Online Scheduling of Bounded Length Jobs to Maximize Throughput

Christoph Dürr (F-Paris) Nguyen Kim Thang (F-Evry)
Łukasz Jeż (PL-Wrocław)

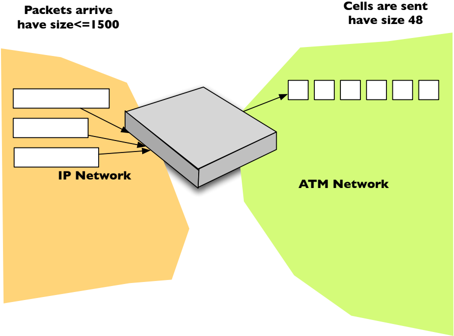
September 23, 2011

The problem

- Every time slot a new product is produced
- It cannot be stored, must be immediately delivered
- Customers arrive on-line. Customer i arrives at time r_i, and promizes to pay w_i Euros if he gets p_i units before deadline d_i (all integers)
- Goal: maximize revenu

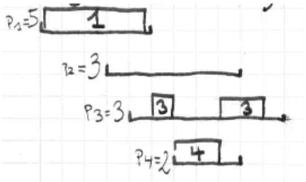
A network motivation

- Consider a gateway between an TCP/IP network and an ATM network
- IP packets arrive (r_i) with a weight (w_i=QoS), a deadline (d_i) and a length ($p_i \leq 1500$)
- ATM cells have fixed small size (48)
- IP packets that want to transite over the ATM network split into unit size cells.
- Goal: maximize total weight of packets sent on-time



A scheduling problem $1|\text{online-}r_i;\text{pmtn}|\sum w_i(1-U_i)$

- A single machine
- Jobs arrive on-line at release times
- One has to find a preemptive schedule which maximizes total weight of jobs completed on-time
- The competitive ratio of an algorithm A is $\max_I \text{OPT}(I)/A(I)$, over all instances I
- The competitive ratio of the problem is $\min_A \text{ratio}(A)$, over all on-line algorithms A



What is known?

General model

- Randomized ratio is unbounded [Koren,Shasha'94]
- For $p_i \leq k$, the deterministic ratio is $\Theta(k/\log k)$.
- For $p_i = k$, the deterministic ratio is at most 5

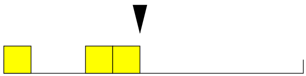
model $w_i = p_i$ (maximize processor usage)

- offline problem is NP-hard [bin packing]
- deterministic ratio is 4 [Lawler'90],[Baruah..'94]

unweighted model $w_i = 1$

- offline problem is polynomial [Lawler'90]
- randomized ratio is at most 130000 [Kalyanasundaram,Pruhs'03]
- For $p_i \leq k$, deterministic ratio is $\Omega(\log k/\log\log k)$ and $O(\log k)$

For the unweighted case deterministic ratio is $O(\log k)$



Available job

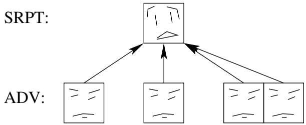
Job j is available for the algorithm at time t, if j is not completed, and $d_j - t$ does not exceed remaining work for j.
SHORTEST REMAINING PROCESSING TIME (SRPT) is the online algorithm which executes always the available job with smallest remaining work.

Let $k = \max p_j$, and $H_k := 1 + \frac{1}{2} + \frac{1}{3} + \ldots \frac{1}{k}$.

Theorem

SRPT is $2H_k$-competitif.

A charging scheme



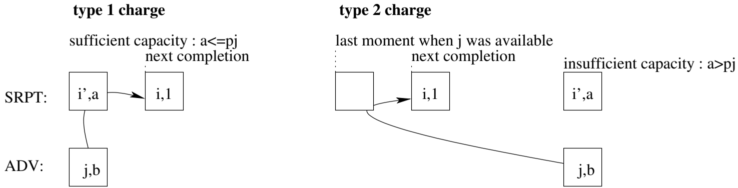
- Fix arbitrary instance. Consider schedules produced both by algorithm and by adversary.
- We denote the p_j units of a job by $(j, p_j), \ldots, (j, 2), (j, 1)$ where in (j, b), b stands for remaining work at moment of execution.
- Every unit (j, b) scheduled by the adversary is charged $1/p_j$ to some job scheduled by the algorithm.
- Every job scheduled by the algorithm will get at most $2H_k$ charges in total.

Crucial observations

If SRPT schedules (i, a) at time t,

1. then if $a > 1$, at time $t + 1$ SPRT will schedule some (i', a') with $a' < a$ since $(i, a - 1)$ is candidate.
2. SRPT will complete some job between t and $t + a$.

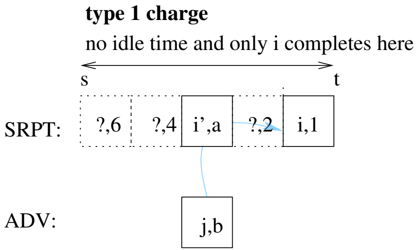
Type of charges



Let (j, b) be a unit scheduled by adversary at time t.
Let (i', a) the unit scheduled by algorithm at the same time.
We call $1/a$ the *capacity* of this unit.

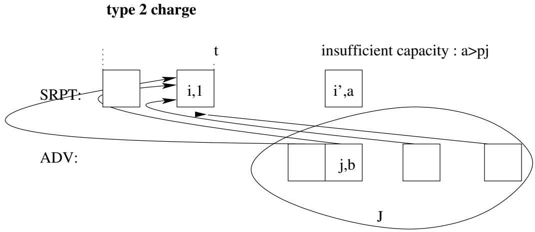
1. if the capacity is sufficient, we charge (j, b) to the next job completed by the algorithm from t on.
2. is insufficient or algorithm is *idle*, then let s be the last moment when j was available for the algorithm. We charge (j, b) to the next job completed by the algorithm from t on.

Bound type 1 charges



- Let i be a job completed by the algorithm at time t.
- Let s be the smallest time such that $[s, t)$ has no idle time, nor completions except i.
- i receives all type 1 charges through units in $[s, t)$.
- i receives at most $1/a$ through (i', a).
- the capacities of units in $[s, t)$ are strictly decreasing.
- Therefore $t - s \leq k$, and the total type 1 charge is at most H_k.

Bound type 2 charges



- Let i be a job completed by the algorithm at time t.
- Let J be the set of units type 2 charged to i.
- Every $(j, b) \in J$ is scheduled not before t
- Key observation: The ℓ-th unit $(j, b) \in J$ satisfies $d_j \geq t + \ell$, and therefore $p_j \geq \ell$ since at t j is not available anymore.
- So

$$\sum_{(j,b)\in J} \frac{1}{p_j} \leq \sum_{\ell=1}^{k} \frac{1}{\ell} = H_k.$$

□

general model (w_j arbitrary)

- No deterministic algorithm is better than $k/\ln k$-competitif.
- Schedule the available job j which maximizes the *Smith*-ratio w_j/q_j, where q_j is the remaining work j. This is $O(k)$-competitif.
- There is a more subtle algorithm with ratio $O(k/\ln k)$ where the hidden constant converges to 1 when k goes to ∞.

Summary

	offline	*randomized*	*deterministic*
general model		unbounded	
bounded proc. time	[knapsack]	$O(\log k)$ $\Omega(\sqrt{\log k/\log\log k})$	$O(k/\log k)$ $\Omega(k/\log k)$
bounded proc. time unit weight	$O(n^4)$	≤ 130000	$O(\log k)$ $\Omega(\log k/\log\log k)$
equal proc. time	$O(n^4)$		$\in [2.598, 5]$
equal proc. time unit weight	$O(n\log n)$		$\longrightarrow 1$
unit proc. time	[matching]		$\in [1.618, 1.83]$