Stochastic Optimization Methods in Scheduling

Rolf H. Möhring
Technische Universität Berlin
Combinatorial Optimization and Graph Algorithms

More expensive and longer ...

Eurotunnel

- Unexpected loss of £ 400,000,000 in first half of 1995
- Delays in project start and excecution

More expensive and longer ...

July 7, 1999

Government & parliament buildings in Berlin

- Not ready for move of the government & parliament
- Expected to be (much) more expensive

Der Tagesspiegel
vom 10. September
1999

Baukommission unglücklich, Büros unfertig

Wieder einmal Verzögerung bei Bundesbauten: Die Räume für die Abgeordnete werden später fertig und kosten mehr

Die Bundestagsabgeordneten müssen nach ihrem Berlin-Umzug voraussichtlich wesentlich länger als erwartet in Provisorien arbeiten. Wie die Baukommission des Bundestags am Donnerstag mitteilte, werden die Parlamentsgebäude um den Reichstag bis zu acht Monate später fertig als ursprünglich geplant. "Wir sind tief enttäuscht und sogar leicht deprimiert", sagte der Kommissionsvorsitzende Kansy.

We are deeply disappointed and even somewhat depressed

Reasons

- Planning assumes certainty about project details
  - deterministic models
- Project excecution is subject to many influences that are beyond control
  - machine breakdowns, weather, illness, …

⇒ leads to underestimation of expected makespan and cost

Fulkerson 1962

Therefore

- Need models and techniques to cope with uncertainty

Coworkers

- M. & Frederik Stork
  - DFG Project "Scheduling Problems with Varying Processing Times"

- M., Andreas Schulz, Martin Skutella & Marc Uetz
  Esther Frostig & Gideon Weiss
  - GIF Project "Polyhedral Methods in Stochastic Scheduling"

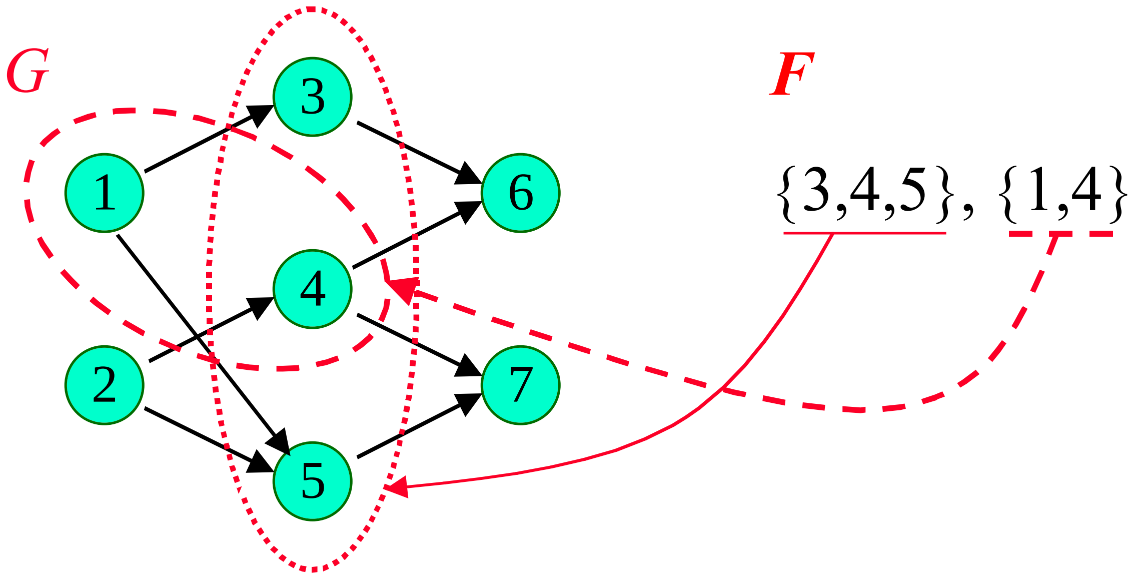
- Background
  - M., Radermacher & Weiss 1982-1986

Overview

- The model
- Classes of policies
- Computation and approximation
- Open problems

The discrete data

- a set *V* of n jobs $j = 1,\ldots,n$ (no preemption)
- a graph (partial order) *G* of *precedence constraints*
- a system ***F*** of forbidden sets (*resource constraints*)



The continuous data

- a joint distribution Q of job processing times
  job j has random processing time X_j with distribution Q_j

- a cost function $\kappa(C_1,\ldots,C_n)$
  depending on the (random) completion times $C_1,\ldots,C_n$

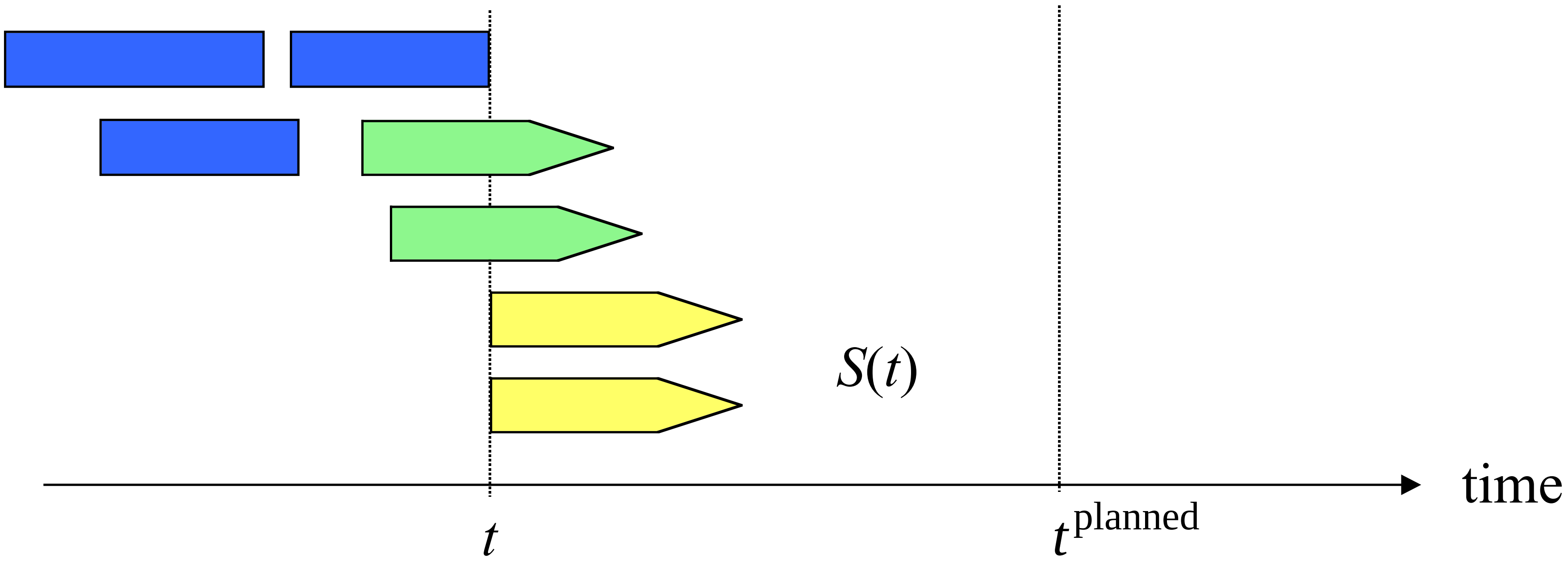
Examples: $C_{\max}$ and $\sum w_j C_j$

The objective

Plan jobs non-preemptively over time and ...

- respect the
  - precedence constraints
  - resource constraints
- minimize
  - expected cost   or
  - other parameters of the cost distribution

Planning with policies — The dynamic view

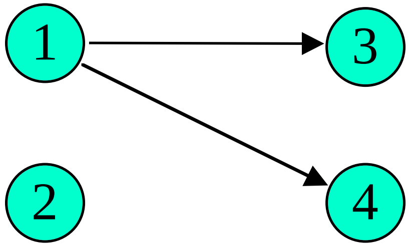


Decision at decision time t non-anticipative

- start set $S(t)$ (possibly empty)
- fix tentative next decision time $t^{planned}$, $t^{next} = \min\{t^{planned}, C^{next}\}$

Minimize expected cost: $\min\{E[\kappa^{\Pi}] \mid \Pi \text{ is a policy}\}$

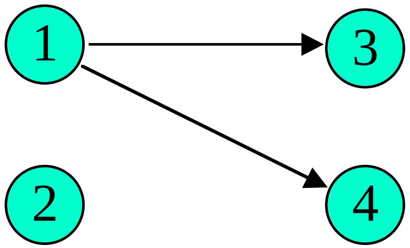
Policies — An Example



- $m = 2$ machines $\Rightarrow F = \{2,3,4\}$ forbidden
- $X_j \sim \exp(a)$, independent
- common due date d
- penalties for lateness: v for job 2, w for jobs 3,4, $v \ll w$

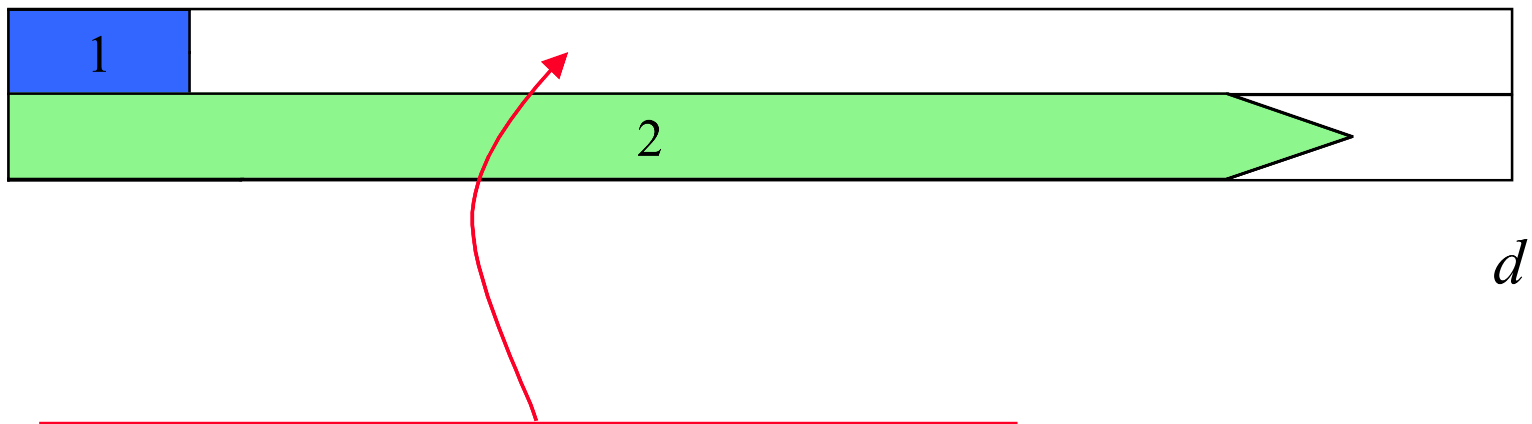
Minimize $E(\Sigma$ penalties $)$

Example: starting job 1 and 2 early



Start jobs 1 and 2 at $t = 0$

Danger: job 2 blocks machine

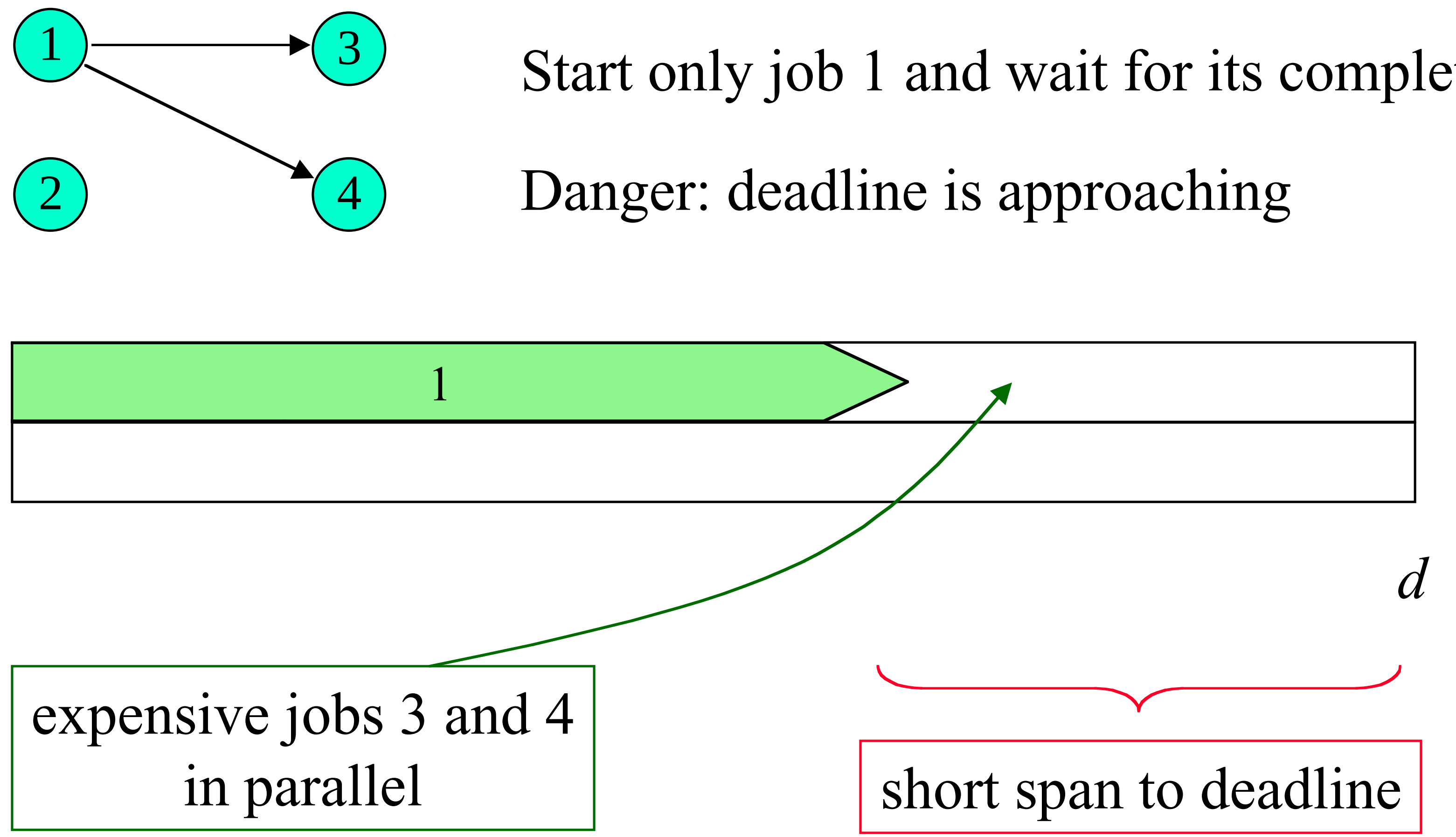


expensive jobs 3 and 4 sequentially

Example: leaving the second machine idle

Start only job 1 and wait for its completion

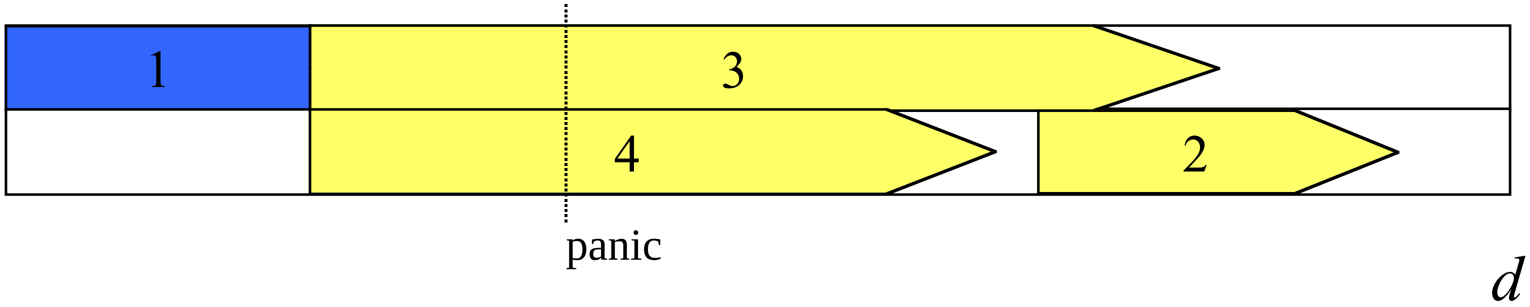
Danger: deadline is approaching



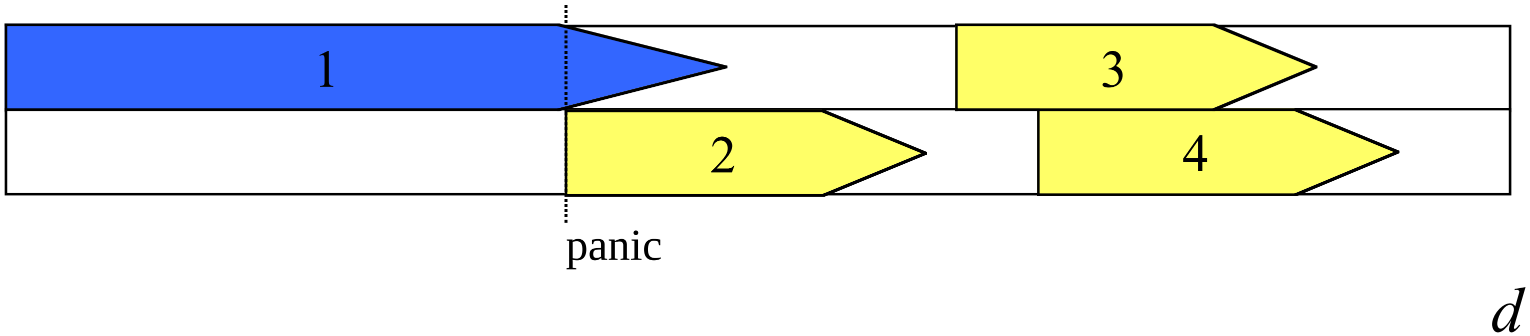
Example: use tentative decision times

Start 1 at time 0. Fix tentative decision time t^{panic}

if $C_1 \leq t^{\text{panic}}$ start 3 and 4 at C_1



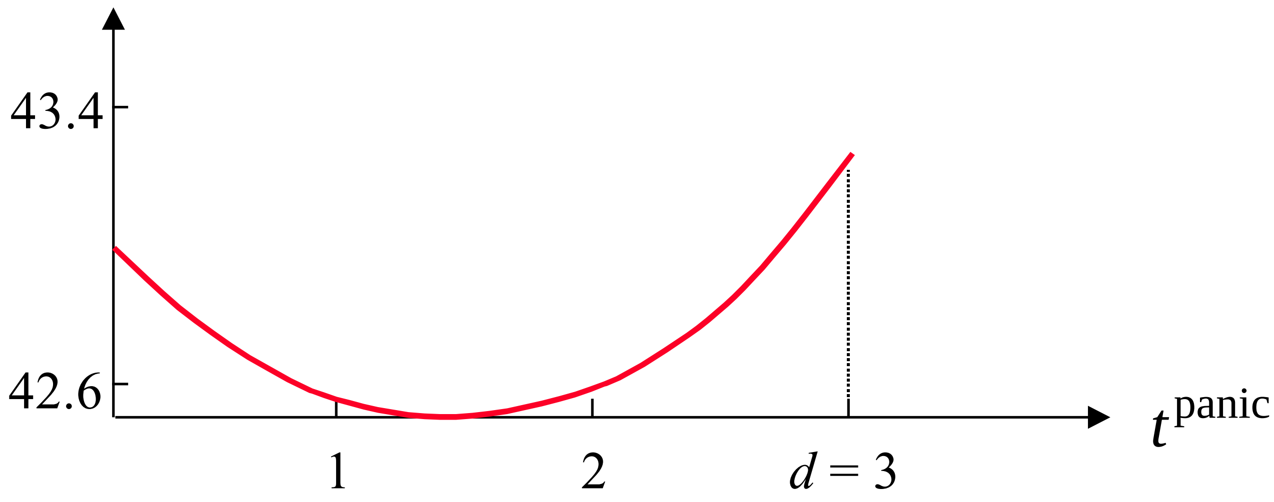
else start 2 at t^{panic}



Jobs may start when no other job ends

Example: best policy uses tentative decision time

Expected cost for $a = 1$, $d = 3$, $v = 10$, $w = 100$



Comparison with Stochastic Programming

2-stage stochastic program:

$$\begin{aligned} &\min \mathrm{E}[f(\xi, x_1, x_2(\xi))] \\ &\text{s.t.} \quad x_1 \in C_1 \\ &\qquad\quad x_2 \in C_2(\xi, x_1) \end{aligned}$$

ξ observation

x_1 first stage decision

x_2 second stage decision

ξ independent from x_1 in this model but not in stochastic scheduling!

Stability of policies

Data deficiencies, use of approximate methods (simulation) require stability condition:

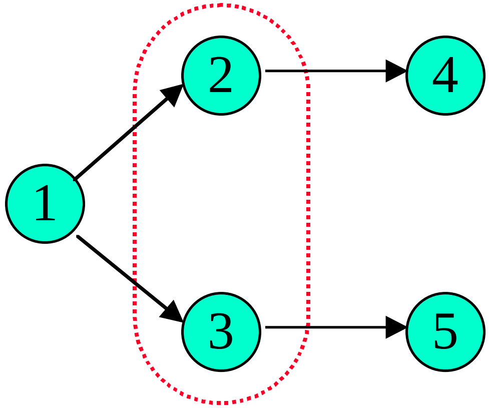
$\tilde{Q}$ approximates Q

$\tilde{\kappa}$ approximates κ

$\Rightarrow$

OPT($\tilde{Q},\tilde{\kappa}$) approximates OPT(Q,κ)

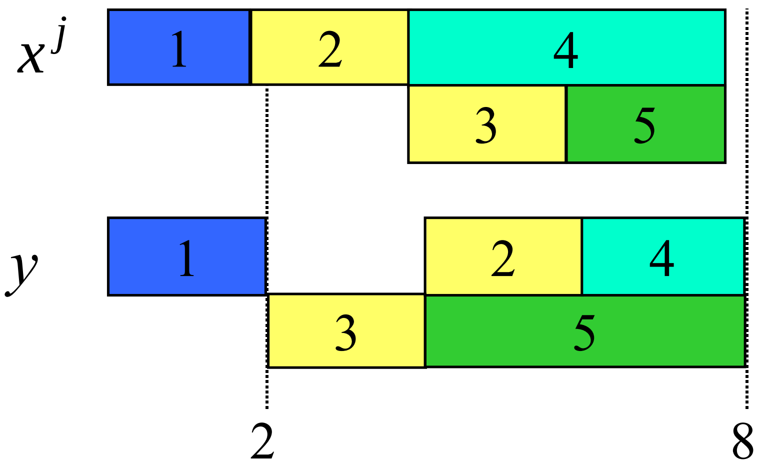
Excessive information yields instability



$$\min E(C_{\max})$$

$$Q^j : \begin{cases} x^j = (2 - \frac{1}{j}, 2, 2, 4, 2) \text{ with probability } \frac{1}{2} \\ y \;\; = (2, \quad\;\; 2, 2, 2, 4) \text{ with probability } \frac{1}{2} \end{cases}$$

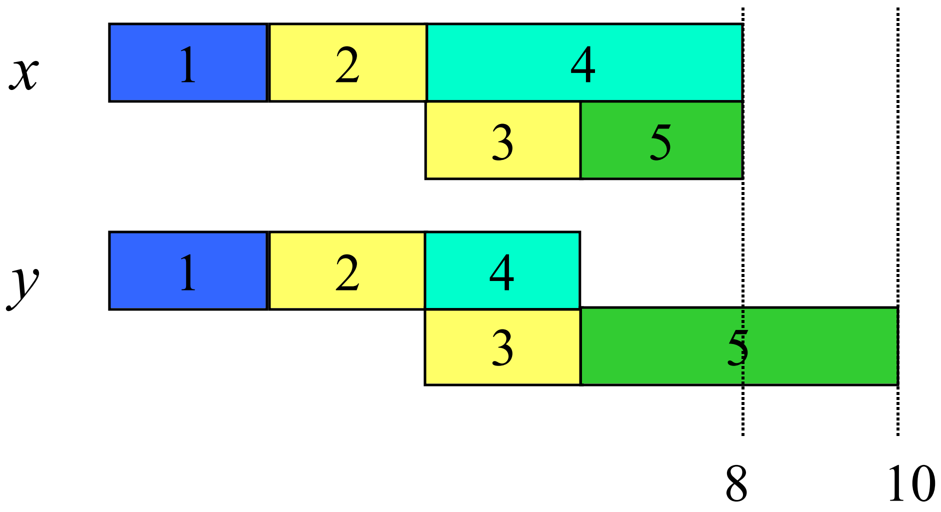
Exploit info when 1 completes



$$\Rightarrow \quad E_{Q^j}(C_{\max}) \xrightarrow[j \to \infty]{} 8$$

$$Q^j \xrightarrow[j\to\infty]{} Q : \begin{cases} x = (2,2,2,4,2) \text{ with probability } \frac{1}{2} \\ y = (2,2,2,2,4) \text{ with probability } \frac{1}{2} \end{cases}$$

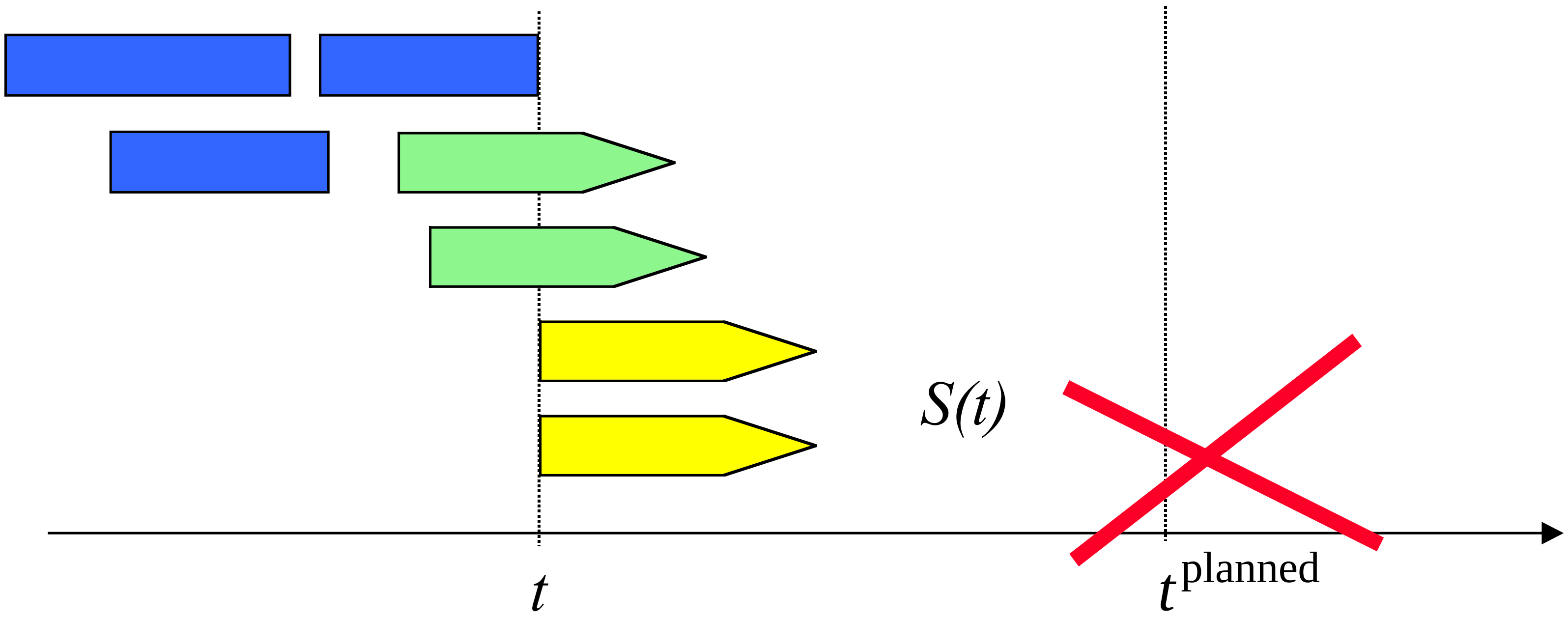
No info when 1 completes.   So start 2 after 1



$$\Rightarrow \quad E_Q(C_{\max}) = 9$$

$$\neq \lim_j E_{Q^j}(C_{\max}) = 8$$

Robust information and decisions



Robust information at time *t*

- which jobs have completed by *t*
- which jobs are running at *t*

Start jobs only at completions of other jobs

Overview

- The model
- Classes of policies
- Computation and approximation
- Open problems

Policies — viewed as functions

$\Pi : \mathbb{R}^n \to \mathbb{R}^n$

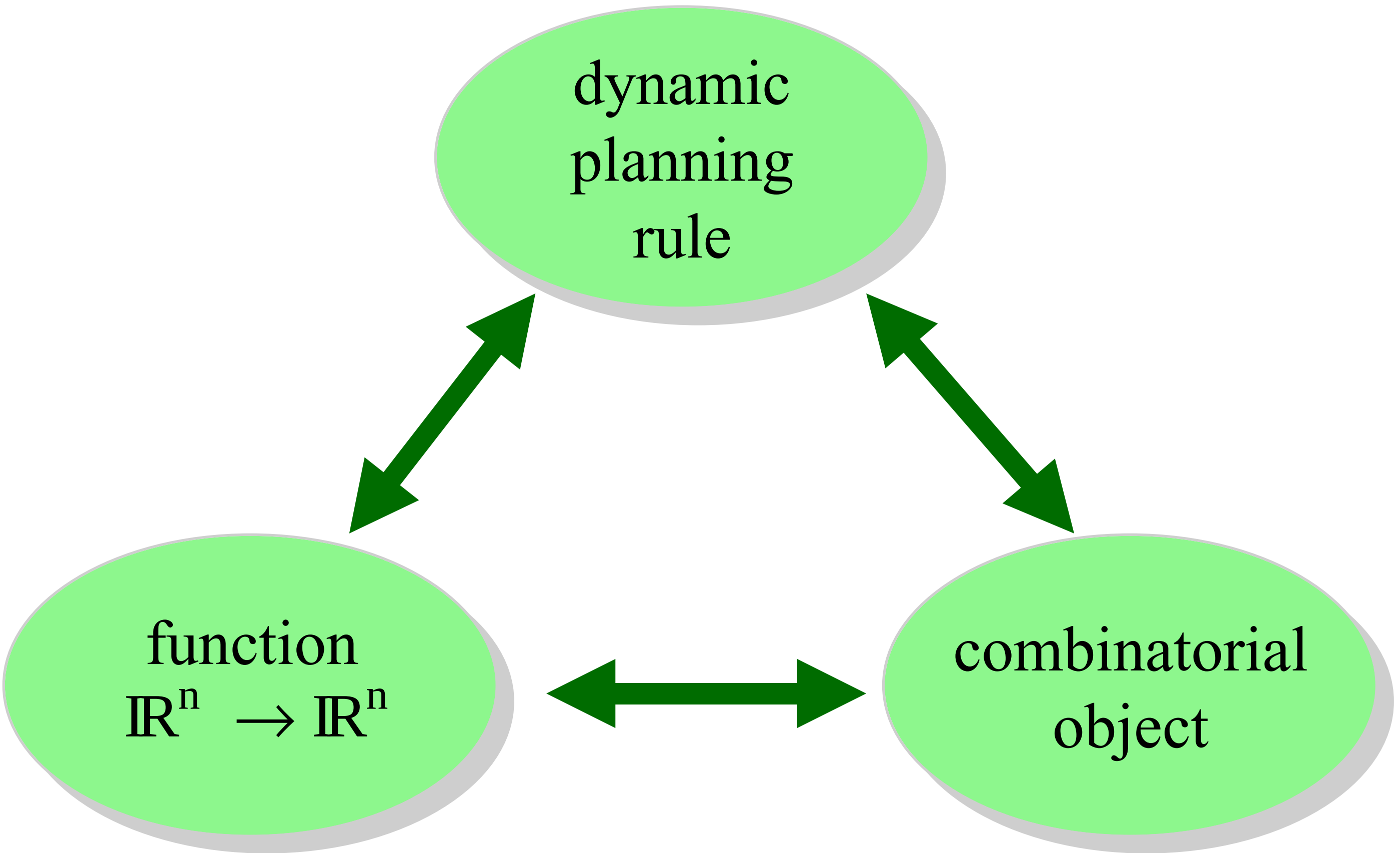
processing time vector $x \to$ schedule $\Pi(x)$

$$(x_1,\ldots,x_n) \to (S_1,\ldots,S_n)$$

Discuss properties of policies like being

- continuous
- convex
- monotone
- ...

A policy is three objects



Classes of policies

- priority policies
- preselective policies
  - earliest start policies (*ES*-policies)
  - linear preselective policies

distinct conflict solving strategies on forbidden sets

- a general class of robust policies

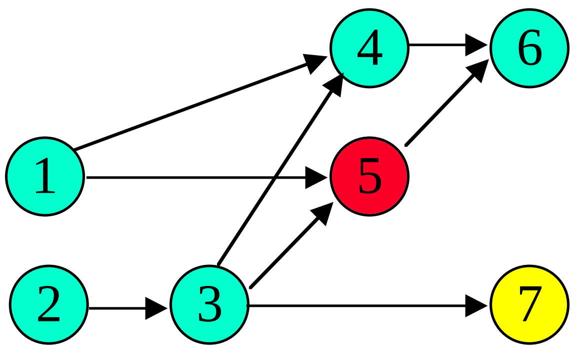
Priority policies

Solve resource conflicts by priorities

At every decision time t, use a priority list $L_t : j_1 < j_2 < \ldots < j_k$
Start as many jobs as possible in the order of L

Greedy use of scarce resources

Priority policies are neither continuous nor monotone (Graham anomalies)

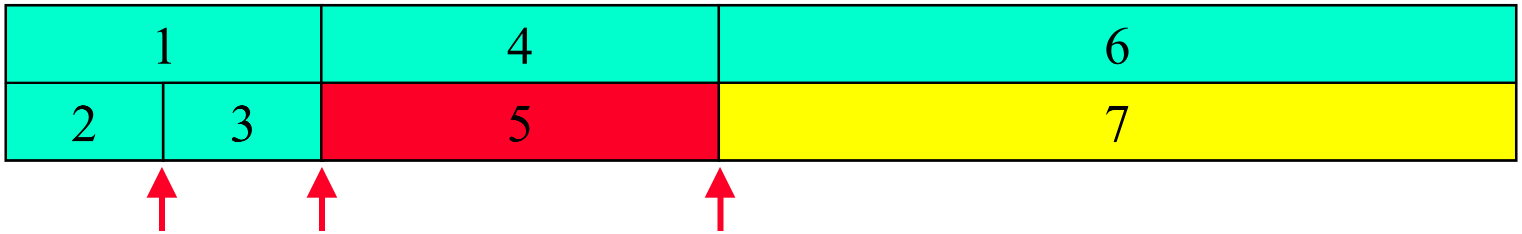


2 identical machines

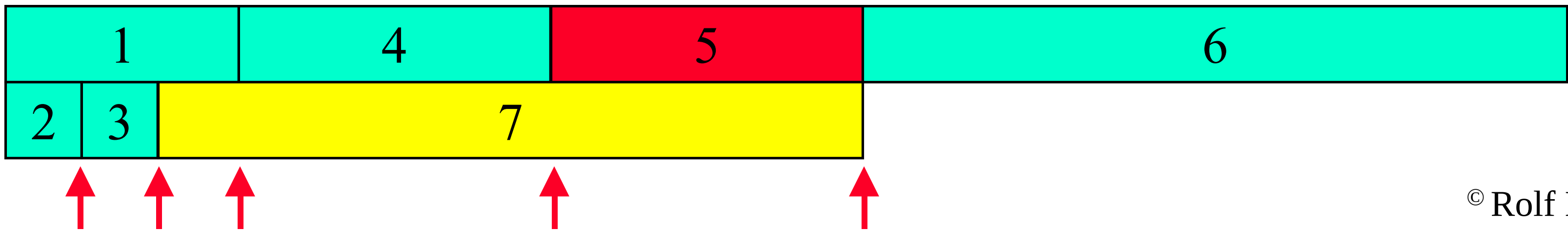
min $C_{\max}$

$L = 1 < 2 < \ldots < 7$

$x = (4,2,2,5,5,10,10)$



$y = x - 1 = (3,1,1,4,4,9,9)$



Classes of policies

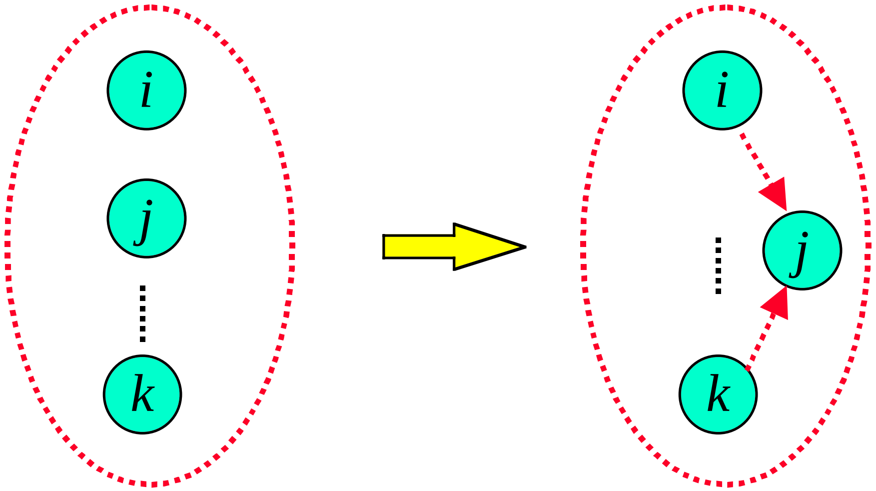
- priority policies
- preselective policies
  - earliest start policies (*ES*-policies)
  - linear preselective policies

distinct conflict solving strategies on forbidden sets

- a general class of robust policies

Preselective policies

Solve resource conflicts by pre-selecting a waiting job

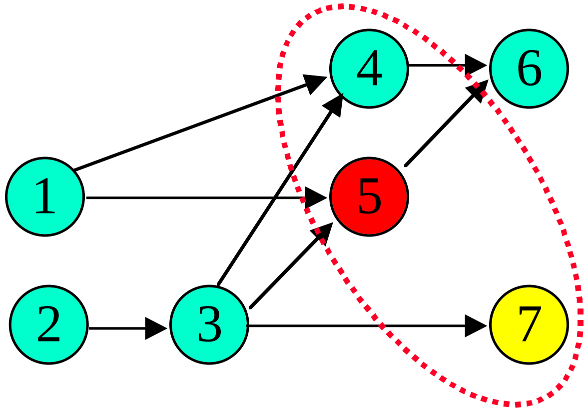


for every forbidden set F,
select waiting job j from F,
j must wait for at least one job from F

delaying alternative waiting condition

Start every job as early as possible w.r.t. to G + waiting conditions

A preselective policy for Graham s example



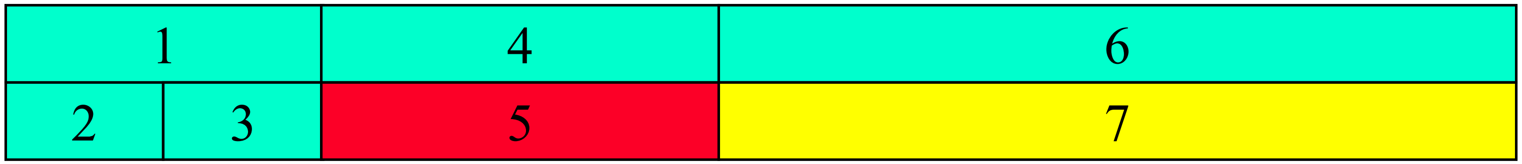
2 identical machines

$\Rightarrow F = \{4,5,7\}$ is only forbidden set

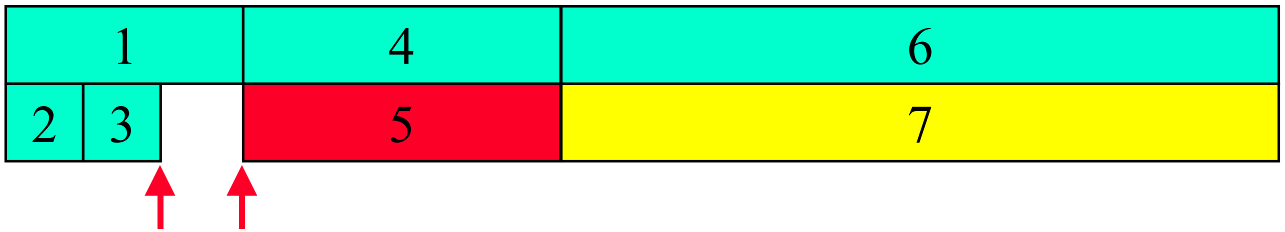
↓

7

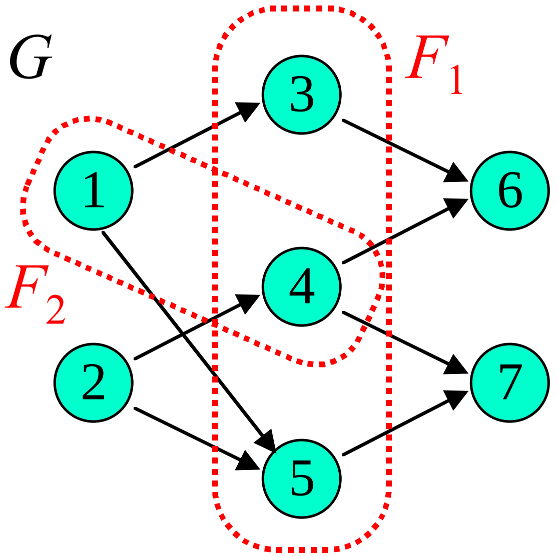
$x = (4,2,2,5,5,10,10)$



$y = x — 1 = (3,1,1,4,4,9,9)$



Preselective policies and AND/OR networks

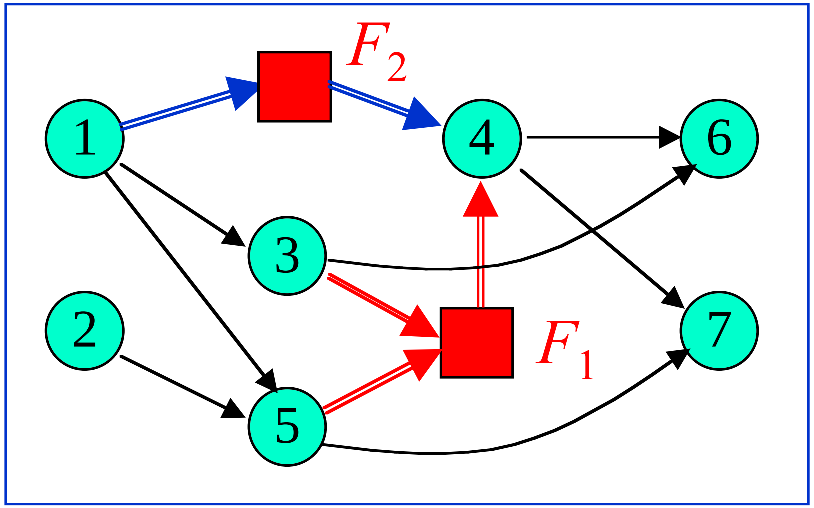


$\boldsymbol{F}$: {3,4,5}, {1,4}

↓ ↓

4 4

waiting jobs
define policy Π



AND/OR network representing Π

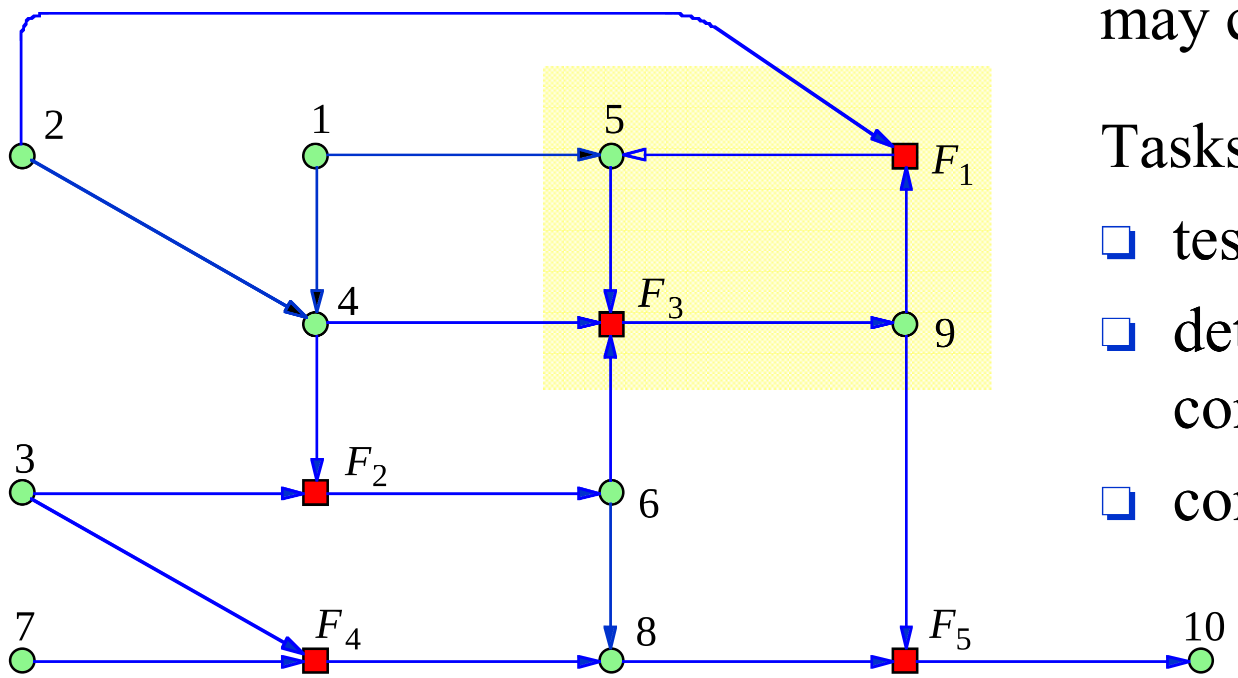
start in Π

= min of longest paths lengths

= min of max of sums of processing times

$\Rightarrow$ Π is monotone and continuous

Problems related to AND/OR networks



may contain cycles

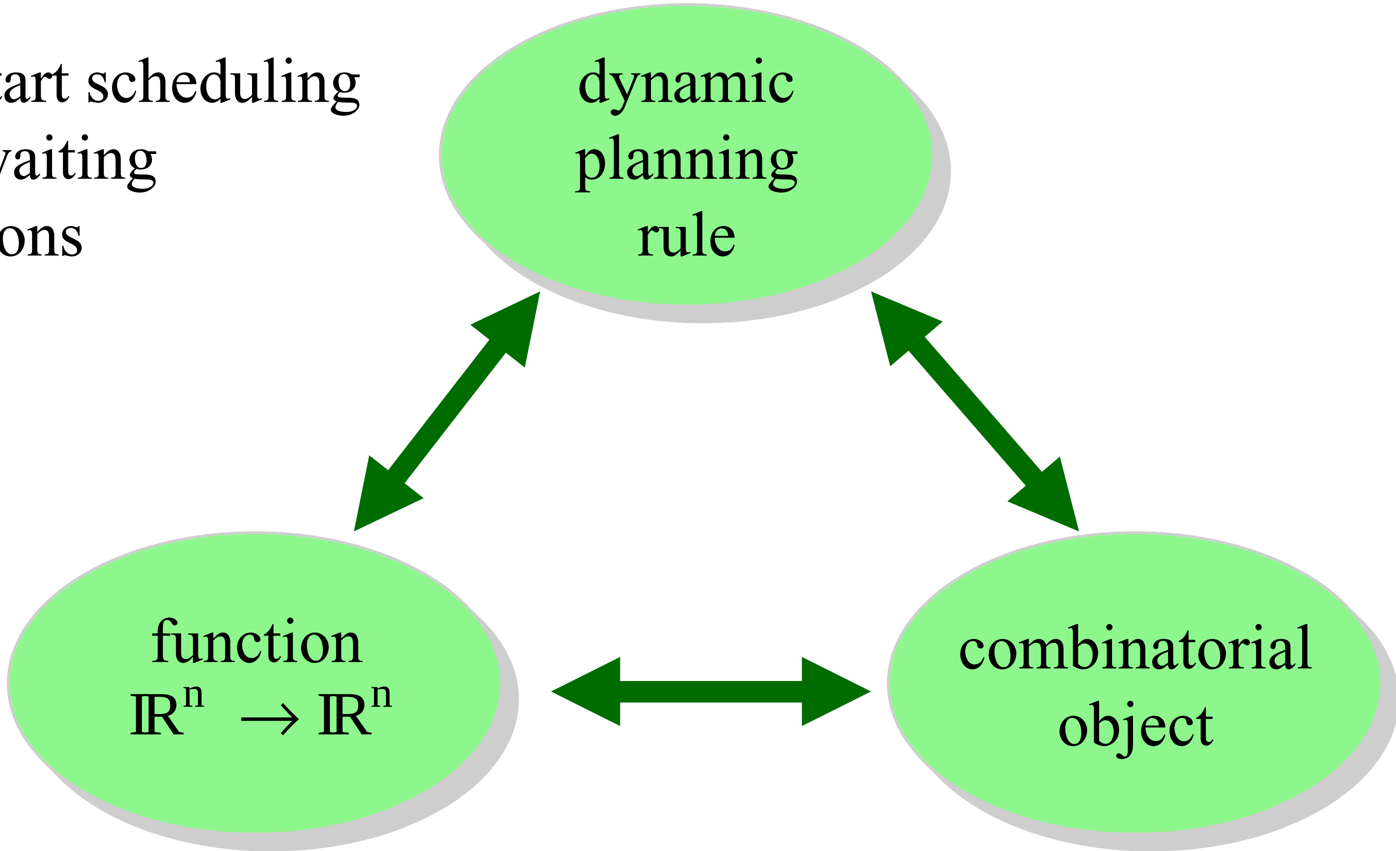
Tasks

- test feasibility
- detect forced waiting conditions (transitivity)
- compute earliest start

Fast algorithms available

3 views on preselective policies

early start scheduling
w.r.t. waiting
conditions



continuous, monotone

AND/OR networks

$\Rightarrow$ stability

Classes of robust policies

- priority policies
- preselective policies
  - earliest start policies (*ES*-policies)
  - linear preselective policies

distinct conflict solving strategies on forbidden sets

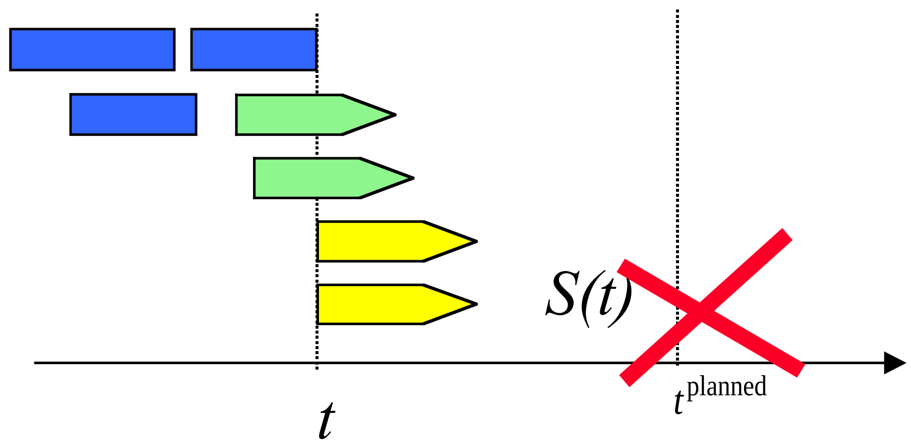
- a general class of robust policies

Set policies: A general class of robust policies

Only exploitable information at time t

- set of completed jobs
- set of busy jobs

Jobs start only at completions of other jobs



Special cases

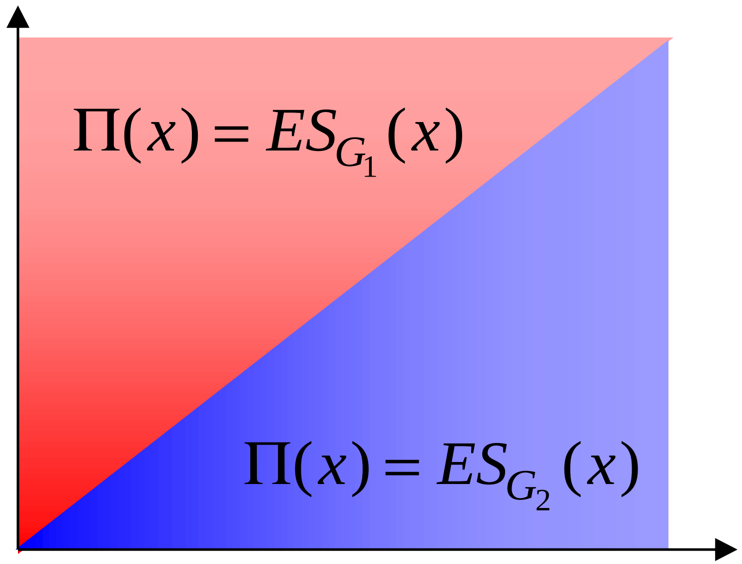
- priority policies
- preselective policies

Set policies behave locally like ES-policies

For every set policy Π, there exists

- a partition of $\mathbb{R}^n$ into finitely many convex polyhedral cones $Z_1,\dots,Z_k$
- and feasible partial orders $G_1,\dots,G_k$

such that $\Pi(x) = ES_{G_i}(x)$ for $x \in Z_i$



Graham anomalies only at boundaries of cones!

Stability for continuous distributions!

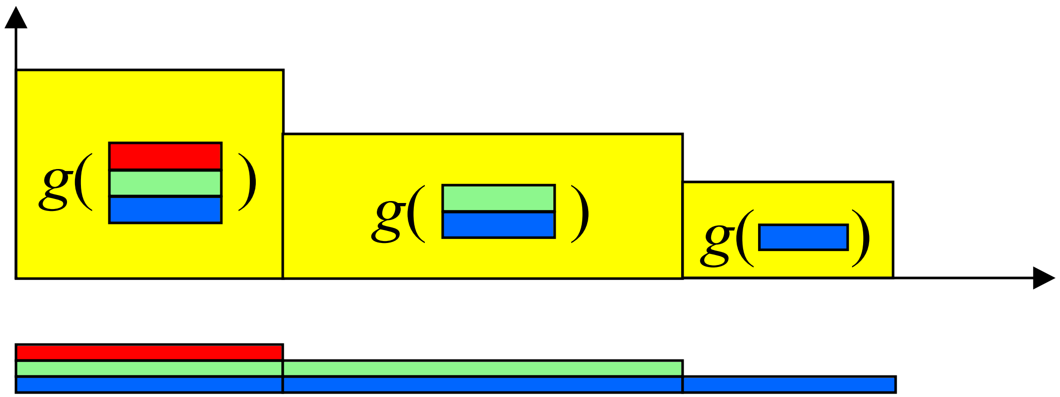
Optimality of set policies

If • all jobs are exponentially distributed and independent
• the cost function κ is *additive*

then there is an optimal set policy Π (among all policies).

κ is *additive* if there is a set function $g: 2^V \to \mathbb{R}$ (the *cost rate*)

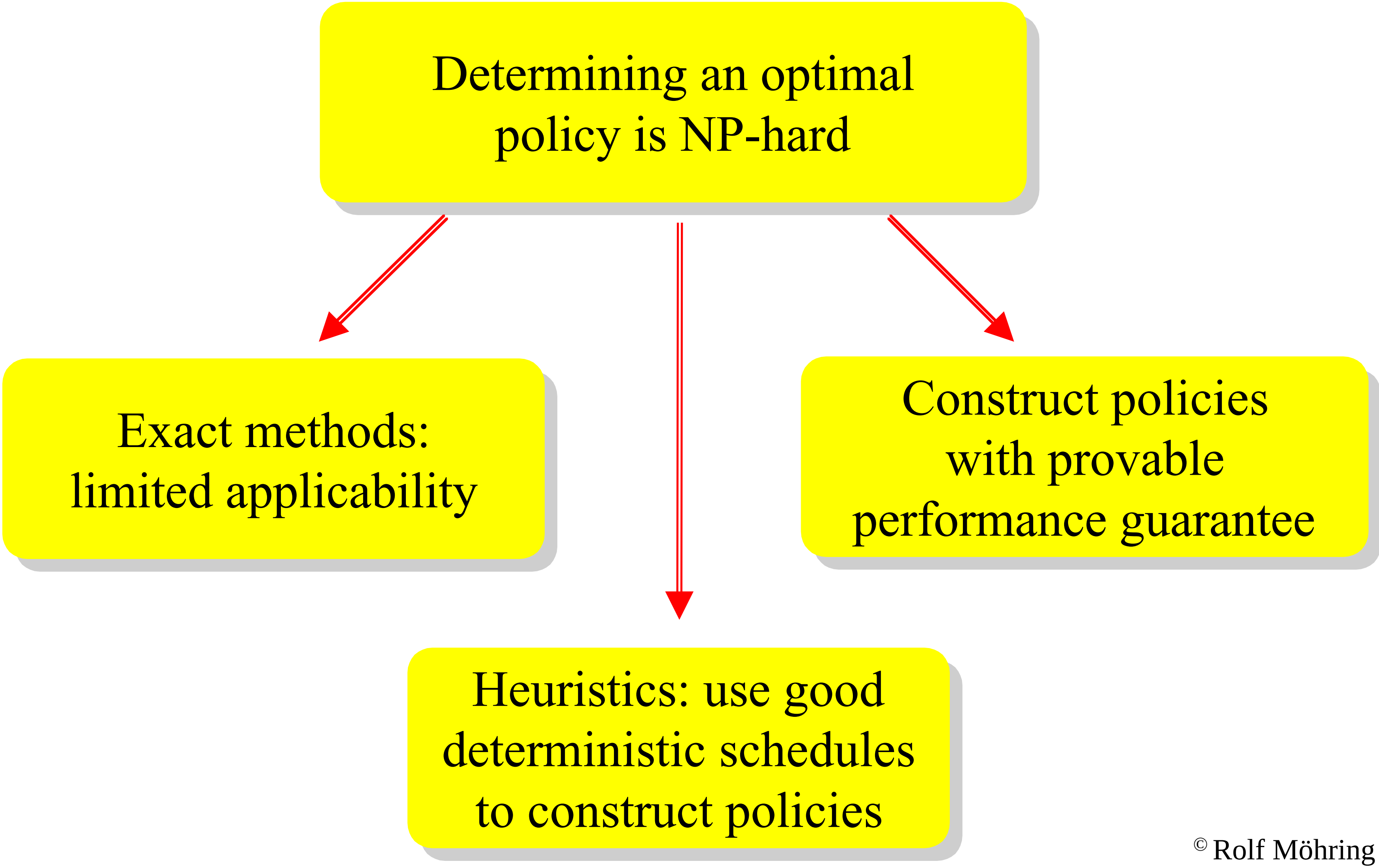
with $\kappa(C_1,\ldots,C_n) = \int g(U(t))dt$ $\quad U(t) =$ set of uncompleted jobs at t



Overview

- The model
- Classes of policies
- Computation and approximation
- Open problems

How to find good policies efficiently?



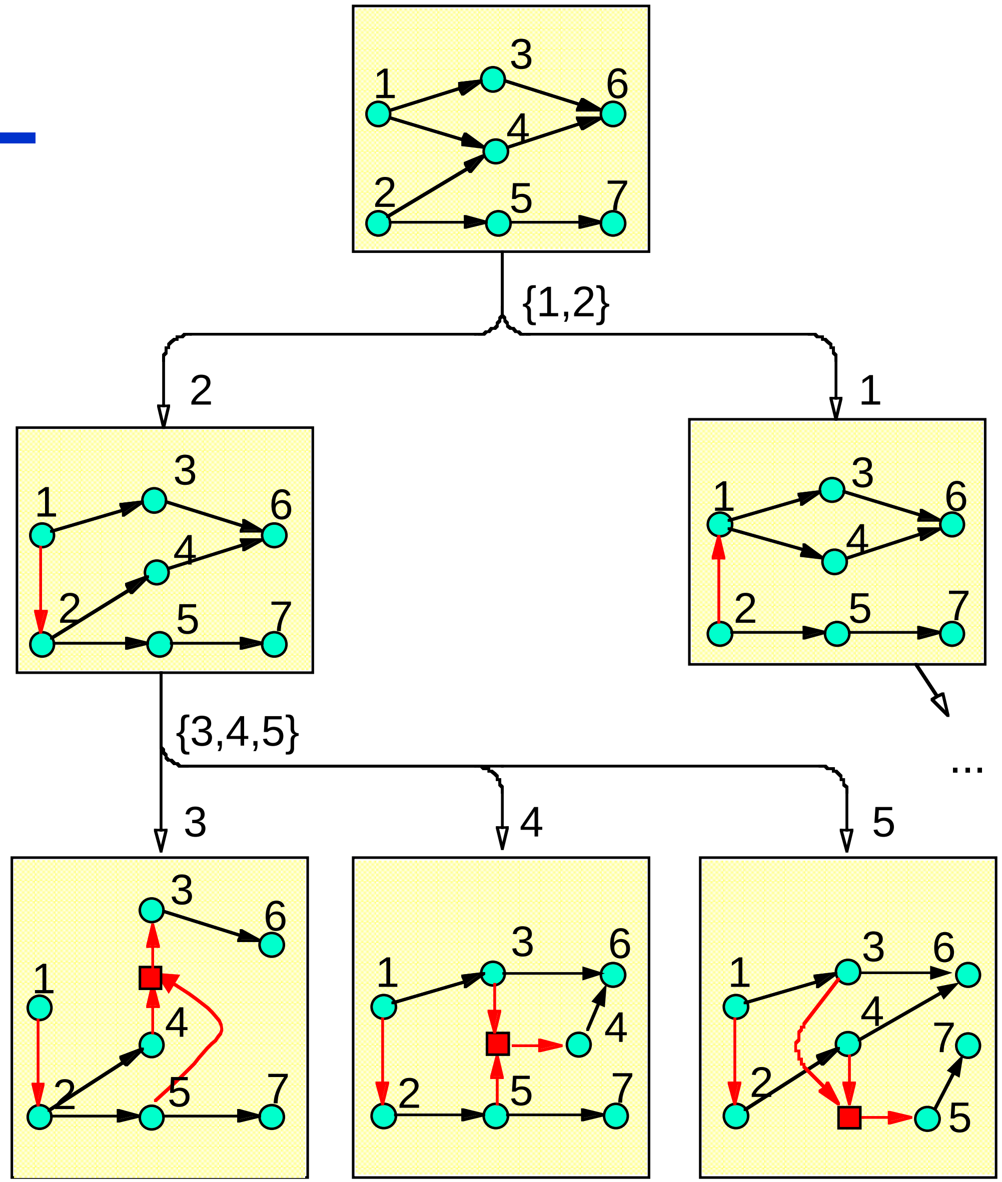
Algorithms

Heuristics

- Extract good policies from several deterministic schedules

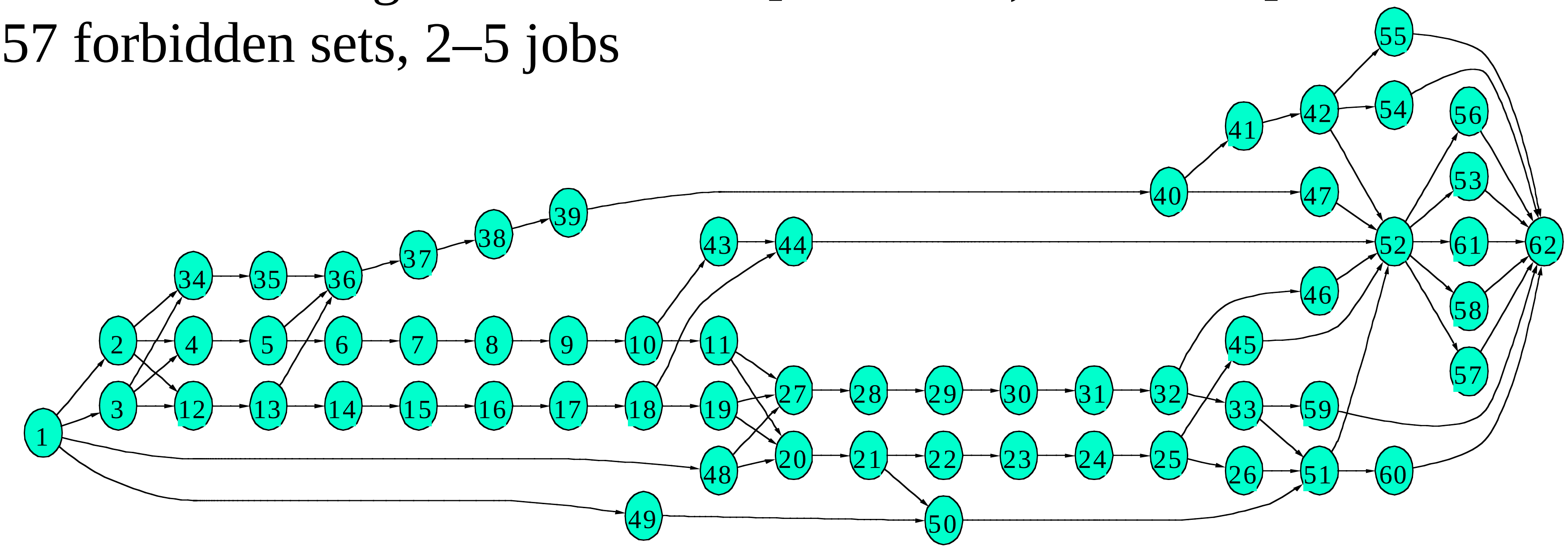
Exact methods

- Branch & Bound and exterior sampling



Computing (linear) preselective policies

Truncated Erlang distribution on [0.2*mean; 2.6*mean]
57 forbidden sets, 2–5 jobs



Optimum deterministic makespan	203	CPU: .17 sec
Optimum expected makespan	243.2	
Optimal preselective policy	Nodes: 115007	CPU: 3772.01 sec
Opt. linear pres. policy	Nodes: 4209	CPU: 49.85 sec

How good are simple policies?

Simple = priority or linear preselective or ??

A simple setting:

- m identical machines
- $\kappa = \sum w_j C_j$

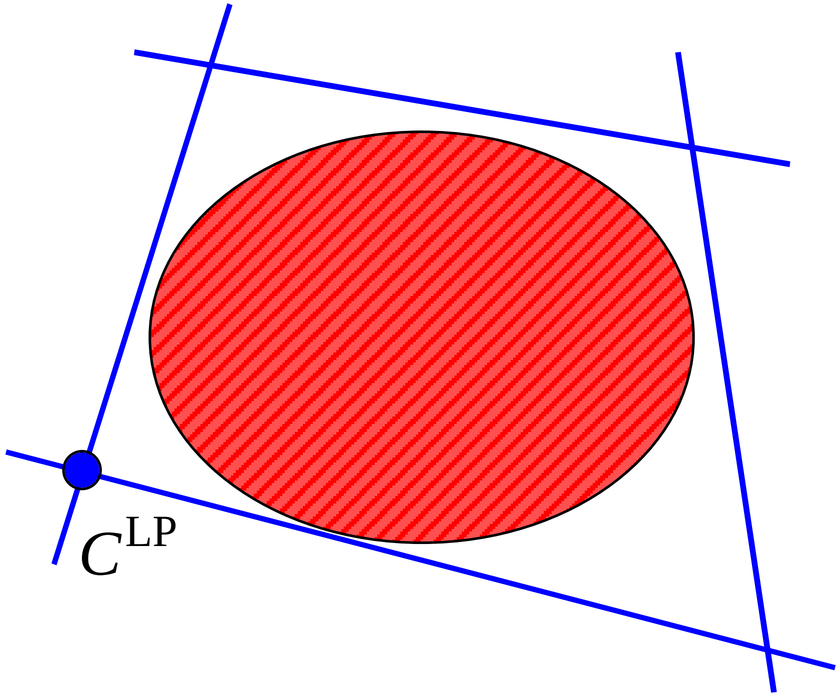
Use ideas from the deterministic case

- LP-relaxation
- LP-guided construction of a list L defining the policy

The LP-based approach

Consider the achievable region

$$(E[C_1^{\Pi}], \ldots, E[C_n^{\Pi}]) \in \mathbb{R}^n \mid \Pi \text{ policy } \}$$



Find a polyhedral relaxation ***P***

Solve the linear program

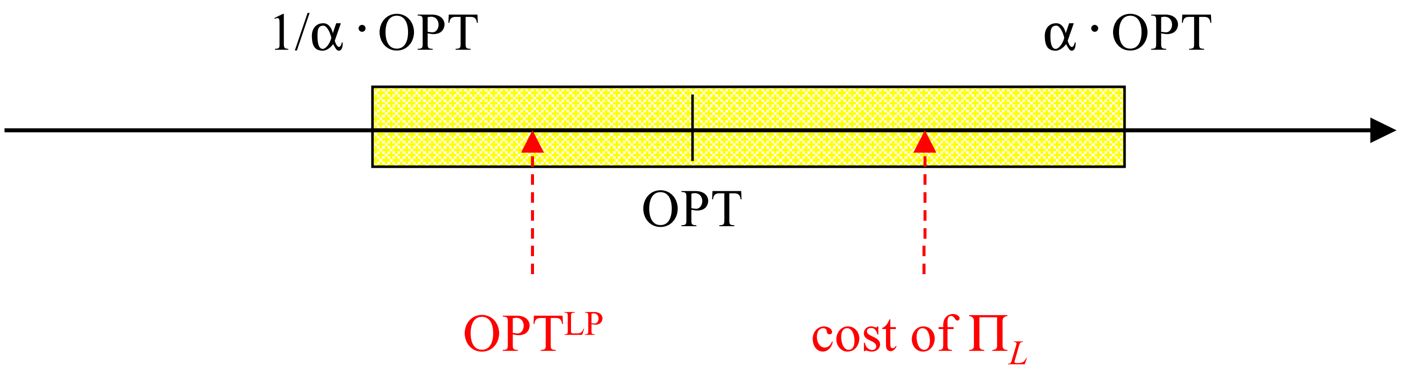
$$\text{(LP)} \;\; \min\{ \sum_j w_j C_j^{\mathrm{LP}} \mid C^{\mathrm{LP}} \in P \}$$

Use the list L: $i_1 \leq \; i_1 \leq \ldots \leq \; i_n$
defined by $C_{i_1}^{\mathrm{LP}} \leq C_{i_2}^{\mathrm{LP}} \leq \ldots \leq C_{i_n}^{\mathrm{LP}}$
as list for priority/lin. pres./other policy

Performance guarantees from the LP

Let Π_L be the policy induced by L: $i_1 \leq \; i_1 \leq \ldots \leq \; i_n$

Hope that $$E[\kappa^{\Pi_L}] \leq \alpha \cdot \text{OPT}^{\text{LP}}, \;\alpha \geq 1$$



The case without precedence constraints

Generalize valid inequalities from deterministic scheduling
Hall, Shmoys, Schulz & Wein 97

$$\sum_{k \in A} E[X_k] E[C_k^\Pi] \geq \frac{1}{2m} \left(\sum_{k \in A} E[X_k] \right)^2 + \frac{1}{2} \sum_{k \in A} E[X_k]^2$$

$$- \frac{m-1}{2m} \sum_{k \in A} Var[X_k]$$

for all $A \subseteq \{1,\ldots,n\}$ and all policies Π

The term $\sum_k Var[X_k]$

Coefficient of variation $CV[X_j] = \dfrac{Var[X_j]}{\left(E[X_j]\right)^2}$

≤ 1 for all distributions that are NBUE

New Better than Used in Expectation

$$E[X_j - t \mid X_j > t] \leq E[X_j] \text{ for all } t > 0$$

Assume $CV[X_j] \leq \Delta$

The modified polyhedral relaxation

Assume $CV[X_j] \leq \Delta$

$$\sum_{k \in A} E[X_k] E[C_k^{\Pi}] \geq \frac{1}{2m}\left(\left(\sum_{k \in A} E[X_k]\right)^2 + \sum_{k \in A} E[X_k]^2\right) - \frac{(m-1)(\Delta-1)}{2m} \sum_{k \in A} E[X_k]^2$$

for all $A \subseteq \{1, ,n\}$ and all policies Π

RHS depends only on $E[X_j]$, LP can be solved in polynomial time

Performance guarantees for NBUE

The LP leads to a linear preselective policy with performance $3 - \frac{1}{m}$

WSEPT leads to a priority policy with performance $2 - \frac{1}{m}$

WSEPT: $\frac{E[X_1]}{w_1} \leq \frac{E[X_2]}{w_2} \leq \ldots \leq \frac{E[X_n]}{w_n}$

Adding release dates:

The LP leads to a linear preselective policy with performance $4 - \frac{1}{m}$

WSEPT may be arbitrarily bad

Dealing with precedence constraints

Skutella & Uetz 00:

Combine valid inequalities for the stochastic case with
delay list scheduling by Chekuri, Motwani, Natarajan & Stein 97

Use inequalities

$$\sum_{j\in A} E[X_j]E[C_j^{\Pi}] \geq \frac{1}{2m}\left(\sum_{j\in A} E[X_j]\right)^2 + \frac{1}{2}\sum_{j\in A} E[X_j]^2 - \frac{m-1}{2m}\sum_{j\in A} Var[X_j]$$

$$E[C_j] \geq E[C_i] + E[X_j] \quad \text{if} \quad i \rightarrow j$$

for constructing the list L from an optimum LP-solution

Delay list scheduling

- Use list L for linear preselective policy
- Use tentative decision times to avoid too much idle time

Consider decision time t

Let i be the first unscheduled and available job in L

Let j be the first unscheduled job in L

if j is available **then** start j at t
charge uncharged idle time in $[r_j, t]$ to j

else if there is at least $\beta \cdot E[X_i]$ uncharged idle time in $[r_i, t]$
then start i at t and charge this idle time to i

else set next tentative decision time to $t + \beta \cdot E[X_k]$ for suitable k

Performance guarantees

LP-based delay list scheduling leads to a policy with performance

$$+\frac{m-1}{m\beta}+(1+\beta)\left(1+\max\left\{1\,,\ \frac{m-1}{m}\cdot\max_j\frac{V[X_j]}{E[X_j]^2}\right\}\right)$$

≤ 5.83 for NBUE processing times

Overview

- The model
- Classes of policies
- Computation and approximation
- Open problems

Open problems

- Better computational methods
- When do tentative decision times help?
  - They help for $P \mid \mid \sum w_j C_j$. What about $P \mid \mid C_{\max}$??
- What are optimal policies for exponential models $P \mid p_j \sim \exp \mid \kappa$
  - LEPT/SEPT optimal for $C_{\max}$ / $\sum C_j$ [Weiss & Pinedo '80]
  - What about $\sum w_j C_j$?
- Detailed policy analysis (cost distribution function)
  - #P complete for earliest start scheduling (PERT model)
  - How to approximate?

Additional information

- Contact the speaker:
  Prof. Dr. Rolf Möhring
  TU Berlin, MA 6-1
  Straße des 17. Juni 136
  10623 Berlin
  Tel. +49 30 - 314 24594, Fax: +49 30 - 314 25191
  email: moehring@math.tu-berlin.de

- Browse our web pages:
  http://www.math.tu-berlin.de/coga/
  in particular
  http://www.math.tu-berlin.de/coga/research/scheduling/