Scenario-Based Robust Optimization of Tree Structures

Christoph Dürr, Sorbonne University

joint work with Spyros Angelopoulos, Alex Elenter and Georgii Melidi

Binary search trees

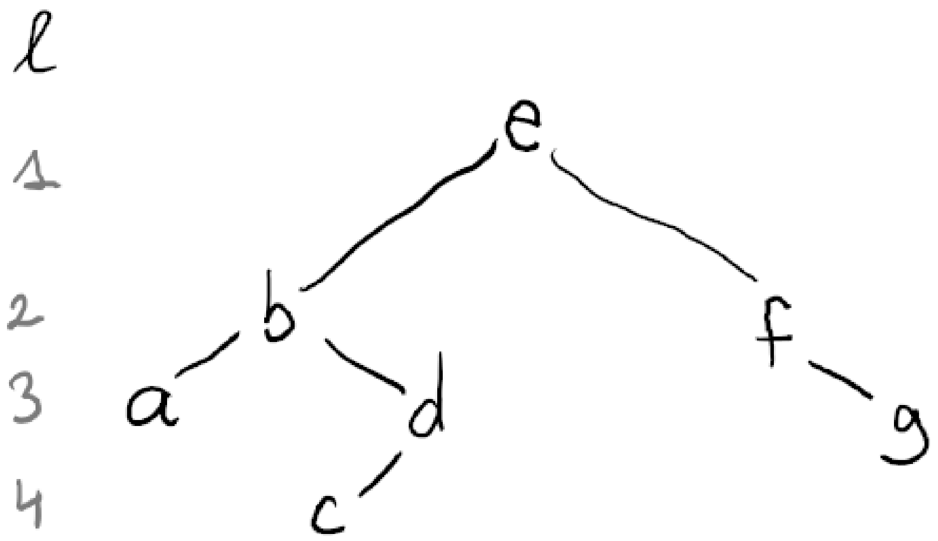
Binary search trees

Input distribution p over an alphabet

Output binary search tree with levels ℓ

Cost $p \cdot \ell := \sum_i p_i \ell_i$

Optimal tree by dynamic programming in time $O(n^2)$ [Knuth 1971]



Our model

Algorithm chooses a tree (ℓ)

Adversary chooses a distribution p^j out of k given scenarios.

Possible measures cost, ratio, regret

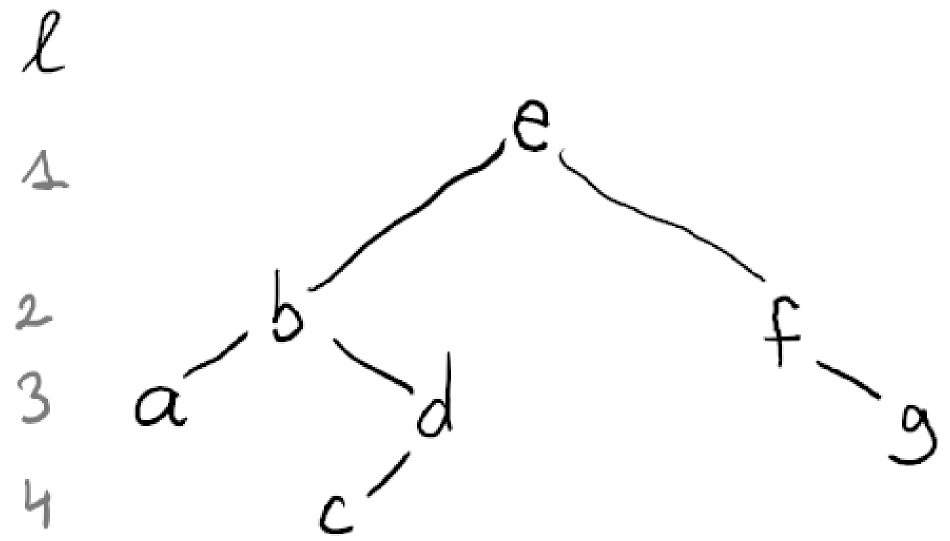
Binary search trees

Input distribution p over an alphabet

Output binary search tree with levels ℓ

Cost $p \cdot \ell := \sum_i p_i \ell_i$

Optimal tree by dynamic programming in time $O(n^2)$ [Knuth 1971]



Our model

$\min_\ell \max_j$

Cost $p^j \cdot \ell$

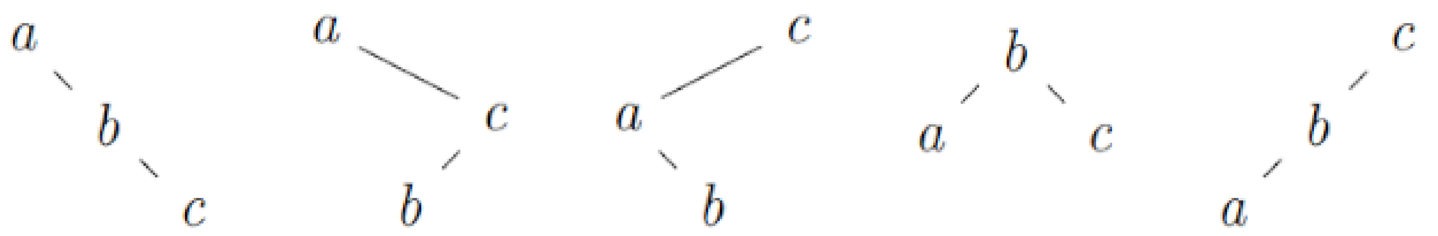
Ratio $\frac{p^j \cdot \ell}{\min_{\ell*} p^j \cdot \ell*}$

Regret

$p^j \cdot \ell - \min_{\ell*} p^j \cdot \ell*$

The measures differ

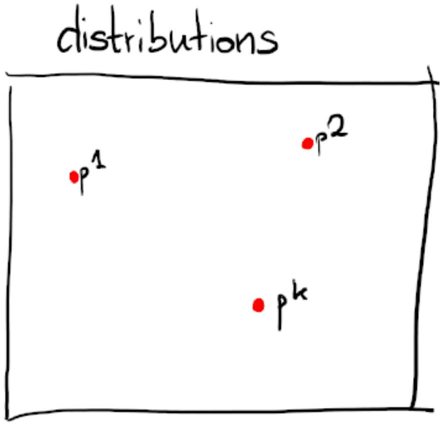
$$p^1 = \left(0, \tfrac{1}{4}, \tfrac{3}{4}\right), \quad p^2 = \left(\tfrac{4}{9}, \tfrac{2}{9}, \tfrac{1}{3}\right)$$

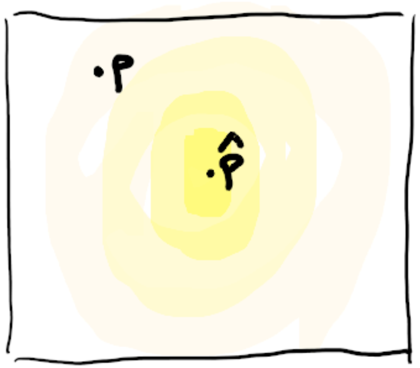


cost for p^1	11/4	9/4	7/4	3/2	5/4
cost for p^2	17/9	16/9	16/9	17/9	19/9
worst cost	11/4	9/4	16/9	17/9	19/9
competitive ratio	11/5	9/5	7/5	6/5	19/16
regret	3/2	1	1/2	1/4	1/3

Related work

- **Binary Search with Distributional Predictions**
  Dinitz, Im, Lavastida, Moseley, Niaparast, Vassilvitskii, 2024
- Predicted distribution $\hat{p}$
- Real distribution p with distance η from $\hat{p}$
- Create binary search tree with cost $O(H(p) + \eta)$ and this is optimal.
- Tree alternates between two strategies





Our results

- It is **NP–hard** to optimize the cost, already for $k = 2$ scenarios
- **Competitive ratio** is $\lceil \log_2(k+1) \rceil$
- **Pareto frontier** of costs can be computed in polynomial time, for constant k

Lower bound on competitive ratio

distributions

k keys

k scenarios

1	2	3	4	5
1	0	0	0	0
0	1	0	0	0
0	0	1	0	0
0	0	0	1	0
0	0	0	0	1

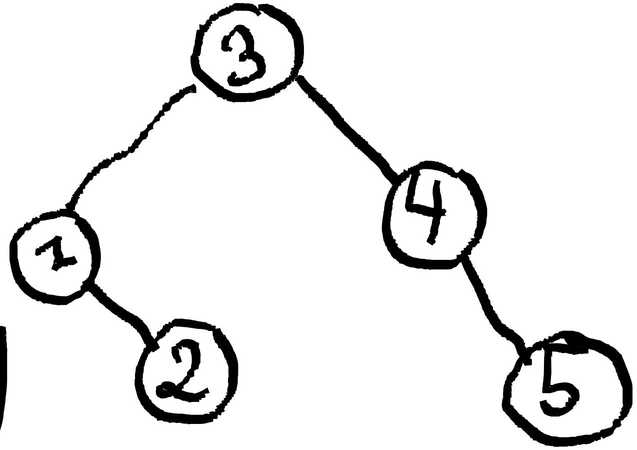
any BST

levels

1

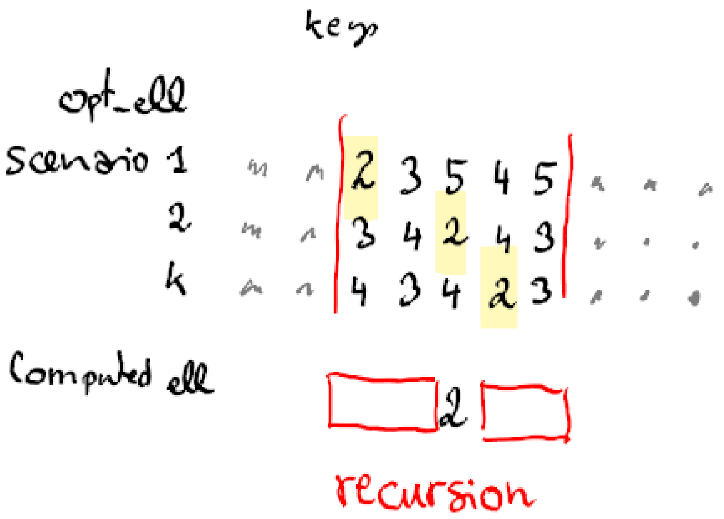
⋮

$\lceil \log(k+1) \rceil$



Upper bound on competitive ratio

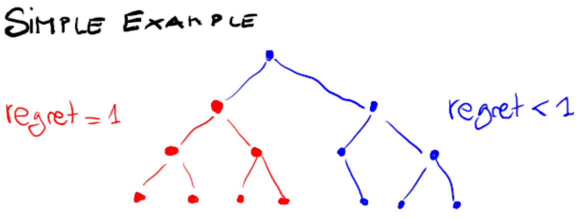
```
# optimal trees for each scenario
opt_ell = [opt_bst(p_i) for p_i in p]

# recursive construction on key interval
def bst(interval, next_level=1):
  # keys with minimum level for each scenario
  for all scenario i:
    S[i] = argmin opt_ell[i][x] for x in interval
    mid = median of S
    ell[mid] = next_nevel
    left, right = interval
    bst(left, mid-1, next_nevel+1)
    bst(mid+1, right, next_level+1)
```

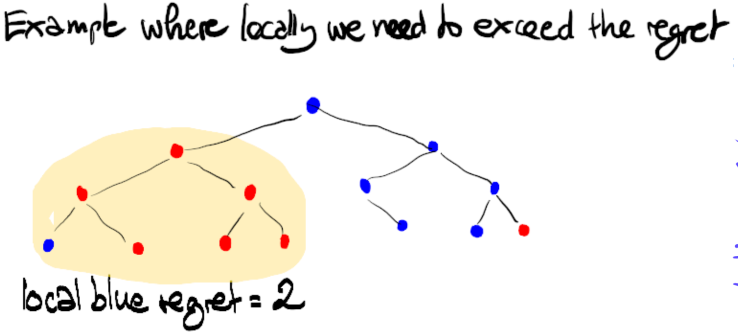


Hard open question

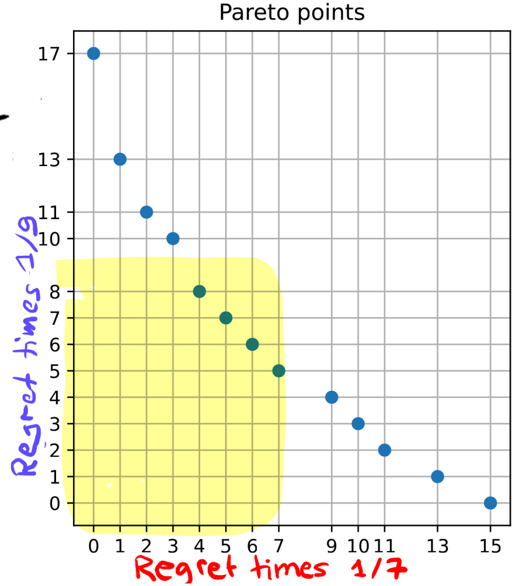
- Suppose we have a keys *be-longing* to scenario 1, and b to scenario 2
- $p^1 = (0, 0, \frac{1}{a}, 0, \frac{1}{a})$
- $p^2 = (\frac{1}{b}, \frac{1}{b}, 0, \frac{1}{b}, 0)$
- **Conjecture** There is a binary search tree with regret 1.

  In other words, a tree with levels ℓ such that for each scenario j we have $p^j \cdot \ell \leq 1 + \min_{\ell^*} p^j \cdot \ell^*$.





Full understanding of the Pareto-curve could be necessary for the proof.



Thank you