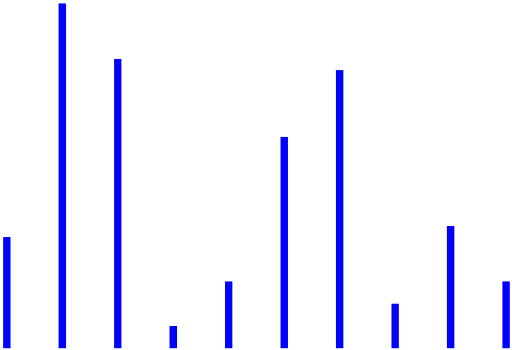
What can we learn from two instances of the same problem?

Rasťo Šrámek

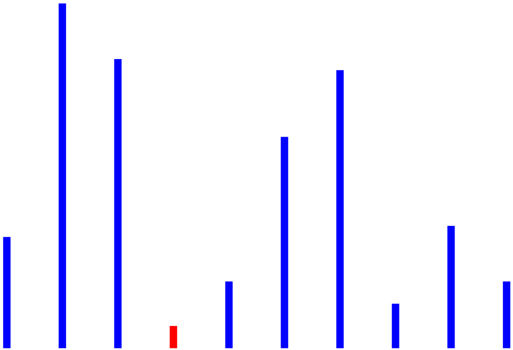
ETH Zürich

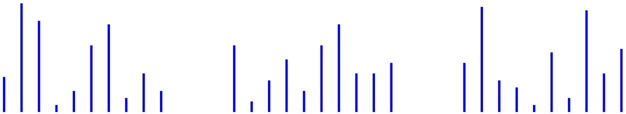
August 30, 2012

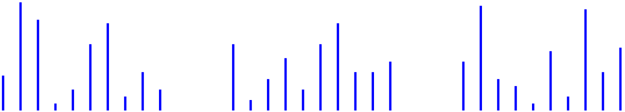
Find the position with minimum value:



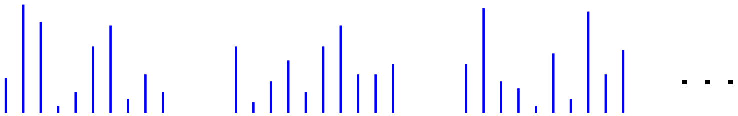
Find the position with minimum value:



$G \Rightarrow$  $\ldots$

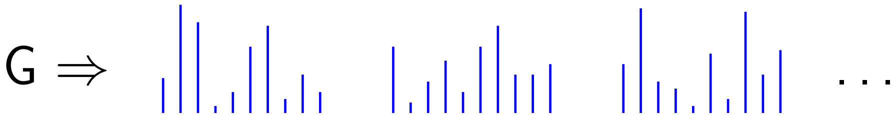
$G \Rightarrow$  $\ldots$

Find the position that will have low value in the future.

$\mathsf{G} \Rightarrow$ 

Find the position that will have low value in the future.

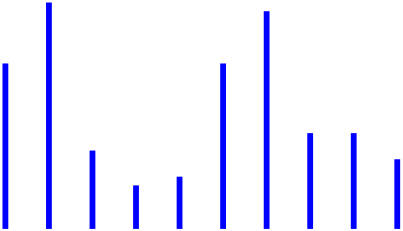
- This is the classical training/testing scenario



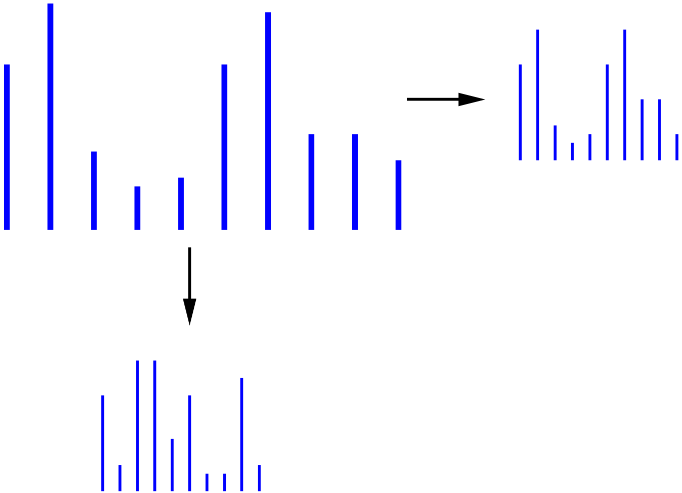
Find the position that will have low value in the future.

- This is the classical training/testing scenario
- Lots of approaches exist, but:
  - Need an expert
  - No self-assessment

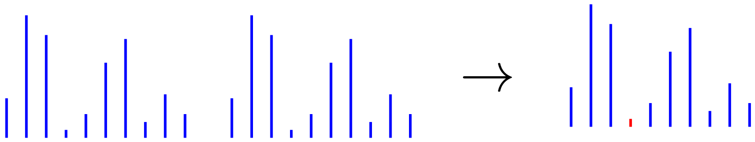
Single instance

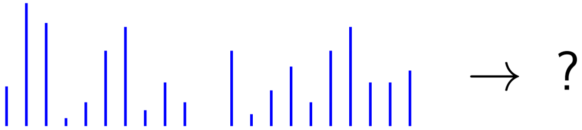


Single instance



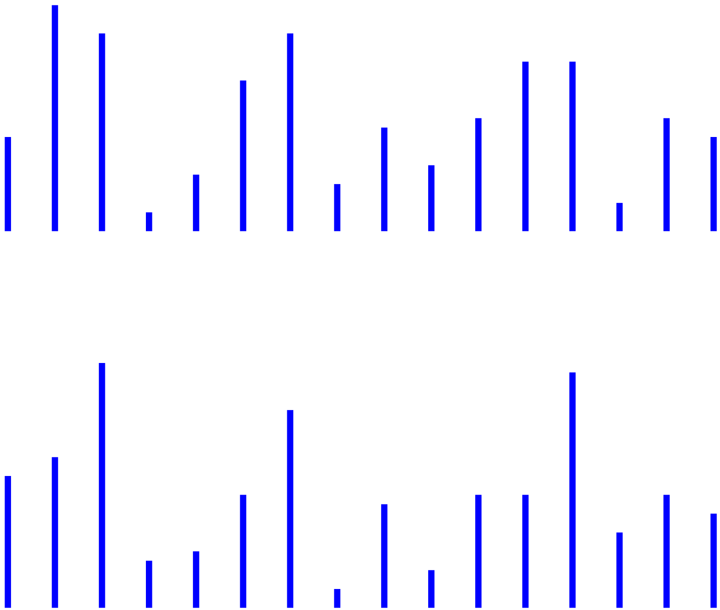
Two instances

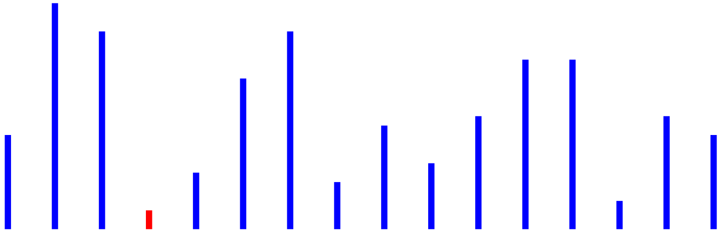


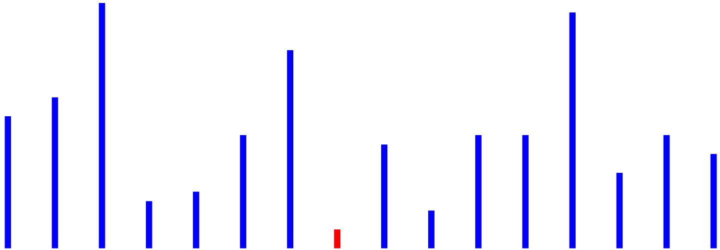


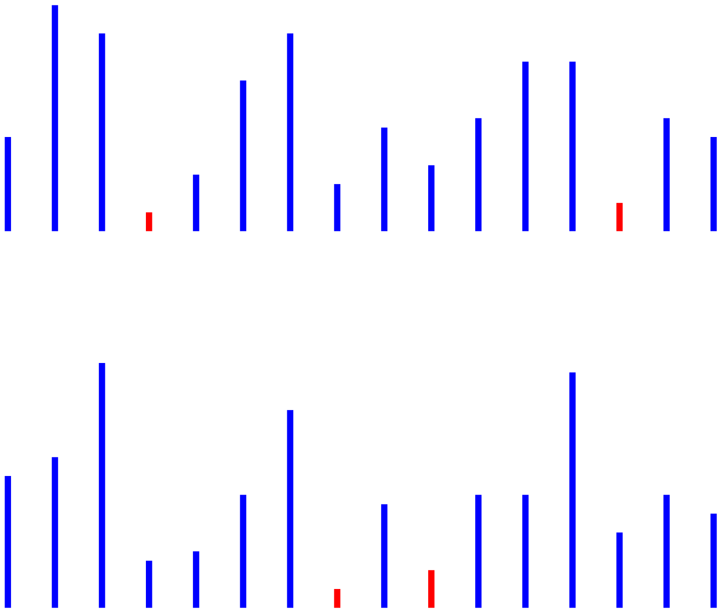
Problem

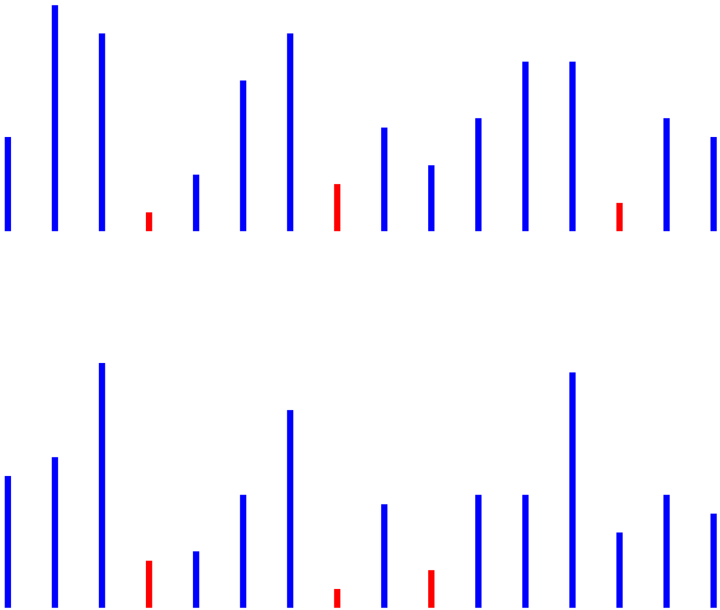
- Given two instances of a problem P, find a generalizing solution
- Related: What is the similarity of the two instances?

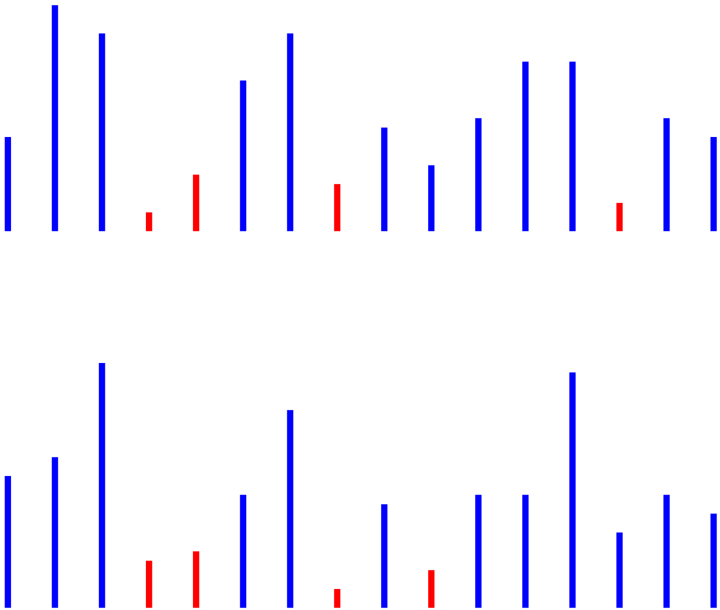


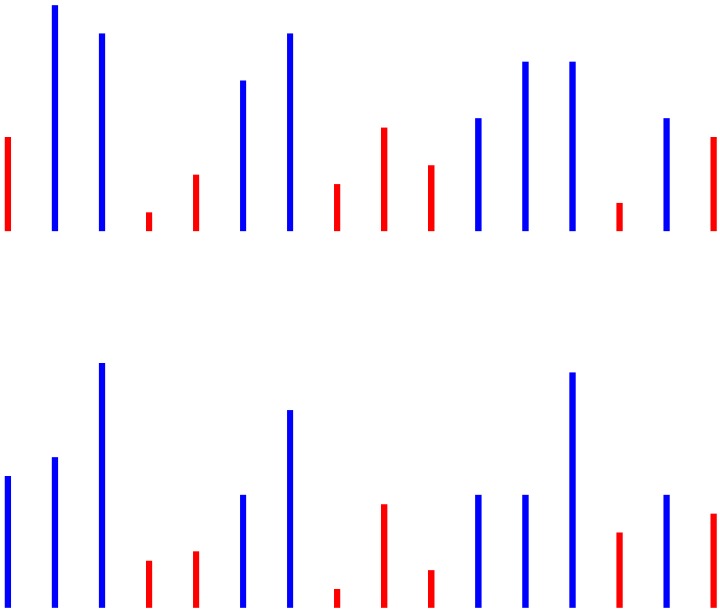


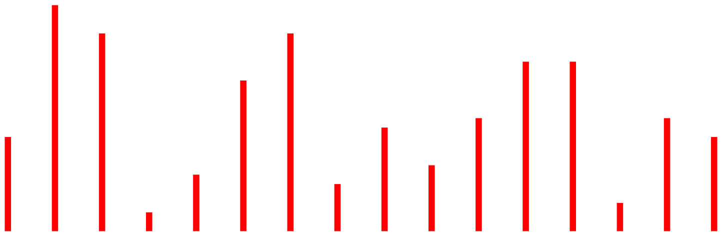


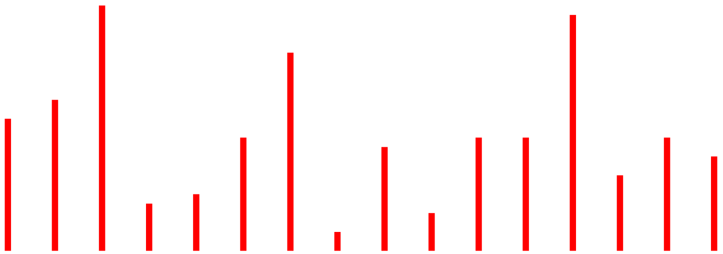












Definition

We call the set of all *approximate solutions* that are not more than ρ times worse than the optimal solution a *ρ–approximation set* and denote it by $A_\rho(I)$.

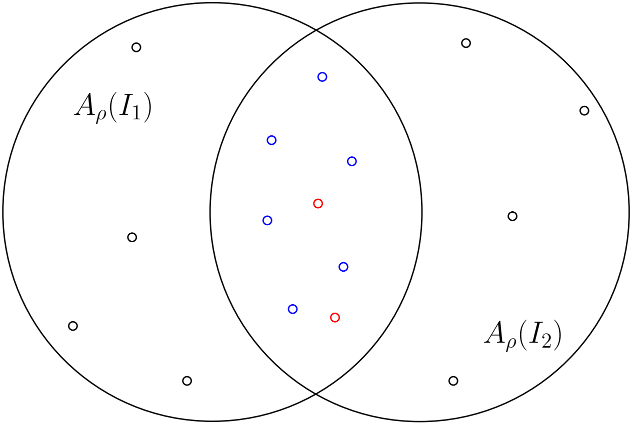
We will consider the intersection of the ρ–approximation sets and choose a solution from there. But which ρ to choose?

Definition

We call the set of all *approximate solutions* that are not more than ρ times worse than the optimal solution a *ρ–approximation set* and denote it by $A_\rho(I)$.

We will consider the intersection of the ρ–approximation sets and choose a solution from there. But which ρ to choose?

Intuition: Choose the ρ for which the intersection is most particular – compared to the case when the instances are not similar.



$$P(\text{a good solution is chosen}) = \frac{g}{g+b}$$

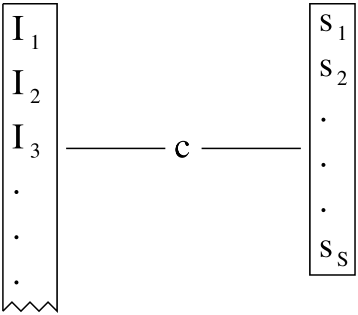
$$\arg\max_{\rho} \frac{g}{g+b} = \arg\min_{\rho} \frac{g+b}{g} = \arg\min_{\rho} \frac{b}{g} =$$

$$= \arg\max_{\rho} \frac{g}{b} = \arg\max_{\rho} \frac{g+b}{b} =$$

$$= \arg\max_{\rho} \frac{|A_{\rho}(I_1) \cap A_{\rho}(I_2)|}{\mathbb{E}|A \cap B|} = \hat{\rho}$$

$$\frac{|A_{\hat{\rho}}(I_1) \cap A_{\hat{\rho}}(I_2)|}{\mathbb{E}|A \cap B|} = U \quad \text{– Unexpected similarity}$$

General combinatorial optimization problem



$c(s_i, I_j)$ – cost of the solution s_i for instance I_j

(just the same as the minimum-in-an-array example)

Framework

We need:

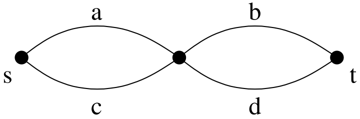
1. Algorithm that counts approximate solutions
2. Algorithm that counts the size of the intersection of the sets of approximate solutions
3. Way to calculate the expected intersection
4. Way to determine $\hat{\rho}$
5. Way to sample from $A_{\hat{\rho}}(I_1) \cap A_{\hat{\rho}}(I_2)$

Expected intersection

- We calculate the expectation over all approximation sets of the same size as $A_\rho(I_1)$ and $A_\rho(I_2)$.

Expected intersection

- We calculate the expectation over all approximation sets of the same size as $A_\rho(I_1)$ and $A_\rho(I_2)$.
- But not all subsets of solutions form a feasible approximation set:



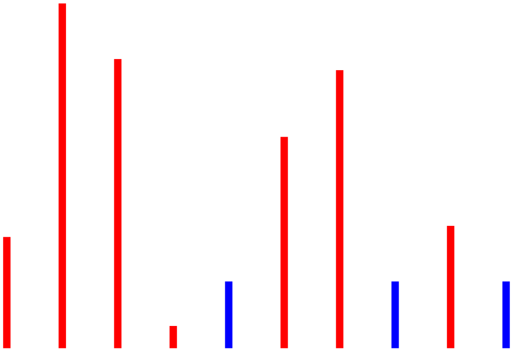
$$a + b < t$$
$$c + d < t$$
$$a + d > t$$
$$c + b > t$$

Some results

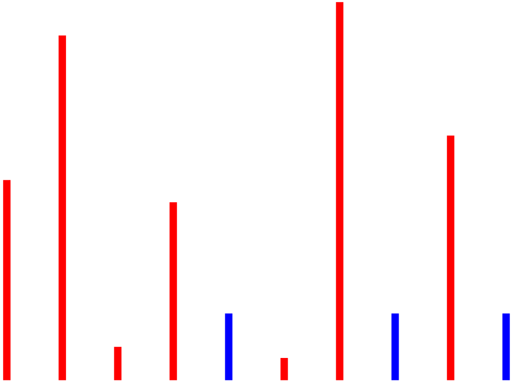
Tests on different instances of different problems (maximum sub-array sum, minimum cut, shortest path, spanning tree, ...) show:

- Averaging $<$ small intersection $<$ optimal intersection
- High unexpected similarity helps all algorithms
- In instances with high randomness, approaches are almost equivalent, but in tests with real-world data, we can extract better solutions

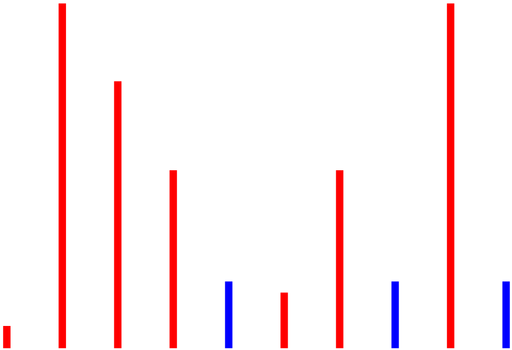
Example of a generator



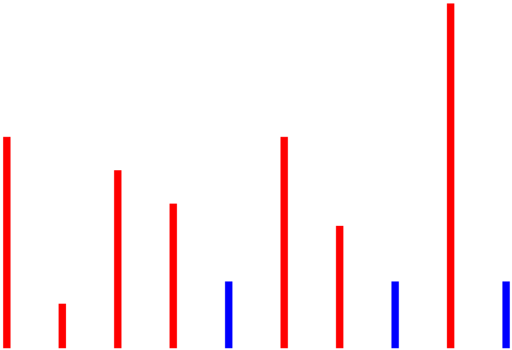
Example of a generator



Example of a generator



Example of a generator



What's ahead?

- Are all solutions in the intersection equal?
- Will more inputs lead to better results?
- Are problems with exponential solution numbers always difficult?
- Where can we exploit problem-based instance similarity?

Thank you!