Open access • Book Chapter • DOI:10.1007/978-3-540-24749-4_43

Topology Matters: Smoothed Competitiveness of Metrical Task Systems
— Source link ↗

Guido Schäfer, Naveen Sivadasan

Institutions: Max Planck Society

Topics: Metrical task system and Competitive analysis

Related papers:

- Online Computation and Competitive Analysis

- Amortized efficiency of list update and paging rules

- Average case and smoothed competitive analysis of the multi-level feedback algorithm

- Competitive Paging with Locality of Reference

- Improved bounds for online stochastic matching

Topology Matters: Smoothed Competitiveness of Metrical Task Systems

Guido Schäfer[*]   Naveen Sivadasan[†]

Borodin, Linial and Saks [6] introduced a general framework to model online problems, called *metrical task systems*. We are given an undirected and connected graph $G = (V, E)$, with node set V and edge set E, and a positive length function $\lambda : E \to \mathbb{R}^+$ on the edges of G. Let n be the number of nodes in G. We extend λ to a metric δ on G. Let $\delta : V \times V \to \mathbb{R}_0^+$ be a distance function such that $\delta(u, v)$ denotes the shortest path distance (with respect to λ) between any two nodes u and v in G. A *task* τ is an n-vector $(r(v_1), \ldots, r(v_n))$ of *request costs*. The cost to process task τ in node v_i is $r(v_i) \in \mathbb{R}_0^+ \cup \{\infty\}$. The online algorithm starts from a given initial position $s_0 \in V$ and has to service a sequence $\mathcal{S} = \langle \tau_1, \ldots, \tau_r \rangle$ of tasks, arriving one at a time. If the online algorithm resides after task τ_{t-1} in node u, the cost to service task τ_t in node v is $\delta(u, v) + r_t(v)$; $\delta(u, v)$ is the *transition cost* and $r_t(v)$ is the *processing cost*. The objective is to minimize the total transition plus processing cost.

Many well-known online problems can be formulated as metrical task systems; for example, the paging problem, the static list accessing problem and the k-server problem. One might as well consider metrical task system as a general scheduling problem. Due to its generality, the competitive ratio of an algorithm for metrical task systems is usually weak compared to the one of an online algorithm that is designed for a particular problem, such as the k-server problem.

A widely accepted measure for the performance of an online algorithm is its *competitive ratio* [11]. Let ALG$[\mathcal{S}]$ and OPT$[\mathcal{S}]$, respectively, be the cost of the online and the optimal offline algorithm on a sequence $\mathcal{S}$. For a cost minimization problem, the competitive ratio c of online algorithm ALG is defined as the supremum over all input sequences $\mathcal{S}$ of the ratio ALG$[\mathcal{S}]$/OPT$[\mathcal{S}]$.

Borodin, Linial and Saks [6] gave a deterministic online algorithm that has a competitive ratio of $2n - 1$ for every metrical task system; this algorithm is known as the *work function algorithm* and we will subsequently use WFA to refer to it. The $2n - 1$ competitive ratio of WFA is optimal. Borodin, Linial and Saks [6] and Manasse, McGeoch and Sleator [10] proved that *every* deterministic online algorithm has competitive ratio at least $2n - 1$ for any arbitrary metrical task system. We emphasize that this lower bound is proven independently of the underlying metric, i.e., it holds for any arbitrary graph G and length function λ.

[*]Dipartimento di Informatica e Sistemistica, Università di Roma "La Sapienza", Via Salaria 113, 00198 Rome, Italy. Email: schaefer@dis.uniroma1.it.

[†]Strand Genomics, 273, Sir. C. V. Raman Avenue, Rajmahal Vilas, Bangalore 560080, India. Email: naveen@strandgenomics.com.

1

It is a known fact that the competitive ratio of an online algorithm often is an overly pessimistic estimation of its actual performance in practice. Sequences that force the online algorithm into its worst case behavior might be artificial and therefore rarely occur in practice. In order to overcome the overly pessimistic viewpoint adopted in worst case analysis, Spielman and Teng [12] proposed *smoothed analysis* which can be seen as a hybrid between average case and worst case analysis. The basic idea is to randomly perturb, or *smoothen*, the input instances and to analyze the performance of the algorithm on the perturbed instances. Intuitively, the smoothed complexity of an algorithm is small if the worst case instances are isolated peaks in the instance/time space.

Based on the idea underlying *smoothed analysis*, Becchetti et al. [3] recently proposed *smoothed competitive analysis* as an alternative to (worst case) competitive analysis of online algorithms. The idea is to perturb an adversarial input sequence $\check{\mathcal{S}}$ slightly at random and to analyze the expected competitive ratio of the algorithm on the perturbed sequences. We use the notation $\mathcal{S} \leftarrow f(\check{\mathcal{S}})$ to refer to a sequence $\mathcal{S}$ that is obtained from an adversarial sequence $\check{\mathcal{S}}$ by perturbing $\check{\mathcal{S}}$ according to a smoothing distribution f. More formally, Becchetti et al. defined the *smoothed competitive ratio* c of an online algorithm ALG with respect to a smoothing distribution f as

$$c = \sup_{\check{\mathcal{S}}} \ \mathbf{E}_{\mathcal{S} \leftarrow f(\check{\mathcal{S}})} \left[\frac{\text{ALG}[\mathcal{S}]}{\text{OPT}[\mathcal{S}]} \right]. \tag{1}$$

Here, we are mainly interested in the asymptotics of the smoothed competitive ratio in the long run. That is, we restrict our attention to sequences $\check{\mathcal{S}}$ whose length exceeds a certain threshold value.

Our contribution. We use the notion of smoothed competitiveness to characterize the performance of WFA. We smoothen the request costs of each task according to an additive symmetric smoothing model. Each cost entry is smoothed by adding a random number chosen from a symmetric probability distribution f with mean zero. Therefore, on expectation each smoothed cost entry coincides with its original cost entry. Our analysis holds for various probability distributions, including the uniform, double exponential and normal distributions. We use σ to refer to the standard deviation of f.

Our analysis reveals that the smoothed competitive ratio of WFA is much better than its worst case competitive ratio suggests and that it depends on certain *topological parameters* of the underlying graph:

- n = number of nodes in G;
- $\lambda_{\min}$ = minimum edge length with respect to λ;
- $\lambda_{\max}$ = maximum edge length with respect to λ;
- Δ = maximum degree of a node in G;
- $\delta_{\max}$ = diameter of G, i.e., the maximum length of a shortest path between any two nodes; more formally, $\delta_{\max} = \max_{(u,v) \in V \times V} \delta(u,v)$;

- $e_{\max}$ = edge diameter of G, i.e., the maximum number of edges on a shortest path (with respect to the number of edges) between any two nodes; observe that $e_{\max}\lambda_{\min} \le \delta_{\max} \le e_{\max}\lambda_{\max}$.

We prove several upper bounds.

1. We show that if the request costs are chosen randomly from a distribution f, which is non-increasing in $[0, \infty)$, the expected competitive ratio of WFA is

$$O\big(1 + \tfrac{\sigma}{\lambda_{\min}} \cdot \log(\Delta)\big).$$

In particular, WFA has an expected competitive ratio of $O(\log(\Delta))$ if $\sigma = \Theta(\lambda_{\min})$. For example, we obtain a competitive ratio of $O(\log(n))$ on a clique and of $O(1)$ on a binary tree.

2. We prove two upper bounds on the smoothed competitive ratio of WFA:

$$O\bigg(\tfrac{\delta_{\max}}{\lambda_{\min}}\big(\tfrac{\lambda_{\min}}{\sigma} + \log(\Delta)\big)\bigg) \quad \text{and} \quad O\bigg(\sqrt{n \cdot \tfrac{\lambda_{\max}}{\lambda_{\min}}\big(\tfrac{\lambda_{\min}}{\sigma} + \log(\Delta)\big)}\bigg).$$

For example, if $\sigma = \Theta(\lambda_{\min})$ and $\lambda_{\max}/\lambda_{\min} = \Theta(1)$, WFA has smoothed competitive ratio $O(\log(n))$ on any graph with constant edge diameter and $O(\sqrt{n})$ on any graph with constant maximum degree. Note that we obtain an $O(\log(n))$ bound on a complete binary tree.

3. We obtain a better upper bound on the smoothed competitive ratio of WFA if the adversarial task sequence only consists of β-*elementary tasks*. A task is β-elementary if it has at most β non-zero entries. (We will use the term *elementary task* to refer to a 1-elementary task.) We prove a smoothed competitive ratio of

$$O\big(\beta \cdot \tfrac{\lambda_{\max}}{\lambda_{\min}}\big(\tfrac{\lambda_{\min}}{\sigma} + \log(\Delta)\big)\big).$$

For example, if $\sigma = \Theta(\lambda_{\min})$ and $\lambda_{\max}/\lambda_{\min} = \Theta(1)$, WFA has smoothed competitive ratio $O(\beta \log(\Delta))$ for β-elementary tasks.

We also present lower bounds. All our lower bounds hold for *any* deterministic online algorithm and if the request costs are smoothed according to the additive symmetric smoothing model. We distinguish between *existential* and *universal* lower bounds. Existential lower bounds only hold for a certain class of graphs while universal lower bounds hold for any arbitrary graph.

4. For a large range of values for $\delta_{\max}$ and Δ, we present existential lower bounds that are asymptotically tight to the upper bounds stated in 2. This means (a) that the stated smoothed competitive ratio of WFA is asymptotically tight and (b) that WFA is asymptotically optimal under the additive smoothing model—no other deterministic algorithm can achieve a better smoothed competitive ratio.

5. We also prove two universal lower bounds on the smoothed competitive ratio:

$$\Omega\big(\tfrac{\lambda_{\min}}{\sigma} + \tfrac{\lambda_{\min}}{\lambda_{\max}}\log(\Delta)\big) \quad \text{and} \quad \Omega\bigg(\min\Big\{e_{\max}, \sqrt{e_{\max} \cdot \tfrac{\lambda_{\min}}{\lambda_{\max}}\big(\tfrac{\lambda_{\min}}{\sigma} + 1\big)}\Big\}\bigg).$$

3

Assume that $\lambda_{\max}/\lambda_{\min} = \Theta(1)$. Then the first bound matches the first upper bound stated in 2 if the edge diameter $e_{\max}$ is constant, e.g., for a clique. The second bound matches the second upper bound in 2 if $e_{\max} = \Omega(n)$ and the maximum degree Δ is constant, e.g., for a line.

6. For β-elementary tasks, we prove an existential lower bound of

$$\Omega\big(\beta \cdot \big(\tfrac{\lambda_{\min}}{\sigma} + 1\big)\big).$$

This implies that the bound in 3 is tight up to a factor of $(\lambda_{\max}/\lambda_{\min}) \log(\Delta)$.

Our smoothed competitive analysis renders meaningless for metrical task systems whose tasks obey a certain combinatorial structure, e.g., for the paging problem, the k-server problem, etc. The reason for this is that our smoothing model destroys zero request costs and thus the underlying combinatorial structure of these problems. As a consequence, the smoothed task sequence cannot be interpreted in terms of the original problem. One way out of this would be to consider zero-retaining smoothing models. However, as will be addressed in the paper, these models cannot yield a smoothed competitive ratio better than $2n - 1$ for any deterministic online algorithm and independent of the underlying metric. Therefore, the general framework of metrical task systems is not suitable to investigate the smoothed competitiveness of these problems.

Nevertheless, numerous other online problems fall into the framework of metrical task systems and we therefore obtain a smoothed competitive analysis for a large class of problems. As an example, one might consider the following online problem of scheduling n jobs on m unrelated parallel machines with predefined set-up costs. Let $[k]$ denote the set $\{1, \ldots, k\}$. The time job $j \in [n]$ needs to be processed on machine $i \in [m]$ is given by its processing time $p_{j,i}$. Moreover, we have a predefined symmetric function $g : [m] \times [m] \rightarrow \mathbb{R}_0^+$, which specifies machine set-up costs. If job $j - 1$ has been processed on machine i', the cost to process job j on machine i is $g(i', i) + p_{j,i}$. We assume that $g(i, i) = 0$ for all $i \in [m]$. The goal is to find an assignment of jobs to machines such that the total set-up plus processing cost is minimized. This problem can be formulated as a metrical task system in a straight-forward way: Each machine $i \in [m]$ corresponds to a node v_i in G. We draw an edge e between nodes v_i and $v_{i'}$ of length $\lambda(e) = g(i, i')$ for all $i, i' \in [m]$, $i < i'$. The arrival of a new job j now corresponds to a task τ_j, where the request cost $r_j(v_i)$ of node v_i in G is given by $p_{j,i}$. Observe that the maximum degree of G is m and the edge diameter is 1. The above mentioned lower bound for metrical task systems implies that every deterministic algorithm for this scheduling problem has a competitive ratio of $\Omega(m)$. As opposed to this, our analysis implies that if the processing times of the jobs are perturbed randomly, the smoothed competitive ratio of WFA is $O(\log(m))$ for this problem (assuming that $\sigma = \Theta(\lambda_{\min})$ and $\lambda_{\max}/\lambda_{\min} = O(1)$). Above we defined G as the complete graph in order to capture all possible set-up functions g. We remark that depending on g, one might be able to construct a refined graph (e.g., the all-pair shortest path graph) that still reflects the set-up function g but allows to relax the condition $\lambda_{\max}/\lambda_{\min} = O(1)$ or/and even leads to an improved smoothed competitive ratio of WFA.

Related work. Several other attempts were made in the past to overcome the overly pessimistic estimation of the performance of an online algorithm by its competitive ratio. One idea was to enhance the capability of the online algorithm by allowing a limited lookahead [1, 2]. Another idea was to restrict the power of the adversary [5, 4, 9]. Yet another idea, was to use a resource augmentation model in which the online algorithm has access to more resources than the optimal offline algorithm [7]. The diffuse adversary model by Koutsoupias and Papadimitriou [8] is another attempt to refine the notion of competitiveness. In this model, the actual distribution of the input is chosen by an adversary from a known class of possible distributions.

We believe that smoothed competitive analysis is a natural alternative to adequately characterize the performance of online algorithms.

References

[1] S. Albers. On the influence of lookahead in competitive paging algorithms. *Algorithmica*, 18(3):283–305, 1997.

[2] S. Albers. A competitive analysis of the list update problem with lookahead. *Theoretical Computer Science*, 197(1–2):95–109, 1998.

[3] L. Becchetti, S. Leonardi, A. Marchetti-Spaccamela, G. Schäfer, and T. Vredeveld. Average case and smoothed competitive analysis of the multi-level feedback algorithm. In *Proceedings of the Forty-Fourth Annual IEEE Symposium on Foundations of Computer Science*, pages 462–471, 2003.

[4] M. Blom, S. O. Krumke, W. E. de Paepe, and L. Stougie. The online TSP against fair adversaries. *INFORMS Journal on Computing*, 13(2):138–148, 2001.

[5] A. Borodin, S. Irani, P. Raghavan, and B. Schieber. Competitive paging with locality of reference. *Journal of Computer and System Sciences*, 50(2):244–258, 1995.

[6] A. Borodin, N. Linial, and M. Saks. An optimal online algorithm for metrical task systems. *Journal of the ACM*, 39:745–763, 1992.

[7] B. Kalyanasundaram and K. Pruhs. Speed is as powerful as clairvoyance. *Journal of the ACM*, 47(4):617–643, 2000.

[8] E. Koutsoupias and C. Papadimitriou. Beyond competitive analysis. In *Proceedings of the Twenty-Fifth Symposium on Foundations of Computer Science*, pages 394–400, 1994.

[9] S. O. Krumke, L. Laura, M. Lipmann, A. Marchetti-Spaccamela, W. de Paepe, D. Poensgen, and L. Stougie. Non-abusiveness helps: An $O(1)$-competitive algorithm for minimizing the maximum flow time in the online traveling salesman problem. In *Proceedings of the 5th International Workshop on Approximation Algorithms for Combinatorial Optimization*, pages 200–214, 2002.

[10] M. S. Manasse, L. A. McGeoch, and D. D. Sleator. Competitive algorithms for on-line problems. In *Proceedings of the Twentieth Annual ACM Symposium on Theory of Computing*, pages 322–333, 1988.

[11] D. Sleator and R. E. Tarjan. Amortized efficiency of list update and paging rules. *Communications of the ACM*, 28:202–208, 1985.

[12] D. A. Spielman and S. H. Teng. Smoothed analysis of algorithms: Why the simplex algorithm usually takes polynomial time. *Journal of the ACM*, 51(3):385–463, 2004.