Online Function Tracking with Generalized Penalties

<u>Marcin Bieńkowski</u>
Institute of Computer Science,
University of Wrocław, Poland

Stefan Schmid
Deutsche Telekom Laboratories / TU Berlin
Germany

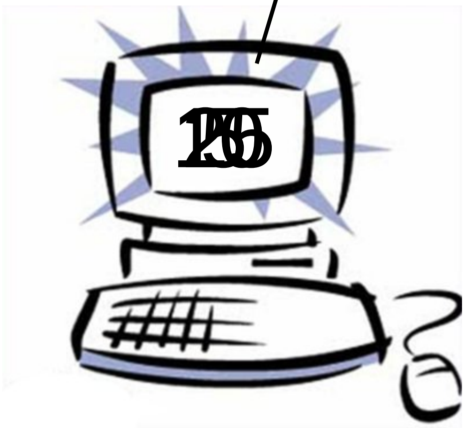
Let’s start with an example

round 4

Knows the measurement
with absolute error |20-25| = 5

measures: 20

Sends update (25)



Sensor

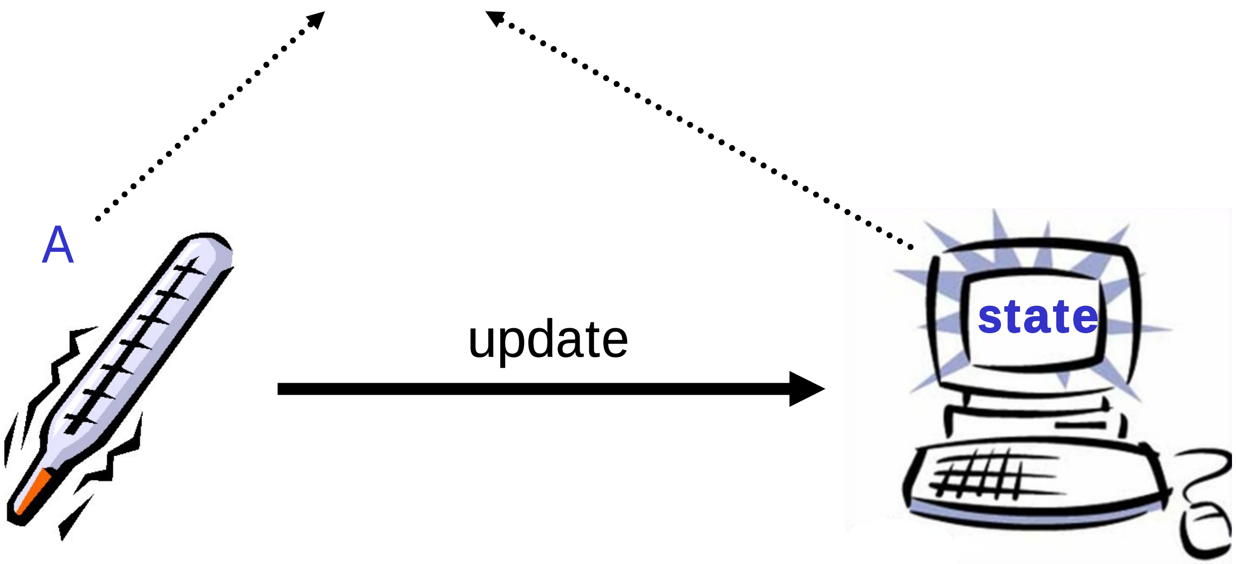
A cost is associated both with
sending updates and inaccuracies

The model and notation

Algorithm's state = the value stored at base station

<u>In one round:</u>

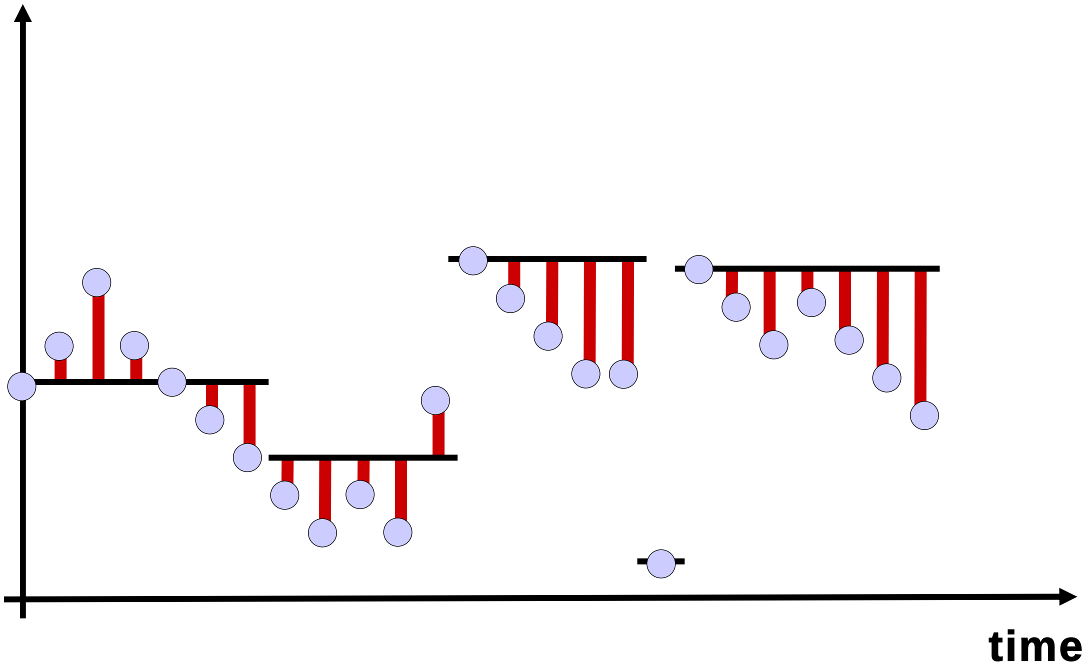
1. New integer value A is observed at sensor node.
2. Sensor may send an update to base station = change state for cost C.
3. We pay penalty Ψ (|A – state|). Ψ is called penalty function.



Note from one reviewer

I don't understand the story behind this model…

… but the problem in this paper is to approximate an arbitrary gradually revealed function by a piecewise constant function



1-lookahead model:

1. observe
2. change state
3. pay for inaccuracies

The goal

Input: sequence σ of observed/measured integral values

Output: schedule of updates

Cost: the sum of update and penalty costs.

Goal: Design effective online algorithms for an arbitrary input.

Online: we do not know the future measurements.

Effective: ALG is R-competitive if for all σ, $C_{\mathrm{ALG}}(\sigma) \leq R \cdot C_{\mathrm{OPT}}(\sigma) + \alpha$

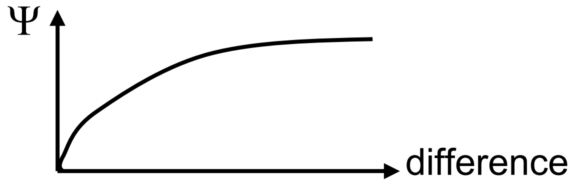
OPT = optimal offline algorithm

R = competitive ratio, subject to minimization.

Our contributions

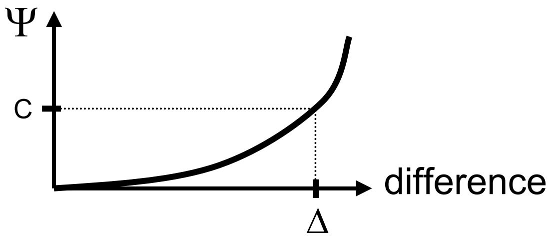
Competitive algorithms for various penalty functions.

- When Ψ is concave:



deterministic $\mathcal{O}\left(\frac{\log C}{\log \log C}\right)$ -competitive algorithm.

- When Ψ is convex:



deterministic $\mathcal{O}(\log \Delta)$-competitive algorithm.

- Matching lower bounds for randomized algorithms.

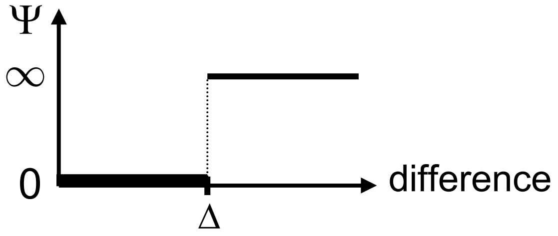
Related results

- Variant of the data aggregation.

- If observed values change monotonically, the problem is a variant of the TCP acknowledgement problem
  $\Rightarrow$ O(1)-competitive solutions exist.

- Yi and Zhang [SODA 2009]: $\mathcal{O}(\log \Delta)$-competitive algorithm for

$$\Psi(x) = \begin{cases} 0 & x \leq \Delta \\ \infty & x > \Delta \end{cases}$$



(their approach works also for multidimensional case)

This talk

This talk: only for the case $\Psi(x) = x$

- When Ψ is concave:

  deterministic $\mathcal{O}\left(\frac{\log C}{\log\log C}\right)$-competitive algorithm.

- When Ψ is convex:

  deterministic $\mathcal{O}(\log \Delta)$-competitive algorithm.

A simplifying assumption

Whenever algorithm in state V observes value A

$$\Psi(|V-A|) \leq C$$

Algorithm MED for concave penalties

,,Accumulate-and-update'' paradigm

One phase of MED

- Algorithm's state = v_{MED}

- Wait till the accumulated penalties (in the current phase) would exceed C and then ...

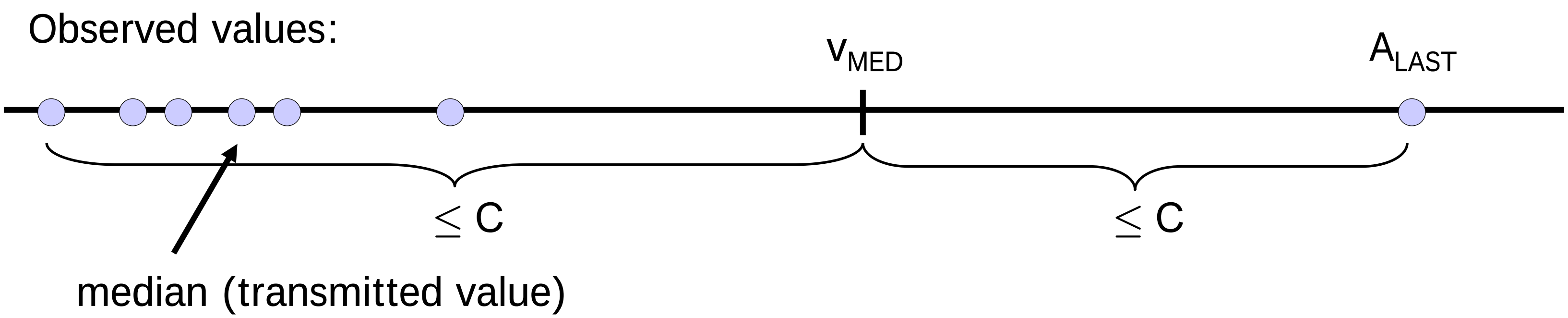
... change state to v'_{MED} = the median of values observed in this phase.

Analysis of MED for $\Psi(x) = x$ (3)

Lemma A: In a phase, MED pays O(C).

Proof:

- Penalties for all rounds but the last one $\leq$ C.
- Sending update in the last round = C
- Penalty for the last round = $|v'_{MED} - A_{LAST}|$



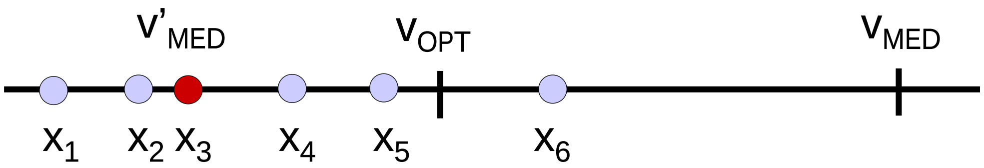
Penalty for the last round $\leq$ 2C

Analysis of MED for $\Psi(x) = x$ (2)

Lemma B: Fix phase P. Assume that OPT does not update in P. Then,

$$\frac{|v'_{\mathrm{MED}} - v_{\mathrm{OPT}}|}{|v_{\mathrm{MED}} - v_{\mathrm{OPT}}|} \leq 2\alpha/(C-\alpha) \text{ , } \quad \text{where } \alpha = C_{\mathrm{OPT}}(P)$$

Proof:



n = 6 observed values

$$\frac{n}{2} \cdot |v'_{\mathrm{MED}} - v_{\mathrm{OPT}}| \leq \sum_{i=1}^{n/2} |x_i - v_{\mathrm{OPT}}| \leq \alpha$$

$$n \cdot |v_{\mathrm{MED}} - v_{\mathrm{OPT}}| \geq \sum_{i=1}^{n} |v_{\mathrm{MED}} - x_i| - \sum_{i=1}^{n} |v_{\mathrm{OPT}} - x_i| \geq C - \alpha$$

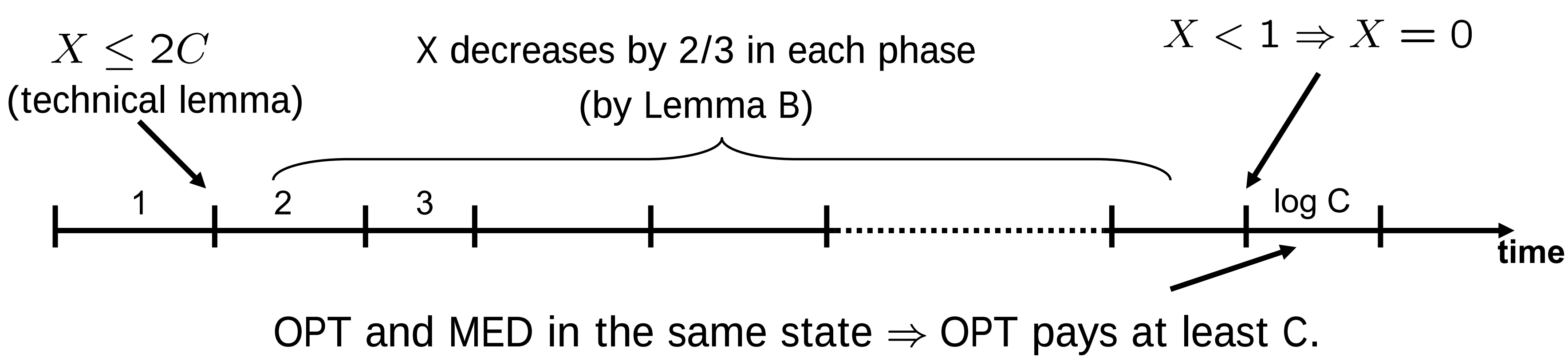
Analysis of MED for $\Psi(x) = x$ (3)

Theorem: MED is O(log C)-competitive

More carefull analysis shows competitive ratio of O(log C / log log C)

Proof: Fix an epoch E of a

- By Lemma A, $C_{MED}(E)$
- To show: there exists phase P, s.t., $C_{OPT}(P) \geq C/4$. ✓
  Assume the contrary ⇒ OPT remains at one state in E.
  Let X = $|v_{MED} – v_{OPT}|$.



Algorithm SET for convex penalties (1)

Same *accumulate-and-update* paradigm as for MED, i.e., in one phase it remains at one state, v_{SET}, till the accumulated penalties would exceed C and then changes state to v'_{SET}.

- For any phase, SET pays O(C) (same argument as for MED)

- How to choose v'_{SET} to build a lower bound for OPT?

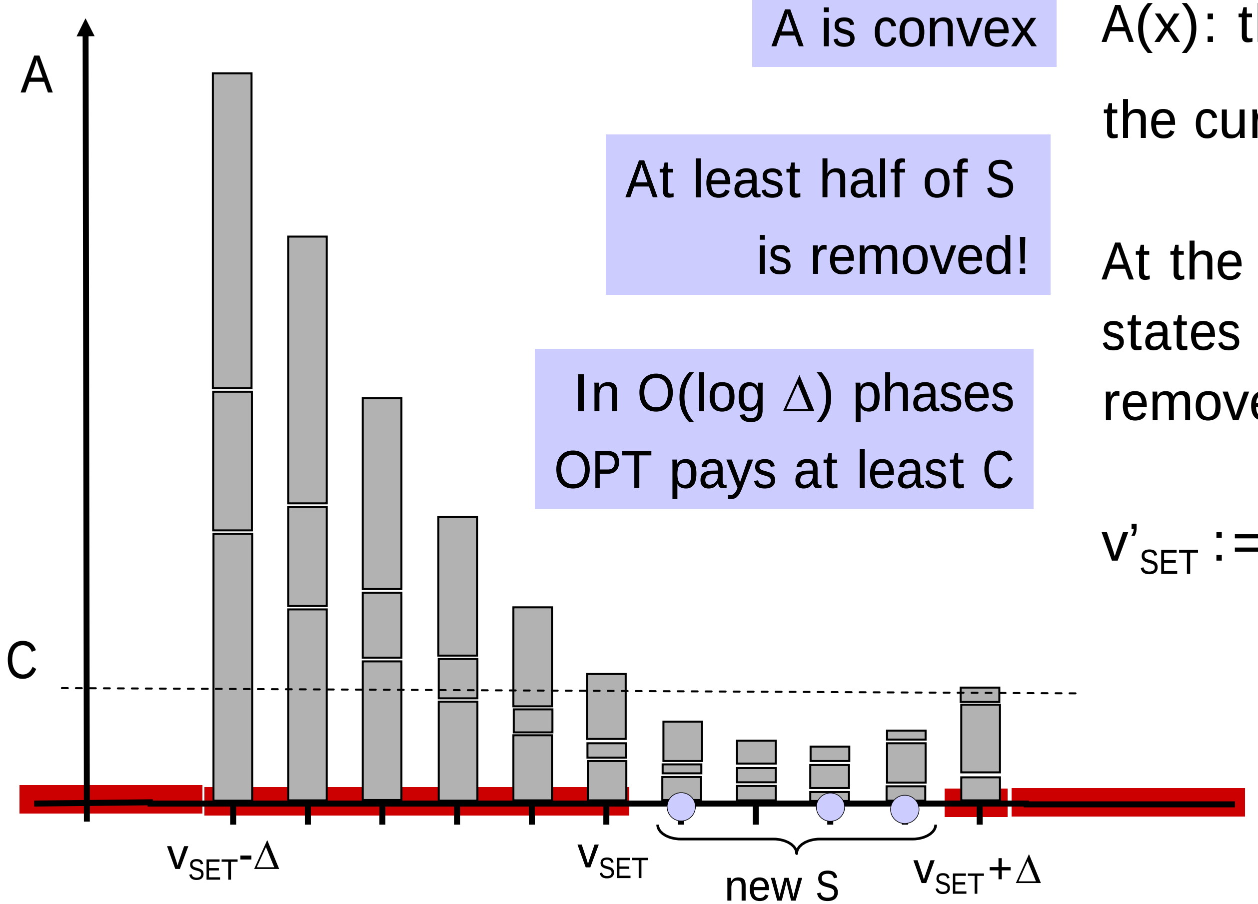
  1. If OPT changes state it pays C

  2. We have to assure that staying in a fixed state is expensive

Algorithm SET for convex penalties (2)

At the beginning, SET changes state to the first observed value.

S = [$v_{SET} - \Delta$, $v_{SET} + \Delta$] = set of *good* states



A(x): the sum of penalties in the current phase if state = x.

At the end of the phase: states { x : A(x) > C } are removed from S.

v'_{SET} := center of new S

Final remarks

- When Ψ is concave: deterministic $\mathcal{O}\left(\frac{\log C}{\log\log C}\right)$ -competitive algorithm

Presented case Ψ(x) = x, but only triangle inequality of Ψ was used.

Same algorithm works for:

• Concave Ψ

• Ψ where triangle inequality holds approximately, e.g., $\Psi(x) = x^d$

- When Ψ is convex: deterministic $\mathcal{O}(\log \Delta)$-competitive algorithm.

Possible to add "hard" constraints "the difference between measured value and state must not be larger than T".

- Matching lower bounds for randomized algorithms.

Randomization does not help (even against oblivious adversaries)

Thank you for your attention!