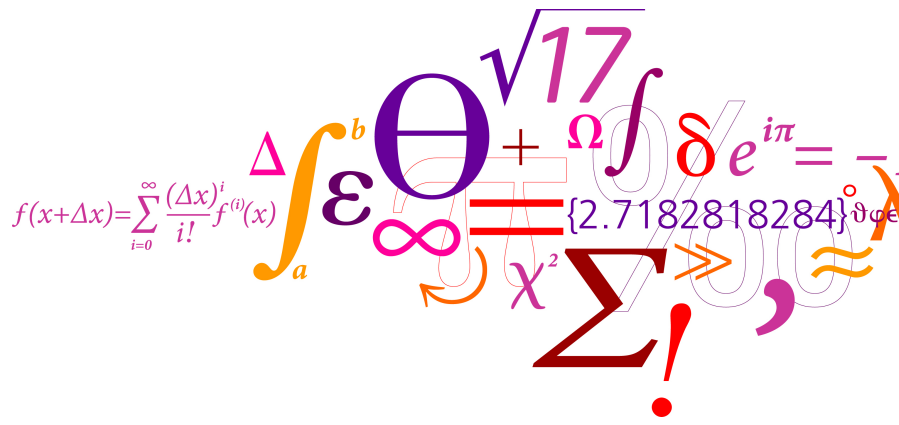
Integrated Disruption Planning

Rolling Stock and Depot Scheduling

Jørgen T. Haahr Richard M. Lusby David Pisinger Jesper Larsen

Department of Management Engineering
Technical University of Denmark

RobustRailS Mini Conference 2015,
August 27st 2015, Kgs. Lyngby



Agenda

Agenda

Integrated Disruption Planning

Research Question...

How does one optimally recover the timetable, rolling stock, depot schedules in a disrupted environment?

Done manually to a large extent

- Complex situation
- Short time

Main collaborators

- DSB & Stog

Reactive Robustness

When things do not go *according to plan*...

- Disruption Causes
  - Infrastructure breakdown
  - Equipment failure
  - Passenger behavior
  - Weather

- Proactive Robustness - prevent disruption

- Reactive Robustness - handle disruption
  - Re-schedule - optimize same problem
    - Run-time requirements - only faster

Reactive Robustness

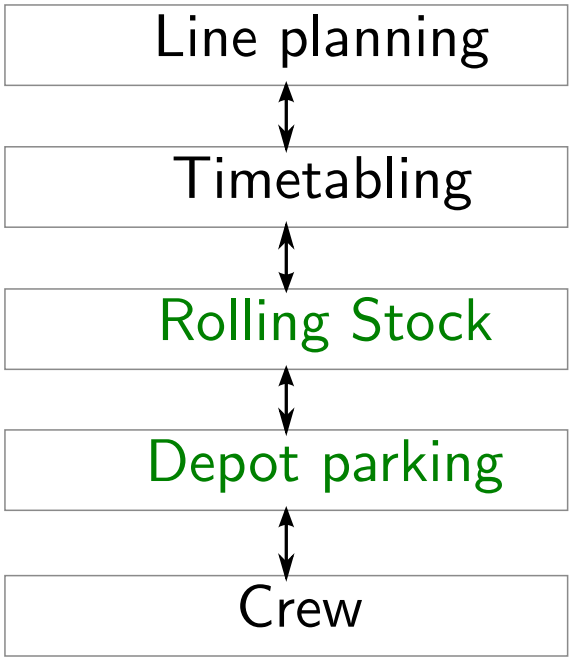
When things do not go *according to plan*...

- Disruption Causes
  - Infrastructure breakdown
  - Equipment failure
  - Passenger behavior
  - Weather

- Proactive Robustness - prevent disruption

- Reactive Robustness - handle disruption
  - Re-schedule - optimize same problem
    - Run-time requirements - only faster

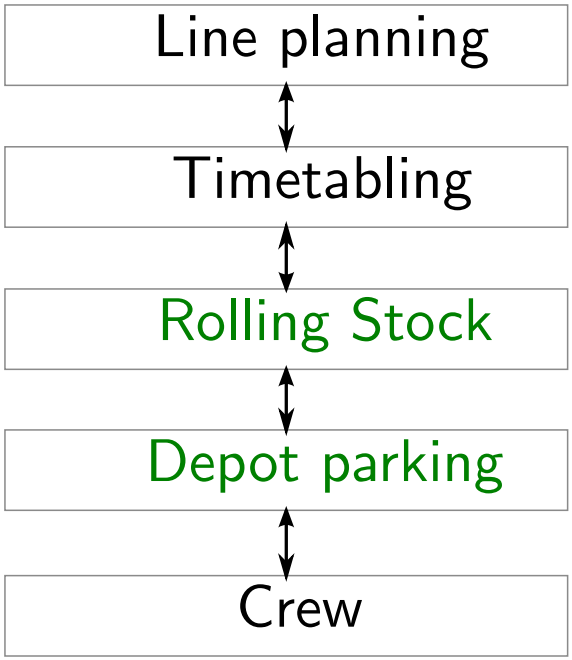
Railway Optimization Problems
Interdepent problems



Our focus

- Rolling Stock
- Depot Parking
- Integration
- Disruption context

Railway Optimization Problems
Interdepent problems



Our focus

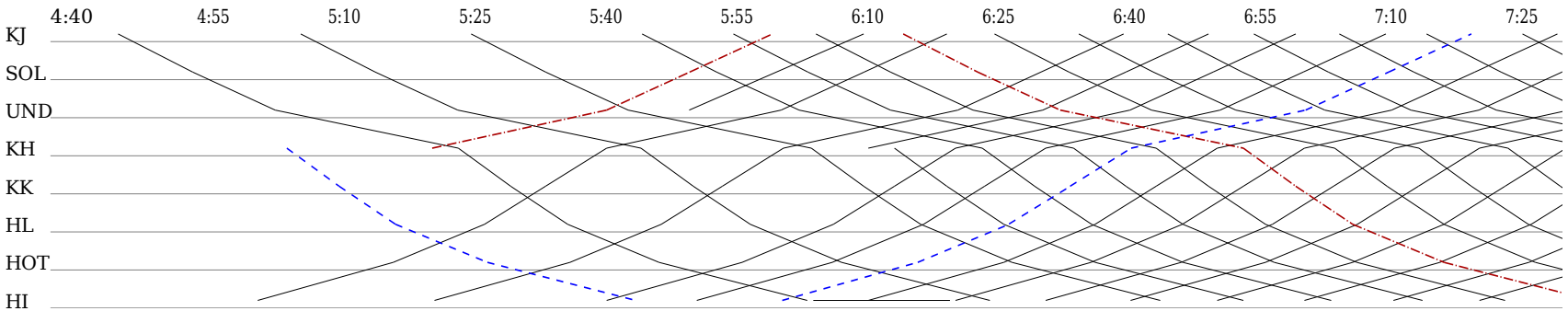
- Rolling Stock
- Depot Parking
- Integration
- Disruption context

Agenda

Rolling Stock (Re-)scheduling

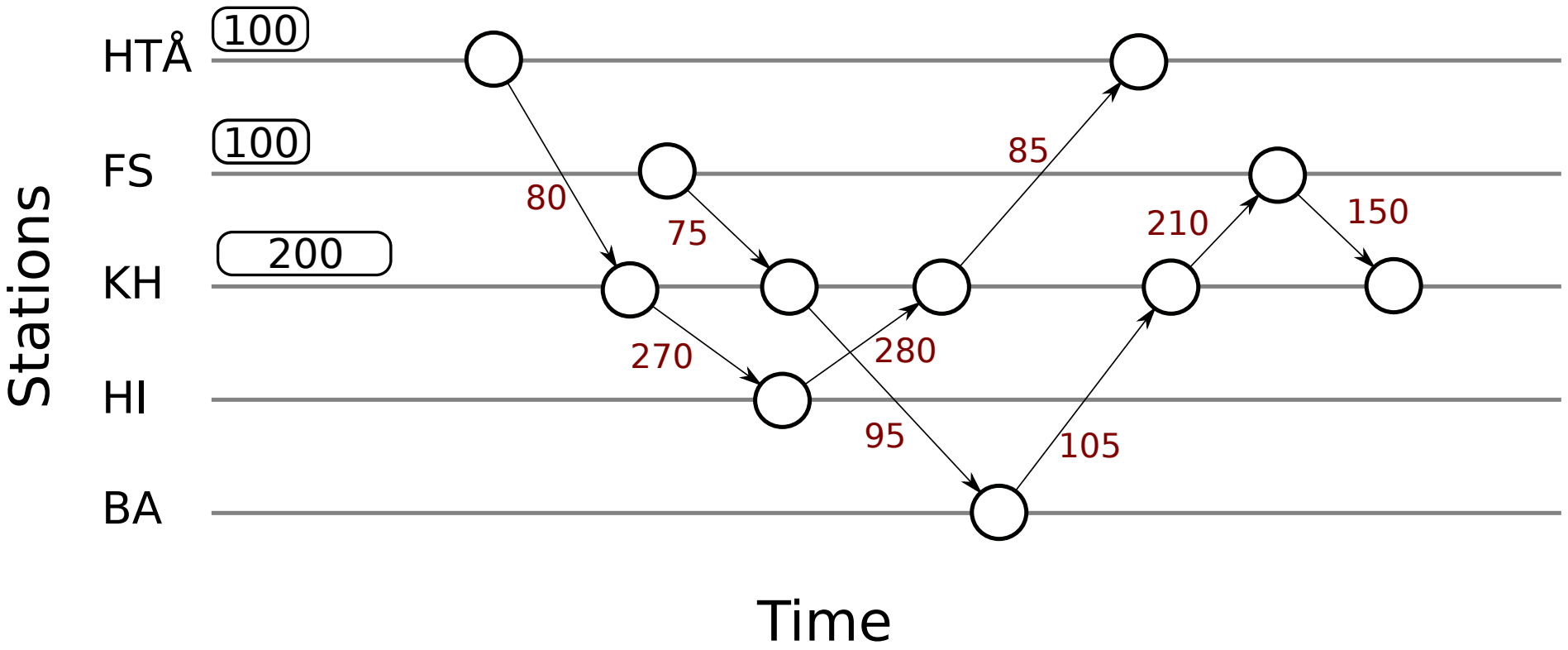
Rolling Stock Circulation

- Timetable - planned trips
- Fleet - units available
- Depots - parking
- Objective
  - Cover trips
  - Satisfy demand
  - Minimize operational cost
  - End-of-day balance
  - Shunting operations



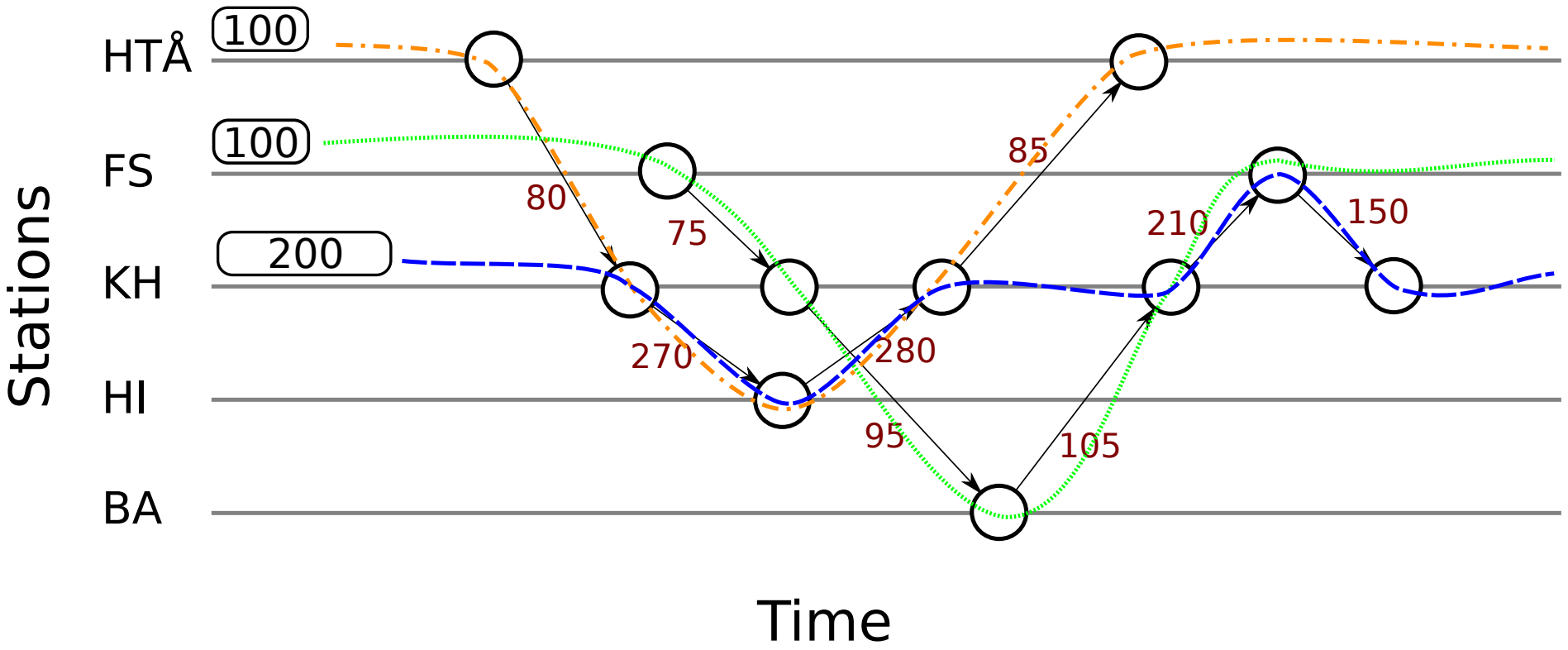
Rolling Stock (Re-)scheduling

Example



Rolling Stock (Re-)scheduling

Example



Rolling Stock (Re-)scheduling

Normal and disruption cases

Instance	Cost	Km	Demand	Shunting	**Cover**	**Time**
NS	639 065	553 310	15 755	70 000	99.9%	465
DSBmon	719 184	555 970	132 214	31 000	98.5%	119
DSBfri	727 159	583 505	119 654	24 000	98.6%	37
DSBsat	418 148	313 469	87 679	17 000	98.3%	10
DSBsun	413 062	297 574	93 489	22 000	98.1%	4

- Planning cases solved in 4-465 seconds
- Balance between objectives
  - High demand cover
- 36 disruption cases solve within 30 seconds on average

Parking the Rolling Stock
A possible dead end

An optimized rolling stock plan

- Valid circulation ✓
- Sufficient aggregated capacity at depots ✓
- Sufficent time for shunting operations ✓
- Sufficient capacity on individual tracks ?
- Any conflict-free track assignment ?

A new (different) plan is needed

Parking the Rolling Stock
A possible dead end

An optimized rolling stock plan

- Valid circulation ✓
- Sufficient aggregated capacity at depots ✓
- Sufficent time for shunting operations ✓
- Sufficient capacity on individual tracks ?
- Any conflict-free track assignment ?

A new (different) plan is needed

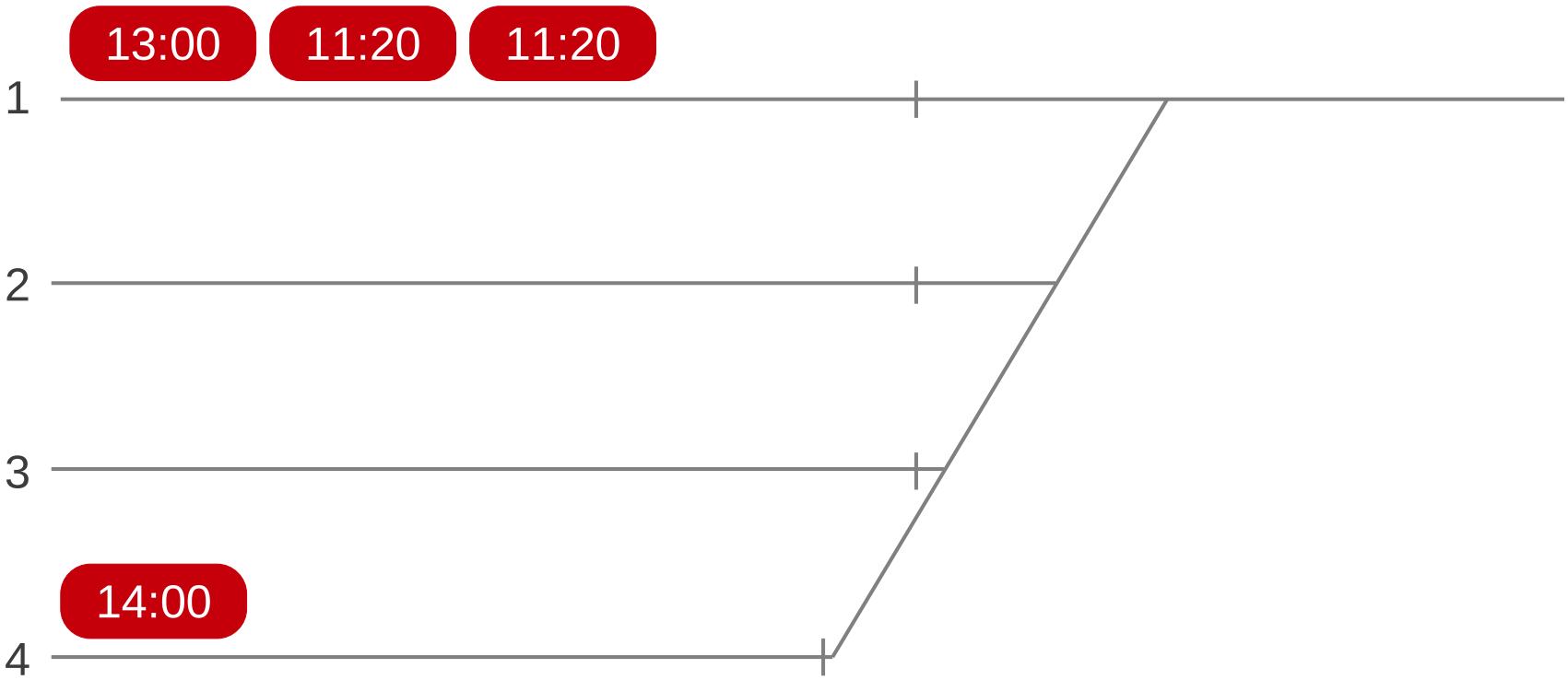
Agenda

Depot Parking
Shunting yard

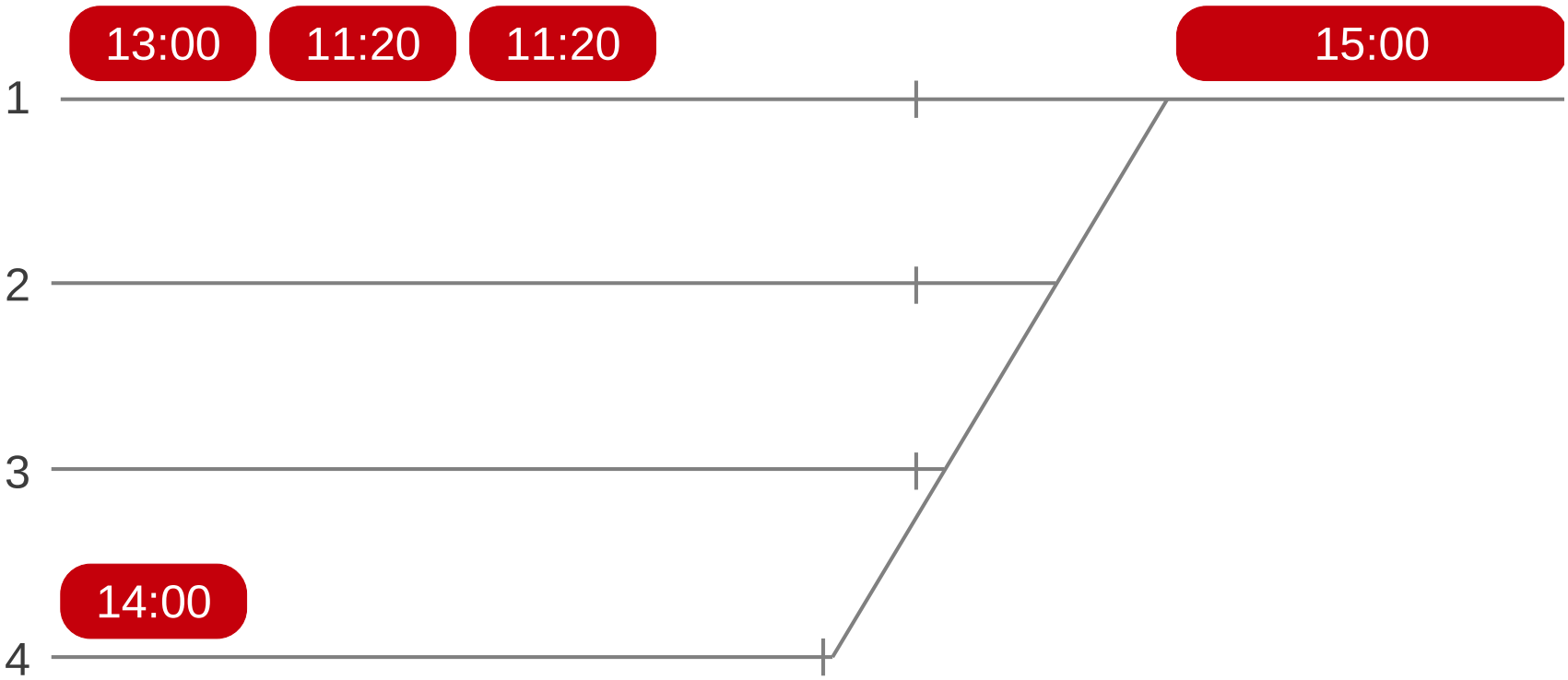
- Given: arrival and departing events
- Need: a unit-to-track assignment

- Tracks - count, length
- Conflicting assignments - LIFO
- Compatibility - electrified, cleaning and more

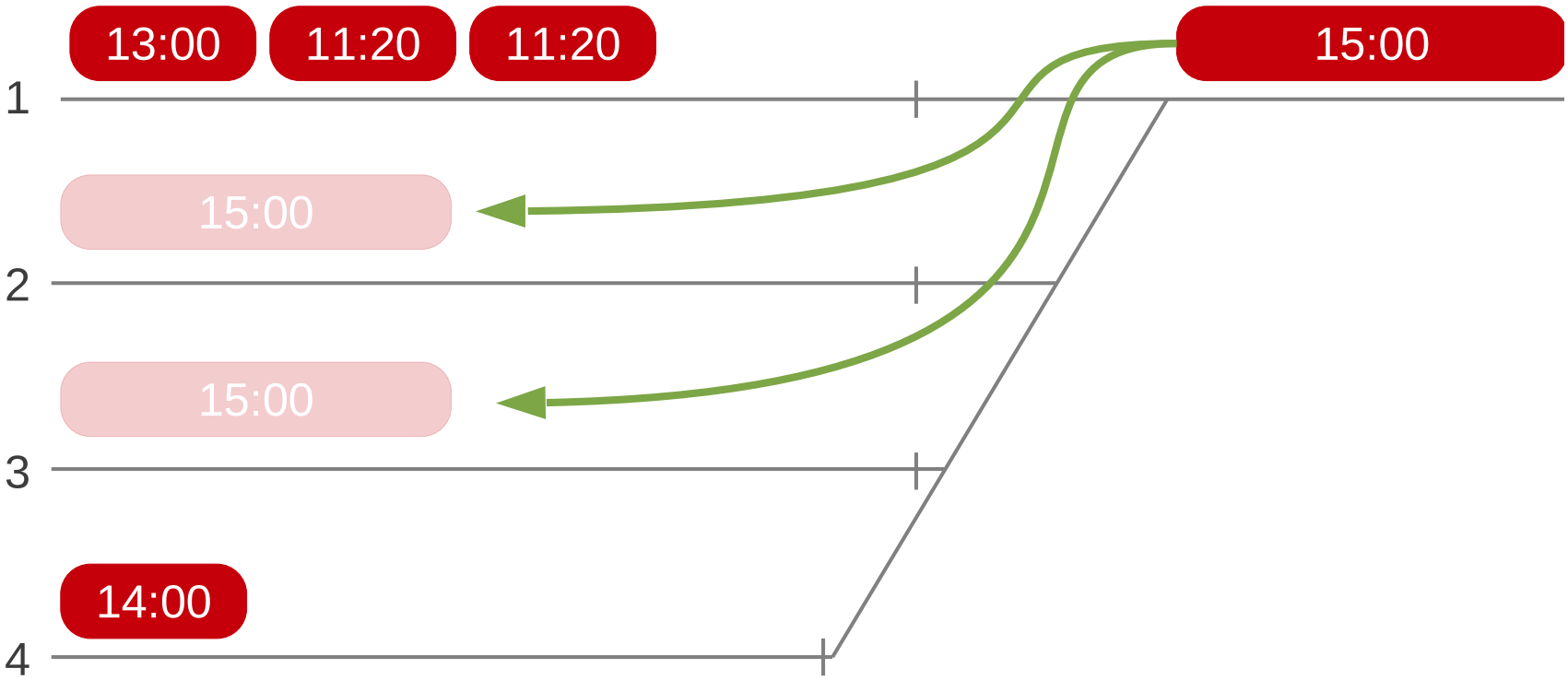
An example
Arrving unit



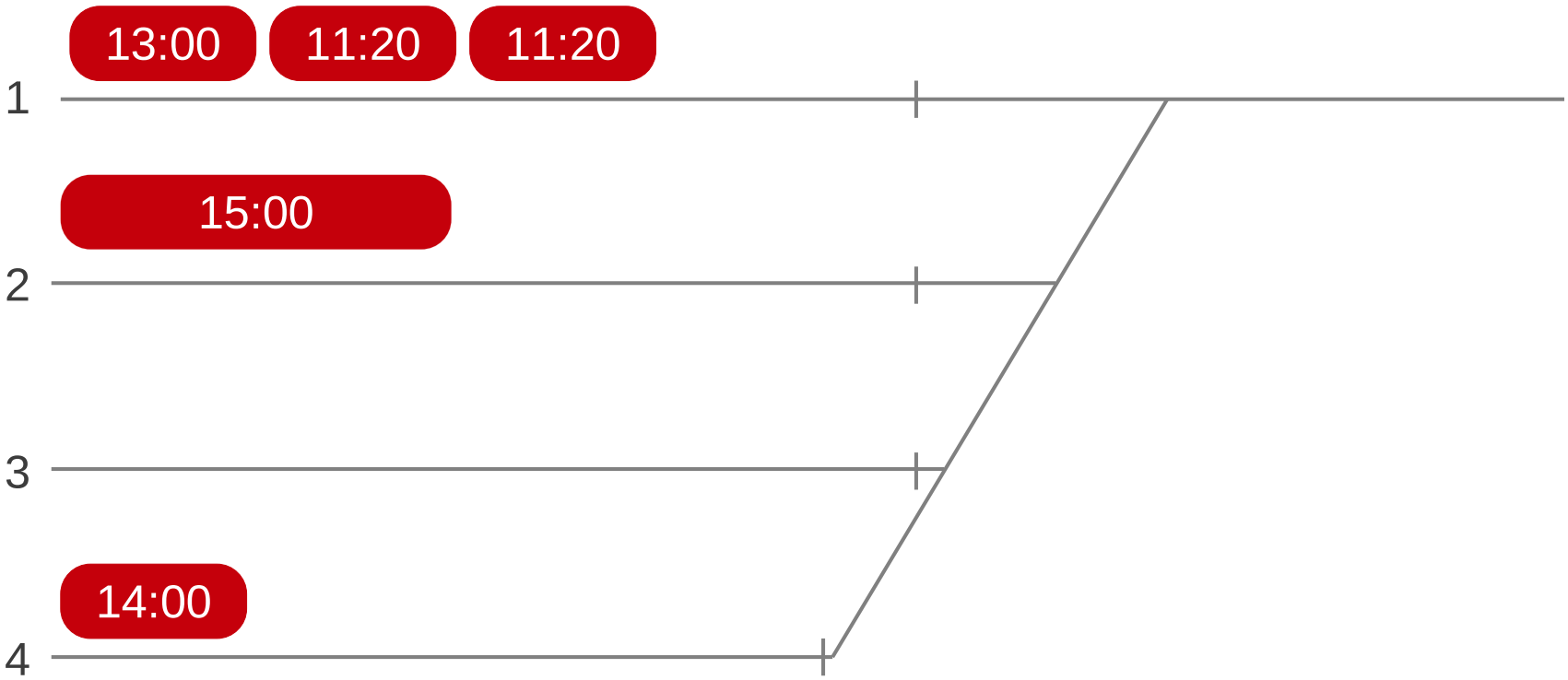
An example
Arrving unit



An example
Arrving unit



An example
Arrving unit



Depot Parking Benchmark

Exact and heuristic methods

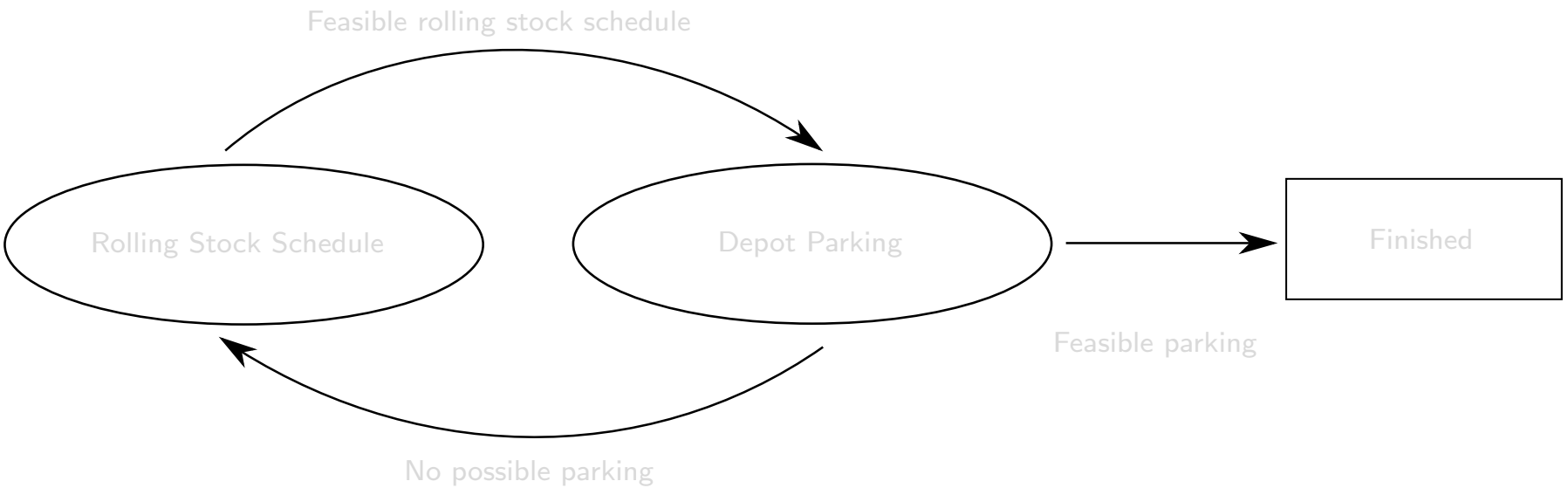
Class	MIP	CP	BAC	CPH	RGCH	TSH
Stog	94	94	94	94	94	93
DSB	0	0	7	0	7	7
NS	0	0	93	110	110	110
NS-hard	0	0	27	79	70	90

- Naive exact methods only work on small instances
- Greedy method and decompositions perform well
  - Within a few seconds

Agenda

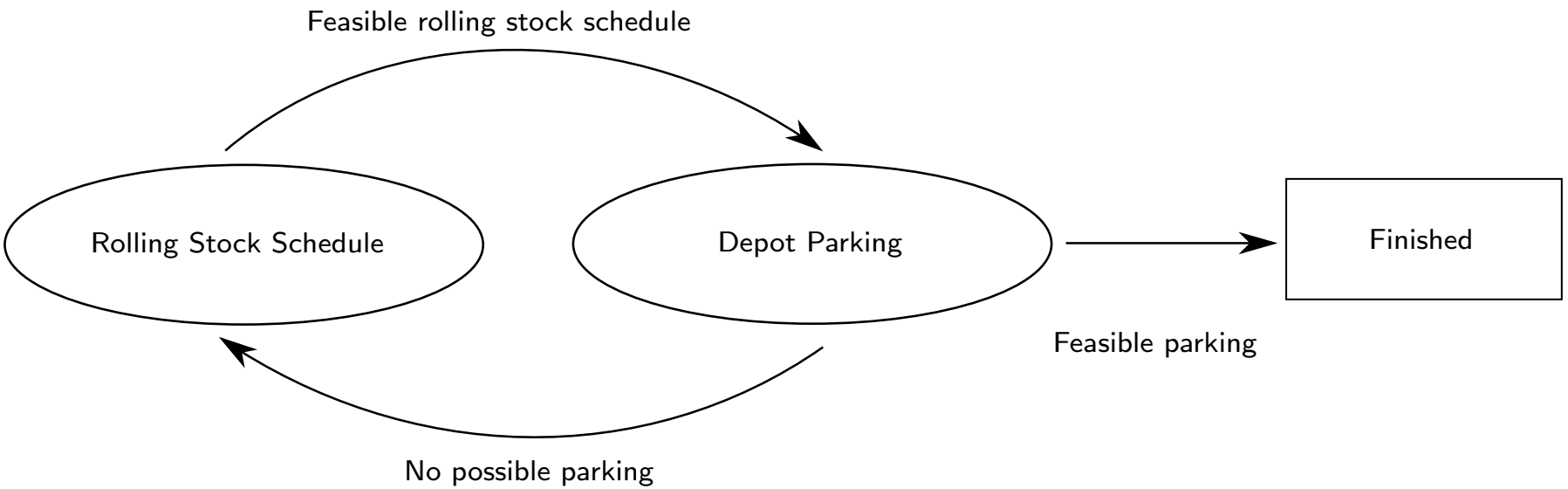
Integration
Rolling Stock and Depot Parking

Rolling stock schedule → shunting events



- Idea: forbid *arrival-and-departure* pattern and resolve
- Maintain optimality

Rolling stock schedule $\rightarrow$ shunting events



- Idea: forbid *arrival-and-departure* pattern and resolve
- Maintain optimality

Rolling Stock Scheduling and Depot Parking Results

DSB S-tog cases

Instance	*Time (s)*	Iterations
Fri1	128	1
Fri2	154	1
Fri3	80	1
Fri4	94	1
Fri5	79	1

Instance	*Time (s)*	Iterations
Sat1	23	1
Sat2	18	1
Sat3	23	1
Sat4	13	1
Sat5	11	1

- Different rolling stock schedules
- No initial parking given
- Fast solution time - but no iterations required

Agenda

Conclusions

Summing up

- Developed methods for rolling stock scheduling and depot parking
- Tested on S-tog, DSB and NS cases
- Integration
  - First work to integrate rolling stock and depot parking (incl timetabling and maintenance),

Future

- Reliability and safety first
- Include more practical constraints
- Commitment from collaborators

Conclusions

Summing up

- Developed methods for rolling stock scheduling and depot parking
- Tested on S-tog, DSB and NS cases
- Integration
  - First work to integrate rolling stock and depot parking (incl timetabling and maintenance),

Future

- Reliability and safety first
- Include more practical constraints
- Commitment from collaborators

Thank You

Questions or comments?