# Synthesis
and
# Stochastic Assessment
of
# Schedules for Lacquer Production

H.C. Bohnenkamp[1] H. Hermanns[2,1] R. Klaren[1] A. Mader[1] Y.S. Usenko[1]

[1]Faculty of Electrical Engineering, Mathematics and Computer Science,
University of Twente, P.O. Box 217, 7500 AE Enschede, The Netherlands

[2]Department of Computer Science, Saarland University, D-66123 Saarbrücken, Germany

# Overview

- Setting the stage
- Ametist - AXXOM - and the lacquer production case
- Modest - Motor - Uppaal - Möbius - and all that
- 1st challenge: Schedulability & Real-Time
  - Schedule synthesis
- 2nd challenge: Performance
  - Stochastic assessment of schedules
- Conclusion

# The AMETIST project

advanced methods for timed systems

- Various academic partners:
  Twente, Nijmegen, Dortmund, Aalborg, Grenoble, Marseille, Weizmann

- Four industrial partners:
  - Axxom (Munich, D)
  - Cybernetix (Marseille, F)
  - Robert Bosch (Stuttgart, D)
  - Terma (Copenhagen, DK)

- Focus:
  - case studies
  - case studies
  - case studies

AMETIST

advanced methods for timed systems

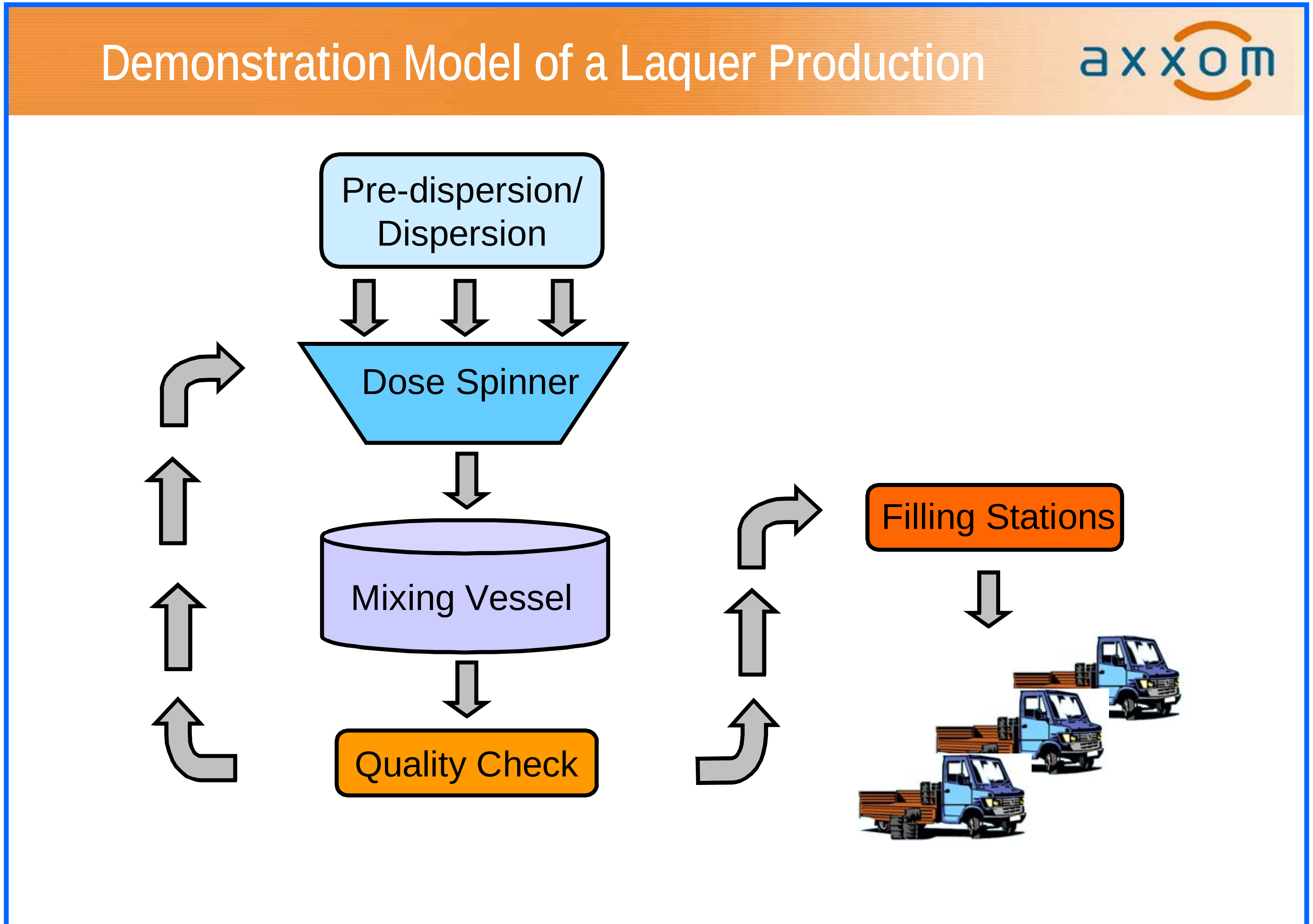
# AXXOM

produces 'value chain optimisation' software for companies like

intricate scheduling problems

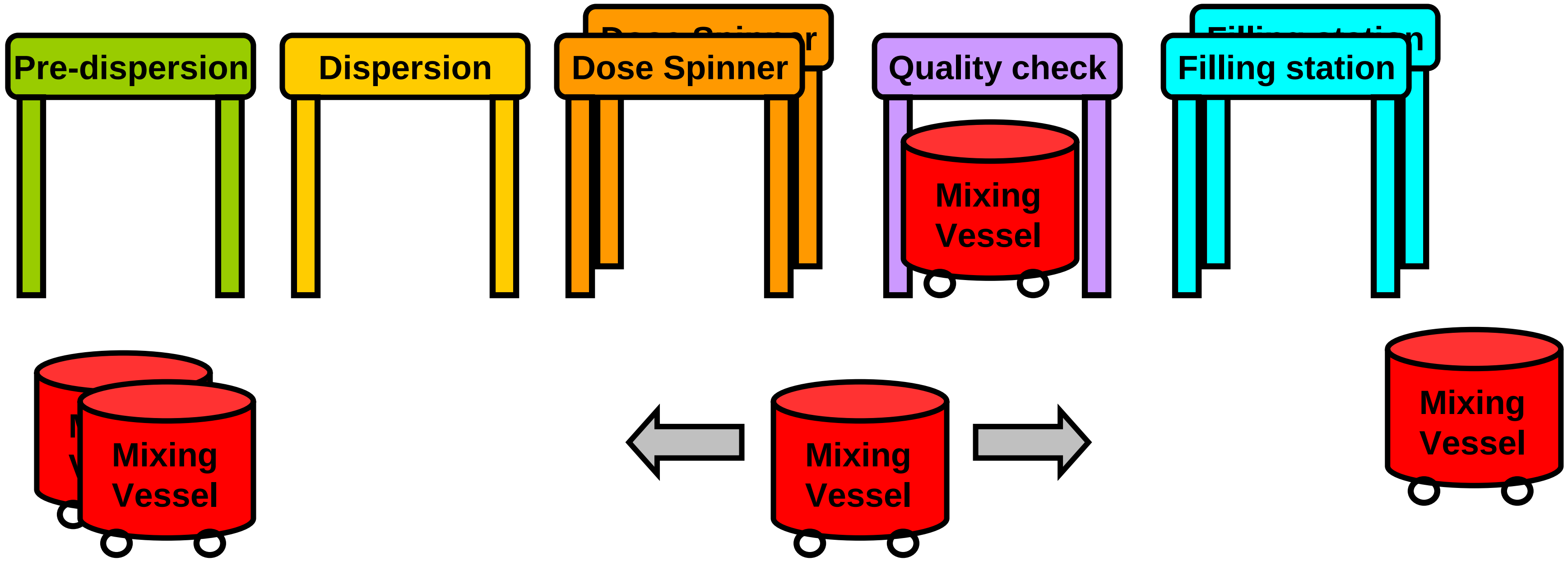
Audi
MAN
DELPHI Automotive Systems
DAIMLERCHRYSLER
STRATEC Medical
Roche
ZEPPELIN CAT
MUK
AlliedSignal Riedel-de Haën
Aventis
DU PONT
DuPont Automotive
Die Unternehmensgruppe Schwabe
SCHWABE
Birkel
Nudeln für Deutschland.
PHILIP MORRIS
Nexans
Alcatel Group
SALAMANDER
INDUSTRIE /// PRODUKTE
SCHERING
BMW
Henkel
SCA
Deutsche Post
AVON Cosmetics

# The AXXOM Case Study



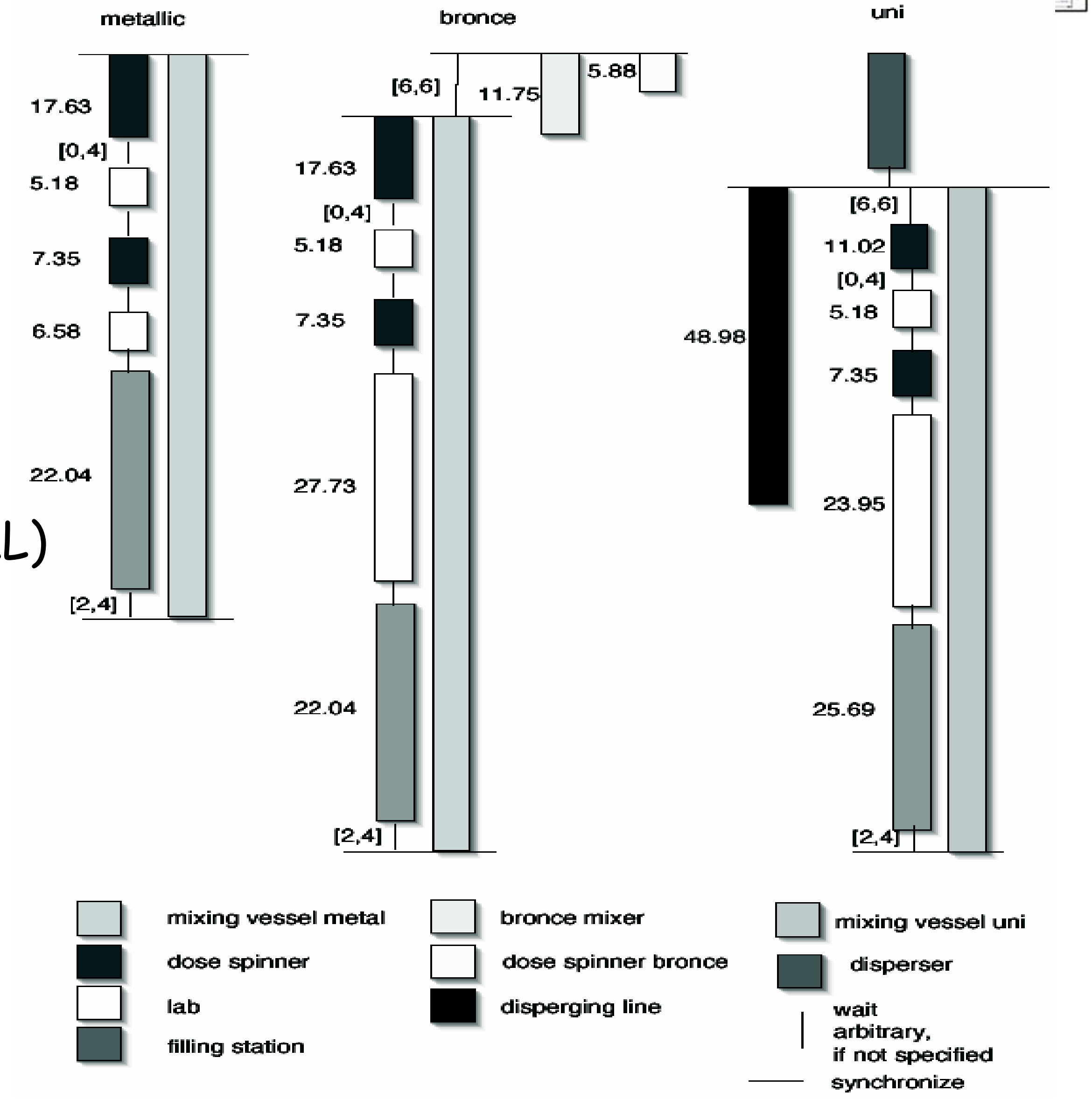
# Pipeless Plant

- Mixing vessels move between stations
- Plant topology (paths, collisions) not considered
- Multiple equal resource instances



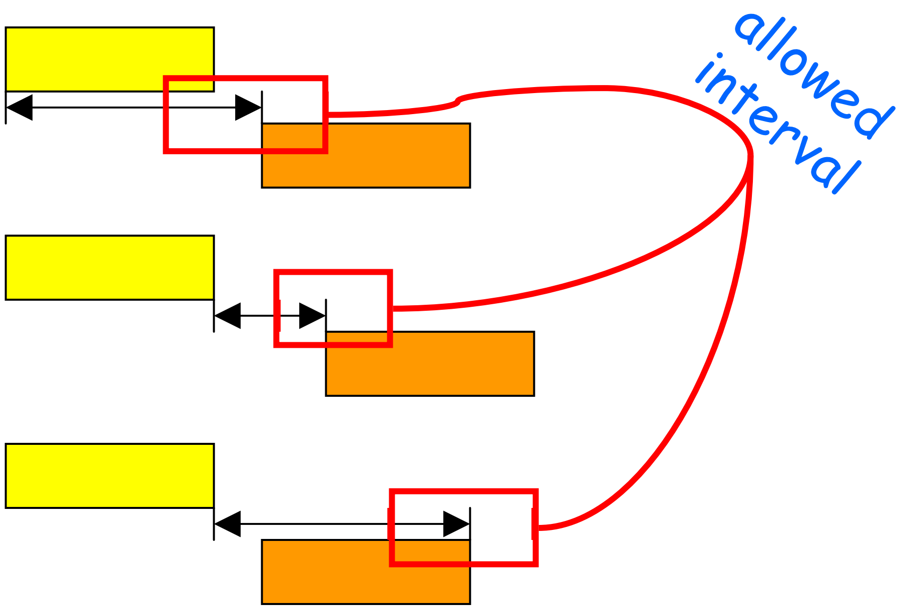
# Problem Statement

- 3 types of jobs: {uni, metallic, bronce}
- 29 jobs:
  - release date,
  - due date,
  - type,
  - batch size (11kL - 19 kL)
- 14 stations, with different capacities
- 6-8 tasks per job, 202 in total
- various time constraints



# Timing Constraints

- Restrictions for pairs of operations:
  - start-start restrictions
  - end-start restrictions
  - end-end restrictions



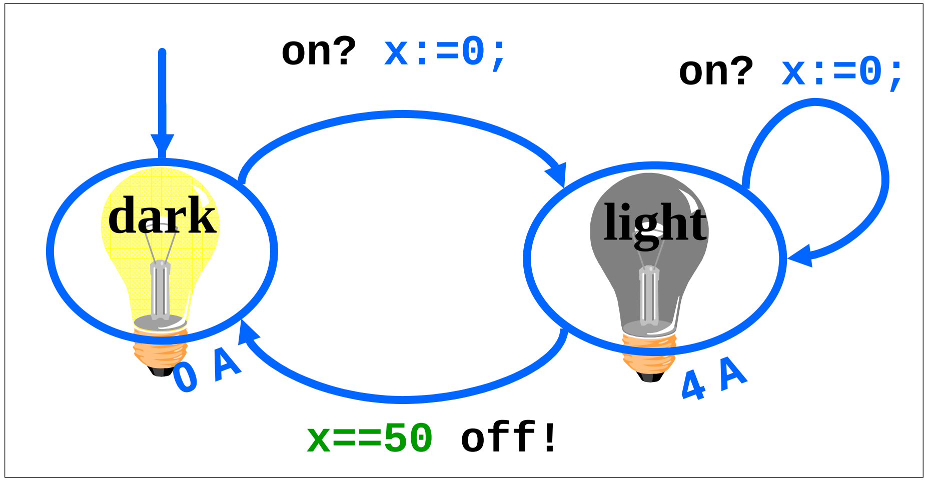

- Motivation:
  - chemical reaction durations,
  - spoiling of products,
  - usage of multiple ressources.

# Overview

- Motivation
- Ametist - AXXOM - and the lacquer production case
- Modest - Motor - Uppaal - Möbius - and all that
- 1st challenge:
  - Schedule synthesis
- 2nd challenge:
  - Stochastic assessment of schedules
- Conclusion

# Timed Automata

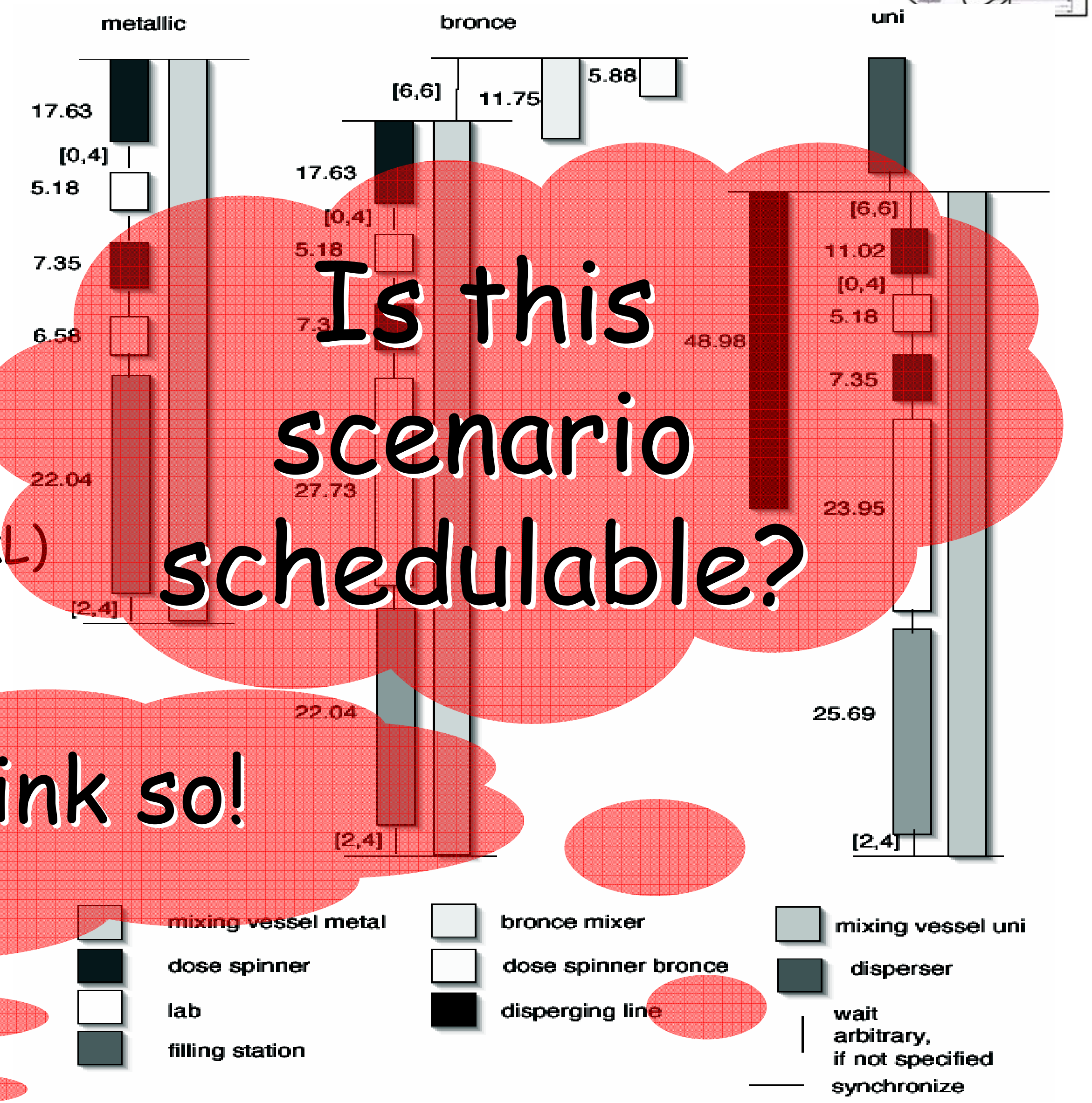
- finite automata
- decorated with clocks
- and with costs



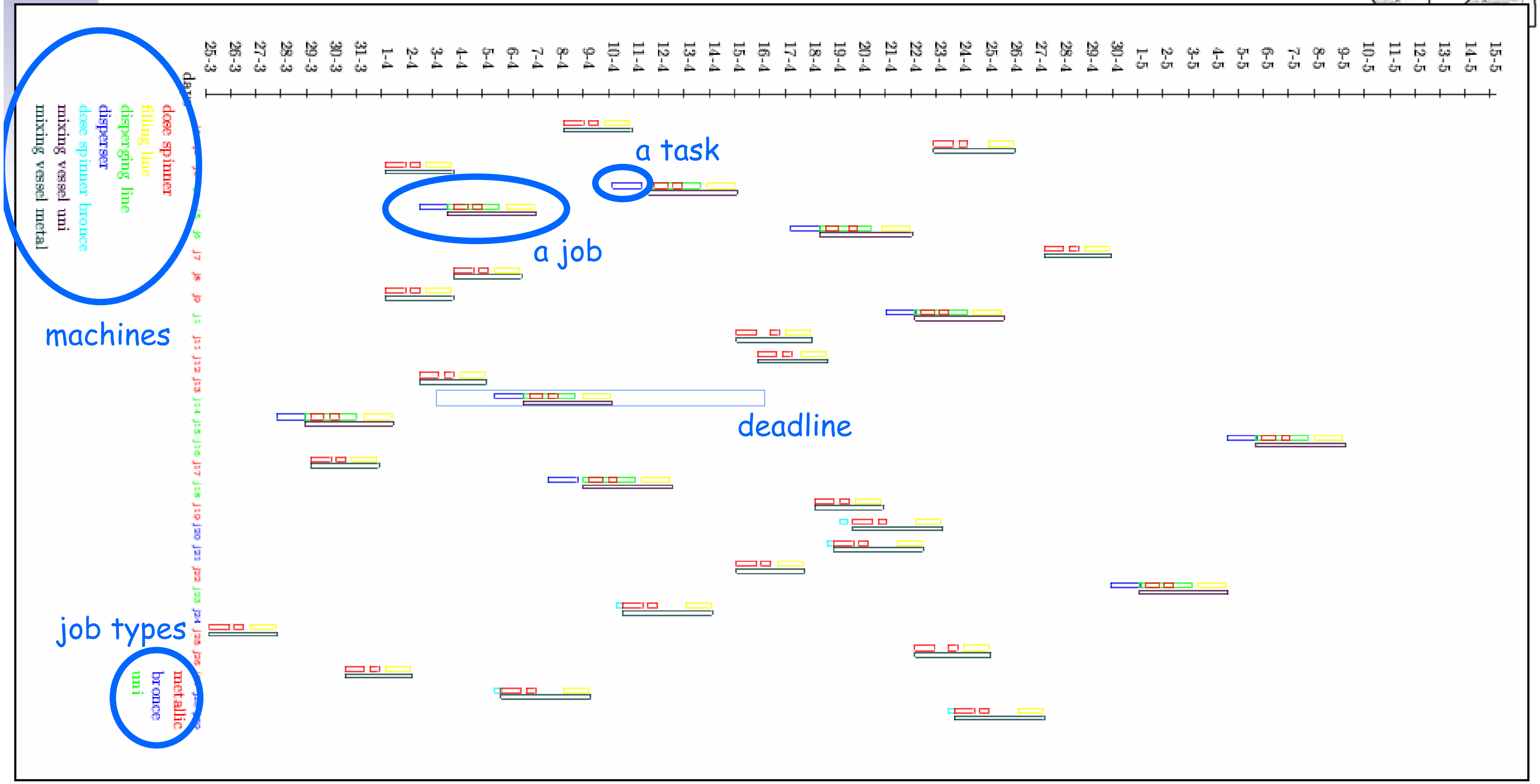

# and Uppaal

# Problem Statement

- 3 types of jobs: {uni, metallic, bronce}
- 29 jobs:
  - release date,
  - due date,
  - type,
  - batch size (11kL - 19 kL)
- 14 stations, with different capacities
- 6-8 tasks per job, 202 in total
- various time constraints



# A synthesized schedule




- 29 jobs, grouped into 3 job types,
- each job type is composed of multiple partially concurrent tasks,
- running on 11 different 'machines'.
- each job has a deadline of 336 hrs (2 weeks)

# Schedule synthesis with UPPAAL

Rough strategy:

- Model
  - machines
  - jobs
  - timing constraints

  as a collection of timed automata

- Feed model into UPPAAL

- Challenge the tool by a (timed) reachability requirement
  "'There is no chance to make all deadlines"

# Schedule synthesis with UPPAAL

Rough strategy:

- Model
  - machines
  - jobs
  - timing constraints

  as a collection of timed automata

- Feed model into UPPAAL

- Challenge the tool by a (timed) reachability requirement

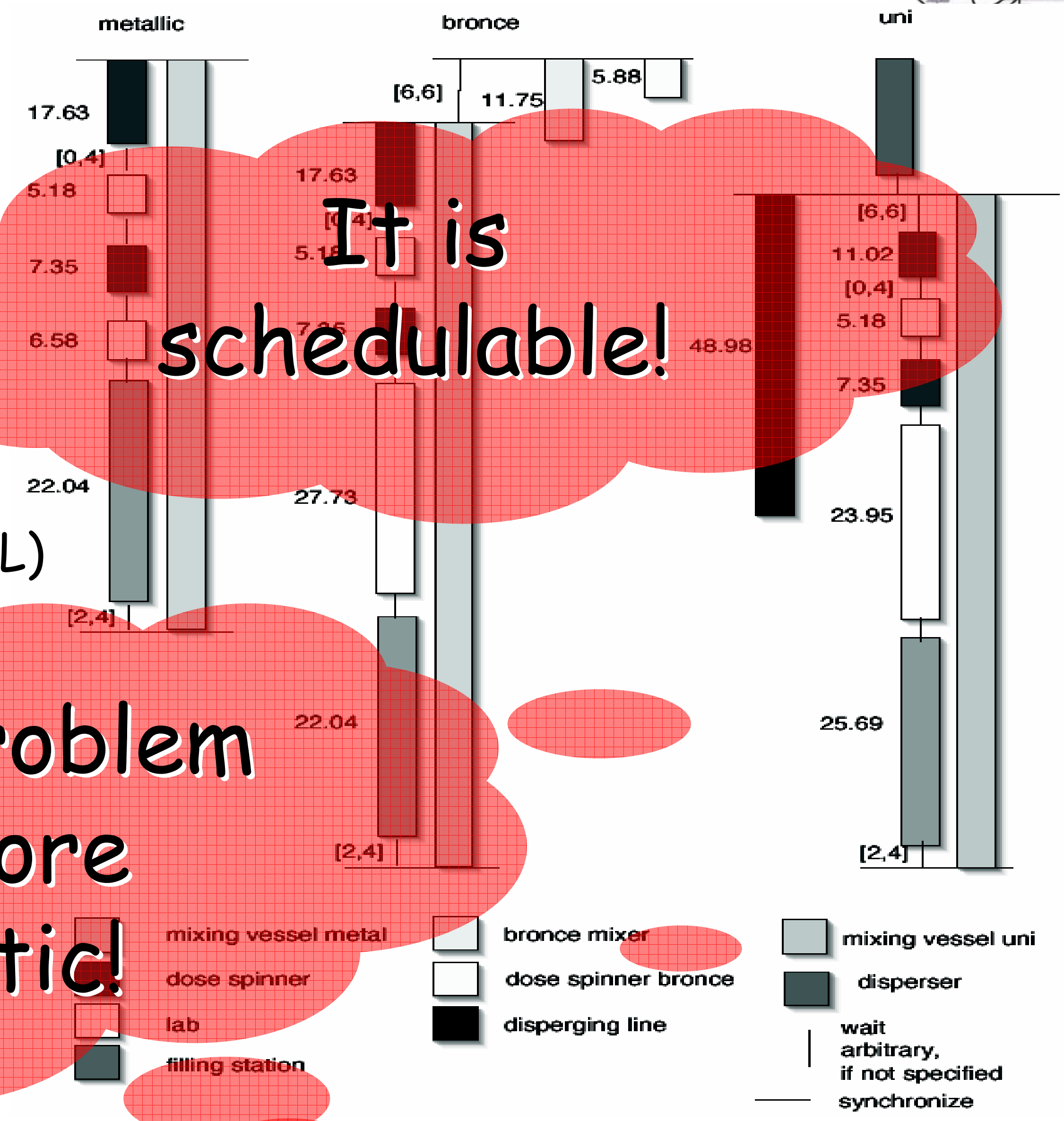
  "'There is no chance to make all deadlines"

- If the tool refutes the requirement:

  Counterexample is a valid schedule

# Problem Statement

- 3 types of jobs: {uni, metallic, bronce}
- 29 jobs:
  - release date,
  - due date,
  - type,
  - batch size (11kL - 19 kL)
- 14 stations, with different capacities
- 6-8 tasks per job, 202 in total
- various time constraints




It is schedulable!

Hoho, the problem is a bit more problematic!

# Overview

- Motivation
- Ametist - AXXOM - and the lacquer production case
- Modest - Motor - Uppaal - Möbius - and all that
- 1st challenge:
  - Schedule synthesis
- 2nd challenge:
  - Stochastic assessment of schedules
- Conclusion

# Performance & Availability Factors

## Performance factor:

- reflects unpredictable perturbations of the production process.
- A performance factor of 0.8 extends the occupation times used for planning by a factor of 1/0.8, i.e., 1.25.

| resource group | description | performance factor | availability factor |
|---|---|---|---|
| ABF001 | filling station 2 | 85 | 86 |
| ABF001_MET | filling station 1 | 85 | 86 |
| DIP002_GRAU | disperser TP2 | 75 | 57 |
| DOK001 | | 85 | 100 |
| DOK002 | | 85 | 100 |
| DVT001 | dose spinner for BR1, BR2, 58BEH001 | 75 | 100 |
| LAB001 | | 85 | 42 |
| MIV007 | 20 m3 Uni-5m | 75 | 86 |
| MIV007 | 20 m3 Uni-5m | 75 | 86 |
| MIV008 | 20 m3 Met-5m | 75 | 86 |
| MIV008 | 20 m3 Met-5m | 75 | 86 |
| MIV008 | 20 m3 Met-5m | 75 | 86 |
| MPA001 | bronce mixer 1 | 75 | 57 |
| MUP001_WEFILL | main disperging line1.1 | 75 | 86 |

## Availabability factor:

- reflects the fraction of time the machine is operational.
- only used if no model of operation hours is available.
- An availability factor of 0.8 extends the occupation times used for planning by a factor of 1/0.8, i.e., 1.25.

**A six-hour disperser job is scheduled for 14 hours**

$$\frac{6}{0.75 \cdot 0.57} = 14.03$$

# Stochastic Perspective

- Both the performance and the availability factors relate to unplanned or unplannable pertubations of the production process.
- They reflect random influences with partially known characteristics.
- This holds in particular for the performance factor, and to a lesser extent for the availability factor.

# Stochastic Perturbations

- It is natural to interpret the availability/performance factor as the ratio of time the system is available/performing.

- In the dependability context this ratio arises as:

Mean up time

$$\frac{\text{MUT}}{\text{MUT} + \text{MTTR}}$$

Mean time to repair

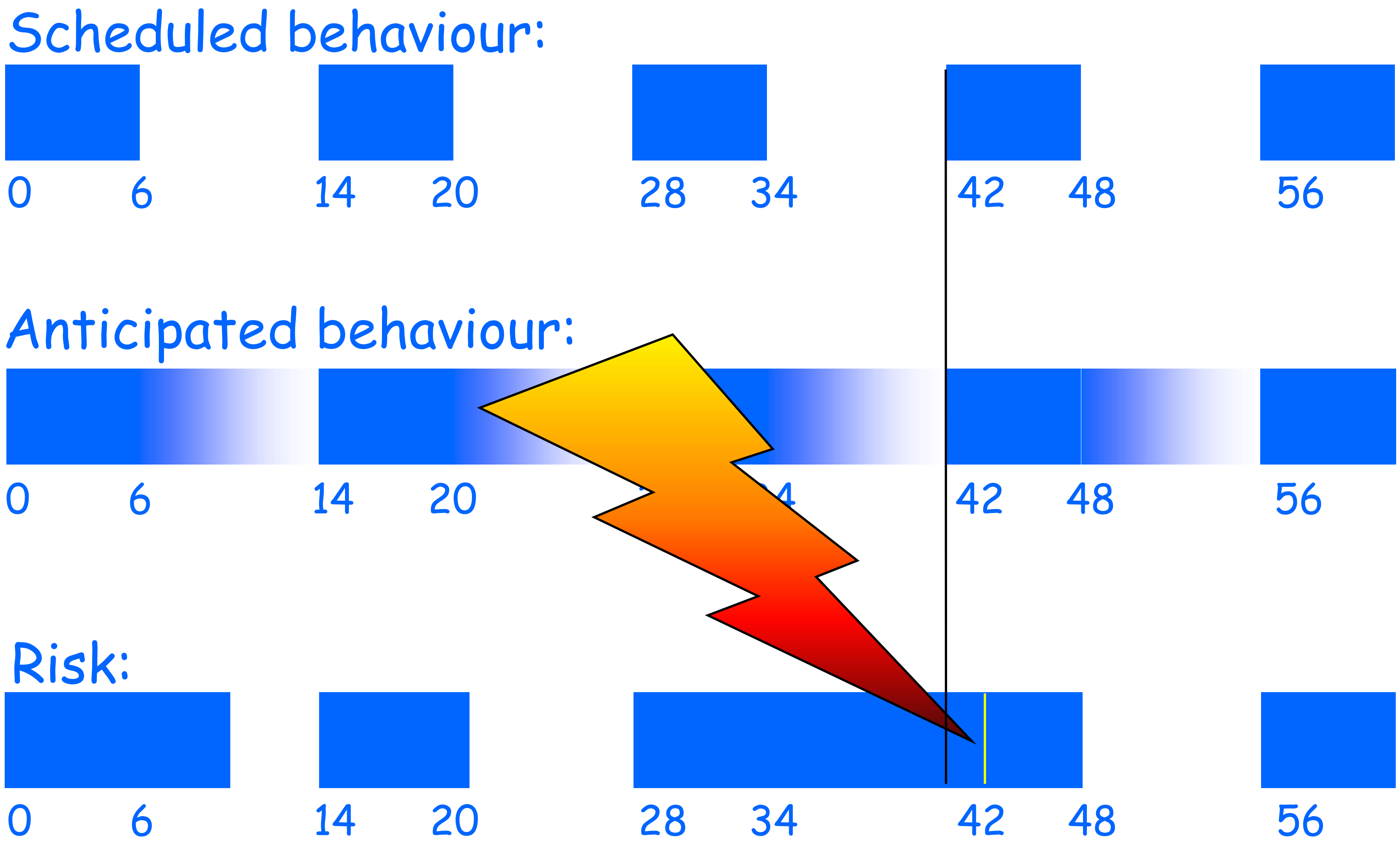
- So a factor of, say, 0.8 relates MUT and MTTR:

$$0.8 = \frac{\text{MUT}}{\text{MUT} + \text{MTTR}} = \frac{80}{80 + 20} = \frac{0.8}{0.8 + 0.2}$$

- If MUT and MTTR are given, the best probabilistic approximation is obtained with negative exponential distributions, parametrized with these mean durations.

- Unfortunately, the means are not given, only their ratio.

# Behaviour of a single machine



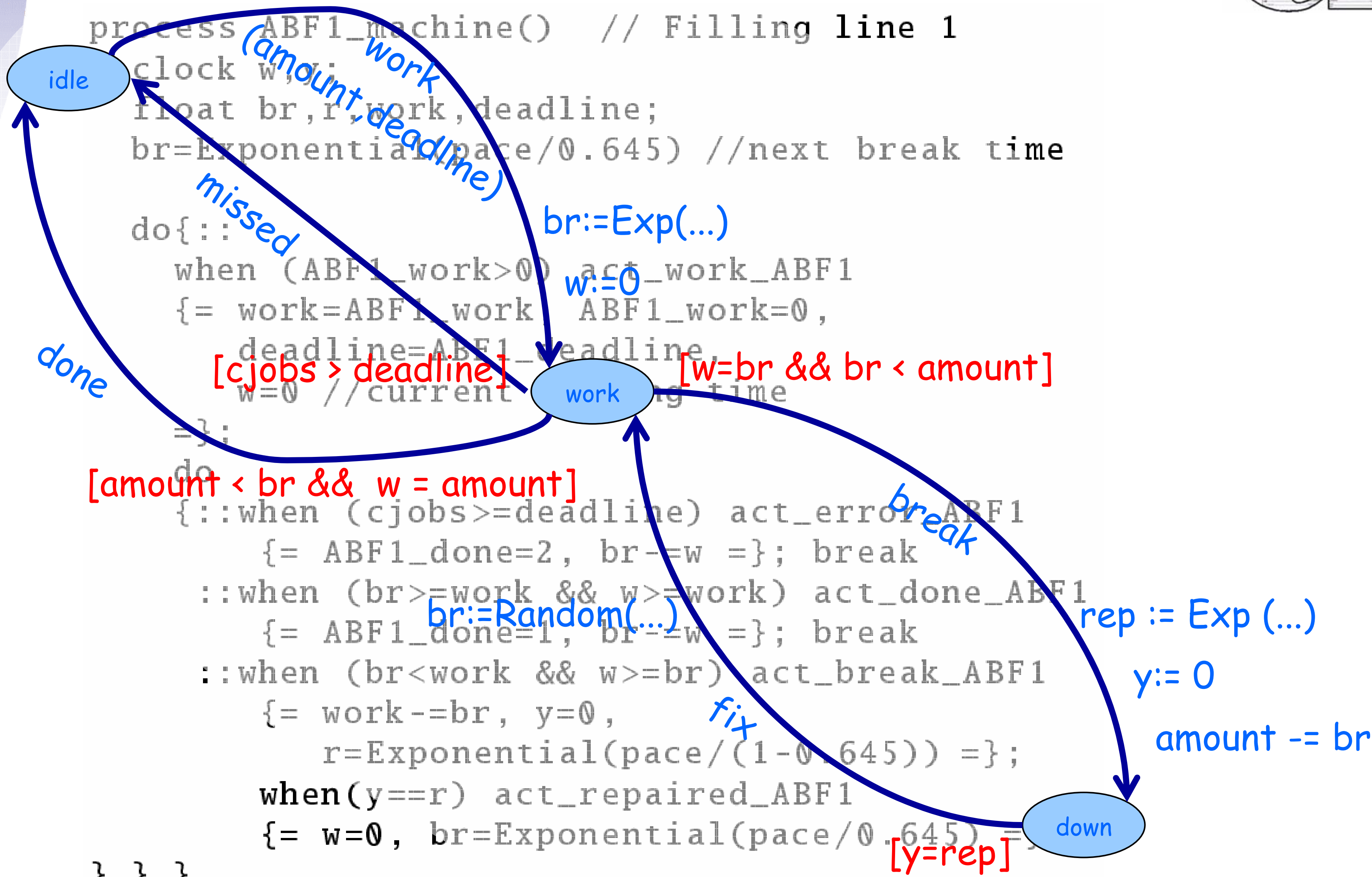
# Our Approach

- Develop a model reflecting the stochastic perturbations.
- Use this model to study a-priori computed valid schedules.
- Quantify the risk
  - to violate the schedule, and
  - to miss deadlines.
- Note: different valid schedules may differ w.r.t. these risks.
- This provides means to rank valid schedules.
- We exercise this approach using Modest.

# Machines in Modest

```
process ABF1_machine()  // Filling line 1
  clock w,y;
  float br,r,work,deadline;
  br=Exponential(pace/0.645) //next break time

  do{::
    when (ABF1_work>0) act_work_ABF1
    {= work=ABF1_work, ABF1_work=0,
       deadline=ABF1_deadline,
       w=0 //current working time
    =};
    do
    {::when (cjobs>=deadline) act_error_ABF1
        {= ABF1_done=2, br-=w =}; break
     ::when (br>=work && w>=work) act_done_ABF1
        {= ABF1_done=1, br-=w =}; break
     ::when (br<work && w>=br) act_break_ABF1
        {= work-=br, y=0,
           r=Exponential(pace/(1-0.645)) =};
        when(y==r) act_repaired_ABF1
        {= w=0, br=Exponential(pace/0.645) =}
} } }
```



# Jobs

```
process Job_type2(int number, float starttime, float earliesttime, float deadline)
{   int mv=0,ds=0;  // which mixing vessel and dose spinner will we get?
    clock c;

    when(cjobs==starttime) {= ii+=1 =}; // starting time according to the schedule

    // disperser for 27
    when(TP2_lock==0) {= TP2_lock=1, TP2_deadline=deadline-49-26-2, TP2_work=27 =};
    when(TP2_done>0) {= TP2_done=0, TP2_lock=0 =};

    // Lock an UNI mixing vessel
    alt{
          :: when(MVU1_lock==0) {= mv=1, MVU1_lock=1 =}
          :: when(MVU2_lock==0) {= mv=2, MVU2_lock=1 =}
    };

    // Two parallel activities:
    par{...
       };
    ...
    // are we on time?
    alt{
          :: when(cjobs<=deadline) {= d+=1, dd+=1 =} ; INC_j(number)
          :: when(cjobs>deadline) ...
    }
}
```

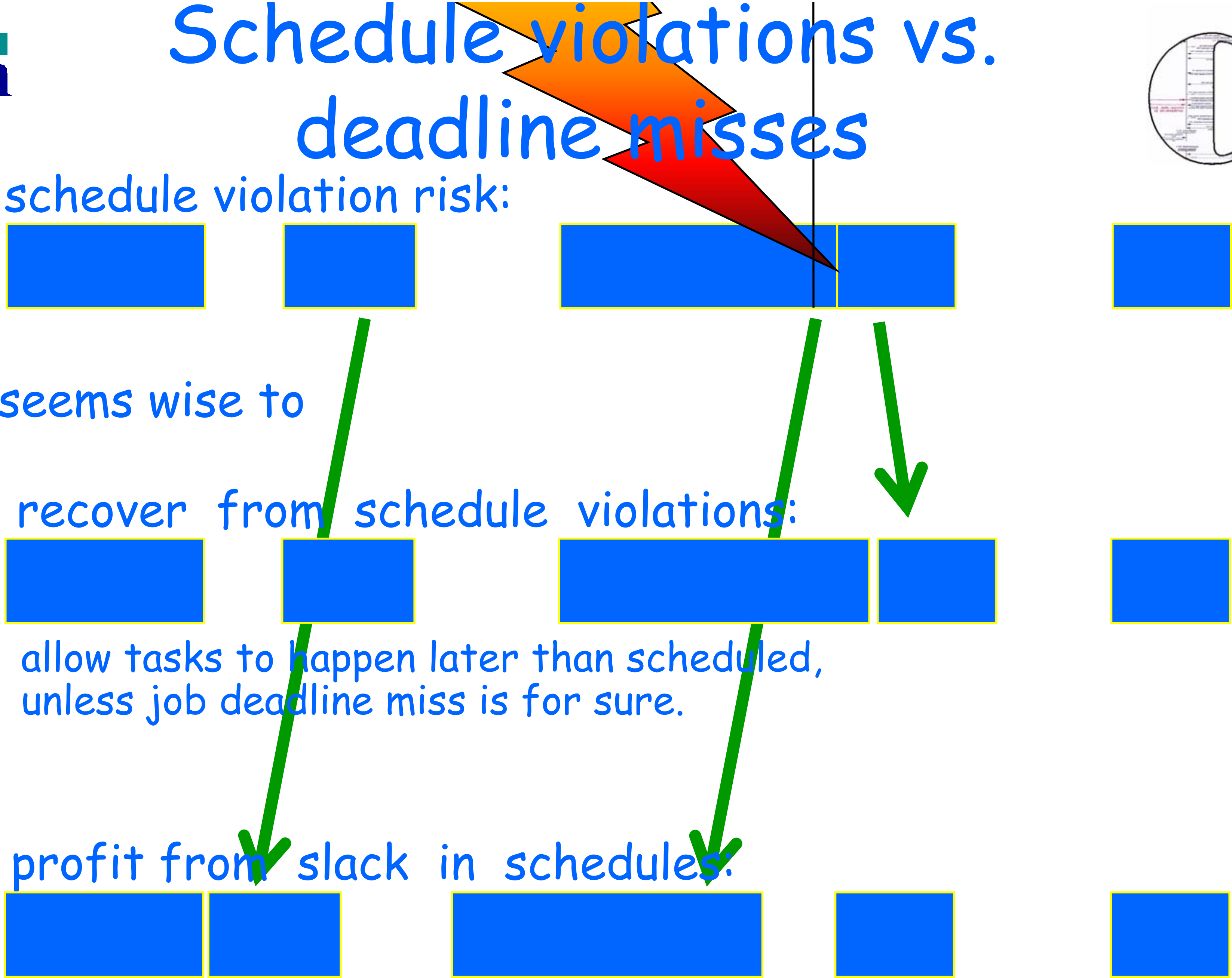
# The system

```
par{
  :: ABF1_machine() :: ABF2_machine()
  :: TP2_machine()
  :: DOK1_machine() :: DOK2_machine()
  :: DVT1_machine(  :: BR1_machine()  :: HDL1_machine()
  :: MVU1_machine() :: MVU2_machine()
  :: MVM1_machine() :: MVM2_machine() :: MVM3_machine()

  :: do {:: tau {= i+=1, d=0, cjobs=0 =};
       par{
          :: Job_type1(17, js17, 101, 101+336)
          :: Job_type2(15, js15,  52,  52+336)
          :: Job_type2( 5,  js5, 191, 191+336)
          :: Job_type2(14, js14, 274, 274+336)
          :: Job_type2(18, js18, 278, 278+336)
          :: Job_type2( 4,  js4, 388, 388+336)
          :: Job_type3(28, js28, 276, 276+336)
           };
           INC_d(d)
        }
}
```

# Schedule violations vs. deadline misses

schedule violation risk:



It seems wise to

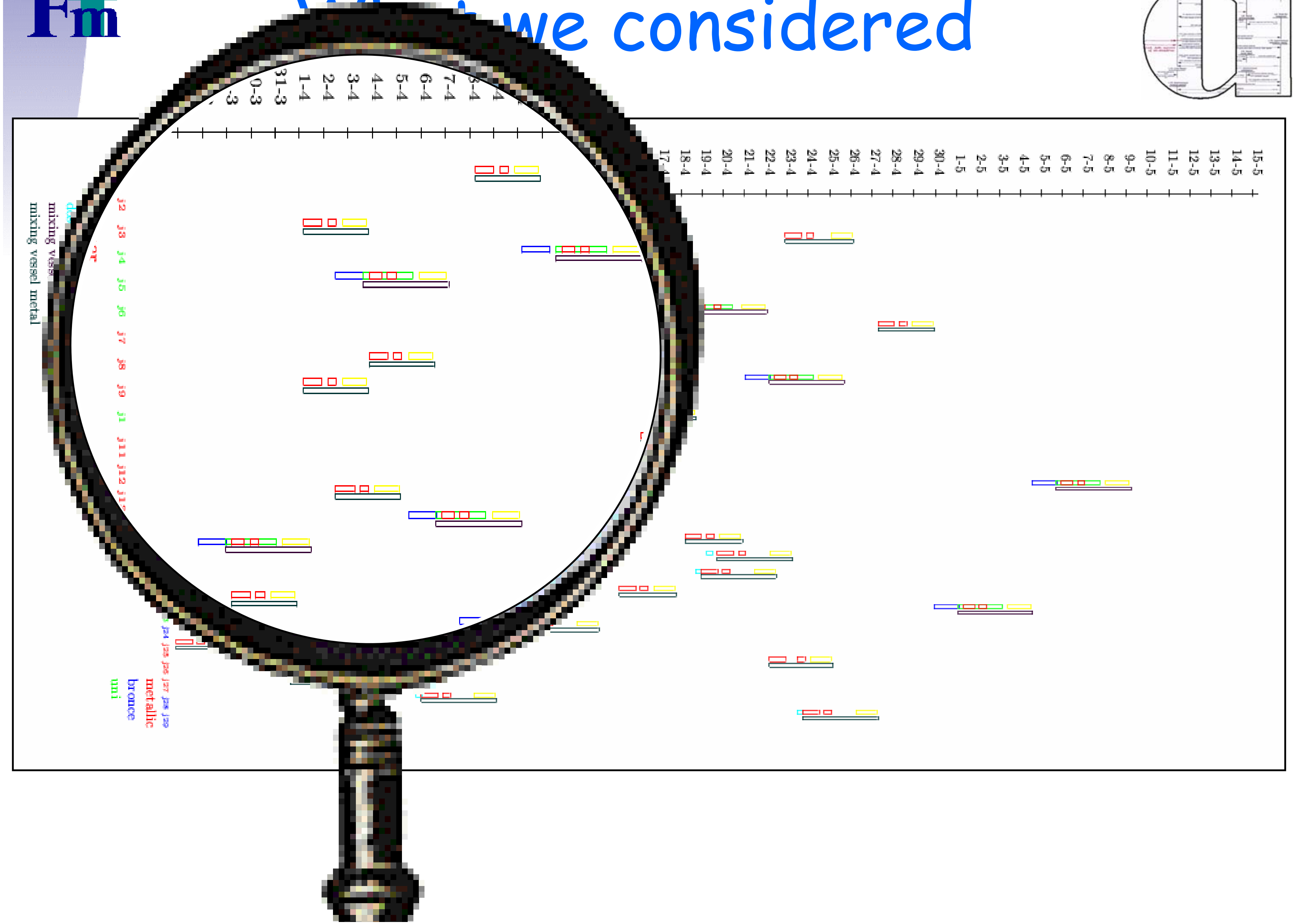
recover from schedule violations:

allow tasks to happen later than scheduled,
unless job deadline miss is for sure.

profit from slack in schedules:

allow tasks to grab machines as early as possible,
respecting scheduled order (not timing).

# What we considered



# Ranking according to individual jobs

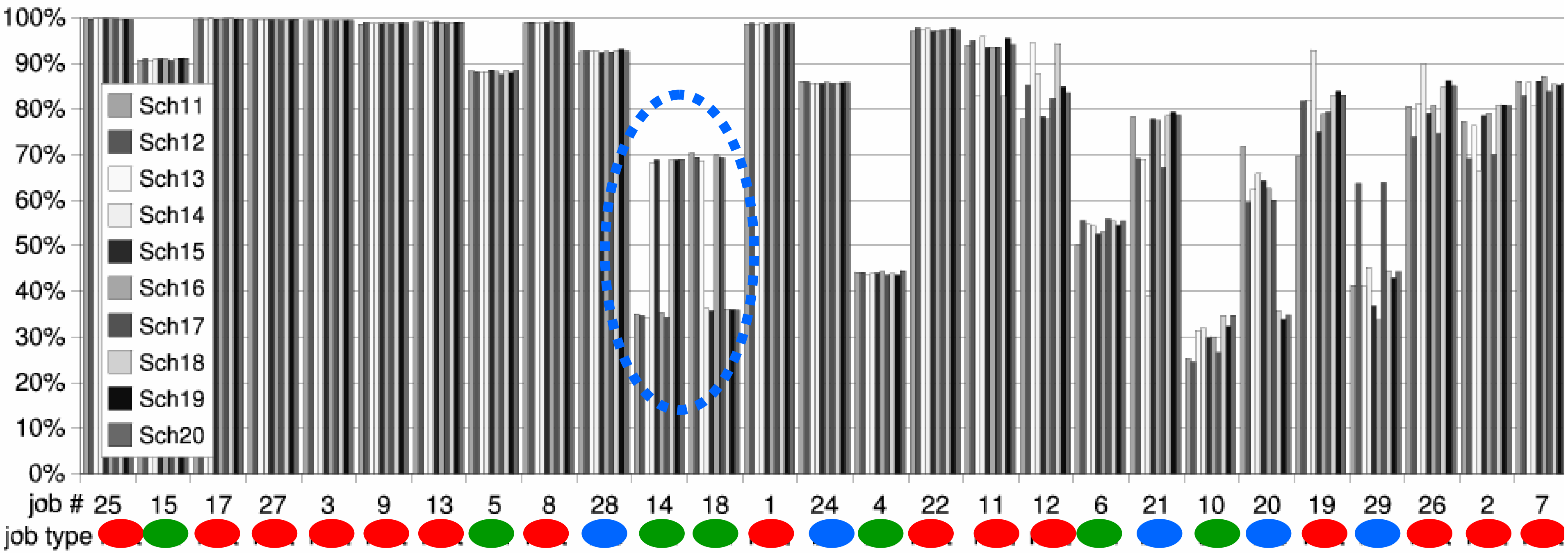
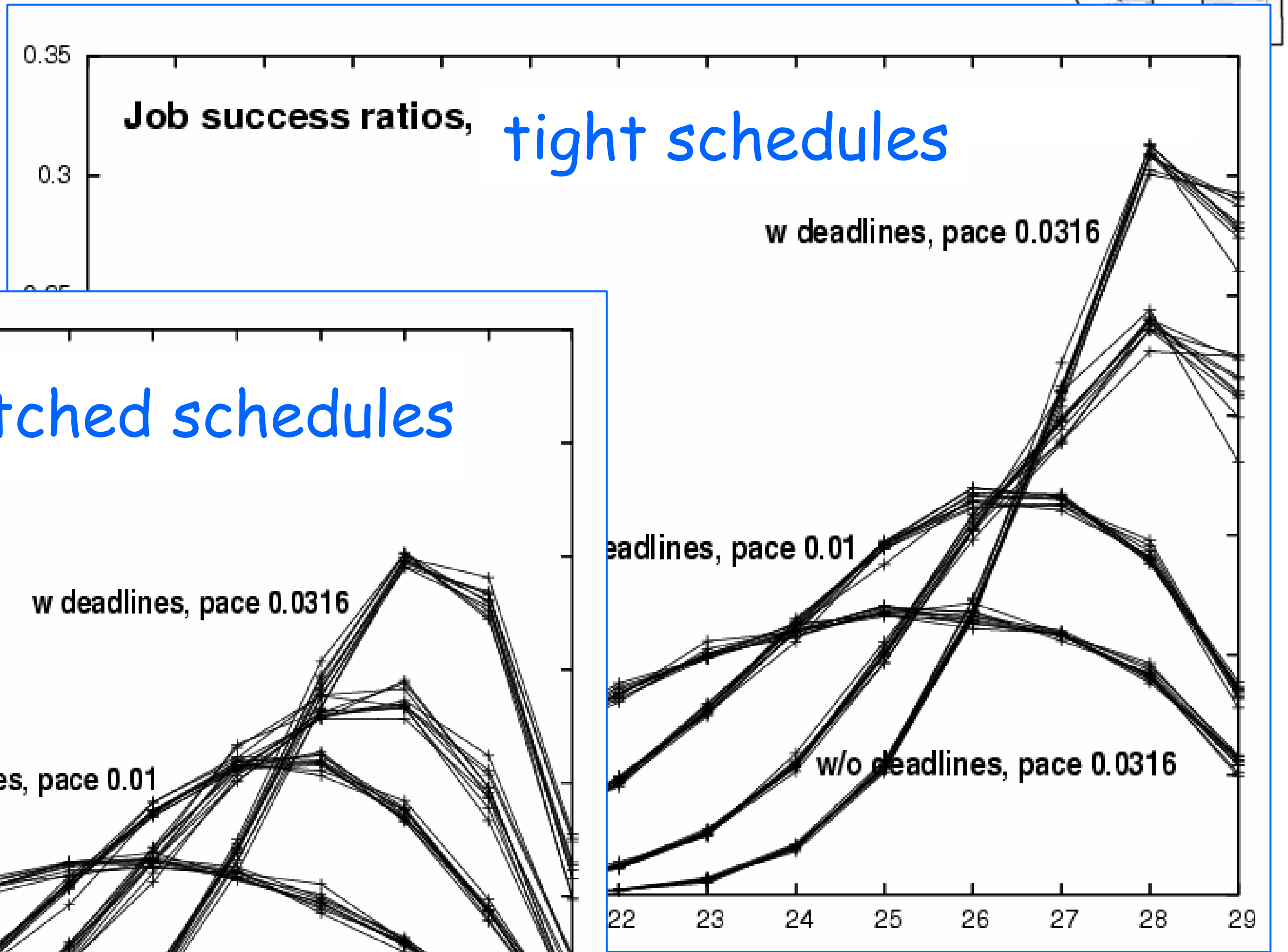
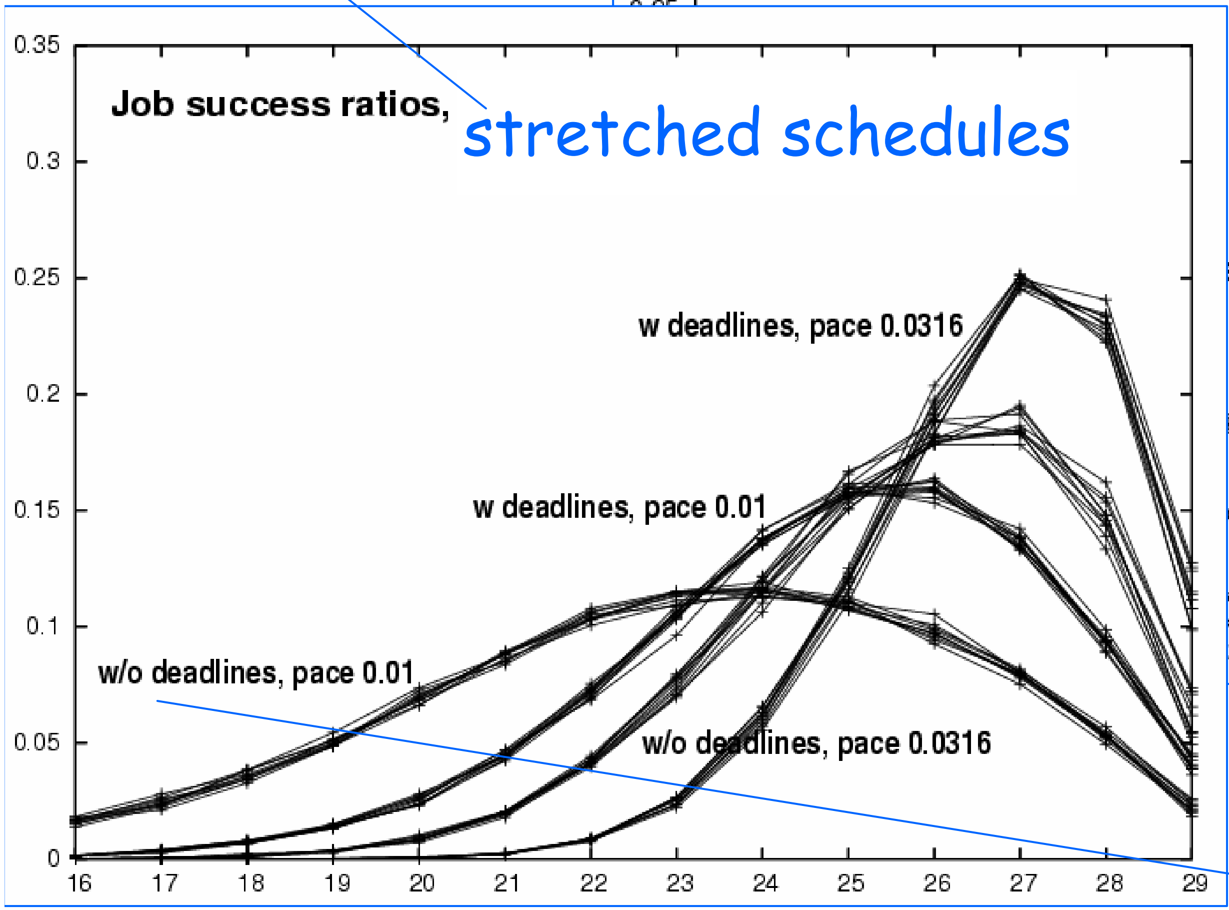



**Figure 8. Success probabilities of the individual jobs.**

# Ranking the schedules

later jobs have
less time remaining
before the deadline




don't throw away
half-finished products

# Overview

- Motivation
- Ametist - AXXOM - and the lacquer production case
- Modest - Motor - Uppaal - Möbius - and all that
- 1st challenge:
  - Schedule synthesis
- 2nd challenge:
  - Stochastic assessment of schedules
- Conclusion