Scheduling machines subject to unrecoverable failures and other related stochastic sequencing problems

Alessandro Agnetis
Università di Siena,
Dipartimento di Ingegneria dell'Informazione e Scienze Matematiche

Scheduling seminar, 9/11/2022

A quiz show

- A contestant is faced with a number of quizzes
- If quiz j is answered correctly, the contestant wins R_j (euros)
- The contestant can continue until he/she fails (and he/she carries home what gained so far)
- The contestant decides *in which order* should he/she answer the quizzes

[Kadane 1969]

	R_j **(€)**
1. Which movie got 14 academy awards nominations (and won 11) in 1997?	1000

	R_j (€)
1. Which movie got 14 academy awards nominations (and won 11) in 1997?	1000
2. What is the title of Leonard Cohen’s 1984 album containing *Hallelujah*?	2000

	R_j (€)
1. Which movie got 14 academy awards nominations (and won 11) in 1997?	1000
2. What is the title of Leonard Cohen’s 1984 album containing *Hallelujah*?	2000
3. What is the name of Charlie Brown’s favorite baseball player?	3000

	R_j (€)
1. Which movie got 14 academy awards nominations (and won 11) in 1997?	1000
2. What is the title of Leonard Cohen's 1984 album containing Hallelujah?	2000
3. What is the name of Charlie Brown's favorite baseball player?	3000
4. Which team won the Italian soccer championship in the year 1991?	5000

	R_j (€)
1. Which movie got 14 academy awards nominations (and won 11) in 1997?	1000
2. What is the title of Leonard Cohen’s 1984 album containing Hallelujah?	2000
3. What is the name of Charlie Brown’s favorite baseball player?	3000
4. Which team won the Italian soccer championship in the year 1991?	5000
5. What is the value of the optimal solution of Muth and Thompson’s 1963 job shop instance?	10000

A quiz show

- For each quiz j, the contestant estimates a *probability* of correctly answering the quiz j (i.e., π_j is her/his confidence in the answer)

	R_j (€)	π_j
1. Which movie got 14 academy awards nominations and won 11 in 1997?	1000	0.8
2. What is the title of Leonard Cohen's 1984 album containing *Hallelujah*?	2000	0.9
3. What is the name of Charlie Brown's favorite baseball player?	3000	0.3
4. Which team won the Italian soccer championship in the year 1991?	5000	0.7
5. What is the value of the optimal solution of Muth and Thompson's 1963 job shop instance?	10000	0.2

A quiz show

- The contestant's goal is to *maximize expected reward*
- In which order should the contestant sequence the quizzes?
- Intuitively, quizzes with high rewards and high probabilities should be chosen first, but they may not be agreeable…

Outline

- Unrecoverable interruptions
- The single-machine case
- The *m*-machine case
- Exponential failures
- Job replication
- Time-critical testing

UNRECOVERABLE INTERRUPTIONS

Uncoverable interruptions

- In certain systems, machines carrying out tasks may *fail* during the execution of a task
- A "failure" may be an actual breakdown, or simply the fact that a machine is withdrawn (without forewarning) by some higher-priority user or process (and never returned)
- The problem becomes to "take home" as much value as possible

Stochastic activities

- A set of *n* jobs (processes, activities) is given to be processed by *m* parallel machines
- There is a *probability* π_j to successfully carry out job *j*
- If an activity succeeds, a *reward* R_j is earned
- If a machine fails, the current job and all subsequently scheduled jobs on the machine are lost

The problem

- The problem is to allocate and sequence the jobs on the machines in order to *maximize expected reward*, accounting for the possibility of unrecoverable interruptions

Expected reward

- Let k jobs be scheduled on a machine, and $\sigma(j)$ be the job in j-th position
- The expected reward on that machine is given by

$$ER(\sigma) = \pi_{\sigma(1)} R_{\sigma(1)} +$$
$$\pi_{\sigma(1)} \pi_{\sigma(2)} R_{\sigma(2)} +$$
$$\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots. +$$
$$\pi_{\sigma(1)} \pi_{\sigma(2)} \ldots \pi_{\sigma(k)} R_{\sigma(k)}$$

Unreliable Jobs Scheduling Problem ($m||ER$)

Given:

- $J=\{1,\ldots,n\}$ job set
- m number of parallel (identical) machines
- π_j success probability of job j (rational)
- R_j reward for job j, if completed

Find an assignment of jobs to the machines and a sequence on each machine maximizing expected reward

SINGLE MACHINE

$1||ER$

- When m=1, the problem is to decide in which order should the jobs be sequenced
- Solved by processing the jobs by nonincreasing values of the ratios:

[Stadje 1995]

$$Z_j = \frac{R_j \pi_j}{1 - \pi_j}$$

	R_j (€)	π_j	Z_j
1. Which movie got 14 academy awards nominations (and won 11) in 1997?	1000	0.8	**4**
2. What is the title of Leonard Cohen's 1984 album containing *Hallelujah*?	2000	0.9	**18**
3. What is the name of Charlie Brown's favorite baseball player?	3000	0.3	**1.28**
4. Which team won the Italian soccer championship in the year 1991?	5000	0.7	**11.6**
5. What is the value of the optimal solution of Muth and Thompson's job shop instance?	10000	0.2	**2.5**

TWDCT $(1||\Sigma w_j(1\text{-}e^{-r\,C_j}))$

- A related problem is the minimization of *Total Weighted Discounted Completion Time*
- A set of jobs is given, each having a processing time p_j and a weight w_j
- Also, a discount factor $r>0$ is given
- The problem is to minimize

$$\sum_{j=1}^{n} w_j (1 - e^{-r C_j})$$

Relation to TWDCT

- A schedule that minimizes

$$\sum_{j=1}^{n} w_j (1 - e^{-rC_j})$$

is a schedule that maximizes

$$\sum_{j=1}^{n} w_j e^{-r(p_1 + p_2 + \ldots + p_j)}$$

Relation to TWDCT

$$\sum_{j=1}^{n} w_j e^{-r(p_1+p_2+\ldots+p_j)}$$

$$\sum_{j=1}^{n} R_j \pi_1 \pi_2 \ldots \pi_j$$

TWDCT and *UJSP*

- Therefore, *TWDCT* reduces to *UJSP* by letting

$$R_j := w_j$$

$$\pi_j := e^{-r p_j}$$

Single-machine problems

- Problem $1||\Sigma w_j(1\text{-}e^{-rC_j})$ is solved ordering the jobs by nonincreasing ratios

$$\frac{w_j e^{-rp_j}}{1-e^{-rp_j}}$$

Single-machine problems

- Problems $1||\Sigma w_j C_j$ and $1||\Sigma w_j(1-e^{-rC_j})$ are the *only* single-machine scheduling problems of the form

$$1||\Sigma\, w_j\, f(C_j)$$

which are solved by means of a simple index rule

[Rothkopf 1984]

J_j	π_j	R_j	Z_j
1	3/4	1	**3**
2	1/2	1	**1**
3	1/6	4	**4/5**

$$\begin{aligned} ER(\{1,2,3\}) &= 3/4 * 1 + \\ &\quad 3/4 * 1/2 * 1 + \\ &\quad 3/4 * 1/2 * 1/6 * 4 \\ &= 11/8 = 1.375 \end{aligned}$$

The selection problem variant

- Suppose now that we can only *select* K out of n jobs, and we still want to maximize the expected reward
- Given K selected jobs, they will then be sequenced according to nonincreasing ratios Z_j
- However, is the selection problem easy or not?

Naïve idea

- Order the jobs by nonincreasing ratios Z_j
- Take the first K jobs of the list…
- Is this the optimal choice?

J_j	π_j	R_j	Z_j
1	3/4	1	**3**
2	1/2	1	**1**
3	1/6	4	**4/5**

$K=2$

$$ER(\{1,2\}) = 3/4 * 1 +$$

$$3/4 * 1/2 * 1 =$$

$$= 3/4 * 3/2 = 9/8 = 1.125$$

J_i	π_i	R_i	Z_i
1	3/4	1	**3**
2	1/2	1	**1**
3	1/6	4	**4/5**

$K=2$

$$ER(\{1,3\}) = 3/4 * 1 + 3/4 * 1/6 * 4 = = 3/4 * 5/3 = 5/4 = 1.25$$

- It can be shown that the naïve heuristic may perform arbitrarily bad !

The selection problem variant

Optimal algorithm [Kadane 1969, Stadje 1995]:

- *Start from the empty set $S=\varnothing$;*
- <u>While</u> $|S|<K$
  - *Add to S the job which maximizes the marginal expected reward;*
  - *Insert the new job in the appropriate position in the optimal sequence*
- <u>endwhile.</u>

General selection problem

- For each job set *S* a submodular cost *c(S)* is defined, and one wants to select a set *S* in order to maximize:

total expected reward – c(S)

- Can the greedy approach be extended to this scenario?
- The idea is to recursively *add the job that maximizes the marginal net expected reward as long as it is positive, then stop*

General selection problem

- This idea yields the optimal solution if $c(S)$ only depends on the *size* $|S|$ (i.e., $c(S)=f(|S|)$ where f is concave)

[Olszewski and Vohra 2016]

- Suppose that each job has a purchasing cost c_j and

$$c(S)=\sum_{j\in S} c_j$$

- Does this greedy approach work?

j	c_j	π_j	R_j
1	9.9	0.9	11.1
2	9.89	0.5	20
3	0.11	0.9	0.22

- The optimal choice in the first step is S={2} which gives

$$0.5*20 - 9.89 = 0.11$$

- However, the optimal solution is S^*={1,3} which gives 0.19

Open problem

Given:

- $J=\{1,\ldots,n\}$ job set
- a single processing machine
- π_j success probability of job j
- R_j reward for job j, if completed
- c_j purchasing cost of job j

Select a subset of jobs and sequence them in order to maximize net expected reward

m MACHINES

$m||ER$

- The problem is strongly NP-hard even for m=2 (reduction from PRODUCT PARTITION)
- We consider simple approximation algorithms for $m||ER$

$$\rho = \frac{ER(\sigma^A)}{ER(\sigma^*)}$$

First heuristic: *Round Robin*

- A very simple heuristic is the Round-Robin heuristic (RR):
  - Sort the jobs by nonincreasing Z_j, and then schedule them on machine 1, 2, …, m, 1, 2,…,m, 1, 2,…
- What is the worst-case behavior of this heuristic?

First heuristic: *Round Robin*

- The worst-case ratio of the Round-Robin heuristic is:

$$\rho_{RR} = \frac{ER(\sigma^{RR})}{ER(\sigma^*)} \geq \frac{1}{m}$$

and the bound is tight

[A., Detti, Pranzo, Sodhi 2009]

Second heuristic: *List Scheduling*

- A more reasonable heuristic algorithm is List Scheduling (LS):
  - order the jobs by a criterion depending on jobs' parameters
  - Sequentially assign the jobs to the machines, choosing the machine on the basis of a simple criterion

List scheduling for $Pm||\Sigma w_jC_j$

- LS has been analyzed for the classical problem $Pm||\Sigma w_jC_j$
  - order the jobs by nonincreasing values of the ratio w_j/p_j (Smith's ratio)
  - the next job on the list is scheduled on the *currently least loaded machine*

List scheduling for $Pm||\Sigma w_jC_j$

- Kawaguchi and Kyan (1986) proved that

$$\rho_{LS} = \frac{z_{LS}}{z^*} \leq \frac{1+\sqrt{2}}{2} \approx 1.207...$$

and the bound is tight

→ Schwiegelshohn (2011) provided a simpler proof of this result

→ By a similar approach, we can derive an approximation result for $m||ER$

List scheduling for $m||ER$

The List Scheduling (LS) algorithm for $m||ER$ is the following:

- order the jobs by nonincreasing values of the ratio $Z_j = R_j \pi_j / (1 - \pi_j)$ and schedule them in this order
- the next job on the list is scheduled on the machine currently having the *highest cumulative probability*

List scheduling for $m||ER$

1. Given an instance I of $m||ER$, there is an instance I' such that $\rho(I') \leq \rho(I)$ and all jobs have the same value of Z_j (*uniform case*)

→ The "worst" instances are those in which all jobs have the same Z-ratio

→ Hence, any job ordering obeys the priority rule

Identical Z-ratios

- If all jobs have the *same* Z-ratio, with no loss of generality we can assume Z=1, i.e.

$$R_j = \frac{1-\pi_j}{\pi_j}$$

and a machine's contribution to expected reward simplifies to

$$ER(\sigma) = 1 - \pi_{\sigma(1)}\pi_{\sigma(2)} \cdots \pi_{\sigma(k)}$$

Identical Z-ratios

- Notice that $ER(\sigma)$ does *not* depend on the sequencing of the jobs on the machine
- The value of the objective function is therefore given by:

$$ER(\sigma) = m - \prod_{j \in M_1} \pi_j - \prod_{j \in M_2} \pi_j - \ldots - \prod_{j \in M_m} \pi_j$$

List scheduling for $m||ER$

2. We can further restrict to instances containing only two types of jobs:

-- *low-value* jobs, i.e., jobs having π_j arbitrarily close to 1

-- *high-value* jobs, i.e., jobs having $\pi_j \approx 0$

Let P_{max} be the maximum cumulative probability of a machine at the end of LS

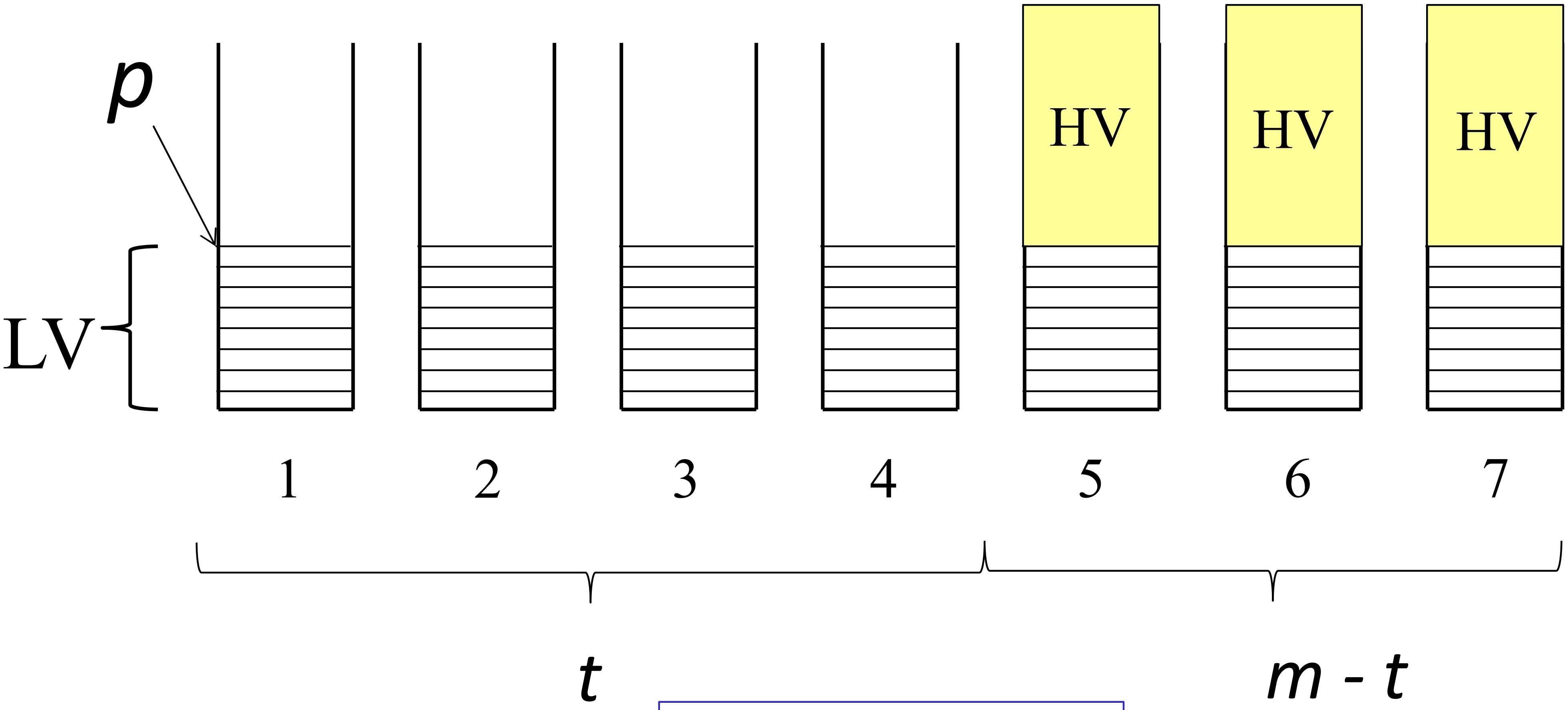
List scheduling for $m||ER$

3. Given an instance I of UJSP, there is an instance I' such that $\rho(I') \leq \rho(I)$, $Z_j=1$ for all jobs and:

a. only *low-value* jobs are scheduled as long as the cumulative probability is larger than P_{max}

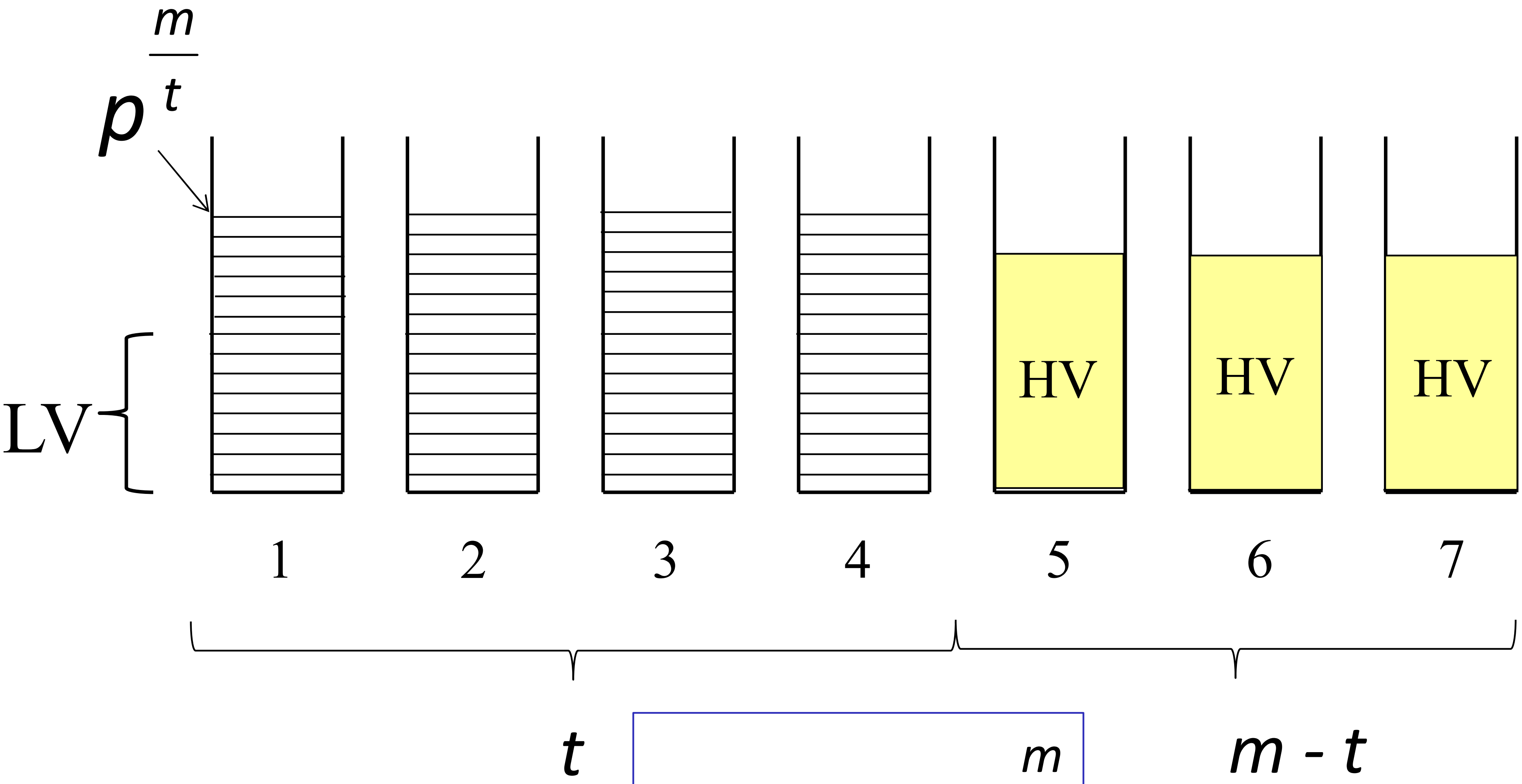
b. on some machines, a single *high-value* job is scheduled thereafter

LS solution



$$\boxed{f = m - tp}$$

Optimal solution



$$\boxed{f = m - tp^{\frac{m}{t}}}$$

List scheduling for $m||ER$

- Hence, the worst-case ratio is a function of $P_{max}=p$ and t

$$\rho = \frac{m - tp}{m - tp^{\frac{m}{t}}} = \frac{\frac{m}{t} - p}{\frac{m}{t} - p^{\frac{m}{t}}}$$

- We find the minimum for $0 \leq p \leq 1$ and $m/t \geq 1$

List scheduling for $m||ER$

m	bound
2	0.85355…
3	0.86179…
4	0.85355…
5	0.85541…
any m	**0.853196**

$$= \frac{2+\sqrt{2}}{4}$$

[A., Detti, Pranzo 2014]

[A. and Lidbetter 2020]

m=3

- The bound for m=3 is:

$$\rho = \frac{3-x}{3-x^3} = 0.86179\ldots$$

where

$$x = \frac{1}{3}\left(1-2\sin\left(\frac{\pi}{6} - \frac{1}{3}\arctan\left(\frac{4\sqrt{2}}{7}\right)\right)\right)$$

[Morandi 2022]

EXPONENTIALLY DISTRIBUTED MACHINE FAILURES ($m|\exp|ER$)

Exponential machine failures

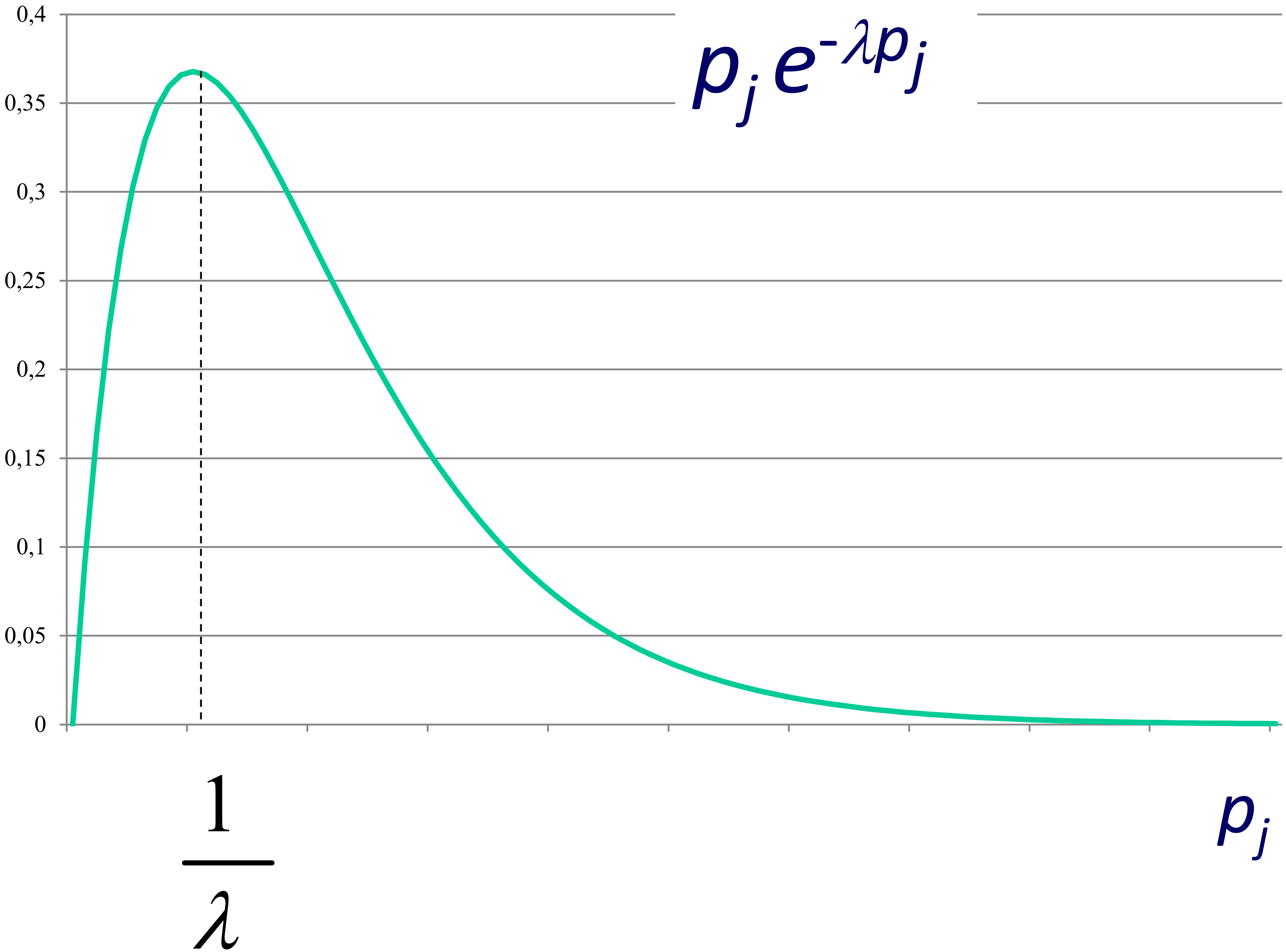
- Each job has *processing time* p_j
- The reward coincides with the *amount of work* $R_j = p_j$
- Machines break down according to an *exponential failure process* with parameter λ, i.e. MTBF = $1/\lambda$
- Hence the success probability of job *j* is

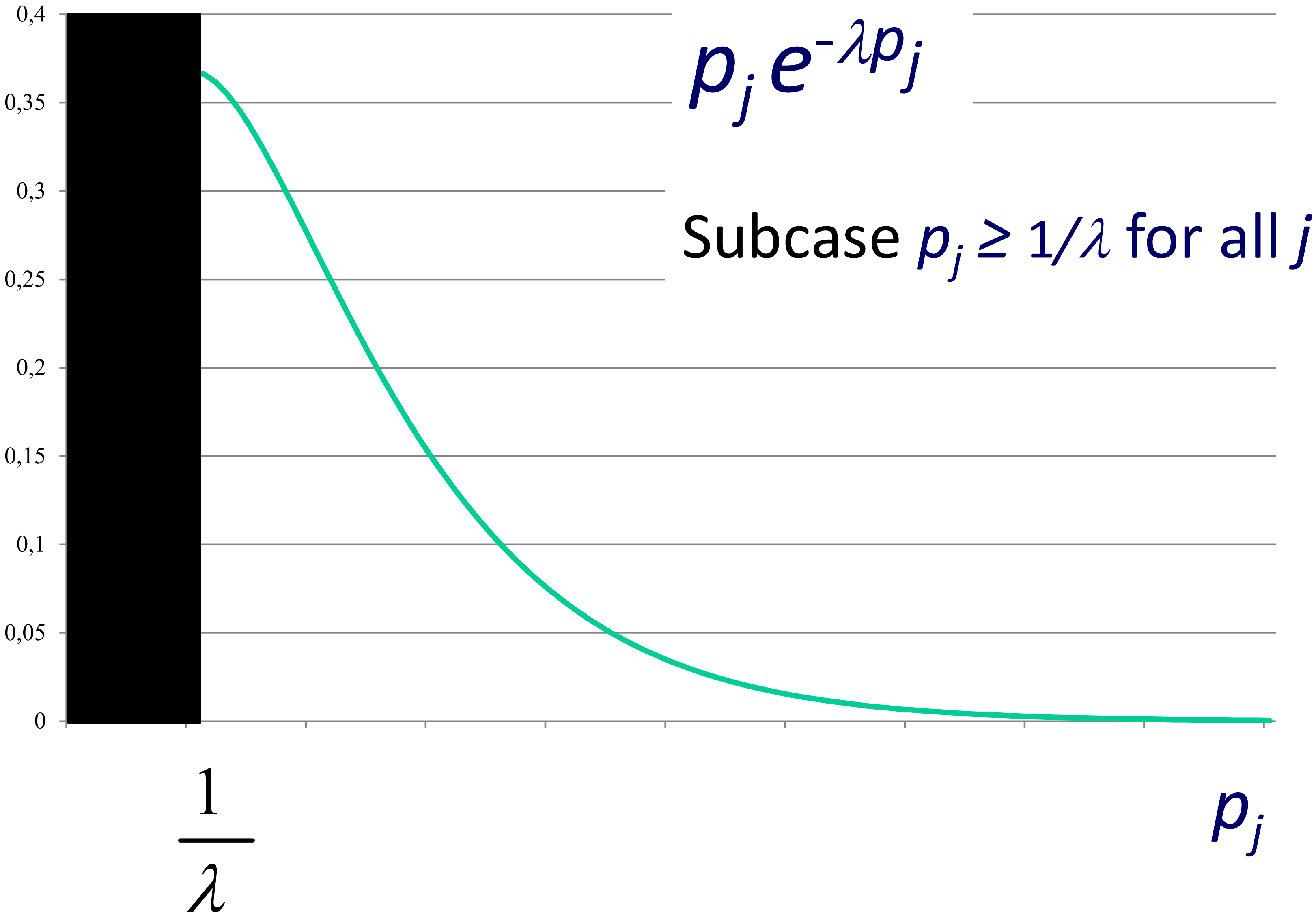
$$\pi_j = e^{-\lambda p_j}$$

Expected amount of work done

- In this case, the reward of job *j* is precisely the job length:
  - $\pi_j = e^{-\lambda p_j}$
  - $R_j = p_j$
- Hence,

$$\pi_j R_j = p_j e^{-\lambda p_j}$$


0,4
0,35
0,3
0,25
0,2
0,15
0,1
0,05
0
$p_j e^{-\lambda p_j}$
$\frac{1}{\lambda}$
p_j


0,4
0,35
0,3
0,25
0,2
0,15
0,1
0,05
0
$p_j e^{-\lambda p_j}$
Subcase $p_j \geq 1/\lambda$ for all j
$\frac{1}{\lambda}$
p_j

Subcase $p_j \geq 1/\lambda$ for all j

- In this case,

$$p_i \leq p_j \leftrightarrow p_i e^{-\lambda p_i} \geq p_j e^{-\lambda p_j} \leftrightarrow \frac{p_i e^{-\lambda p_i}}{1-p_i} \geq \frac{p_j e^{-\lambda p_j}}{1-p_j}$$

and the problem is solved by the round-robin rule, once the jobs are SPT-ordered

Not all $p_j \geq 1/\lambda$

- In the general case is the problem still solvable by means of some list scheduling algorithm?
- No. The jobs with larger $p_j\pi_j$ (candidate to being scheduled first) may also be those with smaller π_j (candidate to being scheduled last)
- The problem $2|\exp|ER$ is NP-hard

[A., Detti, Martineau 2017]

JOB REPLICATION

Job replication

- As a protection from failures, one can think of *replicating jobs* on different machines

[Benoit et al 2011, 2013]

- One copy of each job j is scheduled on each machine, and the revenue R_j is attained if *at least one copy* of job j is successfully carried out

Job replication

π_1	π_2	π_3
π_2	π_3	π_1

Prob. of achieving 1: $\pi_1 + \pi_2\pi_3\pi_1 - \pi_1^2\pi_2\pi_3$

Prob. of achieving 2: $\pi_1\pi_2 + \pi_2 - \pi_1\pi_2^2$

Prob. of achieving 3: $\pi_1\pi_2\pi_3 + \pi_2\pi_3 - \pi_1\pi_2^2\pi_3^2$

Two problems

- Sequence the *m* copies of each job on each machine in order to:

P1) maximize the expected reward (m|rep|ER)

P2) maximize the probability of having a complete kit, i.e., at least one copy of each job (*Kit Availability Maximization problem,* KAMm)

Two problems

- Sequence the *m* copies of each job on each machine in order to:

P1) maximize the expected reward (m|rep|ER)

P2) maximize the probability of having a complete kit, i.e., at least one copy of each job (*Kit Availability Maximization problem,* KAMm)

$m \mid \text{rep} \mid ER$

Given:

- $J=\{1,\ldots,n\}$ job set, m copies of each job
- m parallel (identical) machines
- π_j success probability of *a copy of* job j
- R_j reward for job j, if at least one copy is successfully completed

Find, for each machine, a sequence σ so that the overall expected reward is maximized

2|rep|*ER*

- The first (naïve) idea may be: will the Z-rule work also for 2|rep|*ER*?

$2 | \mathrm{rep} | ER$

j	R_j	π_j	Z_j
1	5	0.9	45
2	2	0.9	18
3	4	0.6	6

Identical schedule

1 (0.9)	**2 (0.9)**	**3 (0.6)**

1 (0.9)	**2 (0.9)**	**3 (0.6)**

Probability of achieving 1:

$0.9+0.9-(0.9)^2 = 0.99$

Probability of achieving 2:

$2(0.9*0.9)-(0.9*0.9)^2 = 0.9639$

Probability of achieving 3:

$2(0.9*0.9*0.6)+(0.9*0.9*0.6)^2 = 0.7358$

Reverse schedule

1 (0.9)	**2 (0.9)**	**3 (0.6)**

3 (0.6)	**2 (0.9)**	**1 (0.9)**

Probability of achieving 1:

$0.9+(0.6*0.9*0.9)-(0.6*0.9^3) = 0.9486$

Probability of achieving 2:

$0.9*0.9+0.6*0.9-(0.6*0.9^3) = 0.9126$

Probability of achieving 3:

$(0.9*0.9*0.6)+0.6-(0.9^2*0.6^2) = 0.7944$

$2|\text{rep}|ER$

j	R_j	π_j	Z_j
1	5	0.9	45
2	2	0.9	18
3	4	0.6	6

ER of the identical schedule = 9.821016

ER of the reverse schedule = 9.7458

$2|\text{rep}|ER$

j	R_j	π_j	Z_j
1	5	0.9	45
2	2	0.9	18
3	4	**0.8**	**16**

ER of the identical schedule = 10.38218

ER of the reverse schedule = 10.436

$2|\mathrm{rep}|ER$ – uniform case

- A special case is the *uniform case*, i.e., all jobs have $Z_j=1$
- In this case, *all reverse schedules are optimal*

$2|rep|ER$

- However, if all jobs have Z_j=1, except for one that has Z_K=1-ε, then the problem becomes NP-hard (ordinary sense)

  [A., Benini, Detti, Hermans, Pranzo 2022]

- Open as for strong NP-hardness

$2|rep|ER$

- A simpler problem: fix the order on M_1 and find the optimal order on M_2
- $P_j^{(1)}$: cumulative probability up to j on M_1
- The optimal sequence on M_2 is found by solving an instance of $1||ER$ with $R_j'=R_j\,(1-P_j^{(1)})$

Heuristic for 2|rep|*ER*

- In view of this result, a somewhat natural (meta)heuristic approach is:

→ for each job sequence on machine M_1, optimize on M_2

→ explore sequences on M_1 through a pairwise interchange-based tabu search algorithm

Quadratic formulation

Max $\sum_{j=1}^{n} R_j({P_j}^{(1)} + {P_j}^{(2)} - {P_j}^{(1)}{P_j}^{(2)})$

${s_{ij}}^{(1)} + {s_{ji}}^{(1)} = 1$ for all *i,j*

${s_{ij}}^{(2)} + {s_{ji}}^{(2)} = 1$ for all *i,j*

${P_j}^{(1)} \leq \pi_j$ for all *j*

${P_j}^{(2)} \leq \pi_j$ for all *j*

${P_j}^{(1)} \leq \pi_j {P_i}^{(1)} + 1 - {s_{ij}}^{(1)}$ for all *i,j*

${P_j}^{(2)} \leq \pi_j {P_i}^{(2)} + 1 - {s_{ij}}^{(2)}$ for all *i,j*

Quadratic formulation

Max $\sum_{j=1}^{n} R_j ({P_j}^{(1)} + {P_j}^{(2)} - {P_j}^{(1)} {P_j}^{(2)})$

${s_{ij}}^{(1)} + {s_{ji}}^{(1)} = 1$ for all *i,j*

${s_{ij}}^{(2)} + {s_{ji}}^{(2)} = 1$ for all *i,j*

${P_j}^{(1)} \leq \pi_j$ for all *j*

${P_j}^{(2)} \leq \pi_j$ for all *j*

${P_j}^{(1)} \leq \pi_j {P_i}^{(1)} + 1 - {s_{ij}}^{(1)}$ for all *i,j*

${P_j}^{(2)} \leq \pi_j {P_i}^{(2)} + 1 - {s_{ij}}^{(2)}$ for all *i,j*

Computational experiments

- Rewards ~ [10,100]
- Probabilities~ $I_p=[\pi_{min}, \pi_{max}]$
  [0.1,1]; [0.5,1]; [0.9,1]
- $n \in \{10, 20, 30, 40, 50\}$
- The QP has been solved by Gurobi (v.9), on a pc with Windows 10, 3.5Ghz Intel Core i7, 64GB RAM
- time limit: 20 minutes

Computational experiments

n	I_p	TS time	Avg gap TS-QP (%)	UB_{3AP} time	Avg gap UB_{QP}-UB_{3AP} (%)	Avg gap UB_{Best}-TS (%)
10	0.1-1	0.24	0,00	0,22	-14,78	0,46
20	0.1-1	3,76	0,15	1,05	20,44	16,93
30	0.1-1	11,80	0,79	1,60	32,36	22,10
40	0.1-1	27,00	2,46	4,44	35,38	26,75
50	0.1-1	55,92	3,48	13,95	44,21	26,42
10	0.5-1	0,25	0,04	0,94	-7,95	4,98
20	0.5-1	3,31	0,36	1,02	17,92	21,75
30	0.5-1	11,47	2,12	1,52	30,21	26,57
40	0.5-1	26,46	4,60	3,98	35,14	28,46
50	0.5-1	55,08	6,48	9,78	39,98	31,19
10	0.9-1	0,24	0,00	0,97	-0,46	0,85
20	0.9-1	3,64	0,08	1,07	2,57	4,13
30	0.9-1	10,84	1,18	1,51	5,24	7,82
40	0.9-1	25,81	2,83	4,81	8,37	10,53
50	0.9-1	54,68	5,22	10,66	9,97	14,52

Z-rule heuristic for $m|rep|ER$

Sequence the jobs on each machine according to the Z-rule

- Complexity $O(n \log n)$
- It can be shown that the approximation is H_m/m where H_m is the m-th harmonic number

Modified Z-rule heuristic

- Sequence M_1 according to the Z-rule (σ_1);
- Given σ_1, optimally sequence $M_2(\sigma_2)$;
- Given $\{\sigma_1,\sigma_2\}$, optimally sequence $M_3(\sigma_3)$;
- …
- Given $\{\sigma_1,\sigma_2 \ldots\sigma_{m\text{-}1}\}$, }, optimally sequence $M_m(\sigma_m)$;

Complexity is $O(n^2 \log n)$

Modified Z-rule heuristic

- Given a schedule S for k-1 machines:

$$S=\{\sigma_1, \sigma_2,\dots,\sigma_{k-1}\}$$

  let $P_j(\sigma)$ denote the cumulative probability of job j in schedule σ and

$$Z_j(S)= Z_j \prod_{\sigma\in S}(1- P_j(\sigma))$$

- the optimal sequence σ_k on machine M_k is obtained by sequencing the jobs according to the *modified Z-ratios* Z_j(S)

Modified Z-rule heuristic

- Let $ER(\sigma_1, \sigma_2, \ldots, \sigma_k)$ denote the expected revenue when k (out of m) machines are accordingly scheduled
- $ER(\cdot)$ is an *increasing, submodular function* over the set of permutations
- Machine M_k is scheduled so to maximize the marginal gain, given the first k-1 schedules >>> Greedy heuristic

Modified Z-rule heuristic

- The problem is in the form:

$$\max \mathrm{ER}(S)$$
$$|S|=m$$

where ER(S) is a submodular function >>> the greedy algorithm is (1-1/e)-approximate [Nemhauser, Wolsey, Fisher 1978]

Mutual best-reply heuristic

- Apply the modified Z-rule heuristic;
- While (*a schedule on some machine M_k can be improved*)

  Reschedule M_k given the sequences on the other machines

- Complexity is open (but converges in a finite number of steps)

Computational experiments

- Rewards ~ [10,100]
- Probabilities~ I_p=[π_{min}, π_{max}]
  [0.1,1]; [0.5,1]; [0.9,1]
- $n \in \{10, 20, 30, 40, 50\}$

m=2

n	I_p	ZRH	MZRH	MBRH	Tabu Search	
		% gap	% gap	% gap	% gap	cpu
10	[0.1, 1]	8.21	6.96	6.53	6.27	0.24
10	[0.5, 1]	8.66	6.94	5.89	5.52	0.25
10	[0.9, 1]	1.38	0.88	0.73	0.65	0.24
20	[0.1, 1]	10.52	8.41	7.29	6.72	3.76
20	[0.5, 1]	10.85	8.60	7.08	6.37	3.31
20	[0.9, 1]	3.37	2.35	1.88	1.60	3.64
30	[0.1, 1]	10.90	8.51	7.40	6.54	11.80
30	[0.5, 1]	11.96	9.40	7.56	6.77	11.47
30	[0.9, 1]	5.82	3.81	2.98	2.47	10.84
40	[0.1, 1]	10.51	8.65	7.11	6.48	27.00
40	[0.5, 1]	11.63	9.31	7.39	6.46	26.46
40	[0.9, 1]	7.53	5.40	4.07	3.43	25.81
50	[0.1, 1]	11.57	9.03	7.29	6.46	55.92
50	[0.5, 1]	11.68	9.18	7.05	6.27	55.08
50	[0.9, 1]	9.10	6.08	4.75	3.84	54.68

m=3

		ZRH	MZRH	MBRH	Tabu Search	
n	I_p	% gap	% gap	% gap	% gap	cpu
10	$[0.1, 1]$	13.52	10.12	9.43	9.24	1.40
10	$[0.5, 1]$	12.68	8.29	7.51	7.26	1.49
10	$[0.9, 1]$	0.80	0.37	0.33	0.32	1.58
20	$[0.1, 1]$	17.00	11.67	10.53	10.38	12.88
20	$[0.5, 1]$	17.53	11.60	9.93	9.70	13.98
20	$[0.9, 1]$	2.81	1.43	1.19	1.18	15.62
30	$[0.1, 1]$	17.61	11.90	10.43	10.26	48.99
30	$[0.5, 1]$	19.29	12.89	10.86	10.61	55.83
30	$[0.9, 1]$	6.17	3.09	2.50	2.49	65.93
40	$[0.1, 1]$	17.12	11.82	10.21	10.04	128.16
40	$[0.5, 1]$	18.80	12.47	10.30	10.08	139.66
40	$[0.9, 1]$	9.03	4.92	3.94	3.94	174.16
50	$[0.1, 1]$	18.68	12.31	10.55	10.30	263.77
50	$[0.5, 1]$	19.06	12.49	10.26	10.05	301.88
50	$[0.9, 1]$	11.84	6.35	5.04	5.02	394.70

Open problem (m|rep,exp|ER)

Given:

- $J=\{1,\ldots,n\}$ job set
- m machines, subject to exponential failures with frequency λ
- p_j processing time of job j

Sequence the n jobs on each machine to maximize the expected amount of work

(Complexity is open for any fixed $m \geq 2$)

Two problems

- Sequence the *m* copies of each job on each machine in order to:
  - maximize the expected reward ($m|\text{rep}|ER$)
  - maximize the probability of having a complete kit, i.e., at least one copy of each job (*Kit Availability Maximization problem,* KAMm)

Two problems

- Sequence the *m* copies of each job on each machine in order to:
  - maximize the expected reward ($m|\text{rep}|ER$)
  - **maximize the probability of having a complete kit, i.e., at least one copy of each job (*Kit Availability Maximization problem,* KAMm)**

KAM

- With two machines (*KAM2*), any arbitrary reverse schedule is optimal
- With m machines and *only two job types*, the optimal solution can be found in $O(\log m)$

[A., Benini, Detti, Hermans, Pranzo 2022]

Open problem (KAMm)

Given:

- $J=\{1,\ldots,n\}$ job set
- m processing machines
- π_i success probability of job i

Sequence the n jobs on each machine to maximize the probability of having a complete kit

(Complexity is open for any fixed $m\geq 3$)

TESTING PROBLEMS

Sequential testing of *n*-out-of-*n* systems

- A complex system consists of *n* modular components
- The system is *up* iff all of the *n* components are working
- Testing component *j* has a cost c_j and the component works with probability π_j

Sequential testing of *n*-out-of-*n* systems

- As soon as a defective component is found, the test stops (the system is *down*)
- For a given test sequence, the expected testing cost is:

$$C(\sigma) = c_{\sigma(1)} + \pi_{\sigma(1)} c_{\sigma(2)} + \pi_{\sigma(1)} \pi_{\sigma(2)} c_{\sigma(3)} + \ldots + \pi_{\sigma(1)} \pi_{\sigma(2)} \ldots \pi_{\sigma(n-1)} c_{\sigma(n)}$$

Sequential testing of *n*-out-of-*n* systems

Given:

- $J=\{1,\ldots,n\}$ set of components to be tested
- π_j probability that component j is UP
- c_j cost of performing test j

Find a test sequence σ minimizing the expected cost

Sequential testing of *n*-out-of-*n* systems

$$C(\sigma) = c_{\sigma(1)} + \pi_{\sigma(1)} c_{\sigma(2)} + \pi_{\sigma(1)} \pi_{\sigma(2)} c_{\sigma(3)} + \ldots +$$

$$\pi_{\sigma(1)} \pi_{\sigma(2)} \ldots \pi_{\sigma(n-1)} c_{\sigma(n)}$$

$$ER(\sigma) = \pi_{\sigma(1)} R_{\sigma(1)} + \pi_{\sigma(1)} \pi_{\sigma(2)} R_{\sigma(2)} + \ldots +$$

$$\pi_{\sigma(1)} \pi_{\sigma(2)} \ldots \pi_{\sigma(n)} R_{\sigma(n)}$$

- Letting $R_j = c_j / \pi_j$, an instance of the testing problem is formally identical to an instance of $1||ER$ in which we want to *minimize* the objective function

Sequential testing of *n*-out-of-*n* systems

- As a consequence, the optimal testing sequence is obtained by *nondecreasing* values of the ratio [Mitten 1960]

$$Z_j = \frac{R_j \pi_j}{1-\pi_j} = \frac{c_j}{1-\pi_j}$$

Search problem

- An object is hidden in one of n locations, a searcher wants to find it
- Searching location j costs c_j, and there is a probability π_j that the object is hidden there
- What is the location sequence that minimizes expected costs?

Search problem

$$C(\sigma) = c_{\sigma(1)} + (1-\pi_{\sigma(1)})c_{\sigma(2)} + (1-\pi_{\sigma(1)}-\pi_{\sigma(2)})c_{\sigma(3)} + \ldots +$$
$$+(1-\pi_{\sigma(1)}-\pi_{\sigma(2)}-\ldots\ \pi_{\sigma(n-1)})c_{\sigma(n)}$$

- The single-searcher problem can be solved by sequencing the locations in nondecreasing order of the ratio c_j / π_j (Smith's rule)

Time-critical problems

- What happens if strict *time constraints* have to be respected? >>> Time-critical problems
- Each tester/searcher is active on a single component at a time
- All testing/searching operations require the same (unit) time
- As soon as one of the testers [searchers] detects a faulty component [finds the object], the process stops

Time-critical **testing** problem

Given:

- n components
- m testers
- π_j probability that component j is working
- c_j cost for testing j
- T deadline

Assign and sequence the n components to the m testers so that the process is completed within T and the expected costs are minimized

Time-critical testing problem

Given:

- n components
- m testers
- π_j probability that component *j* is working
- c_j cost for testing *j*
- T deadline

Assign and sequence the *n* components to the *m* testers so that the process is completed within *T* and the expected costs are minimized

Time-critical **search** problem

Given:

- n locations
- m searchers
- π_j probability that object is in j
- c_j cost for searching location j
- T deadline

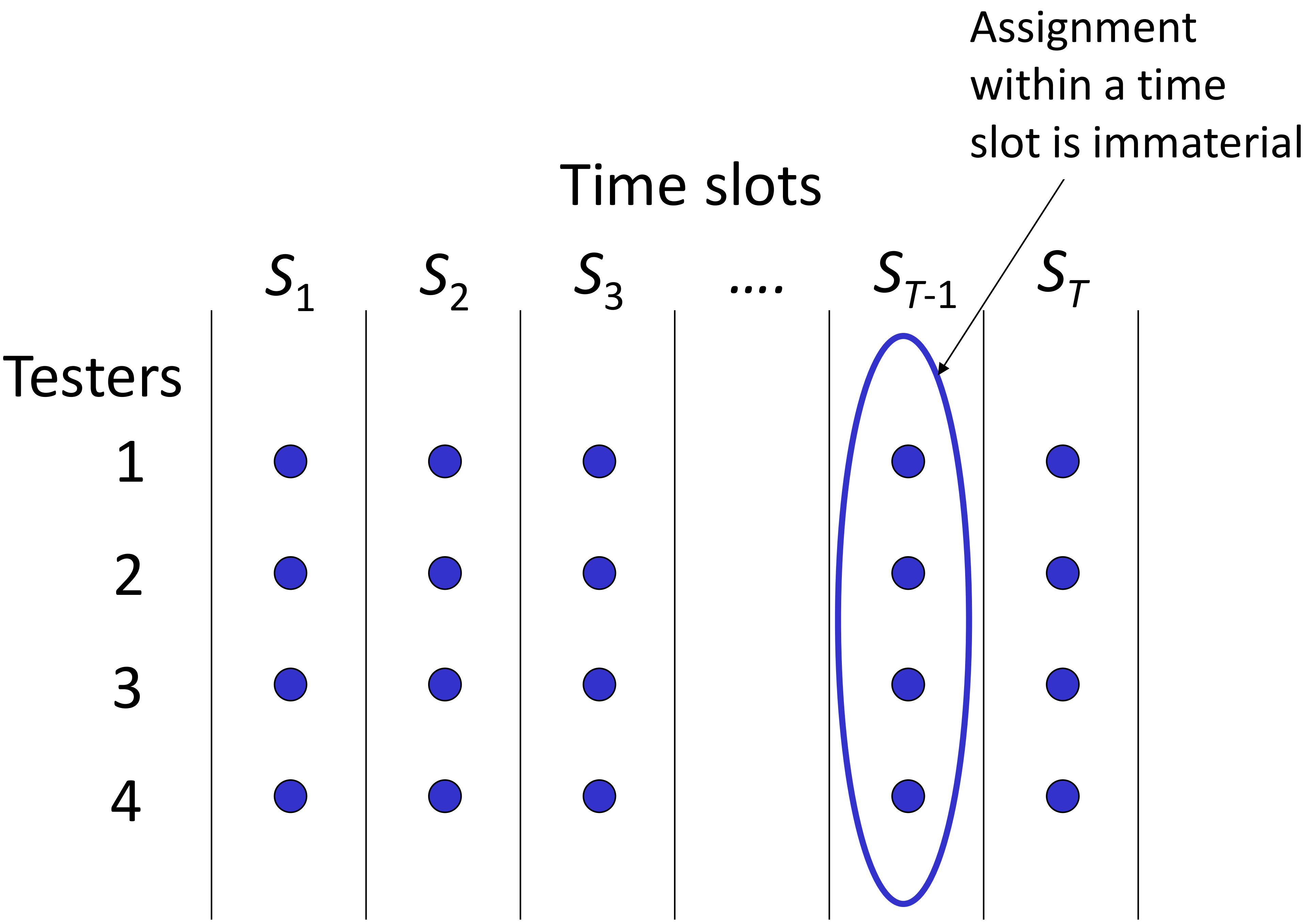
Assign and sequence the n locations to the m searchers so that the process is completed within T and the expected costs are minimized

Time-critical search problem

Given:

- n locations
- m searchers
- π_j probability that object is in j
- c_j cost for searching location j
- T deadline

Assign and sequence the n locations to the m searchers so that the process is completed within T and the expected costs are minimized


Assignment within a time slot is immaterial
Time slots
S_1 S_2 S_3 .... $S_{T\text{-}1}$ S_T
Testers
1
2
3
4

w.l.o.g., $n=mT$

Time-critical testing problem

- Let S_t be the set of components which are tested in slot t

$$f(\sigma) = \sum_{j \in S_1} c_j + \prod_{j \in S_1} \pi_j \sum_{j \in S_2} c_j + \ldots +$$

$$\prod_{j \in S_1} \pi_j \prod_{j \in S_2} \pi_j \ldots \prod_{j \in S_{T-1}} \pi_j \sum_{j \in S_T} c_j$$

Time-critical search problem

- Let S_t be the set of locations which are searched in slot t

$$f(\sigma) = \sum_{j \in S_1} c_j + (1 - \sum_{j \in S_1} \pi_j) \sum_{j \in S_2} c_j + \ldots +$$

$$(1 - \sum_{j \in S_1} \pi_j - \sum_{j \in S_2} \pi_j - \ldots - \sum_{j \in S_{T-1}} \pi_j) \sum_{j \in S_T} c_j$$

Properties

Consider a subset $S \subseteq N$ of tasks [locations], and let, in the testing problem:

$$Z_{TEST}(S) = \frac{c(S)}{1 - \prod_{j \in S} \pi_j}$$

and in the search problem:

$$Z_{SEARCH}(S) = \frac{c(S)}{\sum_{j \in S} \pi_j}$$

Properties

If a partition $\{S_1, S_2, \ldots, S_T\}$ is given, the optimal sequence is attained by scheduling the subsets:

- For the testing problem, by *nondecreasing* values of $Z_{TEST}(S_k)$
- For the search problem, by *nondecreasing* values of $Z_{SEARCH}(S_k)$

These rules generalize the Z-rule and Smith's rule respectively

Time-critical problem

… so the question is: how to find a partition $\{S_1, S_2, \ldots, S_T\}$ so that the expected cost is minimum?

Complexity results

- Both the testing and search problems are NP-hard in the ordinary sense for $T = 2$
- Both the testing and search problems are at least NP-hard for fixed $T \geq 3$

  (open as for strong NP-hardness)
- Both the testing and search problems are strongly NP-hard for every fixed $m \geq 3$

[A., Hermans, Leus, Rostami 2022]

Open problem

- What is the complexity of the time-critical testing problem and of the search problem for $m = 2$?

Partial-order-based formulation for testing problem

Decision variables:

$$\delta_{ij} = 1 \text{ if test } i \text{ is performed before test } j \text{ and 0 otherwise}$$
$$\mu_{\{i,j\}} = 1 \text{ if tests } i \text{ and } j \text{ are performed simultaneously and 0 otherwise}$$
$$\alpha_{in} = \text{probability that test } i \text{ will be performed}$$

Formulation (part 1):

$$\begin{array}{lll}
\min & \sum_{i \in N} c_i \alpha_{in} & \\
\text{s.t.} & \delta_{ij} + \delta_{ji} + \mu_{\{i,j\}} = 1 & \forall\, i, j \in N\colon i \neq j \\
& \mu_{\{i,j\}} + \delta_{ij} + \delta_{jk} - \delta_{ik} \leq 1 & \forall\, i, j, k \in N\colon i \neq j \neq k \neq i \\
& \sum_{j \in N \setminus \{i\}} \mu_{\{i,j\}} = m - 1 & \forall\, i \in N
\end{array}$$

Partial-order-based formulation for testing problem

Decision variables:

$$\alpha_{ij} = \text{probability that test } i \text{ will be performed, given that all tests } j > i \text{ will be successful}$$

Formulation (part 2):

$$
\begin{aligned}
&\alpha_{i0} = 1 && \forall\, i \in N \\
&\alpha_{ij} \geq \alpha_{i,j-1} - \delta_{ji} && \forall\, i \in N \text{ and } j \in \{1, \ldots, n\} \\
&\alpha_{ij} \geq p_j \alpha_{i,j-1} && \forall\, i \in N \text{ and } j \in \{1, \ldots, n\} \\
&\delta_{ij},\ \mu_{\{i,j\}} \in \{0, 1\} && \forall\, i, j \in N
\end{aligned}
$$

Testing is more difficult than search

		Time-critical testing problem						Time-critical search problem					
		Partial order			Assignment			Partial order			Assignment		
m	T	CPU	#	%LP	CPU	#	%LP	CPU	#	%LP	CPU	#	%LP
2	2	0.1	30	27.08	0.1	30	24.45	0.1	10	0.06	0.0	10	17.22
2	3	0.1	30	26.68	0.1	30	55.48	0.1	10	0.63	0.1	10	38.80
2	4	0.2	30	25.77	0.2	30	67.01	0.1	10	0.56	0.1	10	51.24
2	5	14.7	30	24.02	4.5	30	76.34	0.1	10	0.45	0.4	10	60.45
2	6	113.9	29	30.30	152.1	30	80.47	0.1	10	0.50	1.2	10	66.06
2	7	439.3	13	26.76	843.3	7	96.72	0.1	10	0.49	20.7	10	71.32
2	8		0		777.5	1	87.44	0.1	10	0.29	514.5	8	75.35
2	9					0		0.1	10	0.24	207.2	4	77.98
2	10							0.1	10	0.27	261.6	1	81.93
4	2	0.3	30	36.68	0.0	30	28.46	0.0	10	0.18	0.0	10	19.44
4	3	271.1	23	35.20	0.5	30	52.62	0.1	10	0.47	0.1	10	39.75
4	4		0		237.3	30	69.04	0.2	10	1.05	0.7	10	51.05
4	5					0		0.4	10	0.54	349.1	9	60.57
4	6							1.0	10	0.43		0	
4	7							3.1	10	0.39			
4	8							140.9	10	0.22			
4	9							224.7	10	0.59			
4	10							213.0	10	0.35			

Small m and large T more difficult than vice versa

Partial-order-based formulation for time-critical search problem

	$m=2$			$m=4$		
T	CPU	#	%LP	CPU	#	%LP
10	0.1	10	0.27	213.0	10	0.35
15	0.5	10	0.20	1,532.9	1	0.12
20	25.6	10	0.14		0	
25	48.8	10	0.09			
30	173.1	10	0.06			
35	695.5	5	0.05			
40	99.7	1	0.03			
45	890.2	2	0.02			
50		0				

	$T=2$			$T=4$		
m	CPU	#	%LP	CPU	#	%LP
10	0.1	10	0.02	22.6	10	0.25
12	0.5	10	0.01	246.1	10	0.31
14	0.6	10	0.02	370.2	10	0.19
16	0.4	10	0.00	472.6	7	0.09
18	2.0	10	0.01	713.2	5	0.07
20	1.5	10	0.00	757.0	7	0.02
22	1.8	10	0.00	1,015.8	2	0.09
24	3.7	10	0.03	567.2	2	0.02
26	9.3	10	0.00		0	
28	18.9	10	0.00		0	
30	8.4	10	0.00		0	

FURTHER RESEARCH

Further research

- Problems with precedence constraints: solution approaches for $1|\text{prec}|ER$, $1|\text{prec,exp}|ER$
- Analysis of $m|\text{rep,exp}|ER$
- Approximation algorithms for time-critical testing problems
- Time-dependent failure processes

Coauthors

Mario BENINI *(Università di Siena)*

Paolo DETTI *(Università di Siena)*

Ben HERMANS *(KU Leuven)*

Roel LEUS *(KU Leuven)*

Thomas LIDBETTER (*University of Virginia*)

Patrick MARTINEAU *(Polytech Tours)*

Marco PRANZO *(Università di Siena)*

Salim ROSTAMI *(IESEC, Lille)*

Kevin SCHEWIOR *(University of Southern Denmark)*

Manbir S. SODHI *(University of Rhode Island)*

Agnetis, A., Detti, P., Pranzo, M., Sodhi, M.S., Sequencing Unreliable Jobs on Parallel Machines, *Journal of Scheduling*, 2009, 12, 45–54.

Agnetis, A., Detti, P., Pranzo, M., The list scheduling algorithm for scheduling unreliable jobs on two parallel machines, *Discrete Applied Mathematics*, 2014, 165, 2–11.

Agnetis, A., Detti, P., Martineau, P., Scheduling nonpreemptive jobs on parallel machines subject to exponential unrecoverable interruptions, *Computers and Operations Research*, 2017, 79, 109–118.

Agnetis, A., Lidbetter, T., List scheduling is 0.8531-approximate for scheduling unreliable jobs on m parallel machines, *Operations Research Letters*, 48, 405–409, 2020

Agnetis, A., Hermans, B., Leus, R., Rostami, S., Time-critical testing and search problems, *European Journal of Operational Research*, 296(2), 440–452, 2022

Agnetis, A., Benini, M., Detti, P., Hermans, B., Pranzo, M., Replication and sequencing of unreliable jobs on parallel machines, *Computers and Operations Research*, 139, 105634, 2022

1. Which movie got 14 academy awards nominations (and won 11) in 1997?	*Titanic*
2. What is the title of Leonard Cohen's 1984 album containing *Hallelujah*?	*Various Positions*
3. What is the name of Charlie Brown's favorite baseball player?	*Joe Shlabotnik*
4. Which team won the Italian soccer championship in the year 1991?	*Sampdoria*
5. What is the value of the optimal solution of Muth and Thompson's 1963 job shop instance?	*930*