Ordonnancement de projets avec échec des activités

Groupe MOGISA

LAAS – CNRS

11 septembre 2008

Roel Leus

KULeuven, Leuven (Belgium)

Bert De Reyck

London Business School (UK)

University College London (UK)

This talk is largely based on

De Reyck, B. and Leus, R. (2008). R&D-project scheduling when activities may fail. *IIE Transactions* 40(4), 367-384.

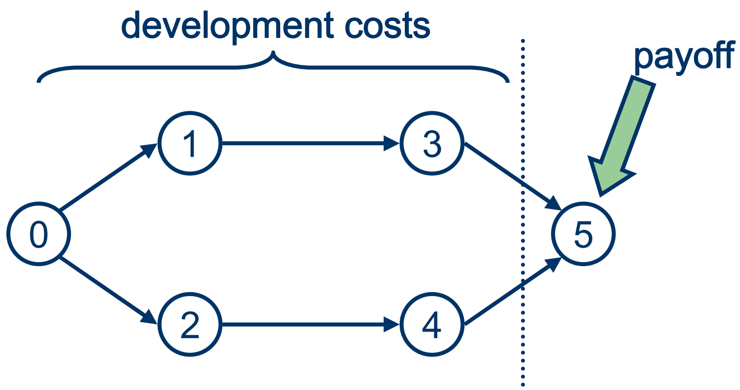
De Reyck, B., Grushka-Cockayne, Y. and Leus, R. (2007). A new challenge in project scheduling: the incorporation of activity failures. *Tijdschrift voor Economie en Management* LII(3), 411-434.

Introduction

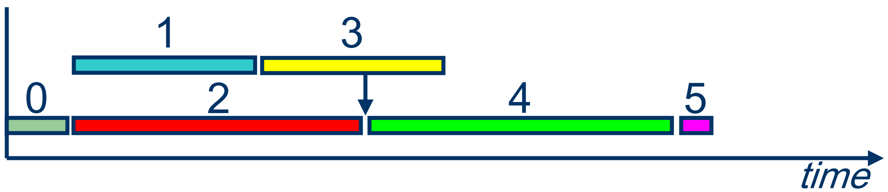
- Project: unique undertaking, aimed at accomplishing a specific non-routine or low-volume task
  $\rightarrow$ Building, software development, infrastructure, …
- Research & Development (R&D) projects, especially NPD: pharmaceuticals, hightech, innovation, ...
- Each project activity has a cost [negative cash flow] and a probability of success – e.g. request for loan, marketing study, apply for building permit, toxicology tests, FDA review, absence of undesirable side-effects, …
- Project pay-off (launch) [positive cash flow] only occurs if *all* activities are successful – especially pharma & agricultural chemicals sector (not modular)
- Time value of money – discounting
- Development of a project schedule with objective: expected NPV (eNPV)

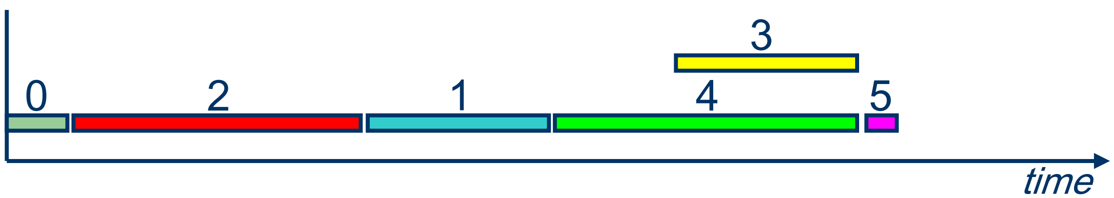
Introduction (2)

precedence network of activities :

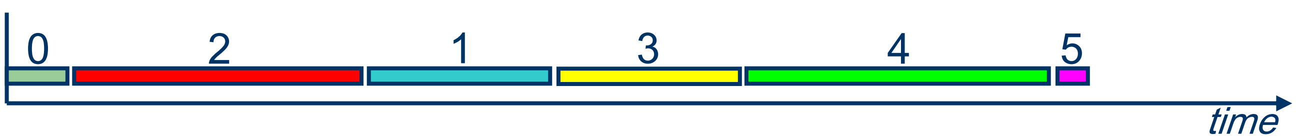


- No (renewable) resource constraints, no duration uncertainty, uncorrelated activity success





"serial" schedule



- Trade-off early project completion *if successful* vs. reduction of costs *if failure*

Problem formulation

N	$= \{0, 1, \ldots, n\}$, the set of project activities
c_i	cash flow of activity $i \in N \backslash \{n\}$, non-positive integer; incurred at the start of the activity
C	integer end-of-project payoff, ≥ 0; received at the start of activity n
d_i	duration of activity $i \in N$
p_i	probability of technical success (PTS) of activity $i \in N \backslash \{n\}$; outcome known at the end of the activity
r	continuous discount rate: the present value of cash flow c incurred at time t equals ce^{-rt}
A	partial order on N representing precedence constraints
δ	project deadline

- We let activity 0 be a dummy representing project start: $c_0 = 0$, $d_0 = 0$, $p_0 = 1$
- Decision variables:

s_i	starting time of activity i; starting-time vector $\mathbf{s}$ is a schedule

- Constraints:
  - A imposes $s_i + d_i \leq s_j$, $\forall (i,j) \in A$
  - deadline δ

Problem formulation & properties

Additional variables:

$$q_i(\mathbf{s}) = \text{probability that all activities ending no later than } s_i \text{ succeed}$$
$$= \prod_{\substack{k \in N: \\ s_k + d_k \leq s_i}} p_k; \text{ remark that } q_n \text{ is a constant.}$$

Objective: maximize the expected net present value (eNPV) of the schedule:

$$\max \qquad q_n C e^{-r s_n} + \sum_{i=1}^{n-1} q_i(\mathbf{s}) c_i e^{-r s_i}$$

The resulting problem is called *R&D Project Scheduling Problem* (*RDPSP*)

Theorem 1: if $r = 0$ and $\delta \geq \sum_{i \in N} d_i$ then an optimal schedule exists that imposes a complete order on N

$\rightarrow$ Problem LCT = “least-cost testing”: solution space restricted to complete orders

Properties (2)

If $r = 0$ and $\delta \geq \Sigma_{i \in N} d_i$ then

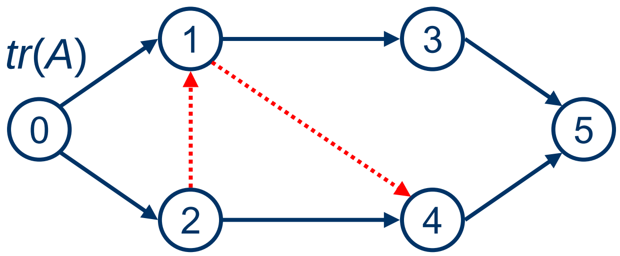
- if $A = \varnothing$ (*no precedence constraints*) then each optimal complete order sequences the activities in non-increasing order of $c_i / (1 - p_i)$ (Mitten, 1960; Butterworth, 1972)
- if A consists of a number of *parallel chains*, a poly-time algorithm exists (Chiu et al., 1999)
- if $G(N,A)$ is *series-parallel*, a poly-time algorithm exists (Monma and Sidney, 1979)

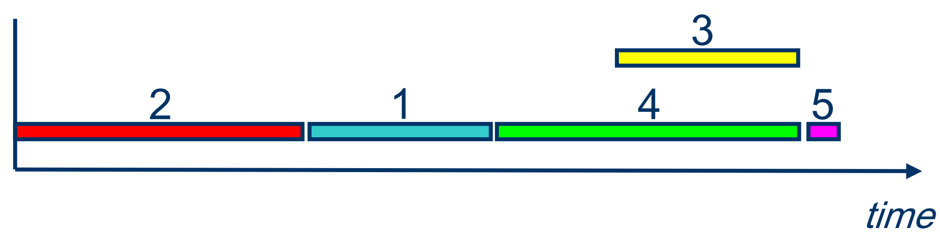
Theorem 2: RDPSP is NP-hard, even if $r = 0$, $C = 0$, all $d_i = 1$ and $\delta \geq \Sigma_{i \in N} d_i$ (reduction from 1 | *prec* | $\Sigma w_j C_j$ (Lenstra and Rinnooy Kan, 1978))

Corollary: LCT is NP-hard under the same conditions as Theorem 2

Order-theoretic approach to scheduling

- Objective function: $\max \quad q_n C e^{-r s_n} + \sum_{i=1}^{n-1} q_i(\mathbf{s}) c_i e^{-r s_i}$
- *E* is an *acyclic extension* of *A* if $E \supseteq A$ and $G(N,E)$ acyclic
- For given extension *E*:
  - values ("information flows") $y_i(E)$ are implicit; we substitute y_i for q_i
  - optimal start times via CPM late-start ($s_0 = 0$ if eNPV ≥ 0; $s_n = \delta$ otherwise)





$A = \{ (0,1),(0,2),(0,3),(0,4),(0,5),(1,3),(1,5),(2,4),(2,5),(3,5),(4,5) \}$

$E = A \cup \{ (1,4),(2,1) \}$

$y_1(E) = p_2$; $y_2(E) = 1$; $y_3(E) = p_1p_2$; $y_4(E) = p_1p_2$; $y_5(E) = p_1p_2p_3p_4$ (with $p_0 = 1$)

- Instead of producing starting times directly, we enumerate all acyclic extensions of *A*. This enumeration is embedded in a B&B procedure.

Computational experiments*

- CPU time is strongly dependent on |*A*| (~ order strength)
- $Imp(\mathbf{s}_1,\mathbf{s}_2) = (g(\mathbf{s}_2) - g(\mathbf{s}_1)) / |g(\mathbf{s}_1)|$
- Truncated B&B. Performance with varying time limits, for $n = 25$:

time limit	opt (/60)	nodes	$Imp(\mathbf{s}_{(0)}, \mathbf{s}_{(i)})$
0	0	0	0.00%
1	20	62,240	+21.67%
5	23	274,900	+23.61%
20	28	996,001	+25.33%
100	35	4,045,108	+27.70%
250	38	8,958,914	+28.60%
1000	41	31,498,214	+31.59%

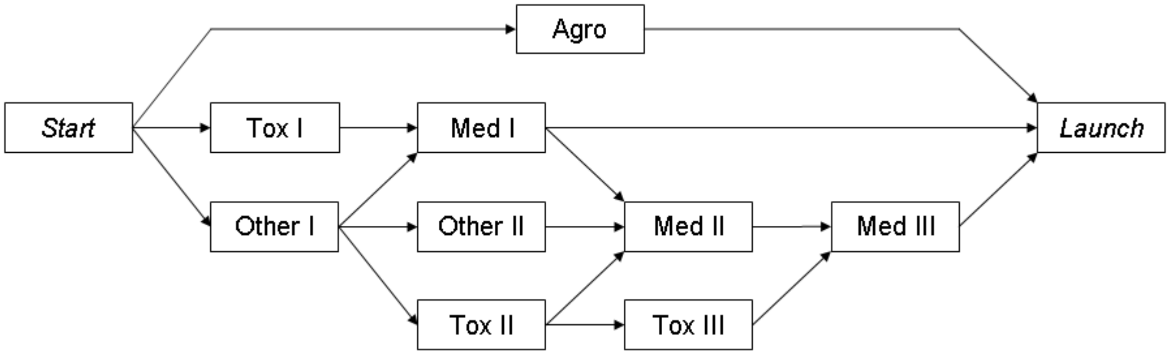
$\mathbf{s}_{(i)}$ = schedule for time limit i ;

$\mathbf{s}_{(0)}$ is a heuristic LB

* Coded in C using MS VC++ 6.0; running on Dell Optiplex GX620, Intel Pentium 4, 2.80 GHz processor, 1 GB RAM

* Instances generated using RanGen (without cash flows and probabilities)

Real-life example



Drug development project, biotech company, Cambridge, England.

task	cash flow (£)	duration (months)	PTS
Agro	−12,000,000	60	100%
Tox I	−300,000	6	75%
Other I	−1,000,000	8	100%
Med I	−200,000	8	80%
Other II	−300,000	8	100%
Tox II	−100,000	6	75%
Med II	−200,000	10	80%
Tox III	−700,000	9	75%
Med III	−400,000	20	60%
Launch	300,000,000	-	-

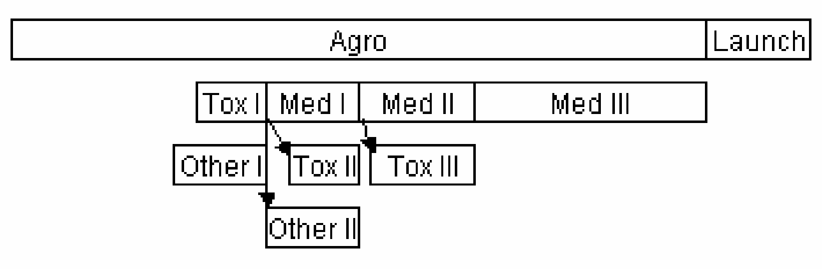
Estimated overall probability of success is 16.2%.

r = 1% per month.

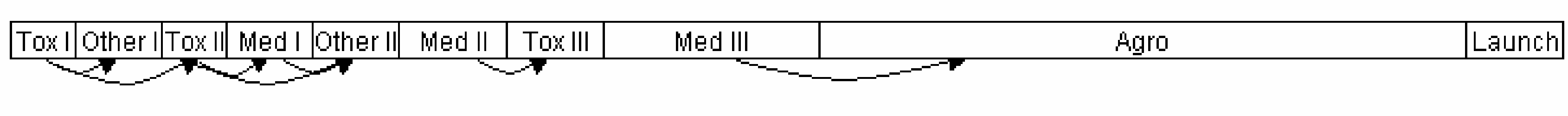
[Crama, P., De Reyck, B., Degraeve, Z. and Chong, W. 2007. R&D project valuation and licensing negotiations at Phytopharm plc. *Interfaces* 37(5), 472–487.]

Real-life example (2)

- CPM Late-start schedule: eNPV of approx. £13 million



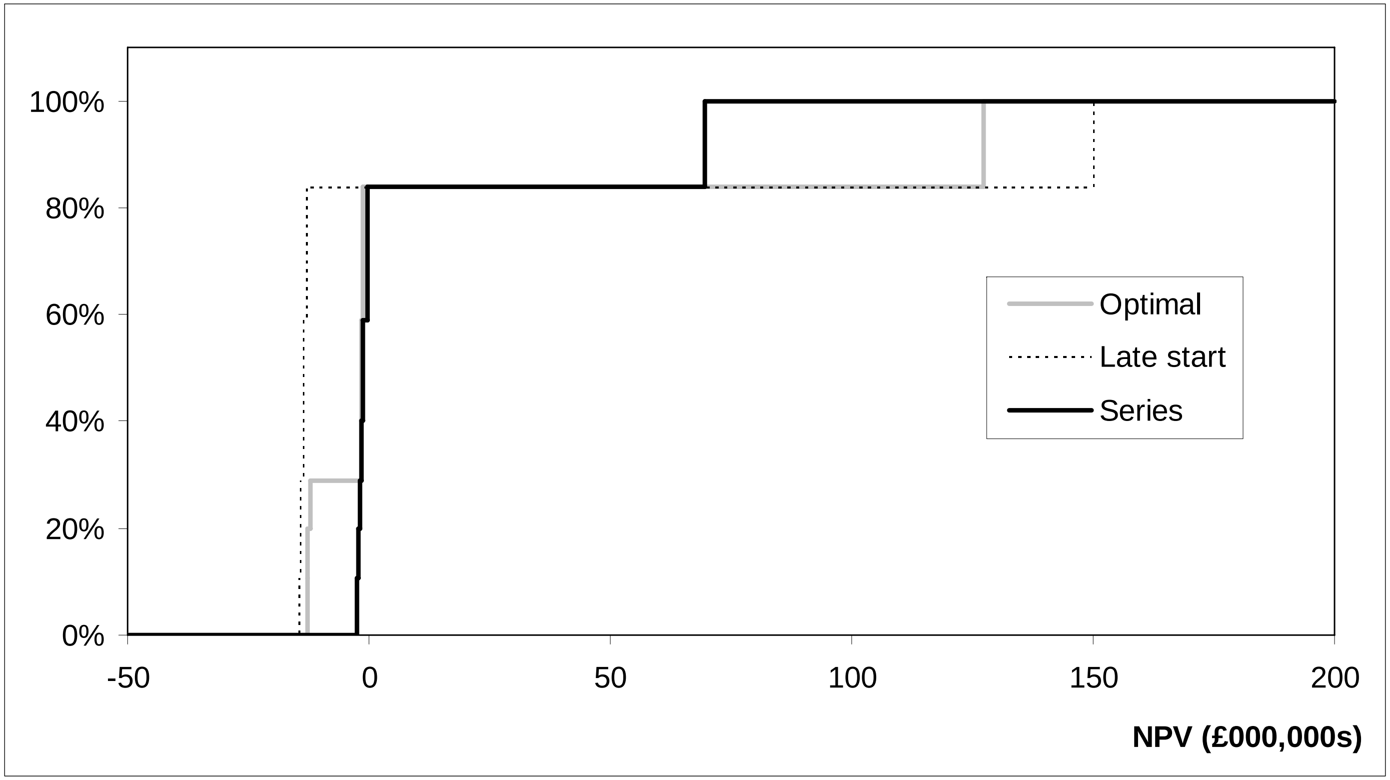
- Serial schedule: eNPV of approx. £10 million



- Optimal schedule: eNPV of approx. £16 million

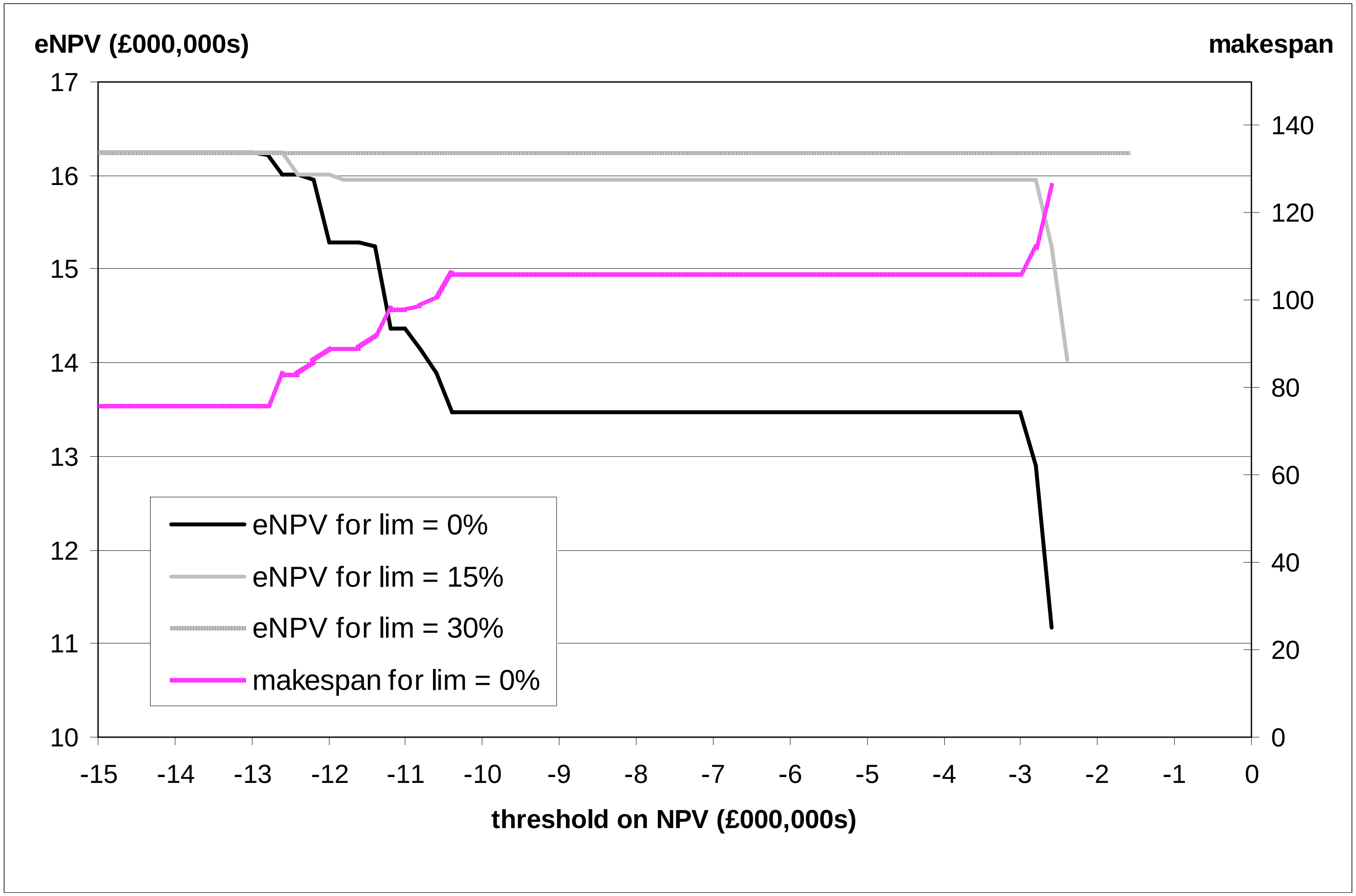
Risk preferences

- *Expected* NPV vs. *actual* project realizations
- cdf of the NPV of a schedule: evaluate entire risk profile
- Determining the cdf of the NPV of an arbitrary schedule in time $O(n \log n)$ (to be compared with stochastic activity durations)



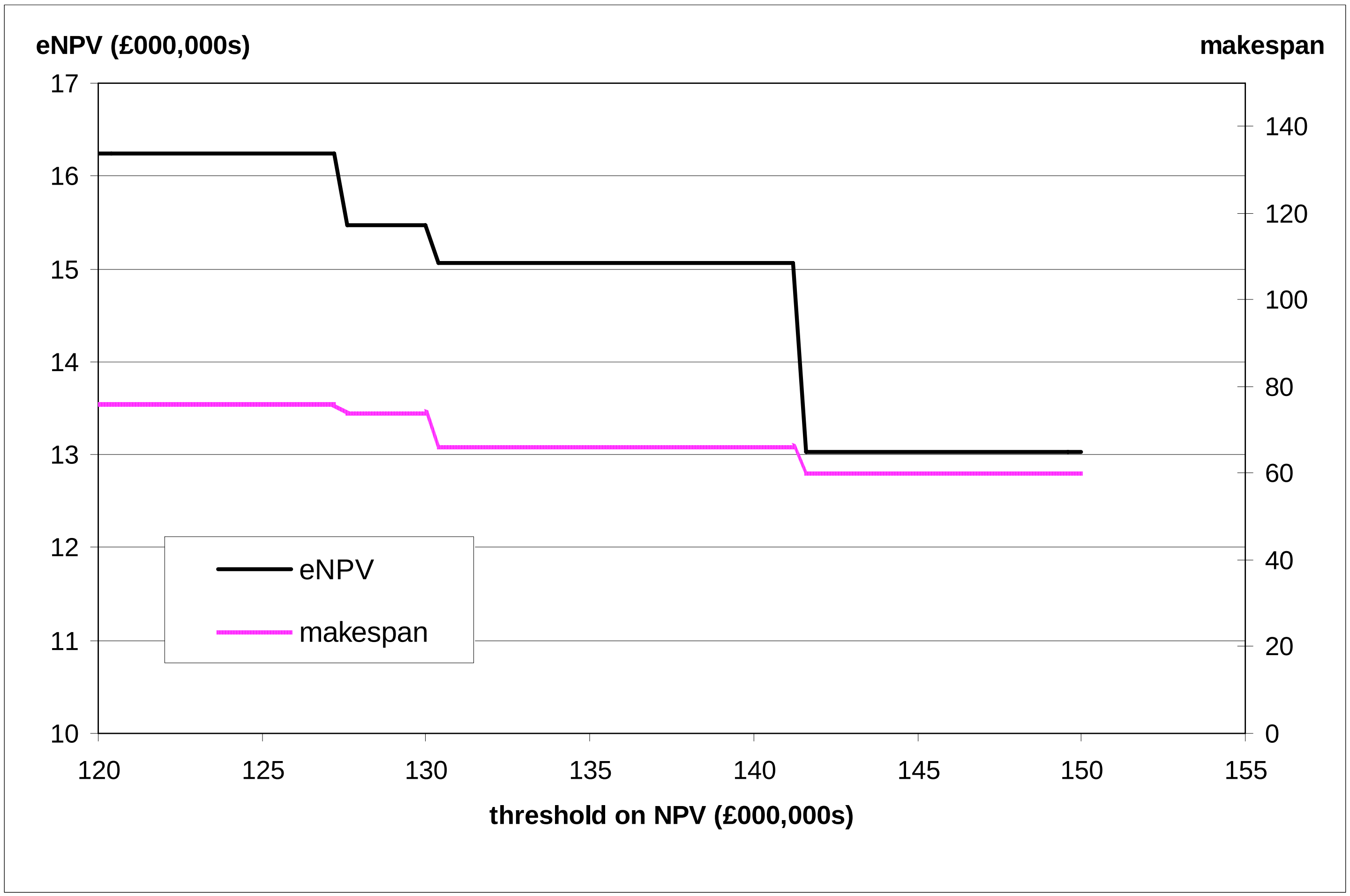
Risk preferences (2)

- « downside risk » = probability that NPV < threshold



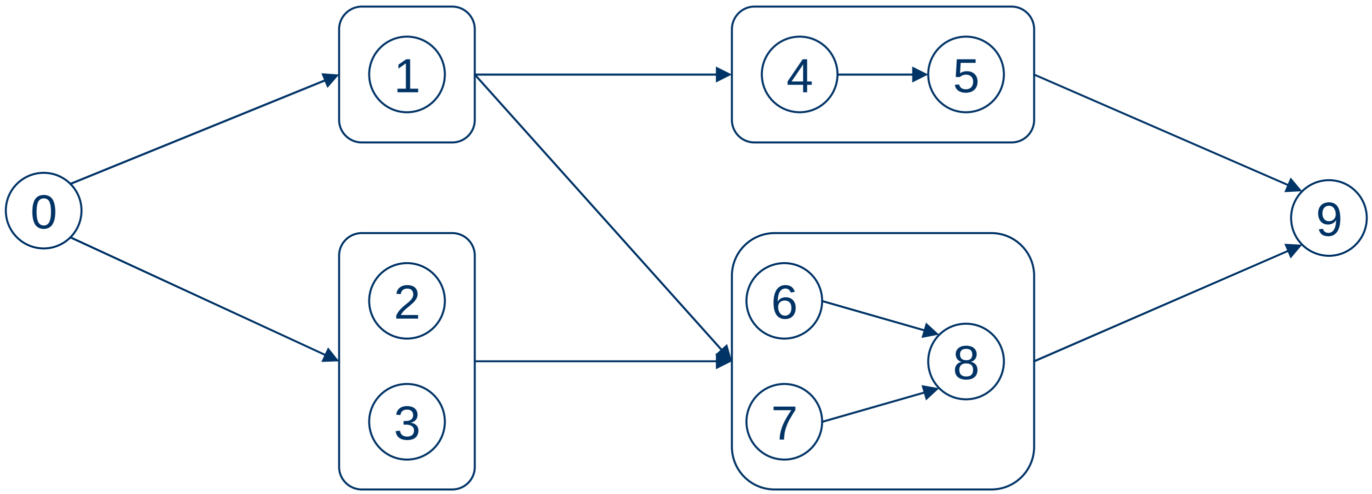
Risk preferences (3)

- « upside potential » = probability that NPV ≥ threshold ?
  Here: NPV *in case of project success* should not be lower than a threshold.



General problem formulation

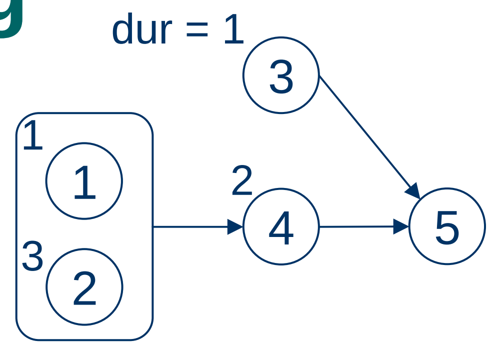
- Literature on sequential testing / scheduling and *k*-out-of-*n* reliability systems:
  1-out-of-*n* = ‘parallel system’; *n*-out-of-*n* = ‘series system’
  $\rightarrow$ only *sequential* testing; no discounting; project success via *no.* of successful act.
- Set of modules $M = \{ 0, 1, ..., m \}$; each module $k \in M$ has a set of activities N_k
- Precedence constraints may apply both between and within modules

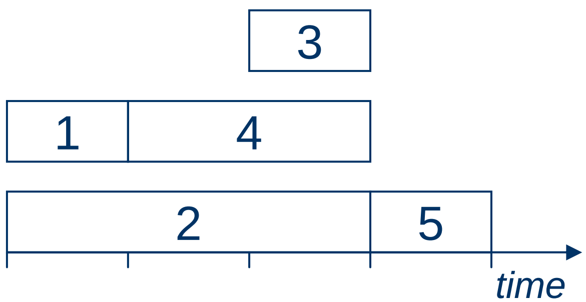


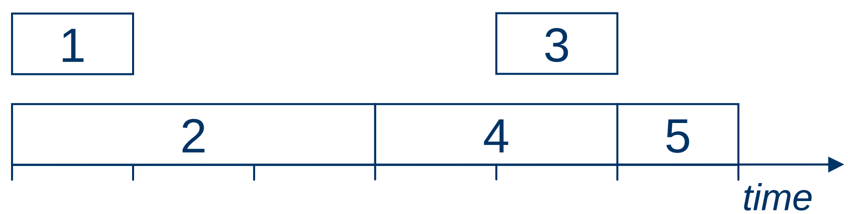
- Each activity has a (fixed) duration, a cost and a probability of success
- A module completes and is successful when one of its activities succeeds
- End-of-project payoff is obtained if all modules are successful

Stochastic scheduling

- What is a solution? A schedule?







- In line with the literature on stochastic programming, especially stochastic scheduling, a solution is a *policy* that defines *actions* at *decision times*
- A *globally optimal* policy is optimal over the class of *all* policies
- In stochastic scheduling, one usually restricts attention to subclasses that have a simple combinatorial representation and where decision points are limited in number
- Activity selection!

Single-module projects

= a 1-out-of-n system: alternative technologies, trials for the same result, or fallback options; *selection* of activities now becomes an issue!

- Define an *elementary policy* as a policy that adheres to a deterministic schedule and imitates it until project completion or the first activity success (*not* selecting a task equates with starting time = + ∞)

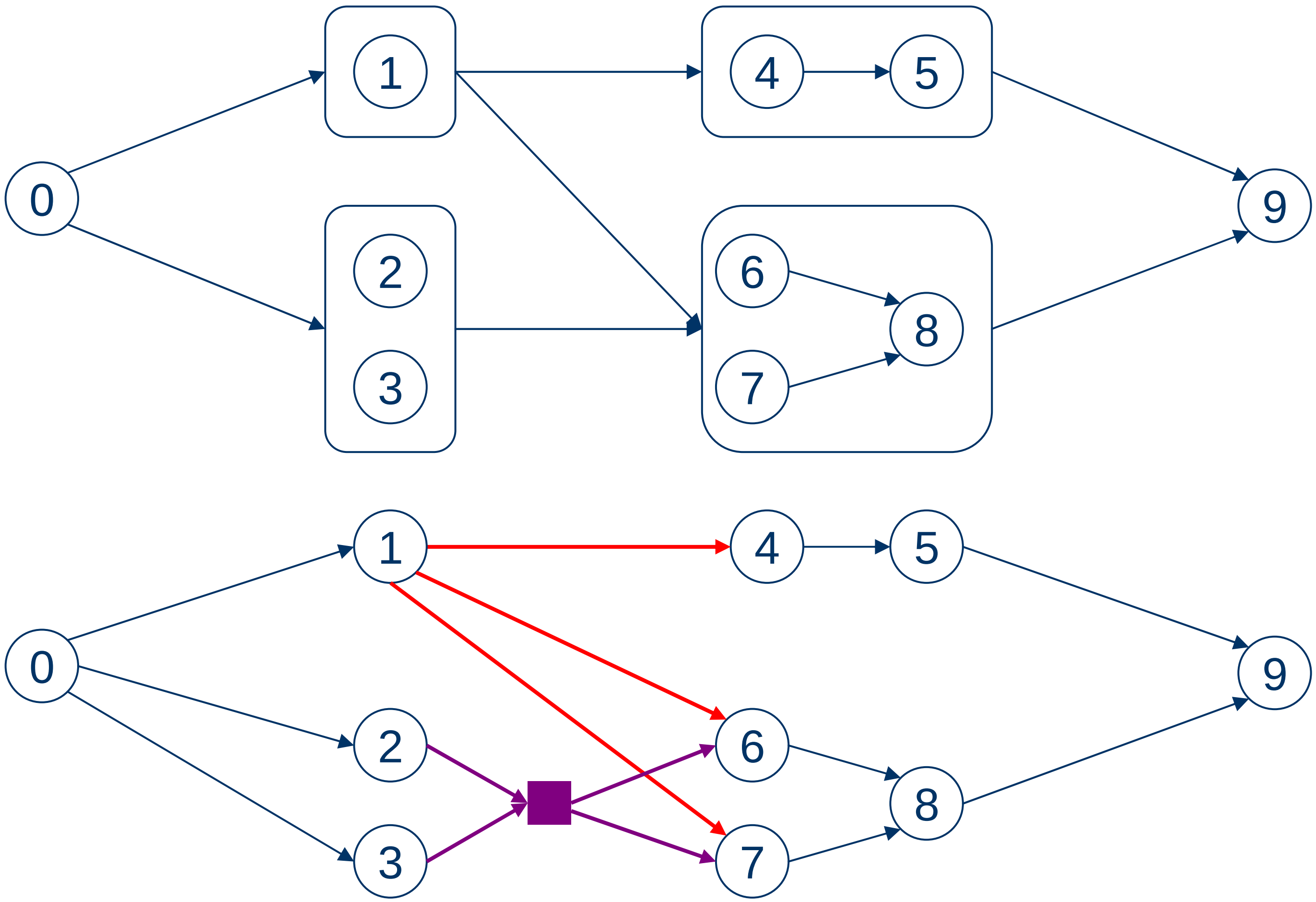
  <u>Theorem 3:</u> an optimal elementary scheduling policy is globally optimal for 1-out-of-n systems

- <u>Theorem 4</u>: if the discount rate is 0 then an optimal schedule exists for 1-out-of-n that imposes a complete order on the set of jobs

- <u>Theorem 5</u>: 1-out-of-n is NP-hard, even if the discount rate is 0

  1-out-of-n is equivalent with n-out-of-n with zero discount rate, for large payoff; Unfortunately, this equivalence no longer holds when the discount rate $\neq 0$ because of the timing of obtaining C

The general problem…



⇒ Use AND/OR-type precedence constraints? Yes, but

– AND/OR precedence constraints combined with non-regular objectives?
Include requirement of “success” in some of the precedence constraints?

Summary & conclusions

- Model for scheduling R&D projects to maximize the expected NPV when the activities have inherent possibility of failure: intermediate step of activity modules
- The problem is NP-hard
- For n-out-of-n systems (RDPSP):
  - Branch-and-bound algorithm
  - Incorporation of risk preferences
- Some characteristics of dominant sets of solutions
  - n-out-of-n: a late-start schedule for an extension of the input graph (determ. NPV)
  - 1-out-of-n: early start may be in order; no determ. NPV anymore ‘
- ‘Stylized’ model: possibly not always of immediate use for decision support