OA, GÜ #1, 23.04.13 (I)

Online Algorithms

ALG: ~~of~~ online algorithm – obj. value: ALG

OPT: objective value of offline optimum

σ: sequence of events

goal: $\frac{ALG(\sigma)}{OPT(\sigma)} \leq c$

c: competitive ratio

The ski rental problem (rep.)

cost: * renting: 1 unit per day
* buying: D units at once

choices: * skier: when to buy
* adversary: length of sequence (# days possible to ski)

(A) Buying day one: $\Rightarrow$ cost D $\Rightarrow$ ALG = D
$\Rightarrow$ adversary: sequence length 1 $\Rightarrow$ OPT = 1
$\Rightarrow$ ratio D

(B) Buying (too) late: adversary ends sequence after skier buys (immediately!)

skier buys after x rentals

Buying at time x: $\frac{x+D}{D}$ OR $\frac{x+D}{x}$

adversary buys right away

adversary never buys

$\Rightarrow$ pick x that minimizes $\max\left\{\frac{x+D}{D}, \frac{x+D}{x}\right\}$

$\Rightarrow$ $x = D$

$\Rightarrow$ competitive ratio: 2

$\rightarrow$ Folie?

Somewhat similar — in fact a generalization:

The Bahncard Problem

Bahncard → 50% price ~~of~~ reduction on train tickets
└ price: 249 €

when to buy ??

<u>Definition</u>: $BP(C, \beta, T)$, the <u>(C, β, T)-Bahncard Problem</u>
with:
- $C > 0$: Bahncard costs
- $\beta \in [0,1]$: reduces price for any ticket $p \mapsto \beta p$
- $T > 0$: Bahncard valid for time T

<u>Examples</u>:
- $BP(249, \frac{1}{2}, $ ~~1~~ $1 \text{ year})$ → „Deutsche Bahn"
- $BP(D, 0, \infty)$ and each ticket 1 € → „Ski Rental Problem"

Folie

<u>goals</u>:
- optimal offline algorithm
- lower bound on any online algorithm
- online algorithm with competitive ratio of lower bound

<u>adversary</u>: gives finite sequence of <u>travel requests</u> $\sigma = \sigma_1 \sigma_2, \ldots$
Each σ_i: pair (t_i, p_i)
t_i: travel time
p_i: regular price of ticket

requests presented in chronological order: $0 \le t_1 < t_2 < \ldots$

online algorithm A: buy ticket for each travel request

BUT: A can decide to buy a Bahncard first

Bahncard bought at time t is valid in $[t, t+T)$

A's cost on σ_i:

reduced price

$$c_A(\sigma_i) = \begin{cases} \beta p_i, & \text{if } A \text{ has a valid Bahncard at time } t_i \\ p_i & \text{otherwise} \end{cases}$$

$\Rightarrow \sigma_i$ reduced request, if A already has a Bahncard

otherwise: regular request (may still buy Bahncard then and pay reduced price)

If A buys Bahncard at times $0 \le \tau_1 \le \tau_2 < \dots < \tau_k$

↳ sequence $\Gamma_A(\sigma) = (\tau_1, -, \tau_k)$ (finite!)

B-schedule of A on σ

total cost of A on σ:

$$c_A(\sigma) = |\Gamma_A(\sigma)| \cdot C + \sum_{i \ge 1} c_A(\sigma_i)$$

partial costs of interest:

$$p^I(\sigma) = \sum_{i: t_i \in I} p_i$$ „cost of all requests in I"

$$c_A^I(\sigma) = \sum_{i: t_i \in I} c_A(\sigma_i)$$ money spent by A on tickets during I

critical cost: $c_{crit} = \dfrac{C}{1-\beta}$

$p^I(\sigma) < c_{crit} \Rightarrow I$ <u>cheap</u>

$p^I(\sigma) \geq c_{crit} \Rightarrow I$ expensive

break-even point for any algorithm

- Buying BC at beginning of expensive intervale saves money in comparison to paying the regular price for all tickets in I

<u>Observation 1</u>: Let σ be a request sequence and $\Gamma_{OPT}(\sigma)$ be an optimal $\mathcal{B}$-schedule for σ. Then we can assume w.l.o.g. that

(a) OPT never buys a Bahncard at a reduced request.

(b) If I is an expensive interval of length at most T then OPT has at least one reduced request in I.

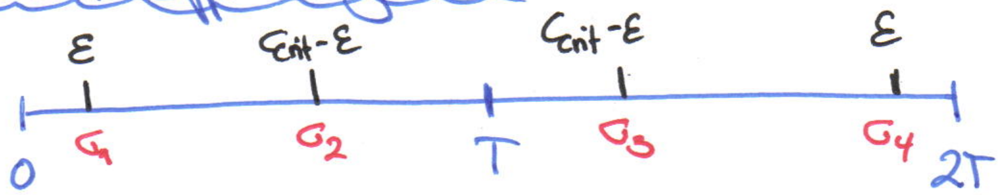
Proof: (a) Postponing the purchase of a new Bahncard until the next regular request cannot increase the cost of a $\mathcal{B}$-schedule.

(b) Otherwise, buying a Bahncard at the first sequence in I would save money, because

$$p^I(\sigma) \geq C + \beta p^I(\sigma) \Leftrightarrow p^I(\sigma) \geq \frac{C}{1-\beta}$$

for any expensive interval.

Optimal Offline algorithm



Two expensive intervals, but OPT buys at $σ_2$
(not whenever it reaches first regular request of an expensive time interval!)

Theorem 2: Given n travel requests, we can compute an optimal B-schedule and its minimal cost in time $O(n)$.

Proof: sequence $σ = σ_1 \dots σ_n$

⇒ construct weighted acyclic directed graph $G_σ$ with

nodes $s = σ_0, σ_1, \dots, σ_n, σ_{n+1} = t$

$s = (0,0)$ $\quad$ $t = (T_n + T, 0)$

two new artificial requests

properties:

* $(s \to^* t)$-paths in $G_σ$ correspond to B-schedules
* shortest $(s \to t)$-path corresponds to optimal —"—

→ for $i = 0, \dots, n$ edge

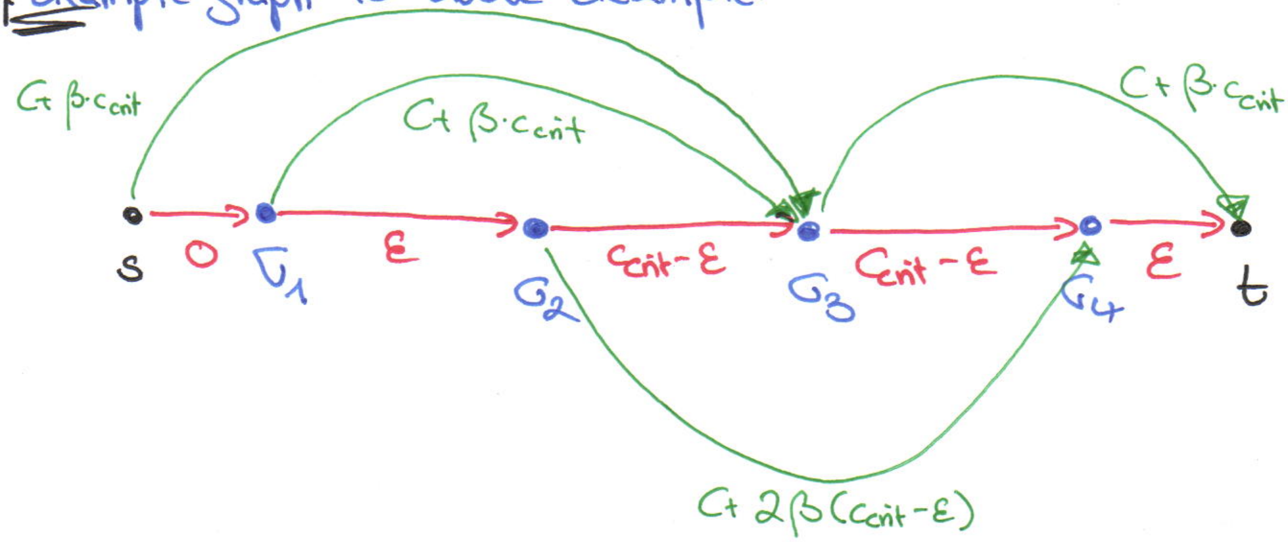
- $σ_i \to σ_{i+1}$ of weight p_i
- $σ_i \to σ_i^{+T}$ of weight q_i

$σ_i^{+T}$: first sequence after (or at) time $t_i + T$

q_i: accumulated cost of buying a BC at $σ_i$ and reduced ticket prices until it expires:

$$q_i = C + \sum_{j:\, t_i \le t_j < t_i + T} β p_j$$

Example graph to above example:



- can compute all edge weights and shortest s-t-path in