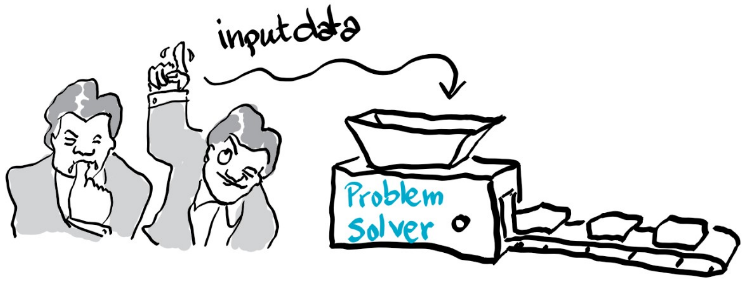
Scheduling with a Processing Time Oracle

C. Dürr
F. Dufossé
N. Nadal
D. Trystram
Ó. Vásquez

ROADEF
APRIL
2021

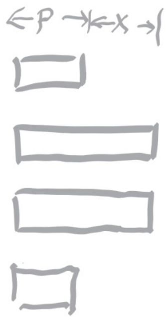
CONTEXT



OUR APPROACH
precise input data
can be queried

THE PROBLEM

single machine



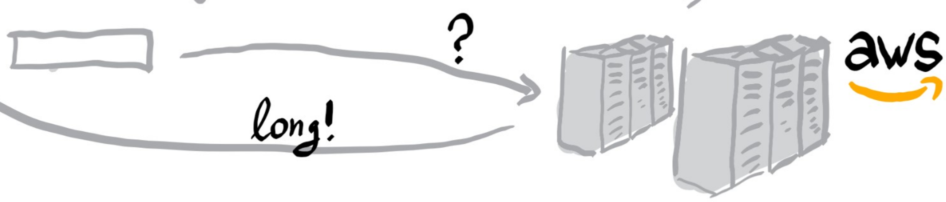
n jobs, known parameters p, x > 0

each job is
- either short : processing time p
- or long : processing time p+x

objectif :

Sum of
Completion
times

Processing time is hidden to the machine
but can be revealed for a given job
by calling a processing time oracle
(such a test lasts 1 time unit)



example

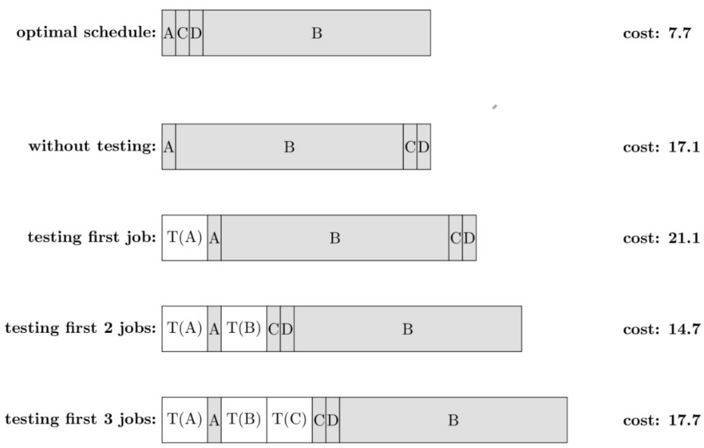


Figure 1: Some schedules with four jobs A, B, C, D.

2 EXTREME STRATEGIES

executing in order of increasing processing time is optimal

not testing any job: worst case is order of decreasing processing times

testing all jobs: permits optimal order but test caused delay.

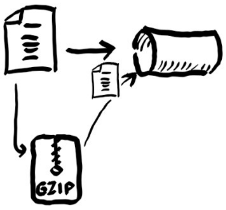
Between these extremes, what is the best strategy?

performance is measured by the COMPETITIVE RATIO $\frac{\text{ALG}}{\text{OPT}}$ = value of game played between algorithm and adversary

Our model differs from

job lengths drawn from same hidden distribution and job weights, test permits to learn these parameters [Levi, Magnanti, Shaposhnik, 2018]

test = different meaning [D. Erlebach, Megow, Meißner, 2018] [Albers, Eckl, 2020]



execute untested → proc. time $= \bar{p}_j$

execute tested → proc. time $\in [0, \bar{p}_j]$

Simplification, Dominance

dominant behavior: $\boxed{T}\boxed{P}$ tested short jobs should be executed immediately

$\boxed{T}$ $\boxed{P+X}$ tested long jobs should be postponed towards the end

Algorithm processes jobs in arbitrary order each job is either tested (T) or executed untested (E)

Strategy of algorithm $\in \{T, E\}^n$

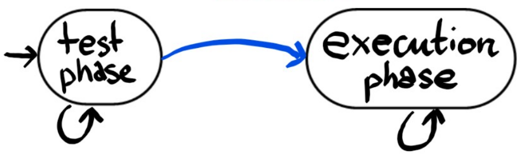
strategy of adversary $\in \{P, X\}^n$ (P short, X long)

NOTATION

CONJECTURE Optimal Algorithm consists in two phases

- test some number of jobs
- execute remaining jobs untested

$T_P T T_P \; E_X E_X E_P \; X$



Our contributions

performance measure = competitive ratio = $\max_{\text{instance}} \frac{\text{cost of alg.}}{\text{optimal cost}}$ ← doesn't need to test

= price of hidden information

input: n, p, x

non adaptive setting we can compute optimal strategy in time $O(n^2)$

adaptive setting we can compute optimal two phase strategy in time $O(n^3)$

open problem are all optimal strategies two phase?

Non Adaptive Algorithm

example n=2, p=2, x=1

	EE	ET	TE	TT
pp	$\frac{3p}{3p}=1$	$\frac{3p+1}{3p}=\frac{4}{3}$	$\frac{3p+2}{3p}=\frac{5}{3}$	$\frac{3p+3}{3p}=2$
px	$\frac{3p+x}{3p+x}=1$	$\frac{3p+x+1}{3p+x}=\frac{8}{7}$	$\frac{3p+x+2}{3p+x}=\frac{9}{7}$	$\frac{3p+x+3}{3p+x}=\frac{10}{7}$
xp	$\frac{3p+2x}{3p+x}=\frac{11}{7}$	$\frac{3p+2x+1}{3p+x}=\frac{12}{7}$	$\frac{3p+x+2}{3p+x}=\frac{9}{7}$	$\frac{3p+x+4}{3p+x}=\frac{11}{7}$
xx	$\frac{3p+3x}{3p+3x}=1$	$\frac{3p+3x+1}{3p+3x}=\frac{16}{15}$	$\frac{3p+3x+2}{3p+3x}=\frac{17}{15}$	$\frac{3p+3x+4}{3p+3x}=\frac{19}{15}$

strategies lead to schedules of the form

$(T_x)^*(T_p)^*(E_x)^*(E_p)^*$

$O(n^3)$ possible schedules

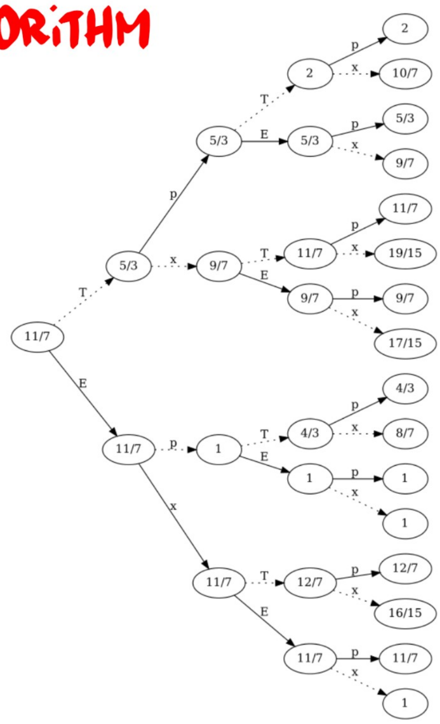
but using 2nd order analysis we can compute the equilibrium schedule and hence also the optimal strategy for the algorithm in time $O(n^2)$

we can compute the asymptotic competitive ratio, it is

$$\begin{cases} \sqrt{1+\frac{x}{p}} & \text{if } 0 \le x < 2+\frac{1}{p} \\ 1+\frac{x^2-px-1+\sqrt{\Delta}}{2px^2} & \text{if } x \ge 2+\frac{1}{p} \end{cases}$$

for $\Delta = 8p(x-1)x^2 + (1+px-x^2)^2$

Adaptive Algorithm



Compute Optimal Strategy

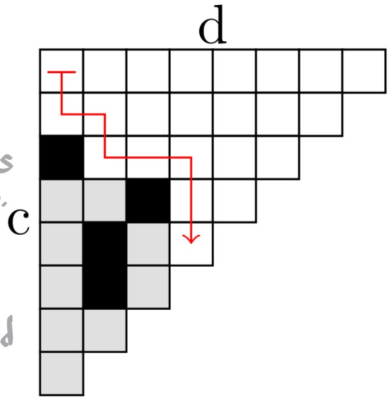
We represent interaction between algorithm and adversary by a walk on a grid

- T_p : step down
- T_x : step right
- switch to execution phase : stop walk

generates final comp. ratio

= Stop Ratio(c,d,e)

cost generated by tests



adversary chooses path
algorithm chooses where to stop

Compute optimal adversarial strategy

Iteratively :

P^* = currently best path
R^* = minimal stop ratio on P^*

marked cells = any path traversing a marked cell has min. stop ratio $\leq R^*$.
= form a combinatorial tableau

P = boundary of marked cells.
$(c,d) \in P$ cell of min. stop ratio R

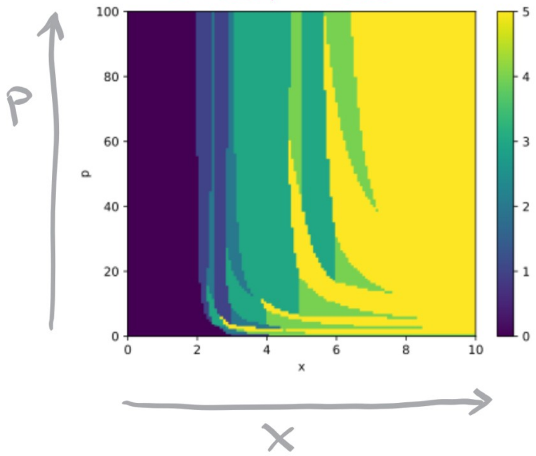
mark(c,d) and to its left and below

if $R > R^*$ then update P^* and R^*

Experiments n=6

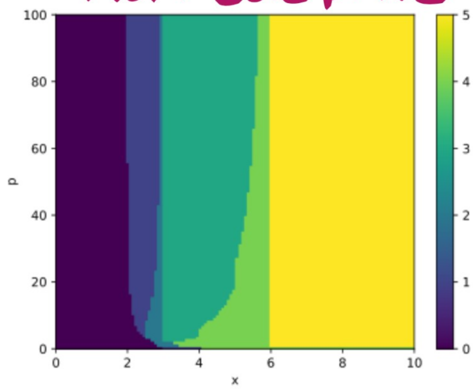
number of tests in the worst case equilibrium schedule

adaptive



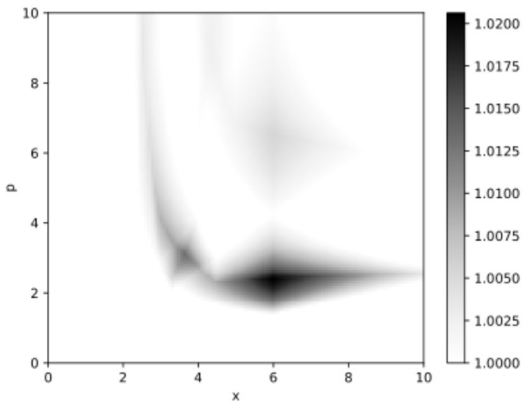
Does not provide insight to show the conjecture

non-adaptive



Regions not convex

price of nonadaptivity



not monotone in p nor in x

Conclusion

- natural next steps: randomized algorithm
  non uniform processing time domain $[\ell_j, u_j]$, testing time t_j
- develop this general framework of
  optimization under explorable uncertainty

THANK YOU FOR YOUR ATTENTION