate between the worst case ($p = 0$) and the average case ($p = 1$). Partial alterations are somewhat easier to analyse: The majority of results on the average-case height of binary search trees is actually not obtained by considering random permutations. Instead, the binary search trees are grown from a sequence of n random variables that are uniformly and independently drawn from $[0, 1)$. This corresponds to partial alterations for $p = 1$. There is no difference between partial permutations and partial alterations for $p = 1$. This appears to hold for all p in the sense that the lower and upper bounds obtained for partial permutations and partial alterations are equal for all p.

Partial Deletions. As the third perturbation model, we introduce ***p*-partial deletions**: Again, we have a random marking M_p^n. Then we remove all marked elements.

Partial deletions do not really perturb a sequence: any ordered sequence remains ordered even if elements are deleted. The reason for considering partial deletions is that they are easy to analyse when considering the stability of perturbation models (Section 7). The results obtained for partial deletions then carry over to partial permutations and partial alterations since the expected heights with respect to these three models are closely related (Section 6).

4 Tight Bounds for the Number of Left-To-Right Maxima

Partial Permutations. The main idea for proving the following theorem is to estimate the probability that one of the k largest elements of σ is among the first k elements, which would bound the number of left-to-right maxima by $2k$.

Theorem 1. *Let $p \in (0,1)$. Then for all sufficiently large n and for all sequences σ of length n,*

$$\text{ltr-perm}_p(\sigma) \leq 3.6 \cdot (1-p) \cdot \sqrt{n/p}\,.$$

The following lemma is an improvement of the lower bound proof for the number of left-to-right maxima under partial permutations presented by Banderier et al. [1]. We obtain a lower bound with a much larger constant that holds for all $p \in (0,1)$; the lower bound provided by Banderier et al. holds only for $p \leq 1/2$.

The idea of the proof is as follows. Let $K_c = c\sqrt{n/p}$ and let $\sigma = (n - K_c + 1, \ldots, n, 1, \ldots, n - K_c)$. The probability that none of the first K_c elements of σ, which are also the K_c largest elements of σ, is moved further to the front is bounded from below by $\exp(-c^2/\alpha)$ for any fixed $\alpha > 1$. In such a case, all unmarked elements of the first K_c elements are left-to-right maxima.

Lemma 1. *Let $p \in (0,1)$, $\alpha > 1$, and $c > 0$. For all sufficiently large n, there exists a sequence σ of length n with* $\text{ltr-perm}_p(\sigma) \geq \exp(-c^2\alpha) \cdot c \cdot (1-p) \cdot \sqrt{n/p}$.

We obtain the strongest lower bound from Lemma 1 by choosing α close to 1 and $c = 1/\sqrt{2\alpha}$. This yields the following theorem.

Theorem 2. *For all $p \in (0,1)$ and all sufficiently large n, there exists a sequence σ of length n with*

$$\text{ltr-perm}_p(\sigma) \geq 0.4 \cdot (1-p) \cdot \sqrt{n/p}\,.$$

Theorem 2 also yields the same lower bound for $\text{height-perm}_p(\sigma)$ since the number of left-to-right maxima of a sequence is a lower bound for the height of the binary search tree obtained from that sequence. We can, however, prove a stronger lower bound for the smoothed height of binary search trees (Theorem 4).

A consequence of Lemma 1 is that there is no constant c such that the number of left-to-right maxima is at most $c \cdot (1-p) \cdot \sqrt{n/p}$ with high probability, i.e. with a probability of at least $1 - n^{-\Omega(1)}$. Thus, the bounds proved for the expected tree height or the number of left-to-right maxima cannot be generalised to bounds that hold with high probability. However, we can prove that with high probability, the height under partial permutations is $O(\sqrt{(n/p) \cdot \log n})$. Clearly, this bound holds for the number of left-to-right maxima as well.

Partial Alterations. Similar to the results for partial permutations, we obtain an upper bound of $3.6 \cdot (1-p) \cdot \sqrt{n/p}$ and a lower bound of $0.4 \cdot (1-p) \cdot \sqrt{n/p}$.

Again, we cannot achieve a bound of $O((1-p) \cdot \sqrt{n/p})$ for the number of left-to-right maxima that holds with high probability, but we can show that the height after partial alteration is $O(\sqrt{(n/p) \cdot \log n})$ with high probability.

5 Tight Bounds for the Height of Binary Search Trees

Partial Permutations. The following theorems is one of the main results of this work. The idea for proving it is as follows: We divide the sequence into blocks $B_1, B_2, \ldots$, where B_d is of size $cd^2\sqrt{n/p}$ for some $c > 0$. Each block B_d is further divided into d^4 parts $A_d^1, \ldots, A_d^{d^4}$, each consisting of $cd^{-2}\sqrt{n/p}$ elements. Assume that on every root-to-leaf path in the tree obtained from the perturbed sequence, there are elements of at most two such A_d^i for every d. Then the height can be bounded from above by $\sum_{d=1}^{\infty} 2 \cdot cd^{-2}\sqrt{n/p} = (c\pi^2/3)\sqrt{n/p}$.

The probability for such an event is roughly $O(\exp(-c^2)^2/(1-\exp(-c^2)))$. We obtain the upper bound claimed in the theorem mainly by carefully applying this bound and by exploiting the fact that only a fraction of $(1-p)$ of the elements are unmarked. Marked elements contribute at most $O(\log n)$ to the expected height of the tree.

Theorem 3. *Let $p \in (0,1)$. Then for all sufficiently large n and all sequences σ of length n,*

$$\text{height-perm}_p(\sigma) \leq 6.7 \cdot (1-p) \cdot \sqrt{n/p}\,.$$

We can also prove the following bound for the tree height: With probability $1 - n^{-\Omega(1)}$, the height is at most $O(\sqrt{(n/p) \cdot \log n})$. More precisely: The probability that the height after partial permutation is at most $c \cdot \sqrt{(n/p) \cdot \log n}$ is at least $1 - n^{-(c/3)^2/\alpha+0.5}$ for sufficiently large n and arbitrary $\alpha > 1$.

As a counterpart to Theorem 3, we prove the following lower bound. Interestingly, the lower bound is obtained for the sorted sequence, which is not the worst case for the expected number of left-to-right maxima under partial permutation; the expected number of left-to-right maxima of the sequence obtained by partially permuting the sorted sequence is only logarithmic [1].

Theorem 4. *Let $p \in (0,1)$. Then for all sufficiently large $n \in \mathbb{N}$,*

$$\text{height-perm}_p(\sigma^n_{\text{sort}}) \geq 0.8 \cdot (1-p) \cdot \sqrt{n/p}\,.$$

Partial Alterations. The results proved for partial permutation can be carried over to partial alterations. This means particularly that we obtain the same upper bound of $\text{height-alter}_p(\sigma) \leq 6.7 \cdot (1-p) \cdot \sqrt{n/p}$ for all $p \in (0,1)$ and all sequences σ with elements from $[n - \frac{1}{2}]$ for sufficiently large n.

Furthermore, we obtain the same upper bound of $n^{-(c/3)^2/\alpha+0.5}$ on the probability that the height after partial alteration is greater than $c \cdot \sqrt{(n/p) \cdot \log n}$

Finally, we have $\text{height-alter}_p(\sigma^n_{\text{sort}}) \geq 0.8 \cdot (1-p) \cdot \sqrt{n/p}$ for all $p \in (0,1)$ and sufficiently large n.

6 Partial Deletions versus Permutations and Alterations

For partial deletions, we easily obtain height-del$_p(\sigma) \leq (1-p)\cdot n$ for all sequences σ of length n and all $p \in [0,1]$ as an upper bound and height-del$_p(\sigma^n_{\text{sort}}) = (1-p)\cdot n$ as a lower bound.

Partial deletions are in some sense the worst of the three models: Trees are usually expected to be higher under partial deletions than under partial permutations or alterations, even though they contain fewer elements. The expected height under partial deletions yields upper bounds (up to an additional $O(\log n)$ term) for the expected height under partial permutations and alterations. The same holds for the number of left-to-right maxima.

Lemma 2. *For all sequences σ of length n and $p \in [0,1]$,* height-perm$_p(\sigma) \leq$ height-del$_p(\sigma)+O(\log n)$. *If σ is a permutation of $[n-\frac{1}{2}]$, then* height-alter$_p(\sigma) \leq$ height-del$_p(\sigma) + O(\log n)$.

The converse is not true, but we can bound the expected height under partial deletions by the expected height under partial permutations or alterations by padding the sequences considered. The following lemma holds also for partial alterations if the sequence σ is a permutation of $[n-\frac{1}{2}]$.

Lemma 3. *Let $p \in (0,1)$ be fixed and let σ be a sequence of length n with* height$(\sigma) = d$ *and* height-del$_p(\sigma) = d'$. *Then there exists a sequence $\tilde{\sigma}$ of length $O(n^2)$ with* height$(\tilde{\sigma}) = d + O(\log n)$ *and* height-perm$_p(\tilde{\sigma}) \in \Omega(d')$.

7 The (In-)Stability of Perturbations

Having shown that worst-case instances become much better when smoothed, we now provide a family of best-case instances for which smoothing results in an exponential increase in height. We consider the following family of sequences: $\sigma^{(1)} = (1)$ and $\sigma^{(k+1)} = (2^k, \sigma^{(k)}, 2^k + \sigma^{(k)})$, where $c + \sigma = (c+\sigma_1, \ldots, c+\sigma_n)$ for a sequence σ of length n. For instance, $\sigma^{(3)} = (4,2,1,3,6,5,7)$. Let $n = 2^k - 1$. Then $\sigma^{(k)}$ contains the numbers $1, 2, \ldots, n$, and we have height$(\sigma^{(k)}) =$ ltr$(\sigma^{(k)}) = k \in \Theta(\log n)$.

Deleting the first element of $\sigma^{(k)}$ roughly doubles the number of left-to-right maxima in the resulting sequence. This is the idea behind the following theorem.

Theorem 5. *For all $p \in (0,1)$ and all $k \in \mathbb{N}$,* ltr-del$_p(\sigma^{(k)}) = \frac{1-p}{p}\cdot((1+p)^k - 1)$.

Since the number of left-to-right maxima is a lower bound for the height of a binary search tree, we obtain height-del$_p(\sigma^{(k)}) \geq \frac{1-p}{p}\cdot((1+p)^k - 1)$.

We conclude that there are some best-case instances that are quite fragile under partial deletions: From logarithmic height they "jump" via smoothing to a height of $\Omega(n^{\log(1+p)})$. (We have $\frac{1-p}{p}\cdot((1+p)^k - 1) \in \Theta(n^{\log(1+p)})$.)

We can transfer this result to partial permutations due to Lemma 3. The result holds also for partial alterations. This means that there are sequences that yield trees of height $O(\log n)$, but perturbing them with partial permutations or partial alterations yields trees of height $\Omega(n^\delta)$ for some fixed $\delta > 0$.

8 Conclusions

We have analysed the height of binary search trees obtained from perturbed sequences and obtained asymptotically tight bounds of roughly $\Theta(\sqrt{n})$ for the height under partial permutations and alterations. This stands in contrast to both the worst-case and the average-case height of n and $\Theta(\log n)$, respectively.

Interestingly, the sorted sequence turns out to be the worst-case for the smoothed height of binary search trees in the sense that the lower bounds are obtained for σ^n_{sort}. This is in contrast to the fact that the expected number of left-to-right maxima of σ^n_{sort} under p-partial permutations is roughly $O(\log n)$ [1]. We believe that for binary search trees, σ^n_{sort} is indeed the worst case.

We performed experiments to estimate the constants in the bounds for the height of binary search trees. The results led to the conjecture that the expected height of trees obtained by performing a partial permutation on σ^n_{sort} is $(\gamma+o(1))\cdot(1-p)\cdot\sqrt{n/p}$ for some $\gamma \approx 1.8$ and for all $p \in (0,1)$. Proving this conjecture would immediately improve our lower bound. Provided that the sorted sequence is indeed the worst case, this conjecture would also improve the upper bound for binary search trees and left-to-right maxima.

The bounds obtained in this work for partial permutations and partial alterations are equal. We suspect that this is always true for binary search trees.

Finally, we are interested in generalising these results to other problems based on permutations, like sorting (Quicksort under partial permutations has already been investigated by Banderier et al. [1]), routing, and other algorithms and data structures. Hopefully, this will shed some light on the discrepancy between the worst-case and average-case complexity of these problems.

References

1. Cyril Banderier, René Beier, and Kurt Mehlhorn. Smoothed analysis of three combinatorial problems. In Branislav Rovan and Peter Vojtás, editors, *Proc. of the 28th Int. Symp. on Mathematical Foundations of Computer Science (MFCS)*, volume 2747 of *Lecture Notes in Computer Science*, pages 198–207. Springer, 2003.
2. Michael Drmota. An analytic approach to the height of binary search trees II. *Journal of the ACM*, 50(3):333–374, 2003.
3. Bruce Reed. The height of a random binary search tree. *Journal of the ACM*, 50(3):306–332, 2003.
4. Daniel A. Spielman. The smoothed analysis of algorithms. In Maciej Liśkiewicz and Rüdiger Reischuk, editors, *Proc. of the 15th Int. Symp. on Fundamentals of Computation Theory (FCT)*, volume 3623 of *Lecture Notes in Computer Science*, pages 17–18. Springer, 2005.
5. Daniel A. Spielman and Shang-Hua Teng. Smoothed analysis of algorithms: Why the simplex algorithm usually takes polynomial time. *Journal of the ACM*, 51(3):385–463, 2004.