KATHOLIEKE UNIVERSITEIT
LEUVEN

R&D Project Planning with Multiple Trials in Uncertain Environments

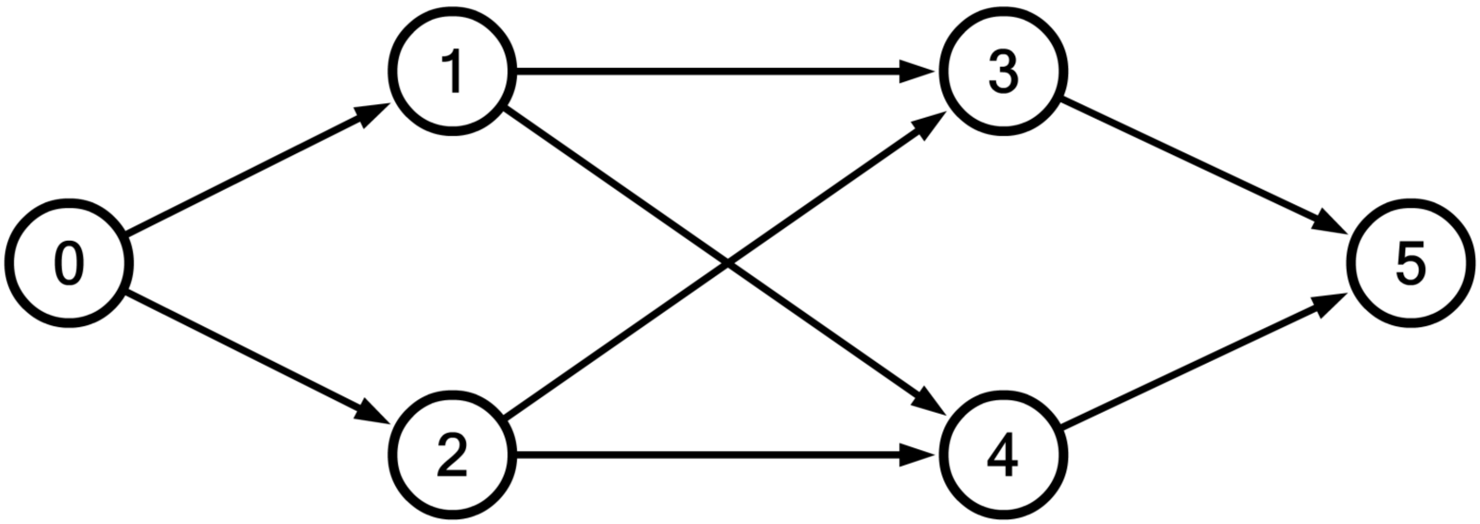
(Liège, ORBEL January 2010)

Stefan Creemers
Roel Leus

Problem Statement

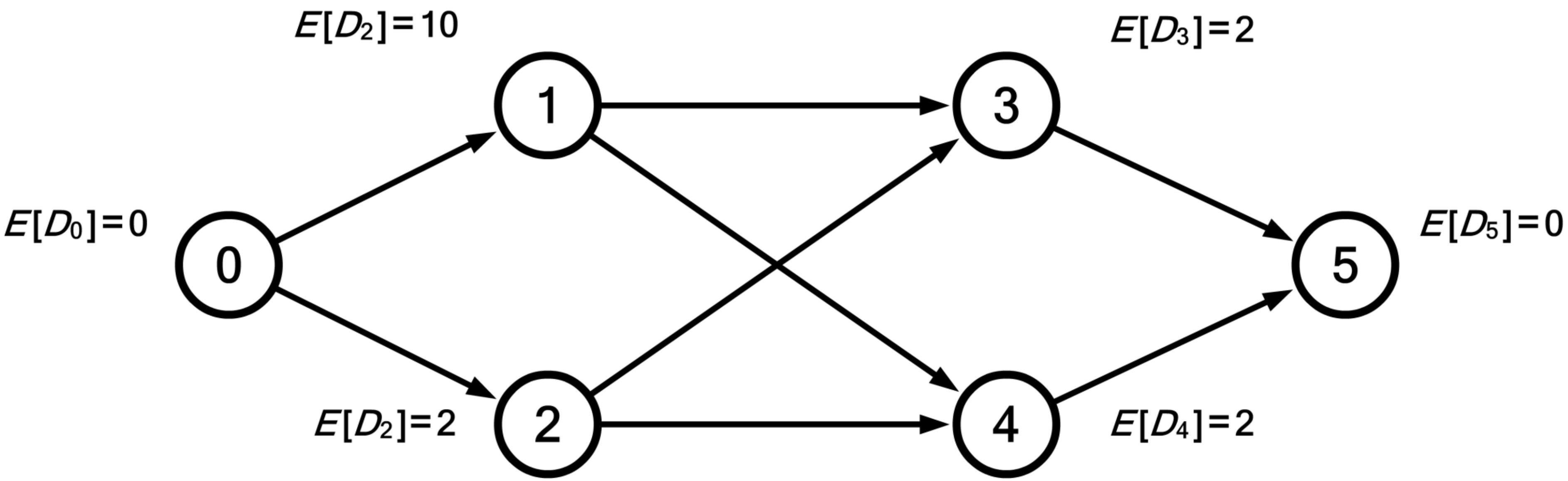
- A *project* = a set of activities that are performed to achieve a common target
- Goal = scheduling the activities to maximize the NPV of projects in which:
  - Activities can fail
  - Activities that pursue the same result are grouped in “modules”
  - Each module needs to be successful for the project to succeed
  - A module is successful if at least one of its activities succeed
    - ⇨ Not all activities in the network have to be started in order for the project to be successful
    - ⇨ Upon failure of all activities in the module, the module fails, resulting in overall project failure
- This is common in R&D (especially in NPD): pharmaceuticals, software development, …

Problem Statement



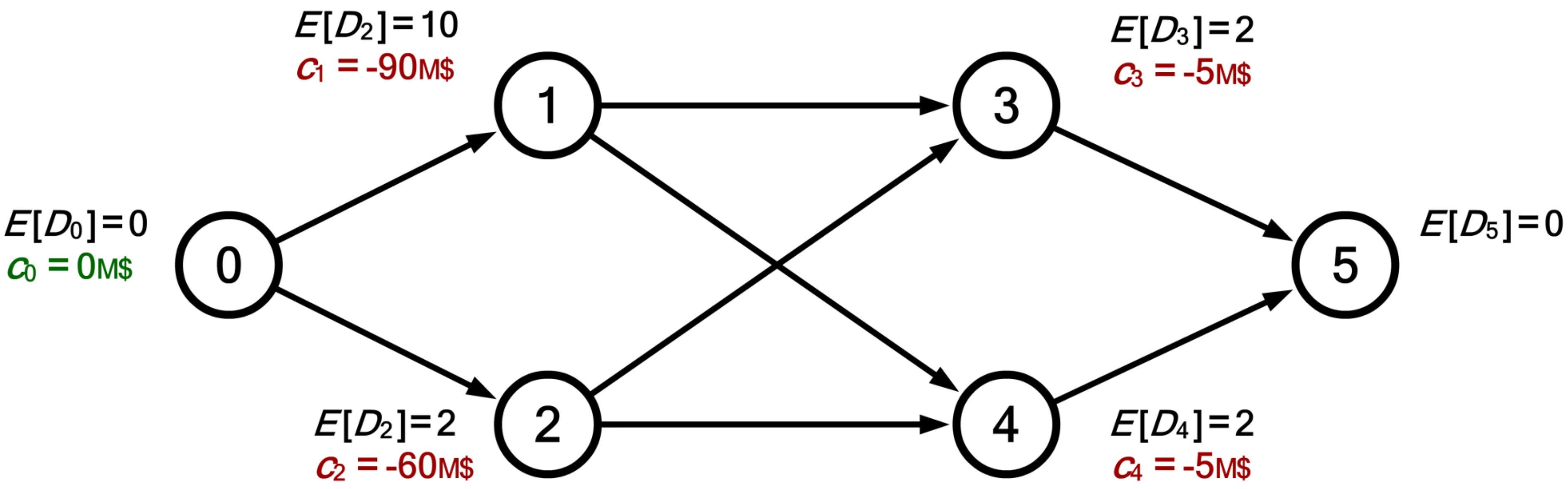
- Project network with *n* activities (activity = on the node)

Problem Statement



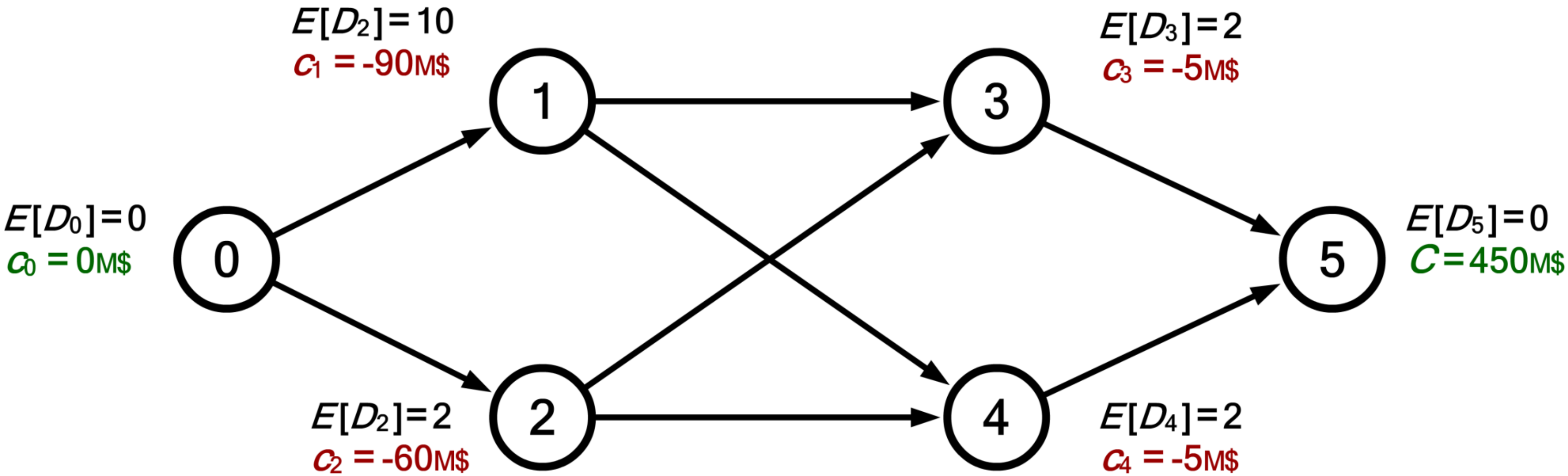
- **Project network with *n* activities (activity = on the node)**
- **Stochastic activity durations: expected duration $E[D_j]$ of activity *j***

Problem Statement



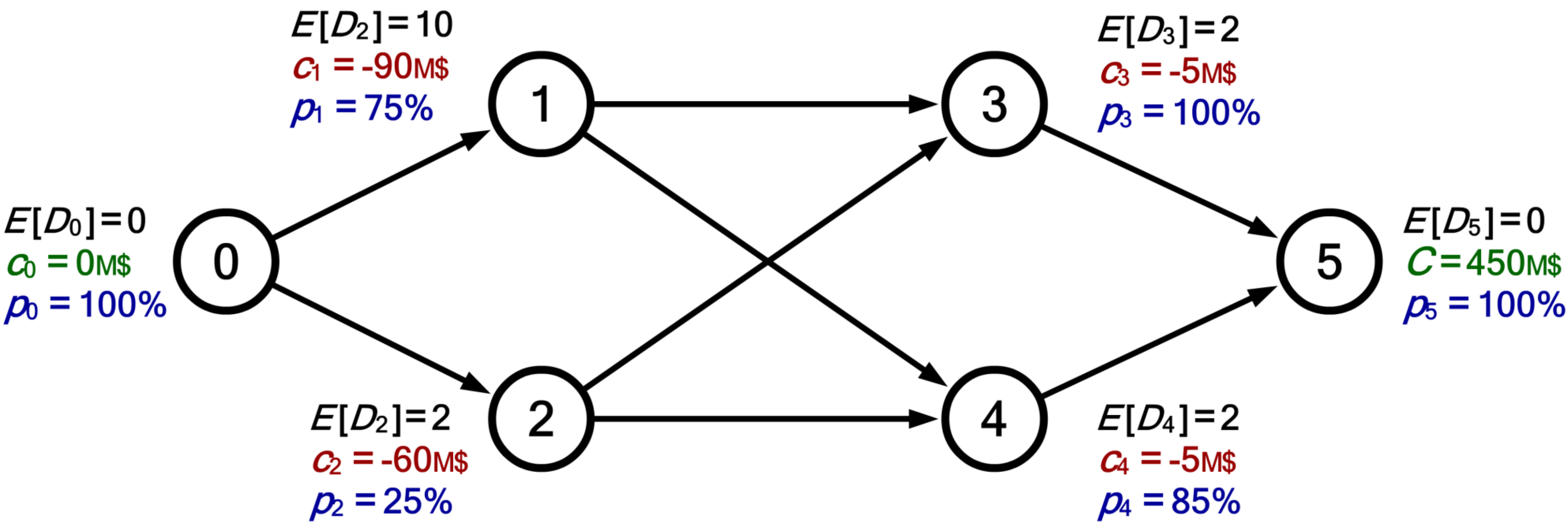
- Project network with n activities (activity = on the node)
- Stochastic activity durations: expected duration $E[D_j]$ of activity j
- Expected-NPV objective: cash flow c_j is incurred at the start of activity j

Problem Statement



- **Project network with *n* activities (activity = on the node)**
- **Stochastic activity durations: expected duration $E[D_j]$ of activity *j***
- **Expected-NPV objective: cash flow c_j is incurred at the start of activity *j***
- **End-of-project payoff *C* obtained upon overall project success**

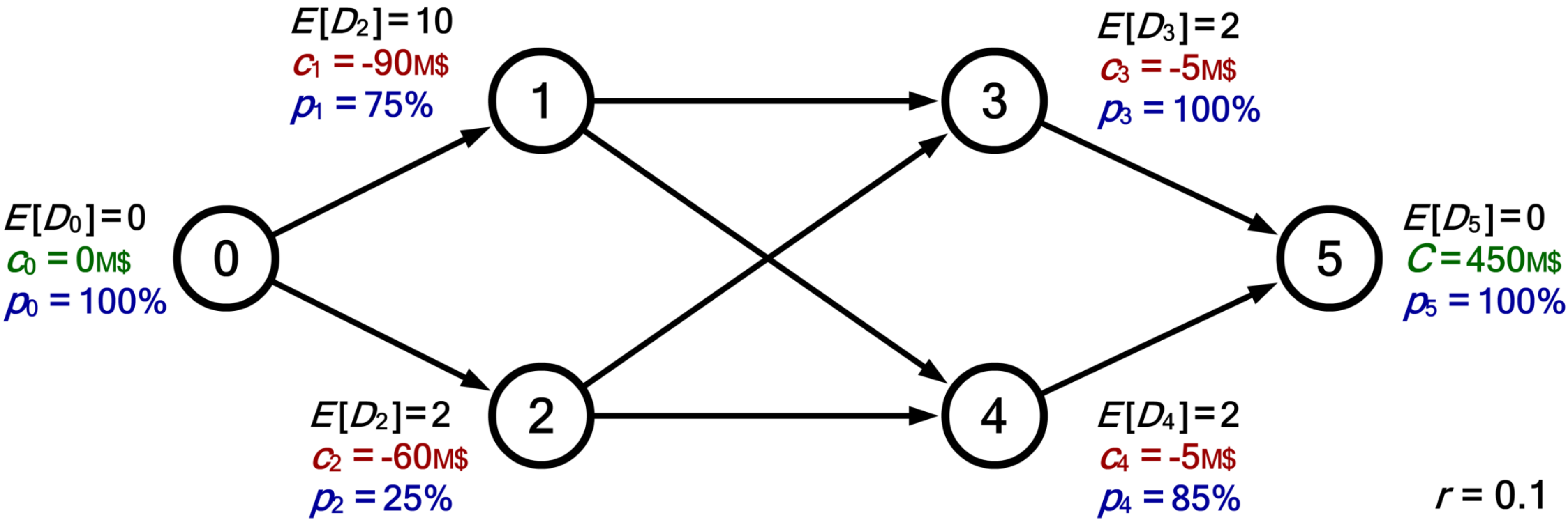
Problem Statement



- **Project network with *n* activities (activity = on the node)**
- **Stochastic activity durations: expected duration $E[D_j]$ of activity *j***
- **Expected-NPV objective: cash flow c_j is incurred at the start of activity *j***
- **End-of-project payoff *C* obtained upon overall project success**
- **Failures: each activity *j* has a probability of technical success p_j**

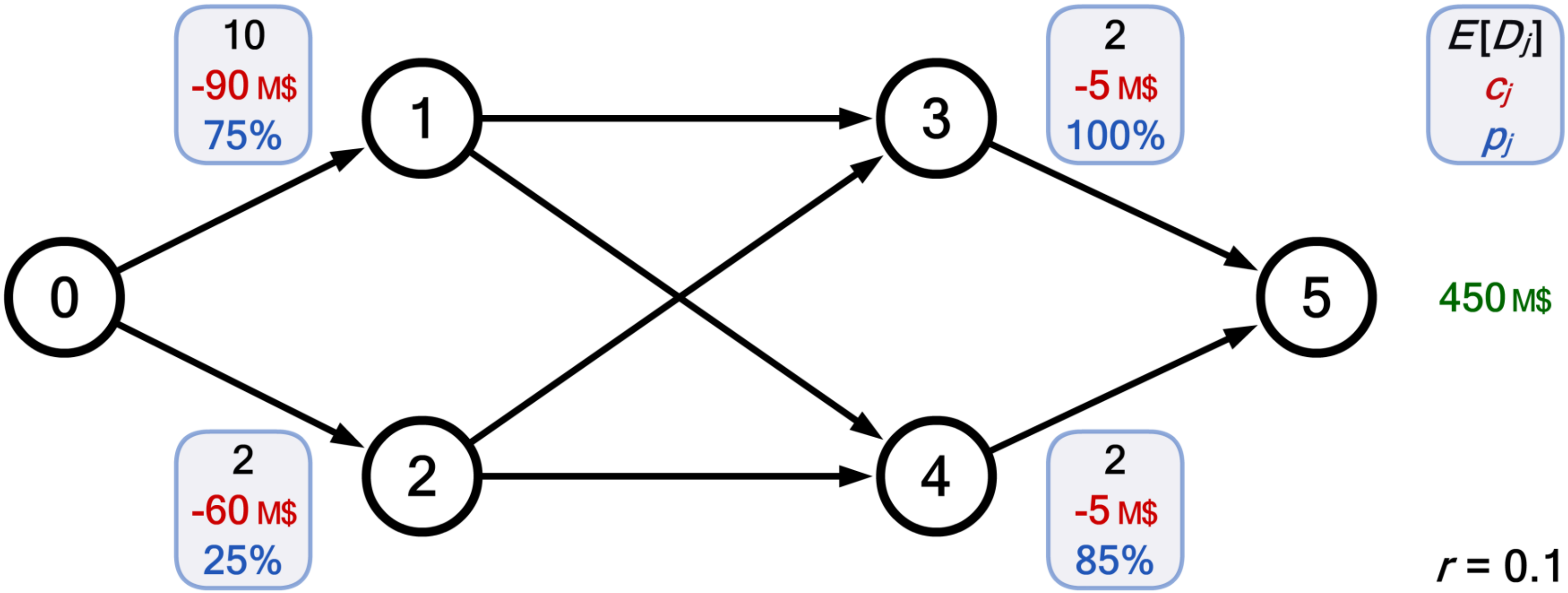
Problem Statement

KATHOLIEKE UNIVERSITEIT LEUVEN



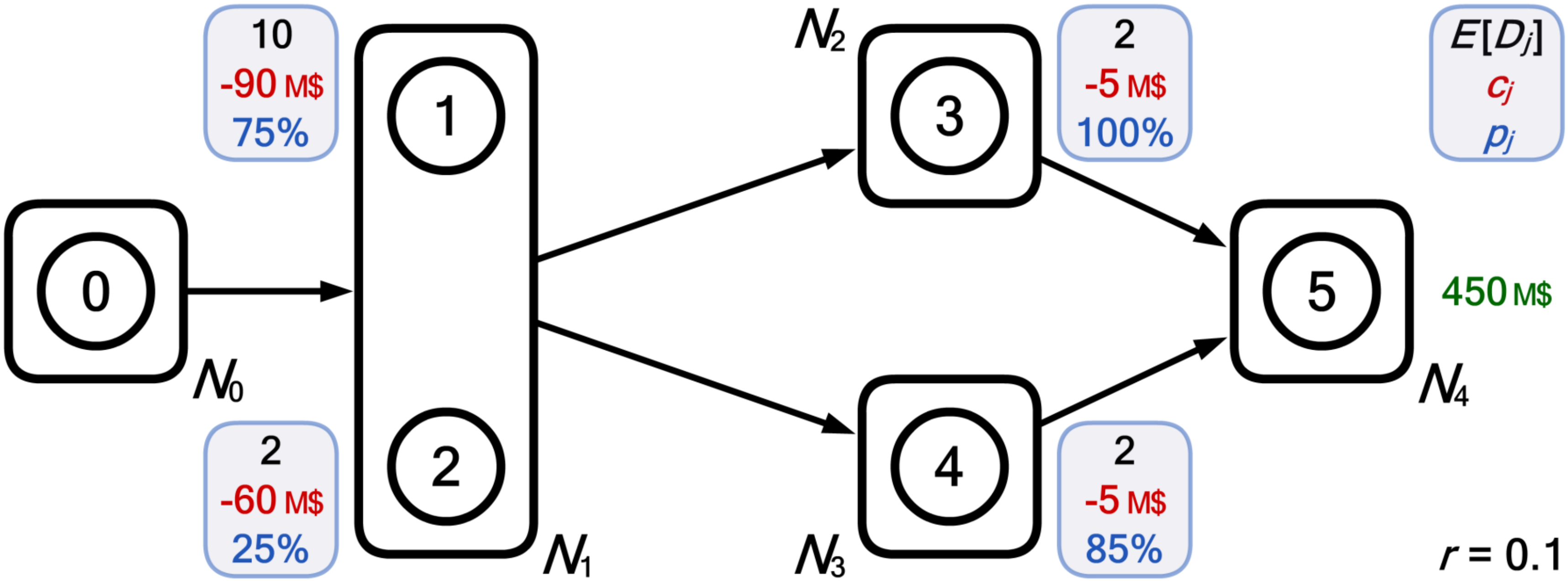
- **Project network with *n* activities (activity = on the node)**
- **Stochastic activity durations: expected duration $E[D_j]$ of activity *j***
- **Expected-NPV objective: cash flow c_j is incurred at the start of activity *j***
- **End-of-project payoff *C* obtained upon overall project success**
- **Failures: each activity *j* has a probability of technical success p_j**
- **Time value of money => discount rate *r***

Problem Statement



- **Project network with *n* activities (activity = on the node)**
- **Stochastic activity durations: expected duration $E[D_j]$ of activity *j***
- **Expected-NPV objective: cash flow c_j is incurred at the start of activity *j***
- **End-of-project payoff *C* obtained upon overall project success**
- **Failures: each activity *j* has a probability of technical success p_j**
- **Time value of money => discount rate *r***

Problem Statement



- **Project network with *n* activities (activity = on the node)**
- **Stochastic activity durations: expected duration $E[D_j]$ of activity *j***
- **Expected-NPV objective: cash flow c_j is incurred at the start of activity *j***
- **End-of-project payoff *C* obtained upon overall project success**
- **Failures: each activity *j* has a probability of technical success p_j**
- **Time value of money => discount rate *r***
- **- *m* modules N_i**

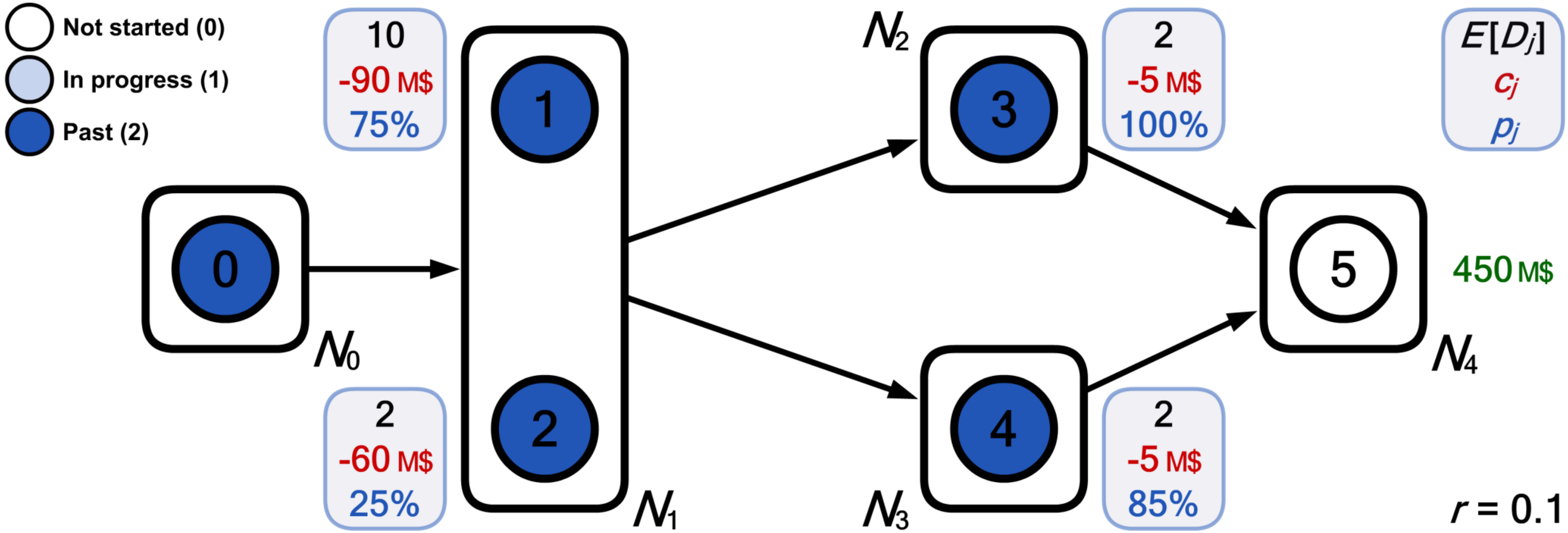
Solution methodology

- Exponentially distributed durations => use of a Continuous-Time Markov Chain (CTMC) to model the statespace
- State of an activity *j* at time *t* can be:
  - Not started
  - In progress
  - Past (successfully finished, failed or considered redundant because another activity of its module has completed successfully)
- Size of statespace has upper bound 3^n. Most states do not satisfy precedence constraints => a strict definition of the statespace is required and provided in Creemers et al. (2010)*

$\Rightarrow$ Backward SDP-recursion

*Creemers S, Leus R, Lambrecht M (2010). Scheduling Markovian PERT networks to maximize the net present value. Operations Research Letters, vol. 38, no. 1, pp. 51 - 56.

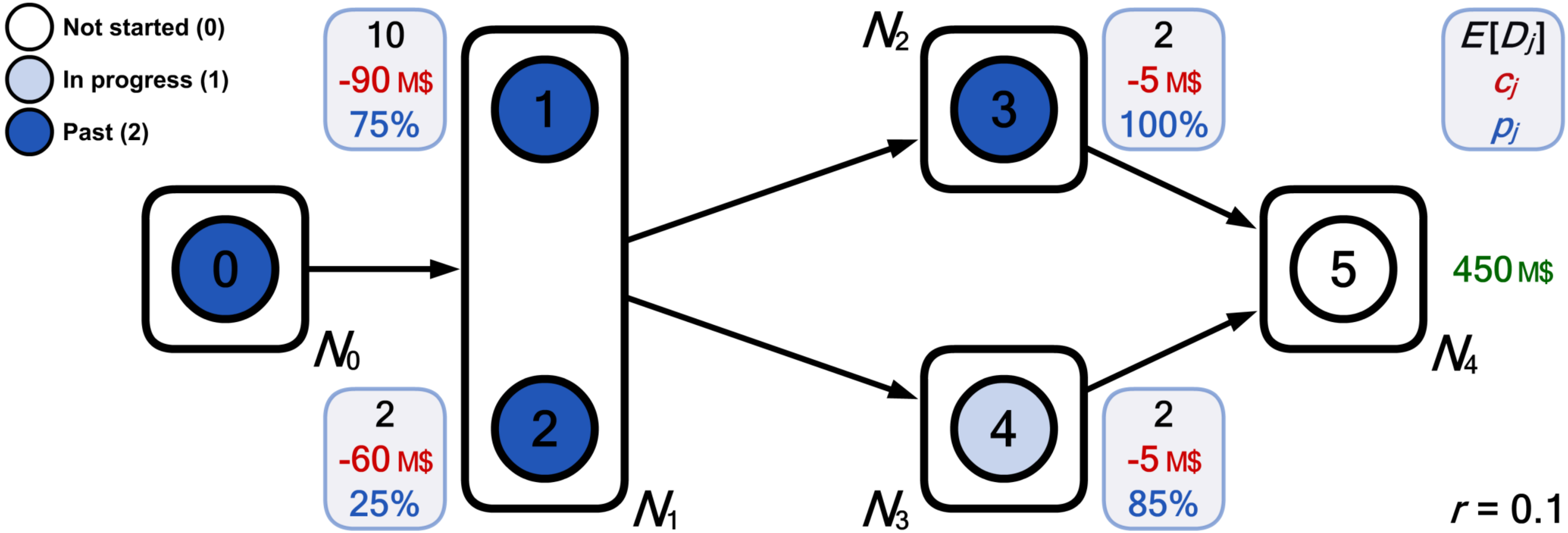
Solution methodology



(2,2,2,2,2,0) [450M$]

Project value upon entry of the final state = project payoff

Solution methodology



(2,2,2,2,2,0) [450M$]

→ (2,2,2,2,1,0) [318.75M$]

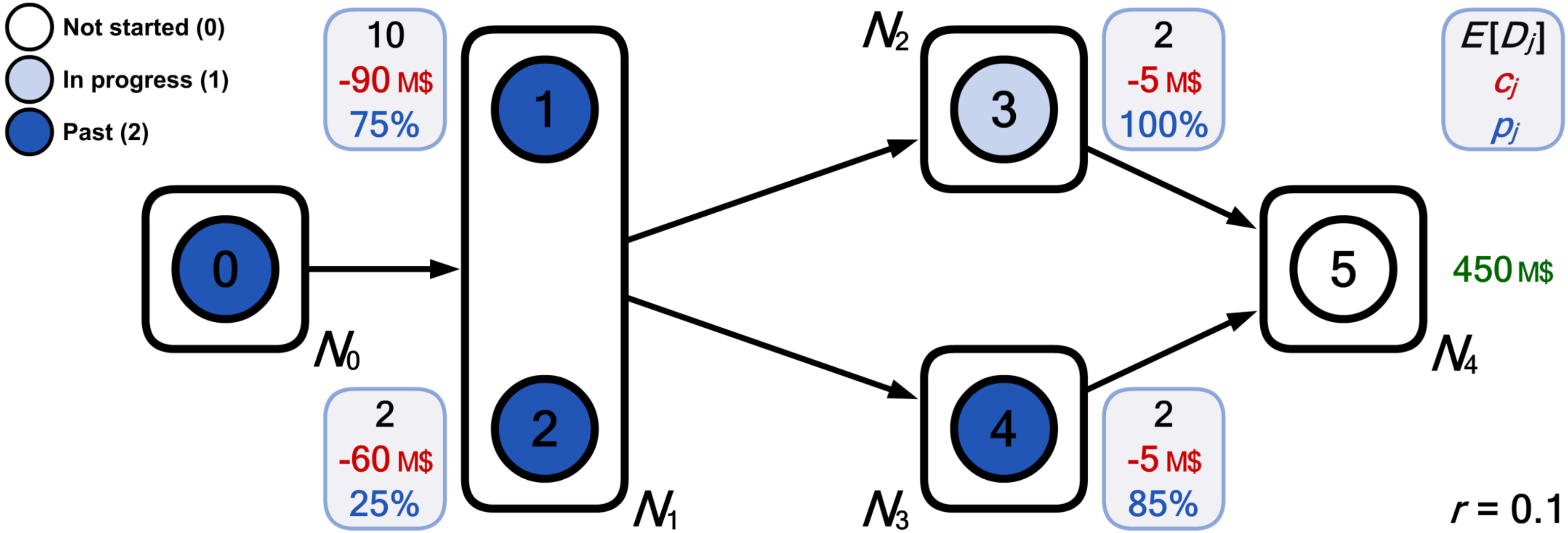
Discount factor: $(1/D_j).(r+(1/D_j))^{-1}$

D_4 = 2 => discount factor = 0.83

NPV upon state entry if success = 375

p_4 = 0.85 => NPV upon state entry = 318.75

Solution methodology



(2,2,2,2,2,0) [450M$]
→ (2,2,2,2,1,0) [318.75M$]
→ (2,2,2,1,2,0) [375M$]

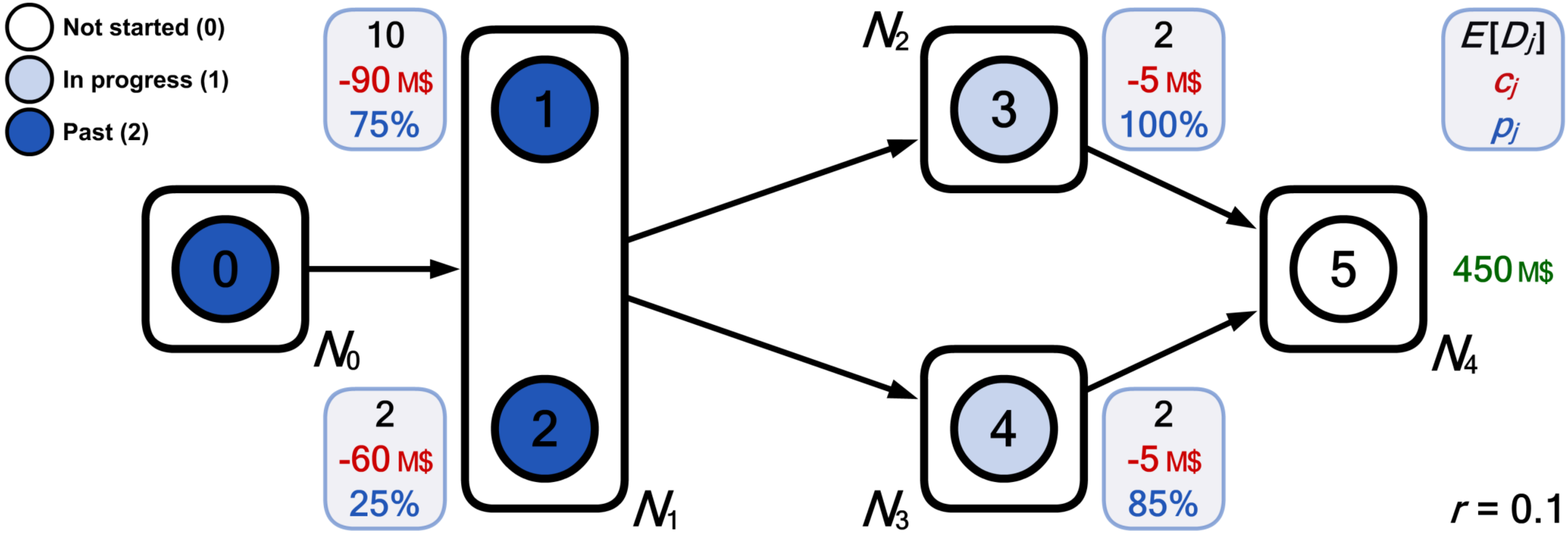
Discount factor: $(1/D_j).(r+(1/D_j))^{-1}$

D_3 = 2 => discount factor = 0.83

NPV upon state entry if success = 375

p_3 = 1.00 => NPV upon state entry = 375

Solution methodology



(2,2,2,2,2,0) [450M$]
→ (2,2,2,2,1,0) [318.75M$]
→ (2,2,2,1,2,0) [375M$]
  → (2,2,2,1,1,0) [289.77M$]

Discount factor = 0.91

Probability of finishing activity *j* first : $(1/D_j).(\Sigma(1/D_j))^{-1}$

=> Probability 3 finishes first = 50% & p_3 = 100%

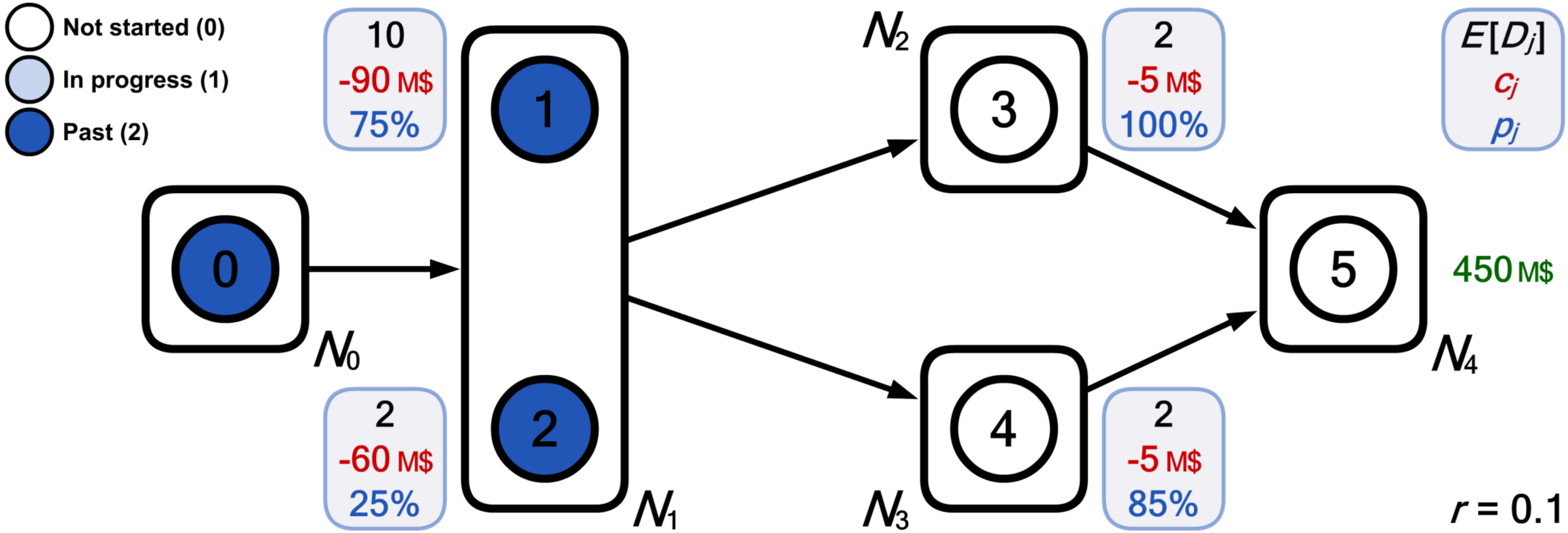
0.5 x 0.91 x 1.00 x 318.75 = 144.89

=> Probability 4 finishes first = 50% & p_4 = 0.85%

0.5 x 0.91 x 0.85 x 375 = 144.89

=> NPV upon state entry = 289.77

Solution methodology

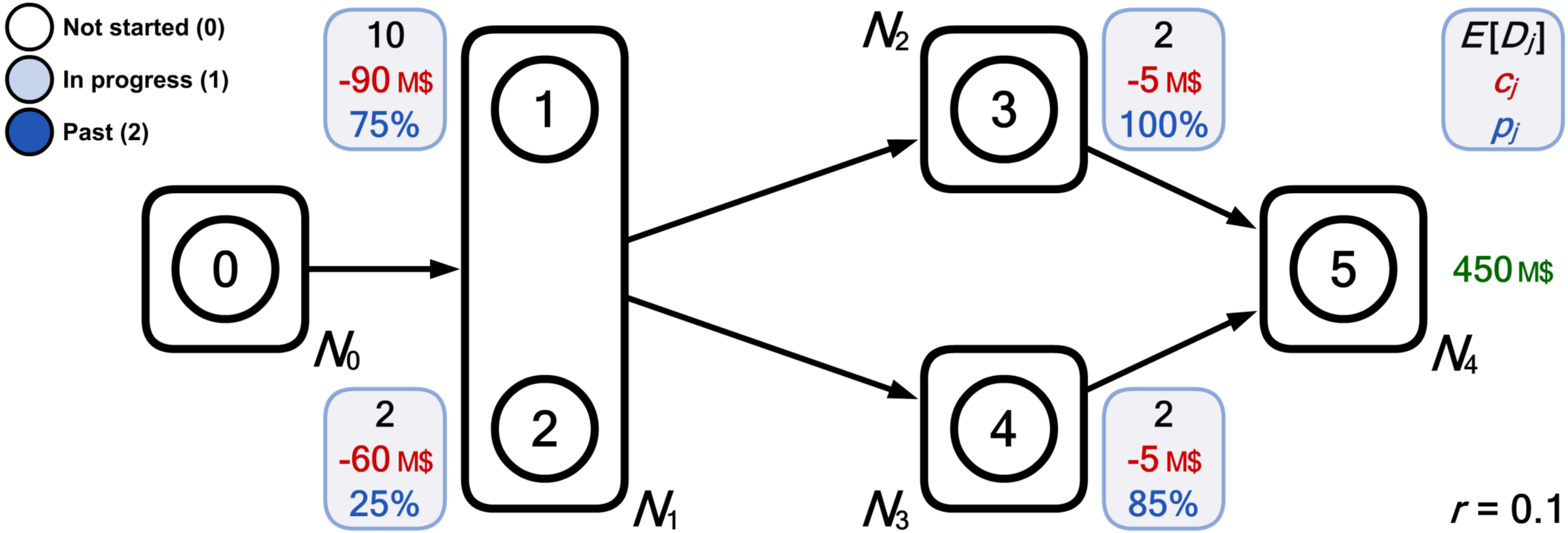


(2,2,2,2,2,0) [450M$]
→ (2,2,2,2,1,0) [318.75M$]
→ (2,2,2,1,2,0) [375M$]
→ (2,2,2,1,1,0) [289.77M$]
→ (2,2,2,0,0,0) [279.77M$]

3 possible decisions (pick the optimal one):

- **Start activity 3 => incur cost c_3 = -5M$**
  => end up in (2,2,2,1,0,0)
- **Start activity 4 => incur cost c_4 = -5M$**
  => end up in (2,2,2,0,1,0)
- **Start activity 3 & 4 => incur cost c_3 + c_4 = -10M$**
  => end up in (2,2,2,1,1,0)[289.77M$]

Solution methodology

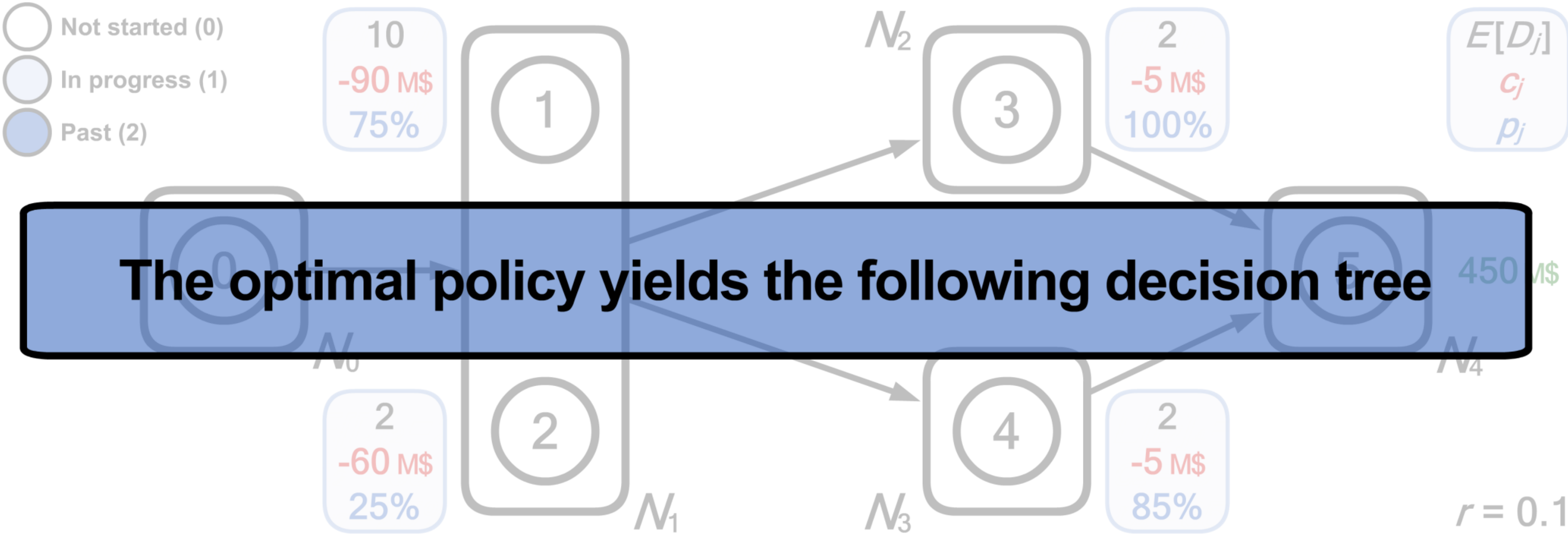


(2,2,2,2,2,0) [450M$]

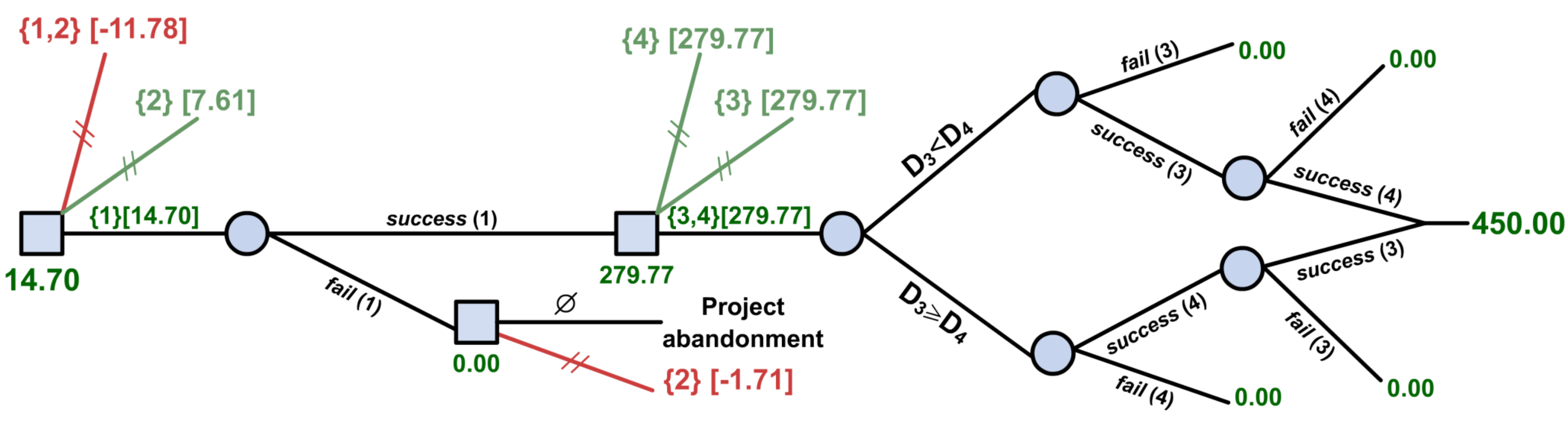
- → (2,2,2,2,1,0) [318.75M$]
- → (2,2,2,1,2,0) [375M$]
  - → (2,2,2,1,1,0) [289.77M$]
    - → (2,2,2,0,0,0) [279.77M$]
      - → (...)
        - → (0,0,0,0,0,0) [14.70M$]

Solution methodology

KATHOLIEKE UNIVERSITEIT LEUVEN



The optimal policy yields the following decision tree



Results & Future Work

- Computational results:
  - 1260 randomly generated projects have been solved to optimality

n	10	20	30	60	90
CPU (sec)	0.00	0.03	1.95	84.04	4100.52

  - Main determinant of computation time = network density (for fixed *n*)
- Future work:
  - Using the model to generate insights
  - General activity durations using Phase-type distributions
  - Renewable resources

Questions?