On the Competitiveness of On-Line Scheduling of Unit-Length Packets with Hard Deadlines in Slotted Time

B. Hajek[1]
Department of Electrical and Computer Engineering
and the Coordinated Science Laboratory
University of Illinois at Urbana-Champaign
e-mail: b-hajek@uiuc.edu

Abstract —

It is shown that the competitive factor for on-line scheduling of unit-length packets with hard deadlines in slotted time is in the interval $[0.5, \phi]$, where ϕ is the inverse of the golden ratio, $\phi = (\sqrt{5}-1)/2 \approx 0.618034$. Moreover, any static priority policy, that in each slot schedules a maximum value packet irrespective of deadlines, achieves competitive factor 0.5.

I. Introduction

Consider on-line scheduling of unit-length packets with hard deadlines by a single server in slotted time. The model considered in this paper consists of a single server with its associated buffer. The server multiplexes packets from several real-time input streams into a single output stream. The system is time slotted, and the unit of time is the service time of a packet. Slot k is the time interval from k to $k+1$. Packets arrive early during a time slot, are stored in the buffer, and may be scheduled in the time slot of arrival. Each packet p arrives with an integer deadline $p.d$, that is the time by which the packet must be scheduled, and a nonnegative value $p.w$. Thus, if packet p arrives in slot k, it is either scheduled in one of the slots $\{k, k+1, \ldots, p.d-1\}$ or dropped from the system by the end of slot $p.d-1$. Once scheduled or dropped, a packet leaves the system and doesn't return. For brevity, the notation $p \leftrightarrow (d, w)$ will be used to denote that a packet p has deadline d and value w.

An *arrival sequence* $\mathcal{A} = (A_0, A_1, \ldots)$ consists of disjoint sets of packets $A_0, A_1, \ldots$, with A_n denoting the set of packets arriving in slot n. A scheduling policy maps each arrival sequence into a specification of which packet, if any, is scheduled in each time slot. A scheduling policy may be idle in some time slots, even if packets are available for service. An on-line scheduling policy is a scheduling policy such that the scheduling decision for each slot n depends only on $A_0, \ldots, A_n$.

For a given sequence of arrivals $\mathcal{A}$ with a finite total number of arrivals, and a given on-line scheduling policy π, the reward received by π is the sum of the values of the packets scheduled by π. There is also a maximum possible reward associated with $\mathcal{A}$, which is the maximum reward over all possible schedules of the packets in A. The *value ratio* for π and $\mathcal{A}$ is the reward obtained by π for $\mathcal{A}$, divided by the maximum possible reward for $\mathcal{A}$. The *competitive factor* of π is obtained as the infimum of the value ratio for π and $\mathcal{A}$, as $\mathcal{A}$ ranges over all arrival sequences with a finite total number of arrivals. The optimal competitive factor η^* for the scheduling problem in this paper is the supremum of the competitive factors , taken over all on-line scheduling algorithms.

The philosophy of competitive optimality (but not under that name) arose in the area of bin packing problems [1, 2] and spread to other problems [3]. The word "competitive" in the context of on-line algorithms was introduced in [4]. Baruha et al. [5] defined and analyzed the competitive factors achievable for a common model of on-line scheduling. In the model of [5], time is not slotted, packets can have arbitrary lengths, and a partially complete task can be dropped in favor of another task. It is shown that no on-line scheduling algorithm can have a competitive factor greater than $\frac{1}{(1+\sqrt{k})^2}$, where k is the ratio of the largest to smallest value-to-length ratio for the tasks. In particular, if no upper-bound is placed on the value ratio, then no on-line algorithm has a positive competitive factor. Baruha et al. [5] in case $k = 1$ and Koren and Shasha [6] for general $k \geq 1$ provide algorithms that meet the upper bound of [5]. See the book of Buttazo [7] for related discussion including applications and a classification of scheduling algorithms.

II. A lower bound based on static priority policies

A scheduling policy is called a static priority policy if in each slot such that some packets are available to schedule, the policy schedules one of the available packets with the maximum value. Except for the fact that expired packets must be dropped, such a policy could completely ignore deadlines. On the other hand, packet deadlines could influence scheduling decisions in multiple packets all have the maximum value. Let π^{SP} denote an arbitrary static priority policy. In this section we prove the following theorem, which shows that $\eta^* \geq 0.5$.

Theorem II.1 *The competitive factor of π^{SP} is 0.5.*

Proof. It is first shown that the competitive factor of π^{SP} is less than or equal to 0.5, by consideration of the

[1]This work was supported by the National Science Foundation under grant NSF ANR 99-80544.

following arrival sequence. Let $\epsilon > 0$. In slot 0 there are two arrivals: $p_1 \leftrightarrow (1,1)$ and $p_2 \leftrightarrow (2, 1+\epsilon)$. There are no arrivals in future slots. Policy π^{SP} schedules p_2 in slot 0, and p_1 expires without being scheduled by π^{SP}. Thus, the reward for π^{SP} is $1+\epsilon$. The optimal reward for $\mathcal{A}$ is $2+\epsilon$, since p_1 can be scheduled in slot 0 and p_2 can be scheduled in slot 1. The reward ratio for π^{SP} and $\mathcal{A}$ is therefore $\frac{1+\epsilon}{2+\epsilon}$, so the competitive factor of π^{SP} is less than or equal to $\frac{1+\epsilon}{2+\epsilon}$. Since ϵ is arbitrary, the competitive factor of π^{SP} is less than or equal to 0.5. This proves half the theorem.

It remains to show that the competitive factor of π^{SP} is greater than or equal to 0.5. Suppose that π is an arbitrary scheduling policy, and $\mathcal{A}$ is an arbitrary arrival sequence with a finite total number of arrivals. Let V denote the set of packets that π schedules and let V^{SP} denote the set of packets that π^{SP} schedules. It must be shown that the rewards satisfy the inequality

$$\sum_{p \in V} p.w \leq 2 \sum_{p \in V^{SP}} p.w. \tag{1}$$

Let $V_1 = V \cap V^{SP}$, and let $V_2 = V - V_1$. Note that V is partitioned into the set of packets V_1 that both policies schedule, and the set of packets V_2 that only π schedules. Given $p \in V_2$, consider the slot that π schedules p. Since p is also available to policy π^{SP} in that same slot, policy π^{SP} must schedule some other packet $\tilde{p}$ in that same slot, such that $\tilde{p}.w \geq p.w$. Let $\tilde{V}_2$ denote the set of all such $\tilde{p}$ as p ranges over V_2. Of course $V_1 \subset V^{SP}$ and $\tilde{V}_2 \subset V^{SP}$. Therefore,

$$\begin{aligned}
\sum_{p \in V} p.w &= \sum_{p \in V_1} p.w + \sum_{p \in V_2} p.w \\
&\leq \sum_{p \in V_1} p.w + \sum_{p \in \tilde{V}_2} p.w \\
&\leq 2 \sum_{p \in V^{SP}} p.w
\end{aligned}$$

and the proof is complete. ■

The definition of competitive factor used in this paper is based on the total rewards received for arrival sequences that have a finite number of packets. An alternative definition of competitive factor of a policy π results by considering arrival sequences with arrivals in possibly an infinite number of slots, and replacing the single value ratio by the the infimum, over all $n \geq 0$, of the cumulative reward of π for slots 0 through n, divided by the maximum possible cumulative reward for slots 0 through n. Call this the *running competitive factor* . While in principle the running competitive factor of a policy can be smaller than the competitive factor , the proof of Theorem II.1 can be easily modified to show that the running competitive factor of π^{SP} is also equal to 0.5.

The following corollary obtains by taking all packets to have the same value, in which case any non-idling policy is a static-priority policy.

Corollary II.1 *For any arrival sequence and any integer n, any non-idling policy schedules at least half as many packets in slots $1, \ldots, n$ as any other policy.*

Given any threshold τ, policy V^{SP} is nonidling for the subset of packets with value greater than or equal to τ. Thus, we have a corollary of the corollary:

Corollary II.2 *For any arrival sequence, any integer n, and any threshold τ, policy π^{SP} schedules at least half as many packets with value greater than or equal to τ in slots $1, \ldots, n$ as any other policy.*

III. An upper bound

In order to prove that $\eta^* \leq c$ for some constant c, it is necessary and sufficient to show that for any on-line scheduling algorithm π, there is an arrival sequence $\mathcal{A}$ with a finite number of arrivals such that the value ratio for $\mathcal{A}$ and π is less than or equal to c. One approach to constructing such arrival sequences is to define an adversary, which for any given policy π sequentially determines the set of arrivals in each slot as a function of the previous arrivals and previous decisions of π. Successively tighter bounds are given in this section by defining a sequence of adversaries.

Adversary One

Suppose two packets arrive in slot 0: $p_1 \leftrightarrow (1,1)$ and $p_2 \leftrightarrow (2, 1+\sqrt{2})$. If π schedules p_2 in slot 0, suppose no more packets arrive. The value ratio is $(1+\sqrt{2})/(2+\sqrt{2}) = 1/\sqrt{2}$.

If π schedules p_1 in slot 0, then suppose one packet arrives in slot 1: $p_3 \leftrightarrow (2, 1+\sqrt{2})$. Note that p_3 is equivalent to p_2, and π can schedule only one of these two packets in slot 1. Intuitively, the purpose of p_3 is to make π regret its decision in slot 0. So, π can schedule either p_2 or p_3 in slot 1. Suppose no more packets arrive. The value ratio is $(1+1+\sqrt{2})/(2(1+\sqrt{2})) = 1/\sqrt{2}$.

Thus, no matter what choices π makes, the value ratio for π and the arrival sequence constructed is $1/\sqrt{2}$. Thus, the competitive factor of any on-line policy π is less than or equal to $1/\sqrt{2}$. Hence $\eta^* \leq 1/\sqrt{2}$.

Adversary Two

Suppose two packets arrive in slot 0: $p_1 \leftrightarrow (1,1)$ and $p_2 \leftrightarrow (2,2)$. If π schedules p_2 in slot 0, suppose no more packets arrive. The value ratio is $2/(1+2) = 2/3$.

Else if π schedules p_1 in slot 0, then suppose two packets arrive in slot 1: $p_3 \leftrightarrow (2,2)$ and $p_4 \leftrightarrow (3,5)$. Note that p_3 is equivalent to p_2. Policy π can schedule either p_2, p_3 or p_4 in slot 1. If π schedules p_4 in slot 1, suppose no more packets arrive. The value ratio is (1+5)/(2+2+5)=2/3.

Else if π schedules p_2, or equivalently p_3, in slot 1, suppose one packet arrives in slot 2: $p_5 \leftrightarrow (3,5)$. Then π can schedule either p_4 or p_5 (they are equivalent) in slot 2. Suppose no more packets arrive. The value ratio is (1+2+5)/(2+5+5)=2/3.

Thus, no matter what choices π makes, an arrival sequence exists for which the value ratio is $2/3$.

Adversaries One and Two are generalized below. Each time π takes a lower valued packet, in the next slot π is faced with a new larger value packet giving π yet another difficult choice.

Adversary N

Let $N \geq 1$. The previous two adversaries correspond to $N = 1$ and $N = 2$, respectively. Let $x_1, \ldots, x_N$ denote constants specified below with $1 < x_1 < \ldots < x_N$. Suppose two packets arrive in slot 0: $p_1 \leftrightarrow (1,1)$ and $p_2 \leftrightarrow (2, x_1)$. If π schedules p_2 in slot 0, suppose no more packets arrive. The value ratio is $x_1/(1 + x_1)$.

Else if π schedules p_1 in slot 0, then suppose two more packets arrive in slot 1: $p_3 \leftrightarrow (2, x_1)$ and $p_4 \leftrightarrow (3, x_2)$. Note that p_3 is equivalent to p_2, and π can't schedule both of these. Policy π can schedule either p_2, p_3, or p_4 in slot 1. If p_4 is scheduled, suppose no more packets arrive. The value ratio is $(1 + x_2)/(2x_1 + x_2)$.

Else if π schedules p_2, or the equivalent packet p_3, in slot 1, suppose two packets arrive in slot 2: $p_5 \leftrightarrow (3, x_2)$ and $p_6 \leftrightarrow (4, x_3)$. Note that p_4 and p_5 are equivalent. If π schedules p_6 in slot 2, suppose no more packets arrive. The value ratio is $(1 + x_1 + x_3)/(x_1 + 2x_2 + x_3)$.

The description for slot n, $2 \leq n \leq N-1$ is as follows. Else if π schedules p_{2n-2}, or the equivalent packet p_{2n-1}, in slot $n-1$, then two packets arrive in slot n: $p_{2n+1} \leftrightarrow (n+1, x_n)$ and $p_{2n+2} \leftrightarrow (n+2, x_{n+1})$. Note that p_{2n} and p_{2n+1} are equivalent. If π schedules p_{2n+2} in slot n, then no more packets arrive. The value ratio is $(1 + x_1 + ... + x_{n-1} + x_{n+1})/(x_1 + ... + x_{n-1} + 2x_n + x_{n+1})$.

The scenario in slot N is as follows. If π schedules p_{2N} in slot $N-1$, then no packets arrive in slot N, as already covered in the previous paragraph with $n = N - 1$. Else if π schedules p_{2N-1} or the equivalent packet p_{2N-2} in slot $N-1$, then one packet arrives in slot N: $p_{2N+1} \leftrightarrow (N+1, x_N)$. Note that p_{2N+1} is equivalent to p_{2N}. Policy π can schedule either of these packets in slot N. No more packets arrive. The value ratio is $(1 + x_1 + ... + x_N)/(x_1 + ... + x_{N-1} + 2x_N)$.

It is shown in the appendix that for any positive integer N there exists a unique choice of positive values for $x_1, ..., x_N$ such that all $N+1$ ratios identified above are equal to each other. Let θ_N denote the common value. The solutions for some small values of N are shown in Table III. As reflected in the table, it is also shown in the appendix that if ϕ denotes the inverse golden ratio, $\phi = \frac{\sqrt{5}-1}{2} \approx 0.618034$, then $\theta_N > \phi$ for all N and $\lim_{N\to\infty} \theta_N = \phi$. Thus, $\eta^* \leq \phi$. Also, for each k fixed, $x_k \to \Phi^k$ as $N \to \infty$, where $\Phi = 1/\phi \approx 1.618034$ is the golden ratio.

Table 1: Some values of θ_N and up to the first four corresponding x_i

N	θ_N	x_1	x_2	x_3	x_4
1	$\frac{1}{\sqrt{2}}$	$1+\sqrt{2}$			
2	$\frac{2}{3}$	2	5		
3	0.649693	1.85464	4.02472	10.21953	
4	0.640388	1.78078	3.56155	8.12311	20.80776
5	0.634559	1.73642	3.29387	6.96623	16.42586
20	0.618536	1.62148	2.63690	4.30838	7.09203
40	0.618038	1.61806	2.61820	4.23669	6.85613
∞	ϕ	Φ	Φ^2	Φ^3	Φ^4

IV. Discussion

This paper shows that $\eta^* \in [0.5, \phi]$. Moreover, any static priority policy π^{SP} achieves the lower bound. This gives the following twist on an old English saying: "A bird in the hand is worth at least half of two in the bush."

It remains to identify η^* exactly. In particular, it would be interesting to see if, as this author suspects, there is an algorithm with competitive factor larger than 0.5. If so, the theory of competitive optimality might steer us to something more useful than π^{SP}.

The upper bound on η^* was proved by the use of an adversary, which for any policy π, determines by a casual, sequential method a suitably bad arrival sequence for π. In principle, this method of constructing upper bounds on η^* is not necessarily tight. That is, there may indeed be a suitably bad arrival sequence for every on-line policy π, but perhaps it cannot be identified by observing π in a causal way.

The notion of competitive optimality may offer good insight for many on-line scheduling algorithms, but it by no means is a panacea. For example, the arrival sequences that make a given policy π perform poorly may be rather rare in practical settings, two algorithms with the same competitive factors may perform much differently in practical settings, and often in practice at least some a priori knowledge is available about the arrival sequence. See [8] for a general discussion of such limitations, and possible refinements to the problem formulation that could make it more practical.

For the particular case of on-line scheduling of unit-length packets with deadlines, an alternative to the use of numerical values is proposed in [9]. Multiclass packets are considered in which each packet has an M-bit class identifier. An optimality property called lex-optimality (short for lexicographic optimality) is defined for on-line scheduling policies. Lex-optimality is a hierarchical sequence of M throughput optimality properties, the jth property keying on the jth bit of the class identifiers.

V. Appendix

Let θ_N denote the bound on the competitive factor η^* yielded by Adversary N, and let ϕ denote the inverse golden ratio. The purpose of this appendix is to show that θ_N is well-defined for all positive integers N, that $\theta_N > \phi$ for all N, and that $\lim_{N\to\infty} \theta_N = \phi$.

For a given N there are $N+1$ equations in $N+1$ unknowns, $\theta, x_1, x_2, \ldots, x_N$:

$$
\begin{aligned}
x_1 &= \theta(1+x_1) \\
1 + x_1 + \ldots + x_{n-2} + x_n &= \theta(x_1 + \ldots + x_{n-2} + 2x_{n-1} + \\
&\qquad \text{for } 2 \le n \le N \\
1 + x_1 + \ldots + x_N &= \theta(x_1 + \ldots + x_{N-1} + 2x_N)
\end{aligned}
$$

Subtracting the $n-1^{th}$ equation from the n^{th} for $2 \le n \le N+1$ yields the equivalent set of equations (taking $x_0 = 1$ by definition):

$$x_1 = \theta(1+x_1) \tag{2}$$

$$\left(\frac{1-\theta}{1+\theta}\right)x_n - x_{n-1} + x_{n-2} = 0 \quad \text{for } 2 \le n \le N \tag{3}$$

$$(1+\theta)x_{N-1} = \theta x_N \tag{4}$$

Given any θ with $0 < \theta < 1$, the N equations given by (2) and (3) sequentially determine $x_1, \ldots, x_N$. Finding an overall solution reduces to finding a value of θ such that (4) is satisfied.

We prove first that there is a unique overall solution subject to the nonnegativity constraint $x_i \ge 0$ for all i and the constraint $0 < \theta_N < 1$. It is shown in addition that such solution also satisfies the strict monotonicity requirement: $1 < x_1 < \ldots < x_n$. For $n \ge 1$ and θ such that $x_{n-1}(\theta) \neq 0$ let

$$r_n(\theta) = \frac{x_n(\theta)}{x_{n-1}(\theta)}.$$

For θ such that r_{n-1} and r_n are both well-defined and finite, (3) yields

$$r_{n+1}(\theta) = \left(\frac{1+\theta}{1-\theta}\right)\left[1 - \frac{1}{r_n(\theta)}\right] \tag{5}$$

Let $a_n = \min\{\theta \ge 0 : x_1 \ge 0, x_2 \ge 0, \ldots, x_n \ge 0\}$. By this definition, $a_1 \le a_2 \le \ldots$. Note that $a_1 = 0$ and $r_1(\theta) = \frac{\theta}{1-\theta}$. By (5) and induction on n, the following is true for $1 \le n \le N$. The function r_n is strictly increasing on $[a_n, 1)$ with value 0 at a_n and limit ∞ at 1, and the value a_{n+1} is given by $r_n(a_{n+1}) = 1$. Equation (4) identifies θ_N as the unique value $\theta \in [a_N, 1)$ at which the increasing function r_N intersects the decreasing function $\frac{1+\theta}{\theta}$. This establishes uniqueness of the overall solution under the nonnegativity constraint. Also, since $\theta_N > a_n$ for $1 \le n \le N$, the solution is such that the x_i's are strictly positive. This and (2)-(4) imply that the solution also satisfies the strict monotonicity requirement.

For given θ, (3) is a set of second order difference equations with constant coefficients, so the following characteristic equation is relevant.

$$\left(\frac{1-\theta}{1+\theta}\right)z^2 - z + 1 = 0 \tag{6}$$

The roots $z_+ = z_+(\theta)$ and $z_- = z_-(\theta)$ are real and distinct for $0.6 < \theta < 1$. See Figure 1. Therefore, for such

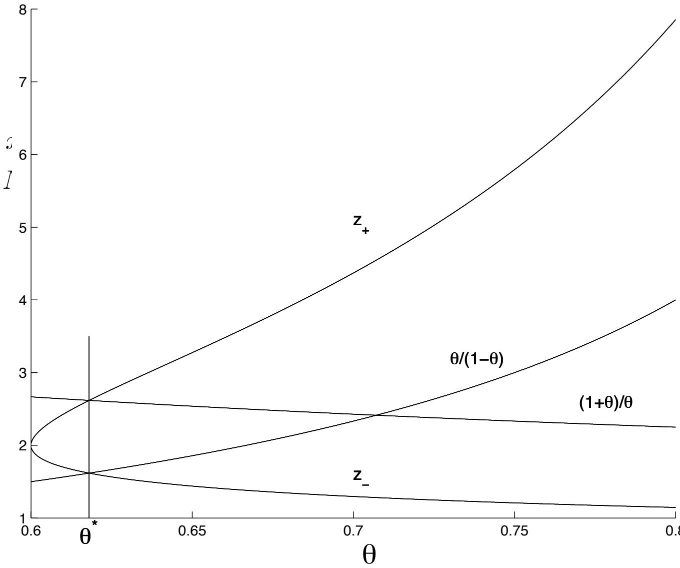

Figure 1: Sketch of z_+, z_-, and two other functions.

θ, the variables $x_1, \ldots, x_n$ can be expressed as

$$x_n = Az_+^n + (1-A)z_-^n \quad 1 \le n \le N \tag{7}$$

where $A = A(\theta)$ is a coefficient also depending on θ, selected so that (2), or equivalently the following, holds:

$$Az_+ + (1-A)z_- = \frac{\theta}{1-\theta} \tag{8}$$

The function z_+ is monotone increasing and the function z_- is monotone decreasing.

We search for a solution satisfying the additional restriction $\theta \in [\phi, 1.0)$. The decreasing function z_- intersects the increasing function $\frac{\theta}{1-\theta}$ at ϕ, and $z_+ > \frac{\theta}{1-\theta}$ for $\theta \in [0.6, 1)$. Therefore, $A(\phi) = 0$ and $0 < A(\theta) < 1$ for $\phi < \theta < 1$.

Turn finally to (4) which can be rewritten as

$$A[1+\theta-\theta z_+]z_+^{N-1} + (1-A)[1+\theta-\theta z_-]z_-^{N-1} = 0 \tag{9}$$

Begin by considering the two terms on the lefthand side of (9). The increasing function z_+ intersects the decreasing function $\frac{1+\theta}{\theta}$ at ϕ, and $z^+ \to \infty$ as $\theta \to 1$. Consequently, the first term on the lefthand side of (9) is 0 for $\theta = \phi$, it is strictly negative for $\phi < \theta < 1$, and it tends to $-\infty$ as $\theta \to 1$. The other term on the lefthand side of (9) is strictly positive and bounded over the interval $[\phi, 1]$. Hence, the lefthand side of (9) is positive for θ sufficiently near ϕ and it tends to $-\infty$ as $\theta \to 1$. Hence, a solution $\theta = \theta_N$ exists in the range $\phi < \theta_N < 1$ for each N. By the uniqueness of overall solutions such that the x_i's are nonnegative, there are no other solutions, even for smaller θ.

It is now easy to determine the limit $\lim_{N\to\infty} \theta_N$. Let $\epsilon > 0$. Then there exists $\delta > 0$ (not depending on N)

such that whenever $\theta \in [\phi + \epsilon, 1)$, the following three inequalities hold:

$$A \geq \delta \qquad 1 + \theta - \theta z_+ \leq -\delta \qquad \frac{z_+}{z_-} > 1 + \delta. \qquad (10)$$

In addition, over the same region, $z_+ > 2$. Therefore, the lefthand side of (9) tends to $-\infty$ uniformly for $\phi + \epsilon \leq \theta < 1$. Therefore, for all large N, the solutions θ_N satisfy $\phi < \theta_N < \phi + \epsilon$. Since ϵ was arbitrary, $\theta_N \to \phi$ as was to be proved.

Acknowledgments

A longer proof that the competitive ratio of π^{SP} is at least 0.5 appeared in the original draft of this paper. The author is grateful to Prof. Jay Sethuraman of Columbia University for suggesting the shorter proof included above, and for other comments.

References

[1] Graham, R.L., "Bounds for certain multiprocessing anomalies," *Bell Sys. Tech. Journal,* vol. 45, pp. 1563-1581, 1966.

[2] A.C. Yao, "New algorithms for bin packing," *Journal of the ACM*, vol 27, pp. 207–227, 1980.

[3] D. Sleator and R. Tarjan, Amortized efficiency of list update and paging rules, *Communications ACM*, vol. 28, February 1985, pages 202–208, 1985.

[4] A. Karlin, M. Manasse, L. Rudolph, and D. Sleator, Competitive snoopy caching, *Algorithmica*, vol. 3, pp. 79-119, 1988.

[5] Baruha, S., G. Koren, D. Mao, A. Raghunathan, B. Mishra, L. Rosier, D. Shasha, and F. Wang, "On the competitiveness of on-line real-time task scheduling" *Journal of Real-Time Systems*, vol. 4, pp. 124-144, 1992.

[6] G. Koren and D. Shasha, "D^{over}: an optimal on-line scheduling algorithm for overloaded real-time systems," In *Proceedings of the IEEE Real-Time Systems Symposium*, pp. 290-299, 1992.

[7] Buttazo, G. C. *Hard real-time computing systems: predictable scheduling algorithms and applications.* Kluwer Publishers, Norwell, MA., 1997.

[8] E. Koutsoupias and C. H. Papadimitriou "Beyond competitive analysis," *Foundations of Computer Science*, 1994. (Updated version available at http://www.cs.berkeley.edu/ christos/)

[9] B. Hajek and P. Seri, "Lex-optimal on-line multiclass scheduling with hard deadlines", 2000. Preprint available at www.comm.csl.uiuc.edu/hajek/