Constraint programming for real-time train circulation management in railway nodes

Joaquín Rodriguez

INRETS (Institut National de Recherche sur les Transports et leur Sécurité)

Outlines

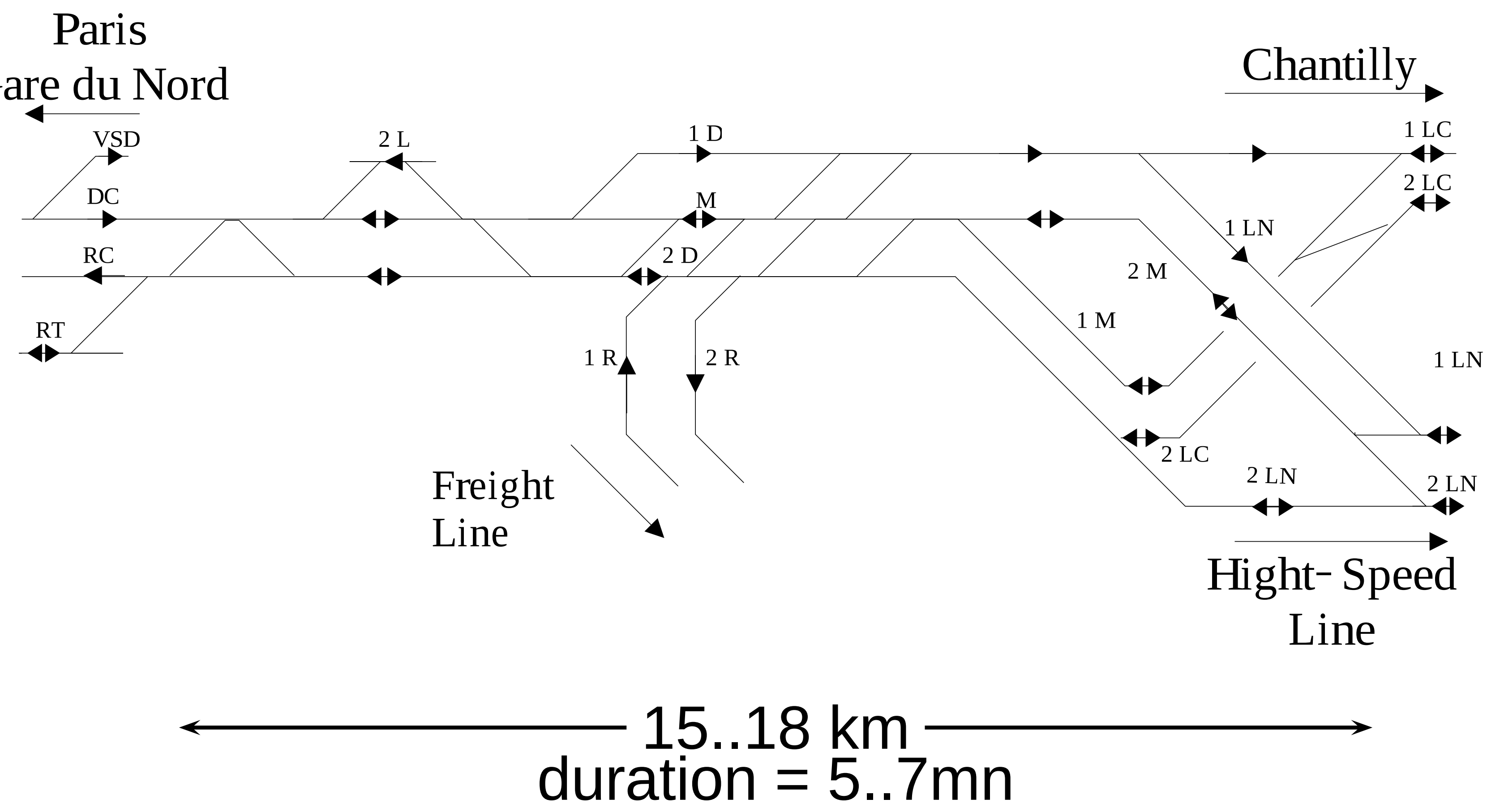
- Introduction
- Real scenario
- Models
  - Simulation model
  - Constraint programming model
- Solution method
- Results
- Conclusions/issues

Introduction

Train circulation management
Paris-Nord Control Center

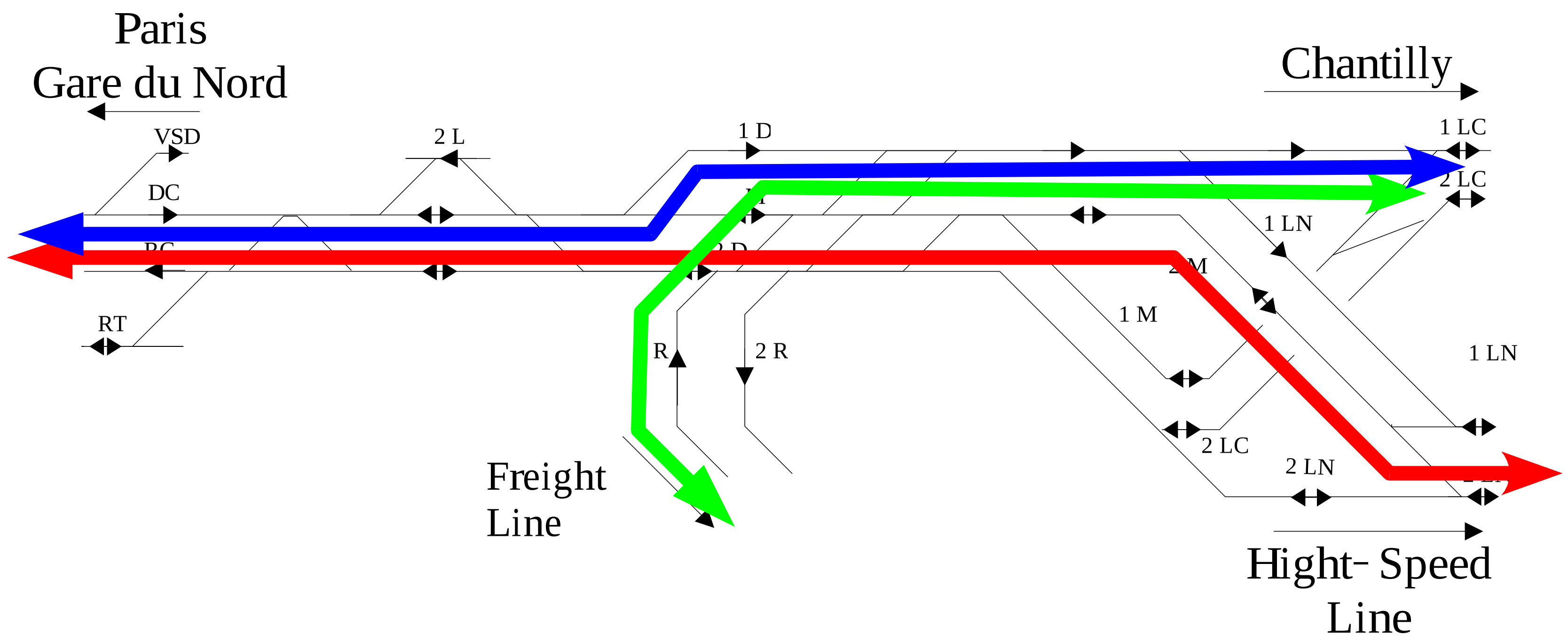
Introduction

Train circulation management
Case study : Pierrefitte-Gonesse node



Introduction

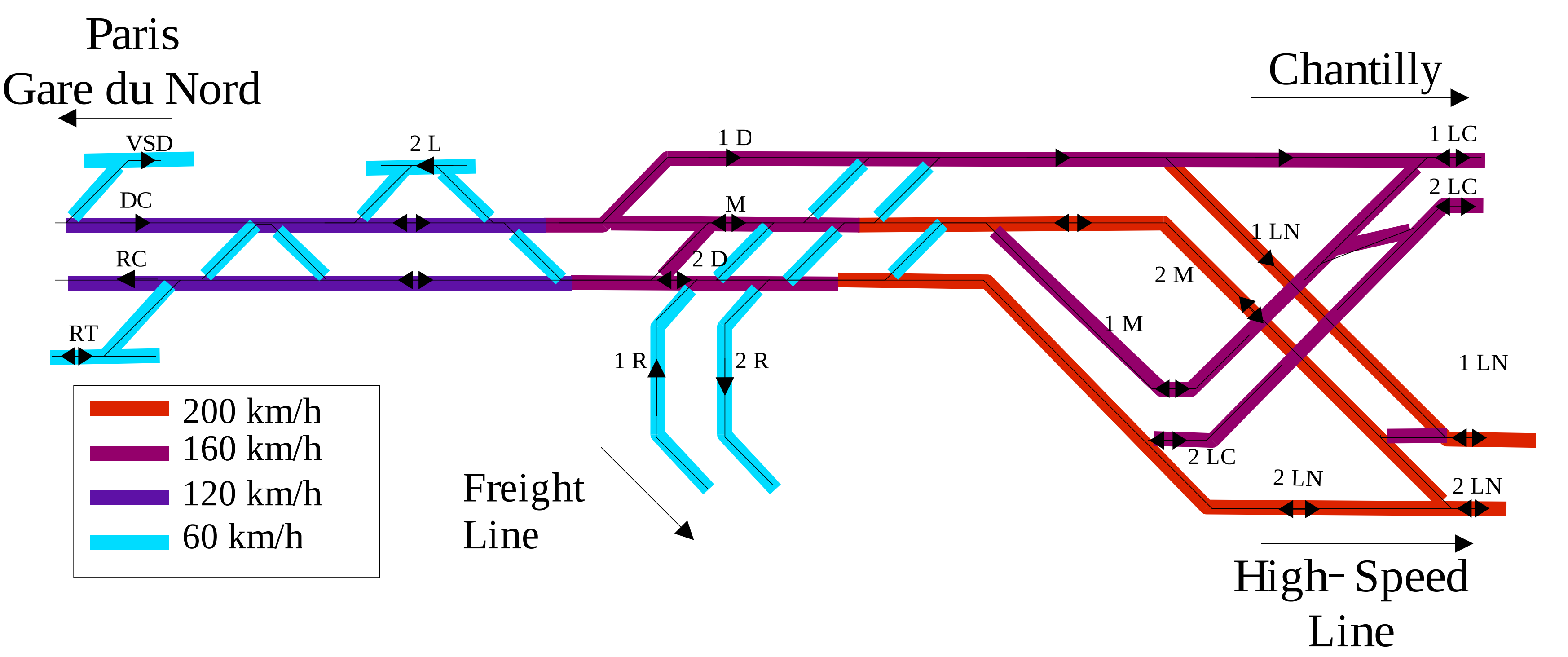
Train circulation management
Case study : Pierrefitte-Gonesse node



Mixed traffic :

- High Speed (North TVG, Eurostar, Thalys)
- Intercity
- Freight

Train circulation management
Case study : Pierrefitte-Gonesse node



Different speed limits

Introduction

Disruptions :

- train delays
- unplaned trains
- unavailable infrastructure devices (points, signals,. . .)
- breakdown (faulty) traction vehicle
- . . .

↪ The forcasted plan can no longer be used

Introduction

Operator has to :

- ◆ Anticipate the conflicts,
- ◆ Use alternative routes, train orders.

Disrupted situations :

- ◆ A lot of information,
- ◆ Little time to make a decision.

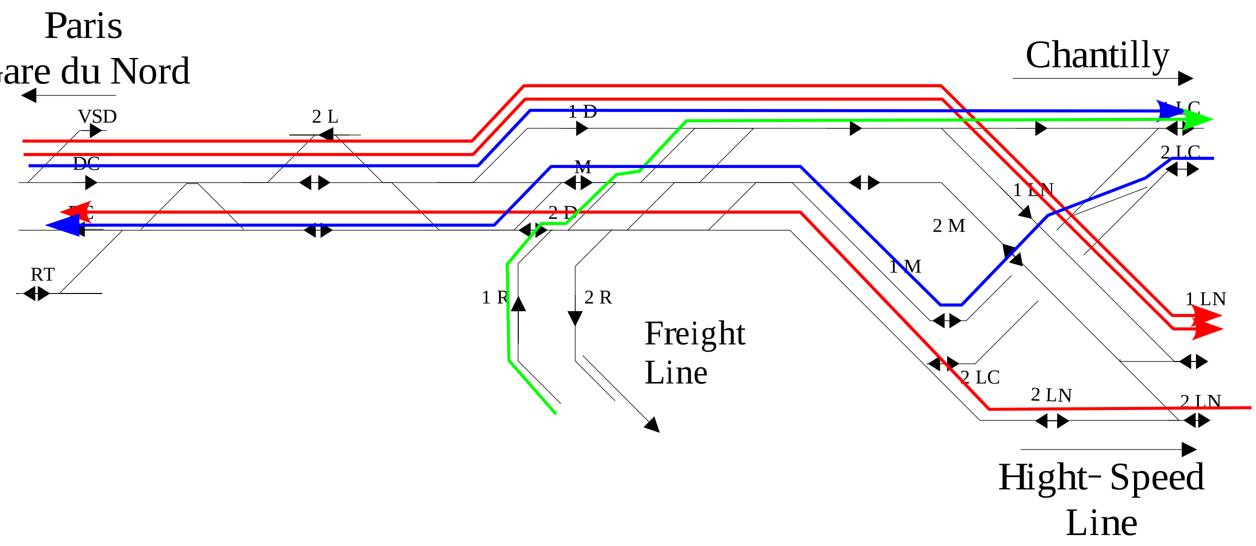
☞ Tool to generate and evaluate a set of solutions

➥ find a **good solution** (i.e. not necessary optimal)

➥ criterion : minimise the sum of the delays caused to the trains

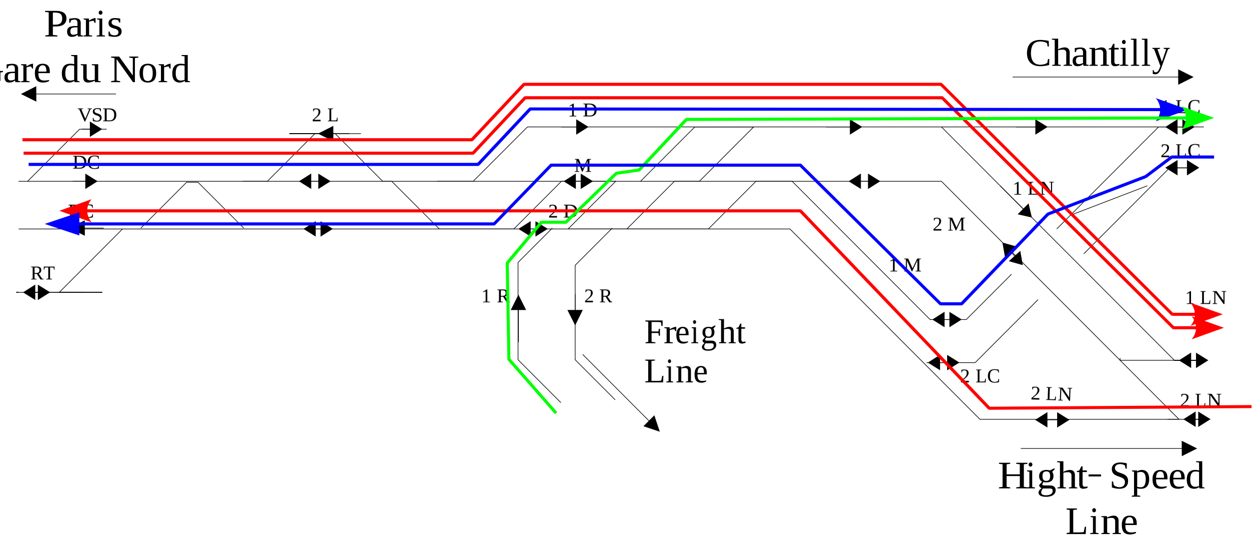
☞ Model based on **simulation** and **constraint programming** (CP)

Case study : Pierrefitte-Gonesse node
6 trains



Real scenario

Case study : Pierrefitte-Gonesse node

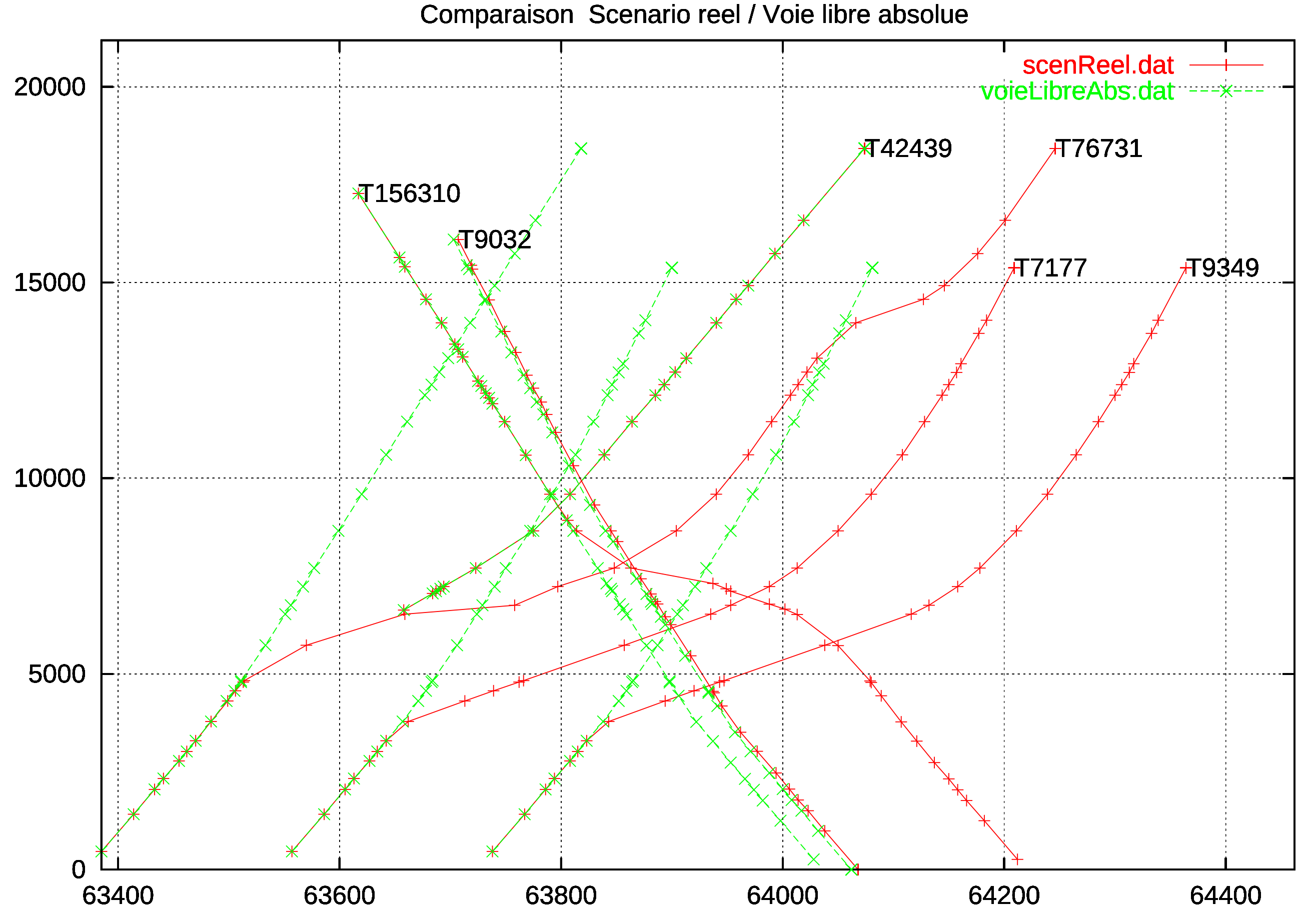


Disruptions :

Trains	Delays(seconds)
North TVG 7177	+7
North TVG 9349	+210
Intercity 76731	+280
Freight 42439	+1958
Intercity 156310	+1610
Eurostar 9032	-215

Real scenario

Simulation of the 6 train scenario



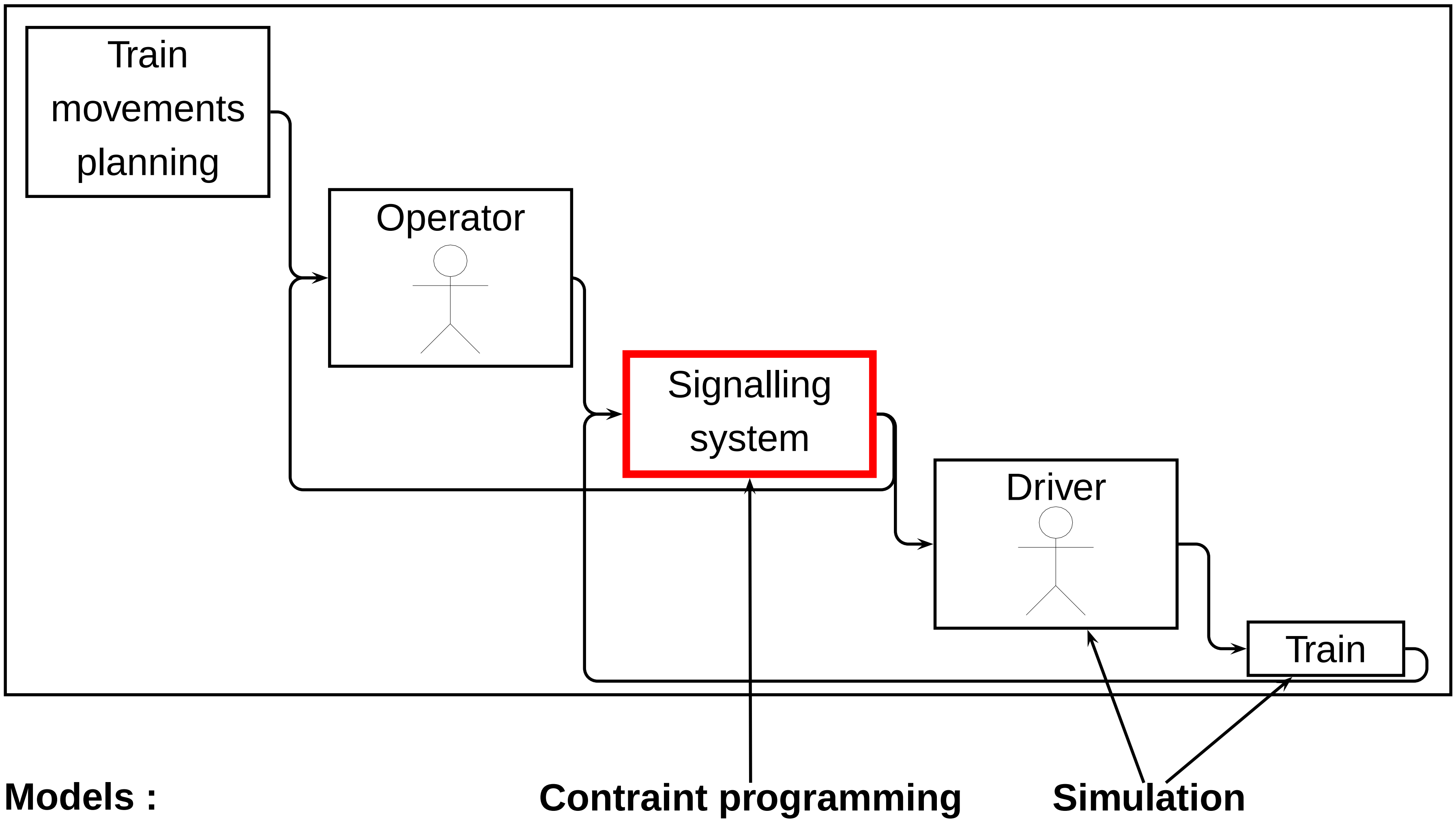
Time-distance graph of : - train runs with maximum speed

- train runs of the simulated scenario

sum of delays = 1210s ($\simeq$ 20 mn)

Models

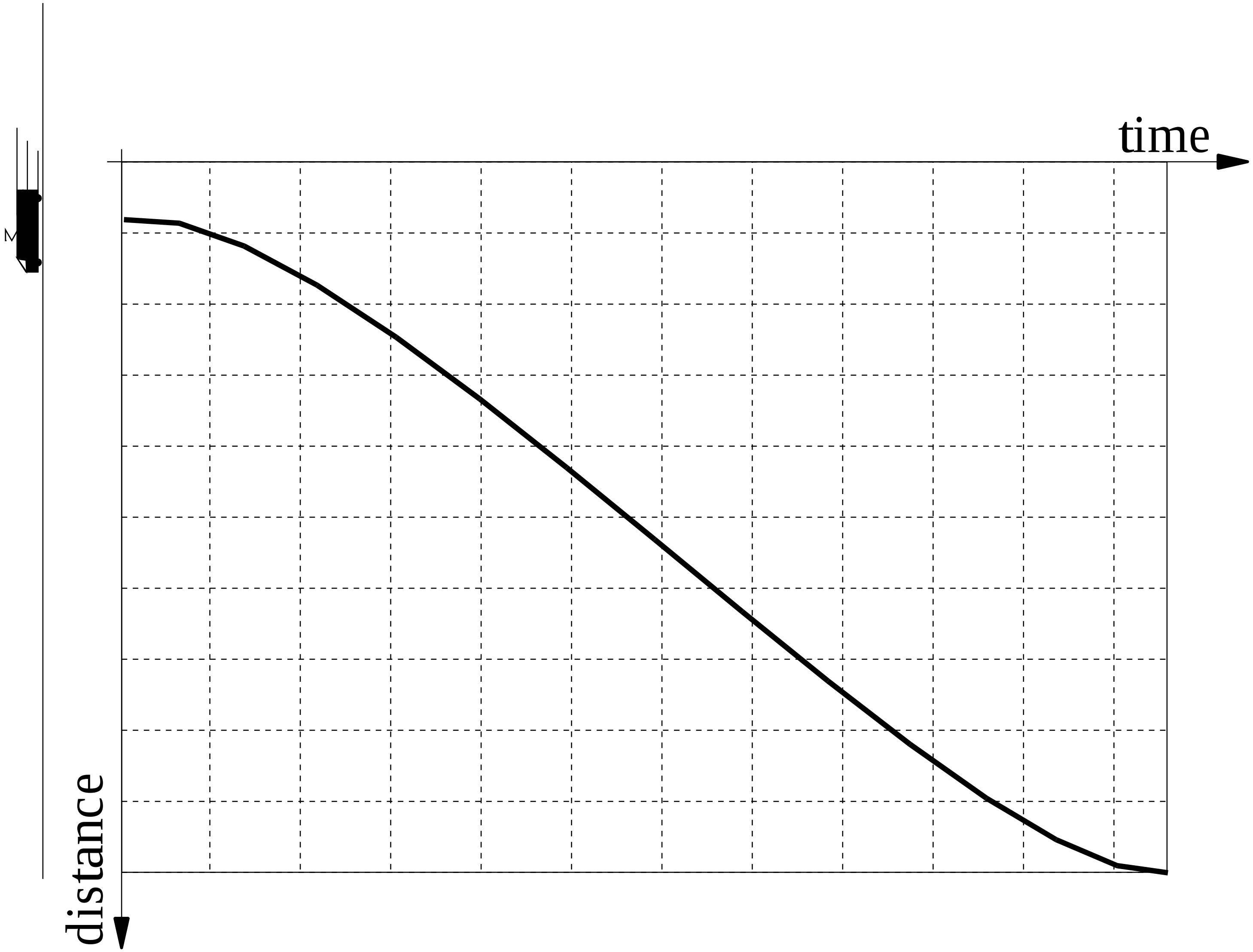
Functional diagram



Simulation model

Train movements

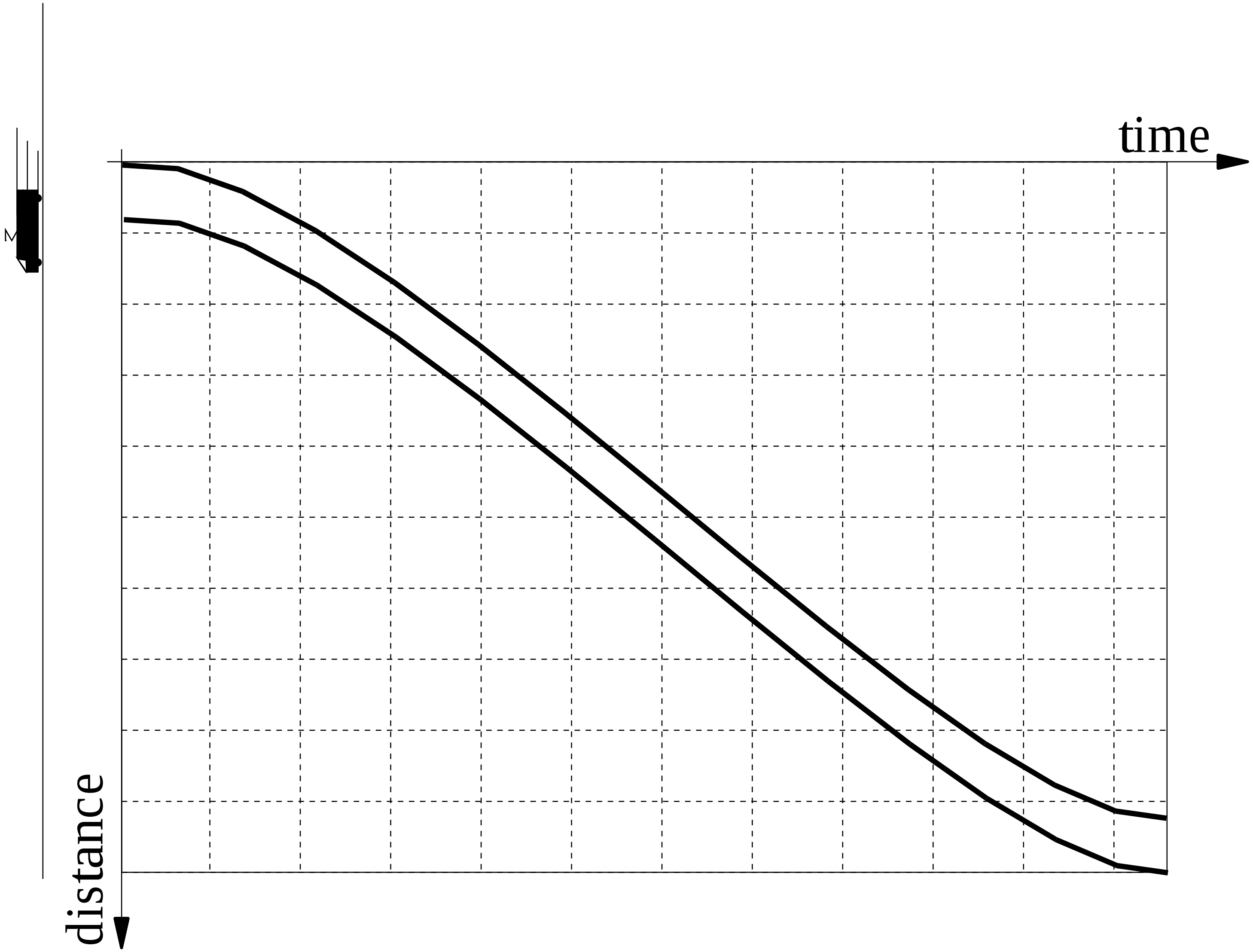
Time-Distance graph of head



Simulation model

Train movements

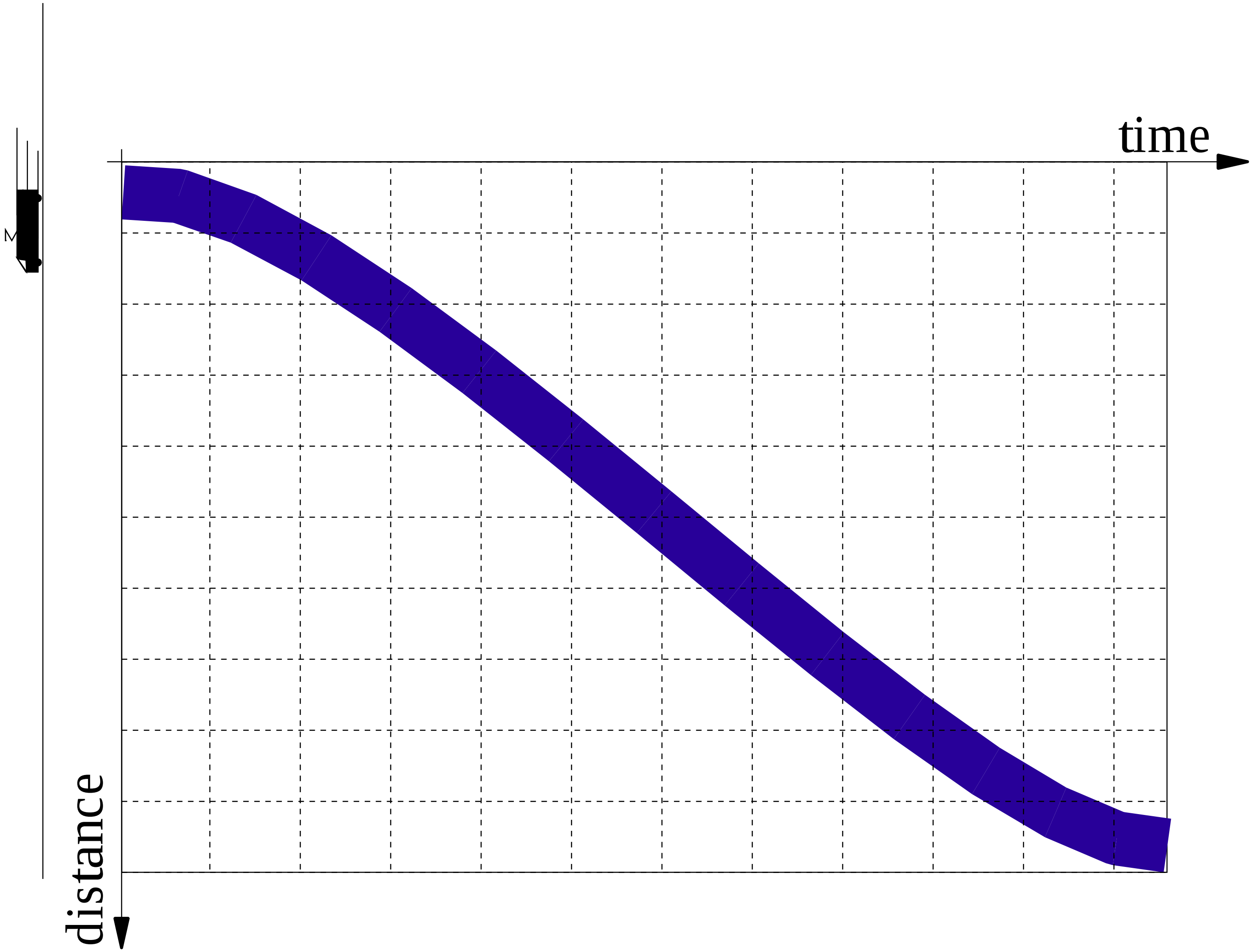
Time-Distance graph of head+tail



Simulation model

Train movements

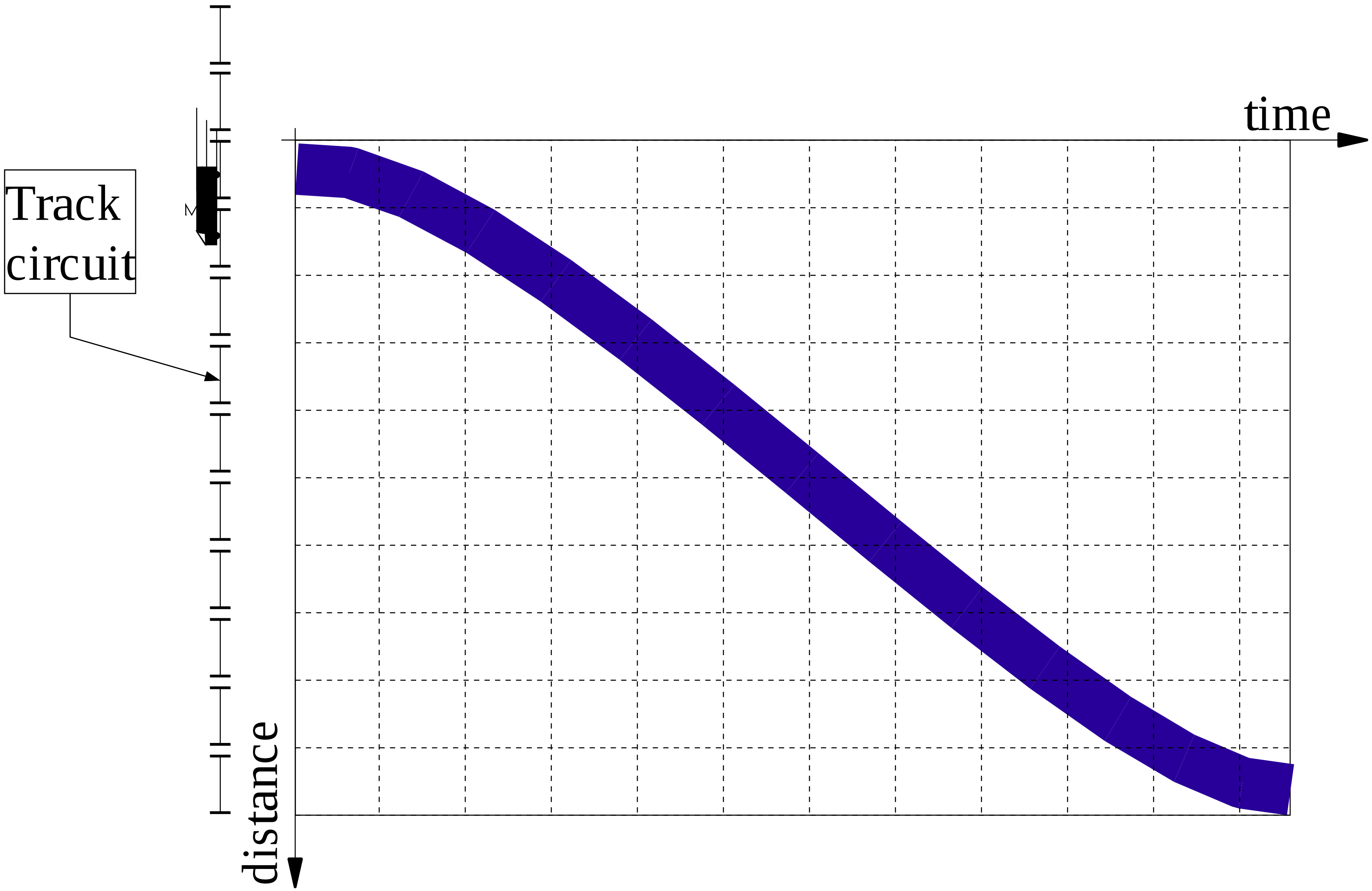
Time-Distance graph of head+tail



Simulation model

Train movements are detected by track circuits

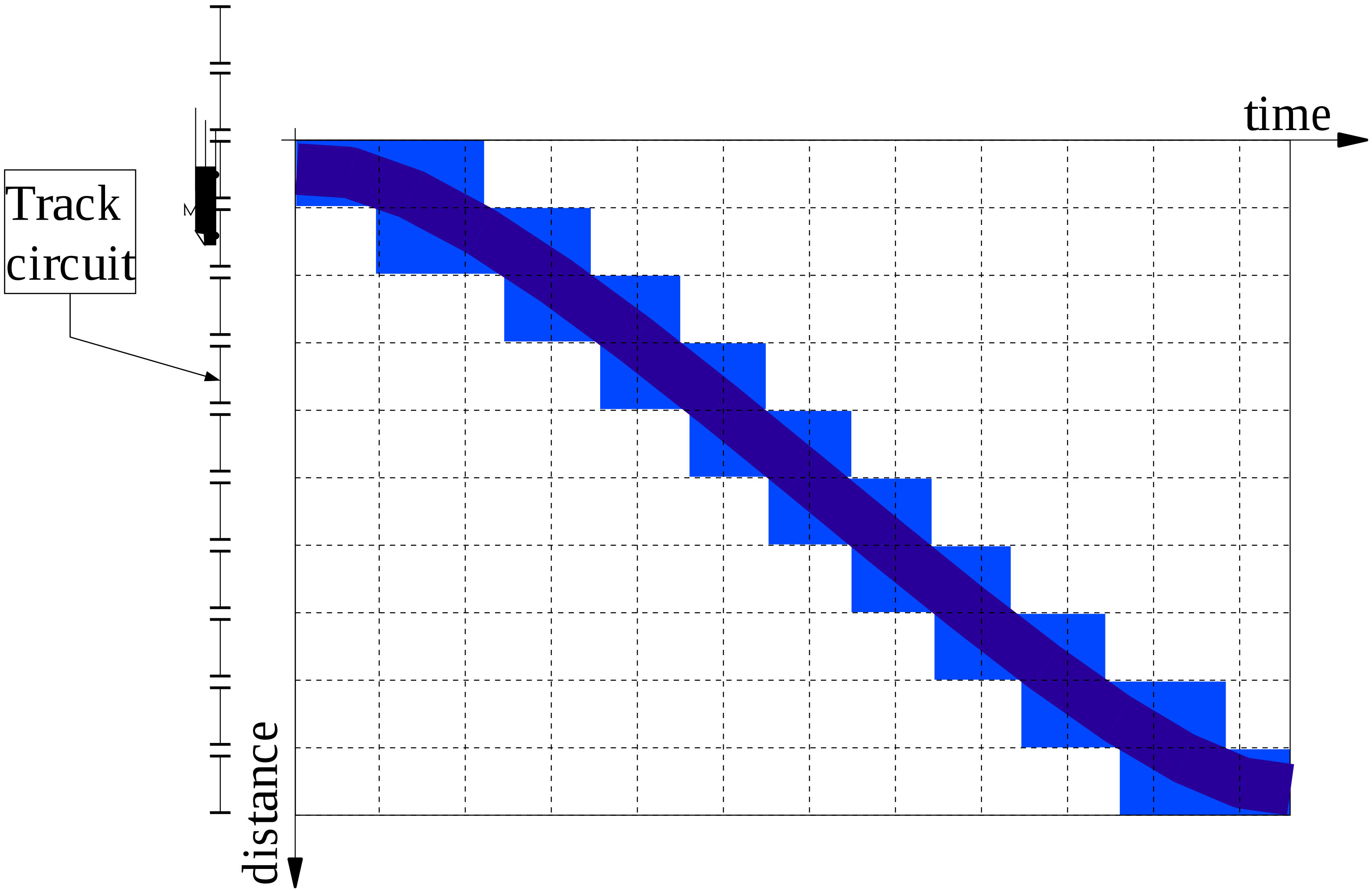
Time-Distance graph of detection



Simulation model

Train movements are detected by track circuits

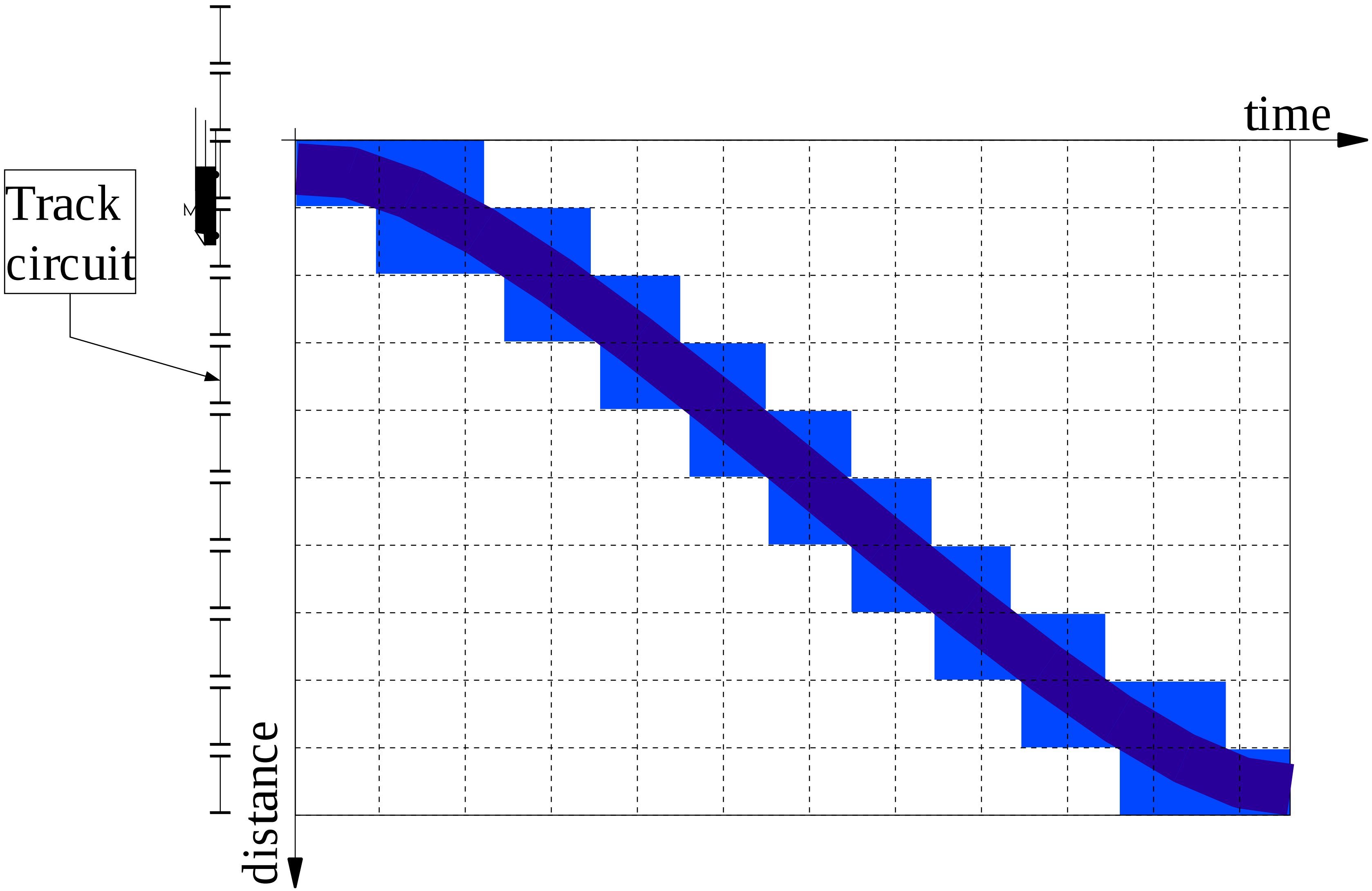
Time-Distance graph of detection



Simulation model

Train movements are detected by track circuits

Time-Distance graph of detection $\rightarrow$ input data of the CP model



Simulation model

for each (train,route)

sequence of events :

- **track circuit start of detection**
- **track circuit end of detection**

Use of Multi-Train railway simulator (SNCF) : SISYFE

Constraint programming model

$\boxed{\text{Train circulation management} \equiv \text{Scheduling Problem}}$

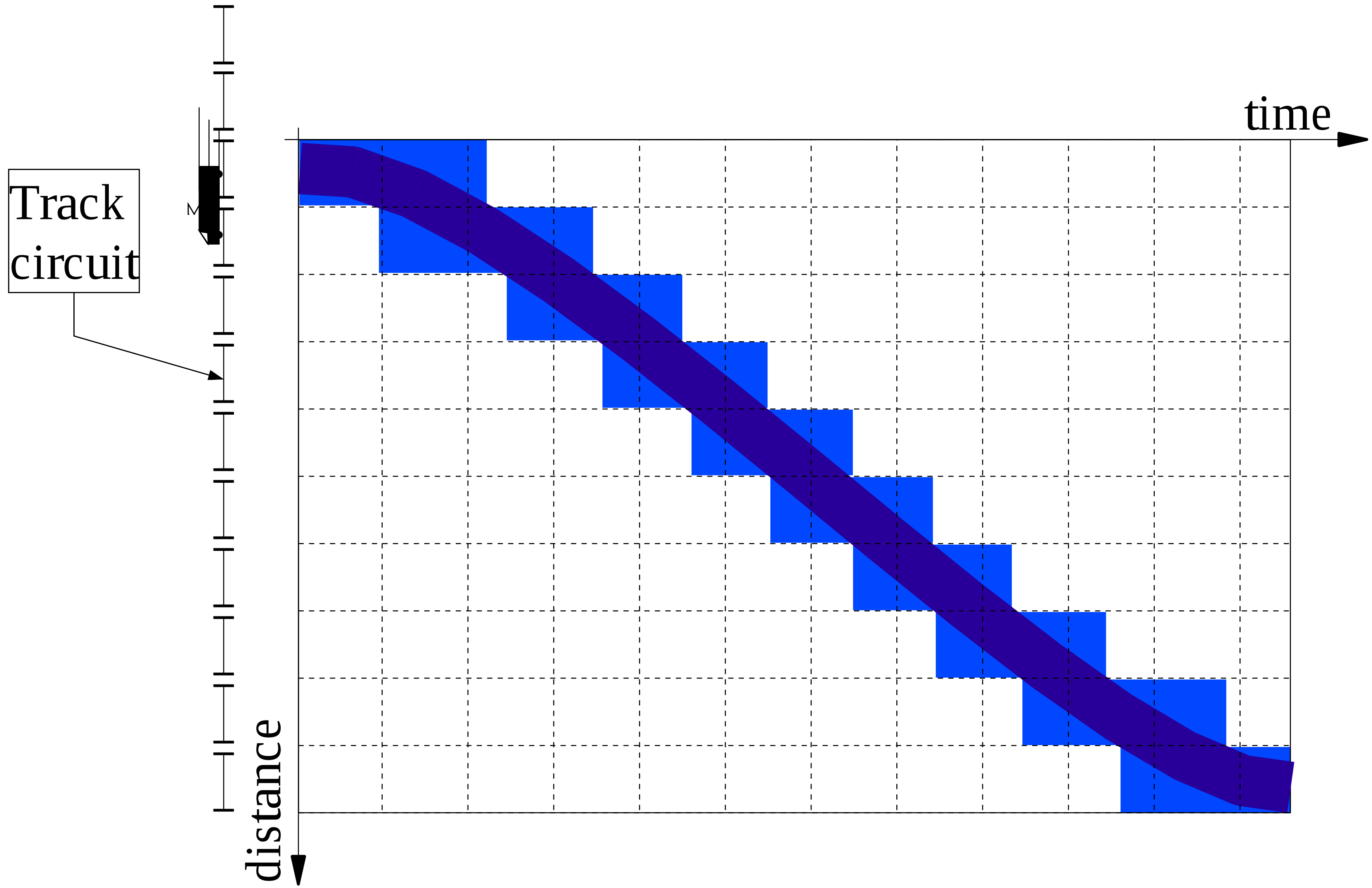
i.e. *allocating scarce resources to activities over time*

Model :

activities	?
resources	?
resource constraints	?
temporal constraints	?

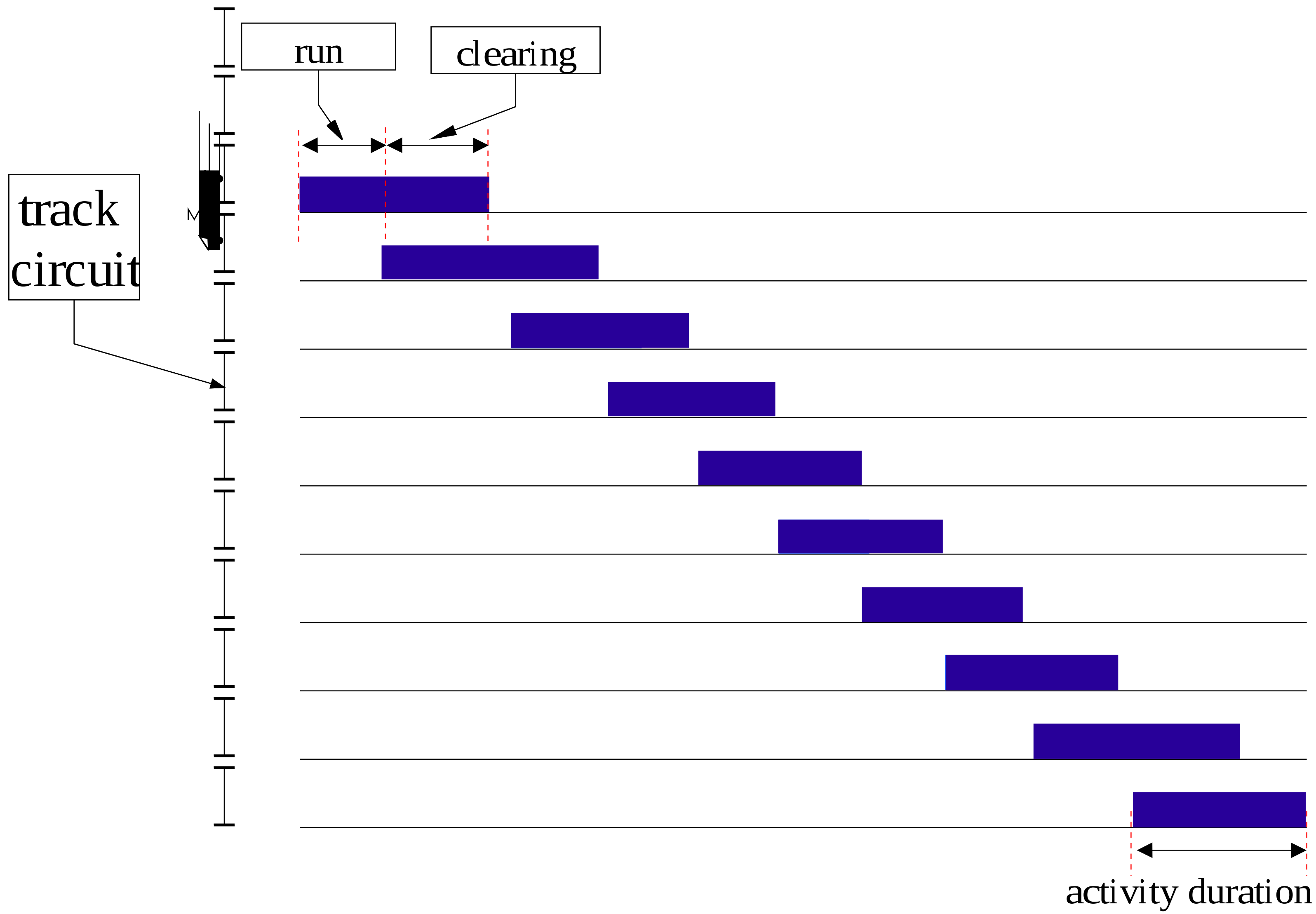
Constraint programming model

Time-Distance graph



Constraint programming model

Time-Distance graph $\rightarrow$ Gantt chart of activities

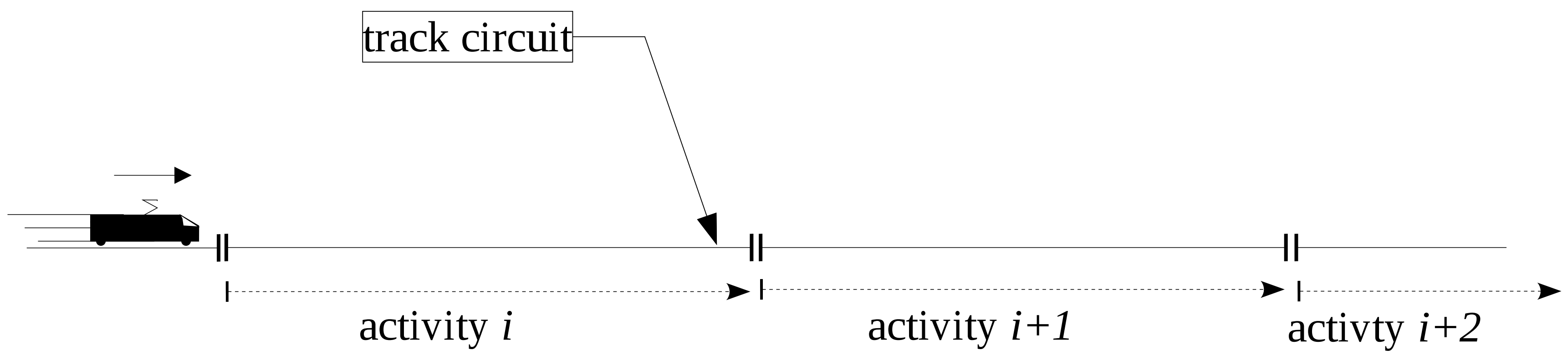


Constraint programming model

A train is a *sequence of activities*

Activity $\mapsto$?

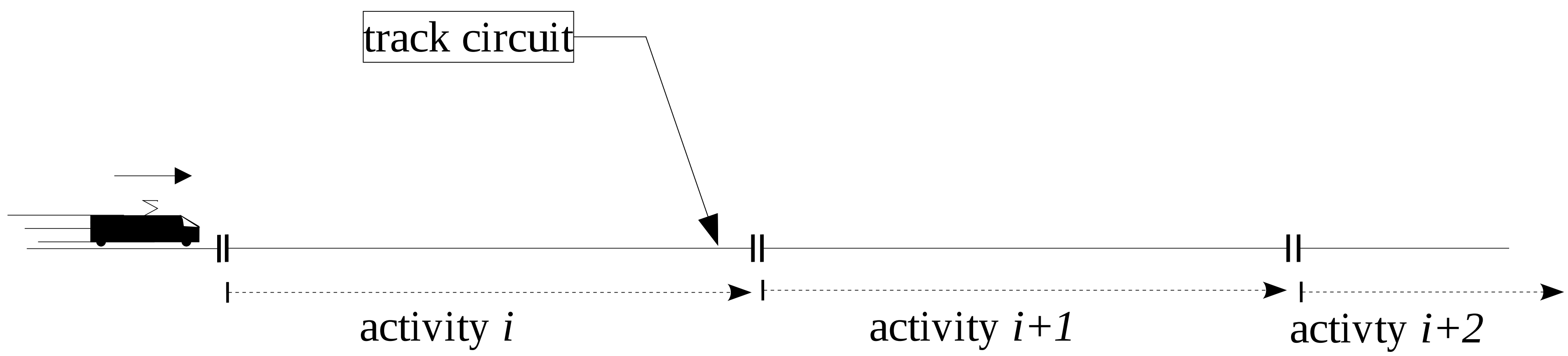
Ressource $\mapsto$?



Constraint programming model

A train is a *sequence of activities*

Activity	$\mapsto$	elementary run
Ressource	$\mapsto$	track circuit



Constraint programming model

Resource constraints

- ✓ capacity(track circuit) = 1

Constraint programming model

Resource constraints

- ✓ capacity(track circuit) = 1
- ✓ requires($activity_i$,tc_i) — tc_i : track circuit variable of the $activity_i$

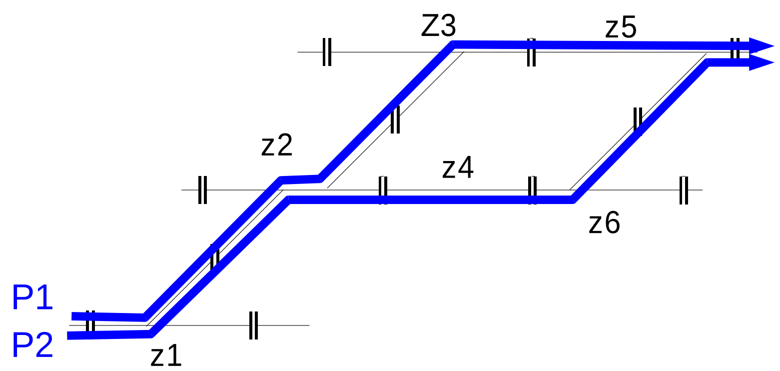
Constraint programming model

Resource constraints

- ✓ capacity(track circuit) = 1
- ✓ requires($activity_i$,tc_i) — tc_i : track circuit variable of the $activity_i$
- ✓ Assignment constraint

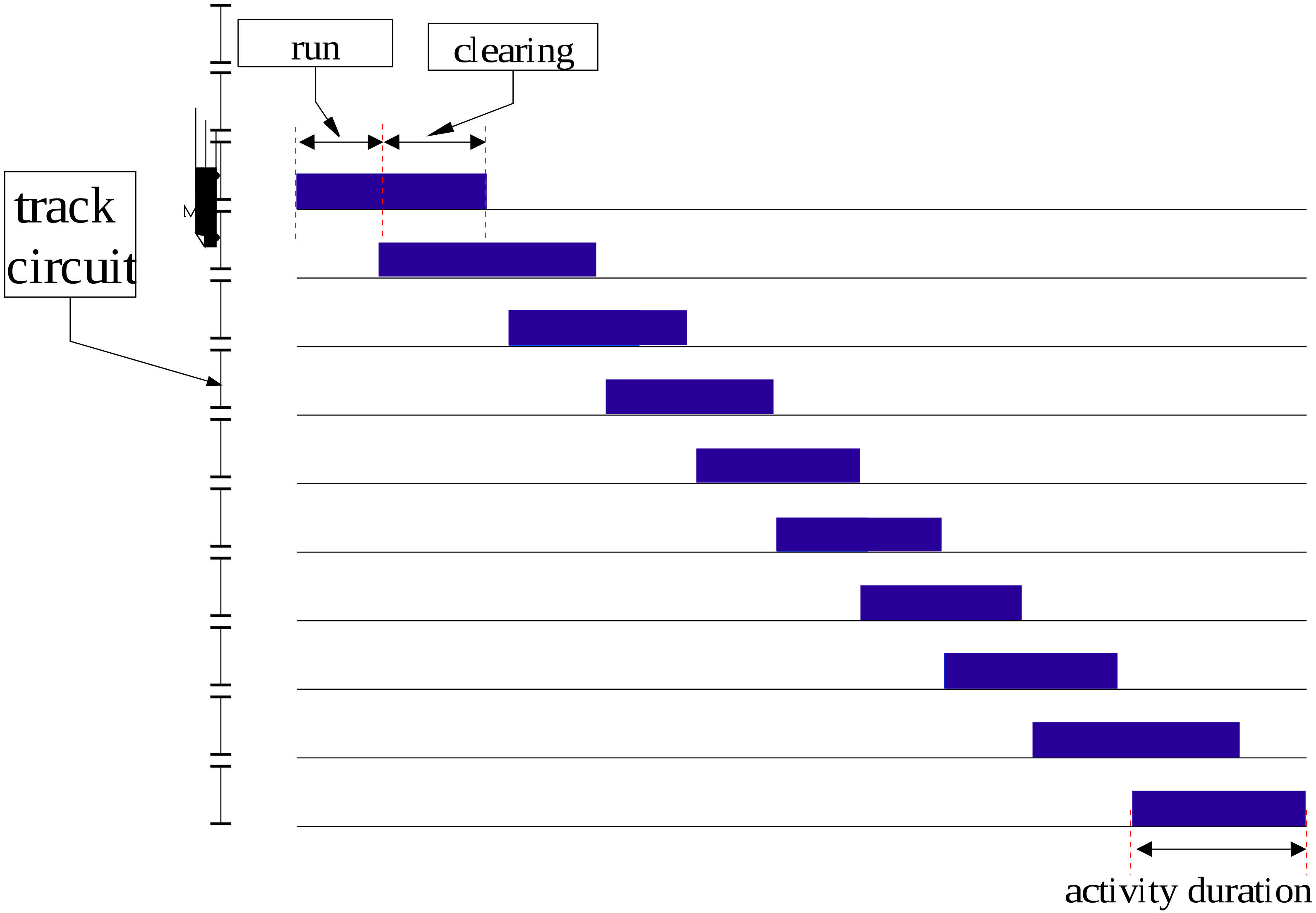
Assignment constraint					
r	tc_1	tc_2	tc_3	tc_4	tc_5
P1	z1	z2	z3	z*	z5
P2	z1	z2	z4	z6	z5

r : the variable of the alternatives routes



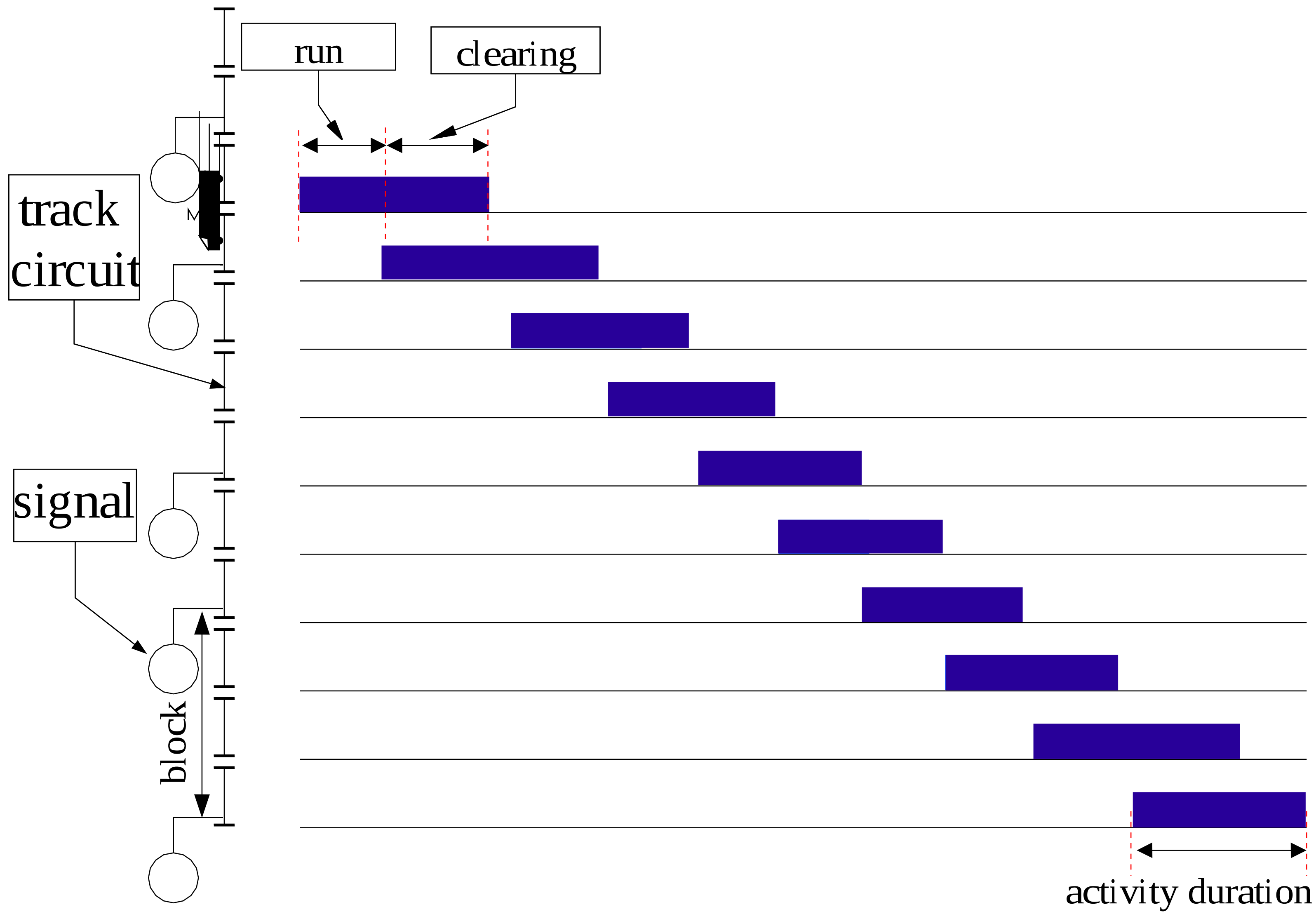
Constraint programming model

Gantt chart of activities : Block signalling system ?



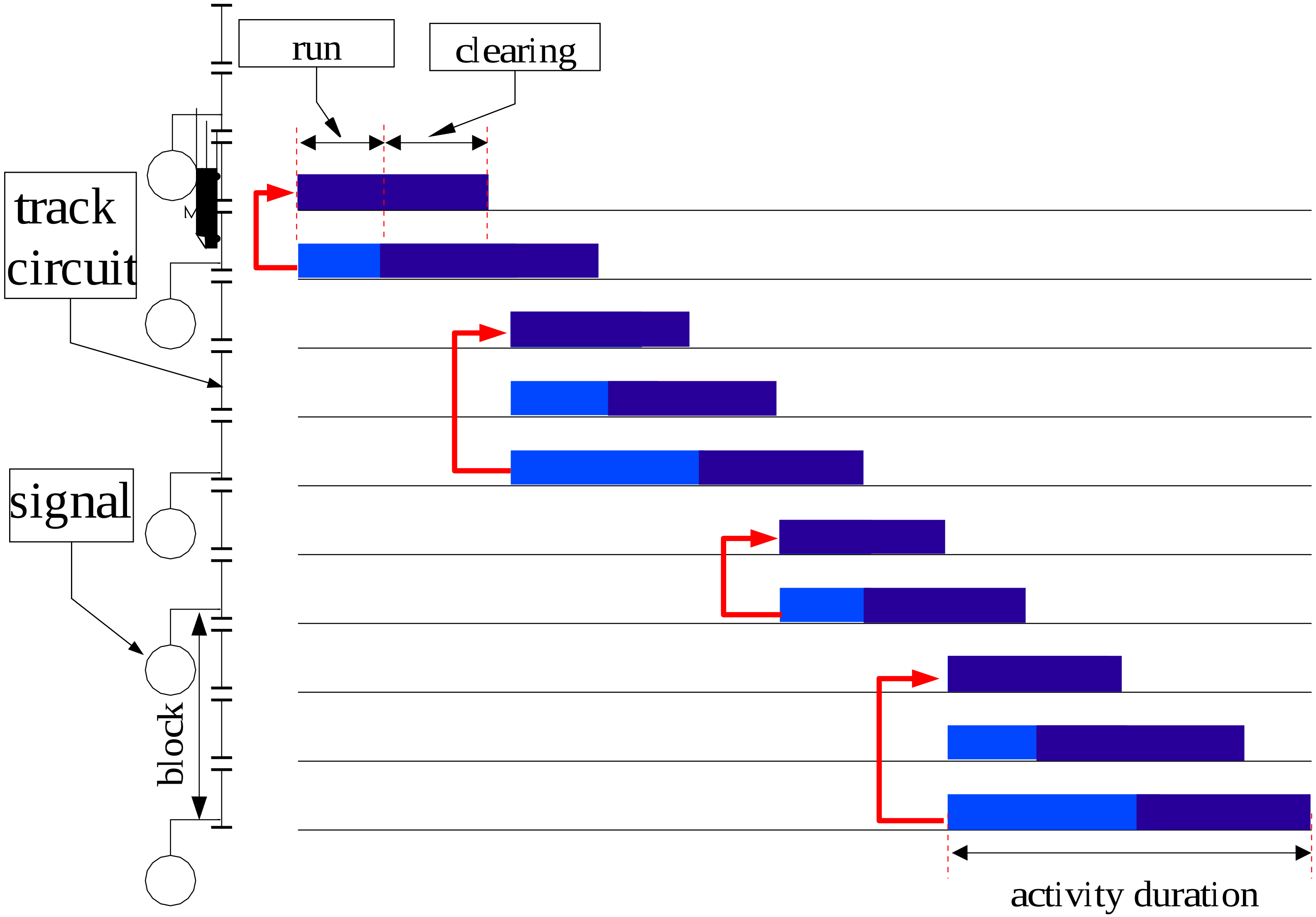
Constraint programming model

Gantt chart of activities : Block signalling system ?



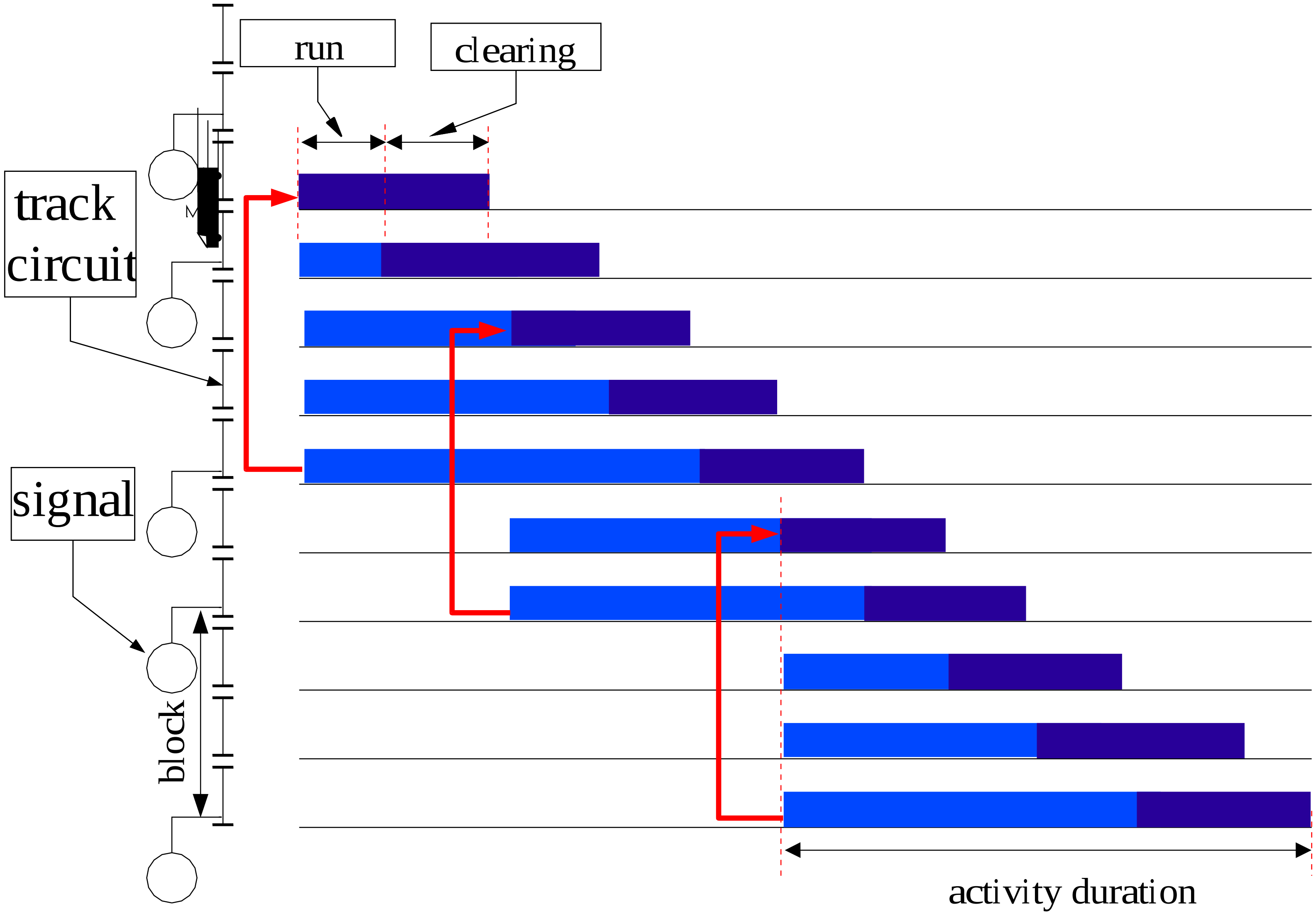
Constraint programming model

Gantt chart of activities : 2 aspect block signalling system



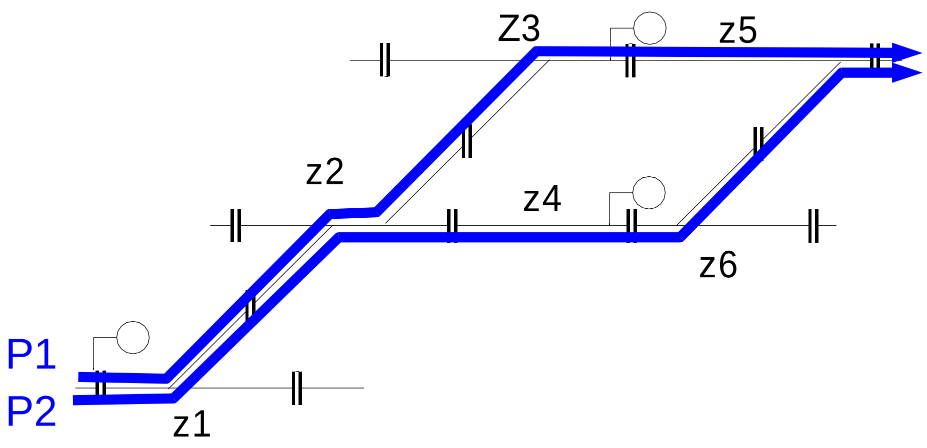
Constraint programming model

Gantt chart of activities : 3 aspect block signalling system

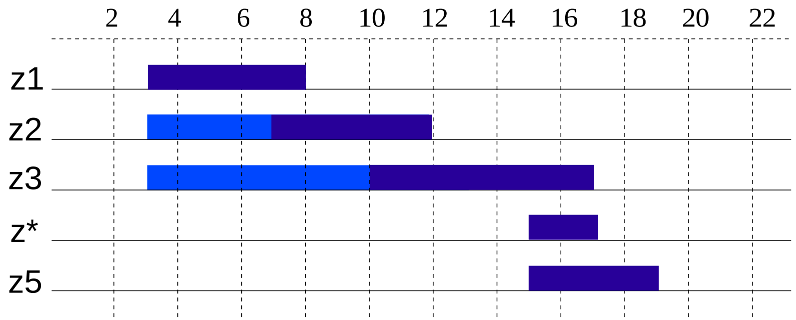


Constraint programming model

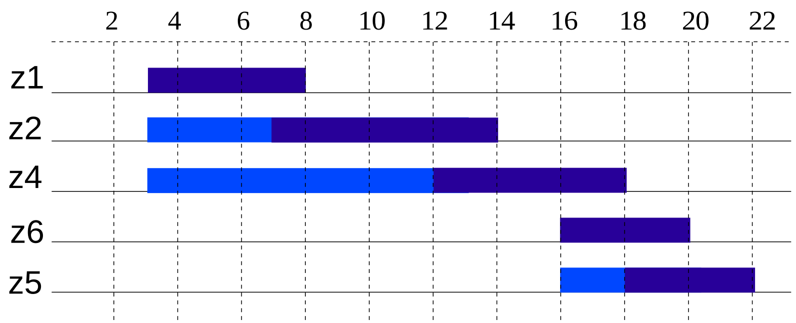
Example of routes P1,P2



Gantt chart of route P1



Gantt chart of route P2



Constraint programming model

Temporal constraints

✓

earliest start time constraint					
r	est_1	est_2	est_3	est_4	est_5
P1	3	3	3	15	15
P2	3	3	3	16	16

est_i : earliest start time of the $activity_i$

Constraint programming model

Temporal constraints

✓

earliest start time constraint					
r	est_1	est_2	est_3	est_4	est_5
P1	3	3	3	15	15
P2	3	3	3	16	16

est_i : earliest start time of the $activity_i$

✓

earliest finish time constraint					
r	eft_1	eft_2	eft_3	eft_4	eft_5
P1	8	12	17	17	19
P2	8	14	18	20	22

eft_i : earliest finish time of the $activity_i$

Constraint programming model

Temporal constraints

- ✓ $ft_i = eft_i + \sum_{j=1}^{i} d_j$

ft_i : finish time of $activity_i$

d_i : delay of $activity_i$

Constraint programming model

Temporal constraints

✓ $ft_i = eft_i + \sum_{j=1}^{i} d_j$

ft_i : finish time of $activity_i$

d_i : delay of $activity_i$

✓

Block synchronization constraint					
r	s_1	s_2	s_3	s_4	s_5
P1	1	1	1	4	5
P2	1	1	1	4	4

s_i : index of the activity to synchronize the $activity_i$

Constraint programming model

Temporal constraints

✓ $ft_i = eft_i + \sum_{j=1}^{i} d_j$

ft_i : finish time of $activity_i$

d_i : delay of $activity_i$

✓

Block synchronization constraint					
r	s_1	s_2	s_3	s_4	s_5
P1	1	1	1	4	5
P2	1	1	1	4	4

s_i : index of the activity to synchronize the $activity_i$

✓ $st_i = est_i + \sum_{j=1}^{s_i-1} d_j$

st_i : start time of $activity_i$

Solution method

Tree search

1. Each node :
   (a) labelling procedure
   (b) consistency (propagation) procedure
2. Optimisation : each time a solution is found, post a constraint on the criterion variable

Variable choice order :

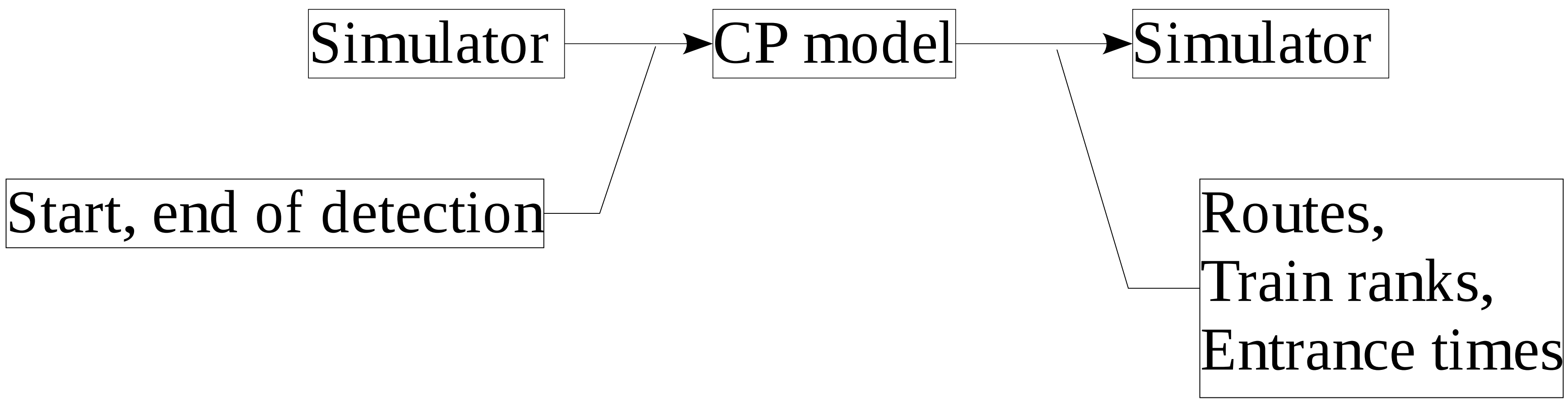
1. Resource assignment variables of activities (tc_i)
   (a) Minimum size domain
2. Time variables of activities (st_i, ft_i)
   (a) Minimum earliest value

Solution method

Implementation :

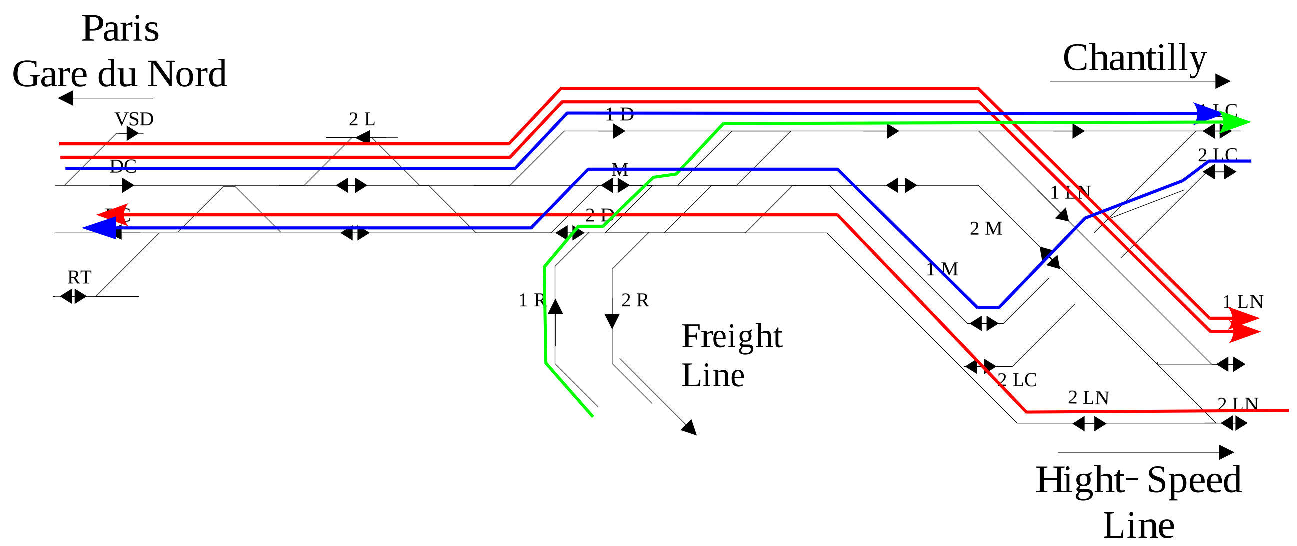
C++ Library Ilog Solver, Scheduler, Views

Validation of solutions :



Results

Real scenario :



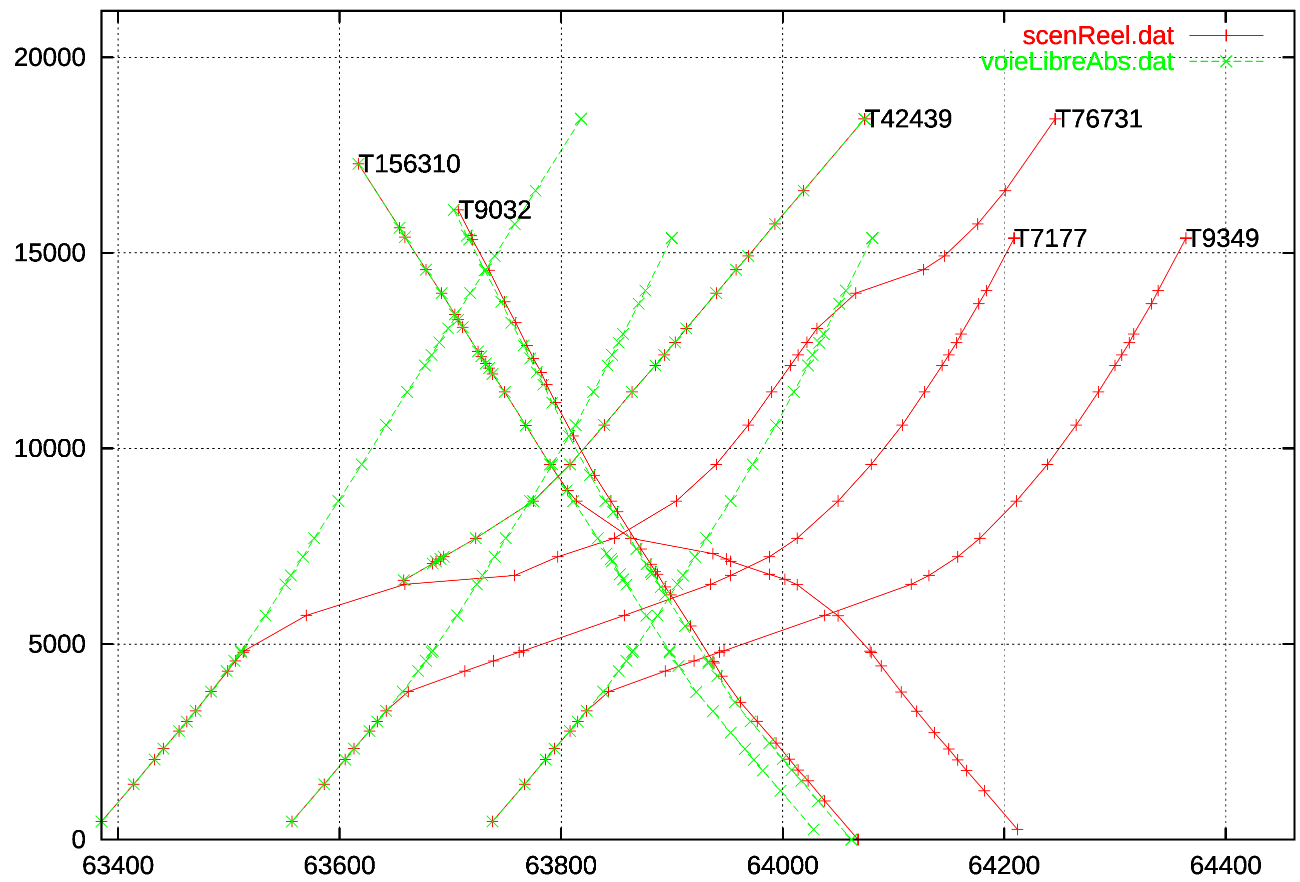
Size of the instance problem :

- Number of activities per train : [20,36]
- Number of variables : 3019
- Number of constraints : 1737
- Number of alternative routes per train : [4,8]

Assignment problem (6 trains) : 25920 combinations

Results

Simulation of the 6 train scenario



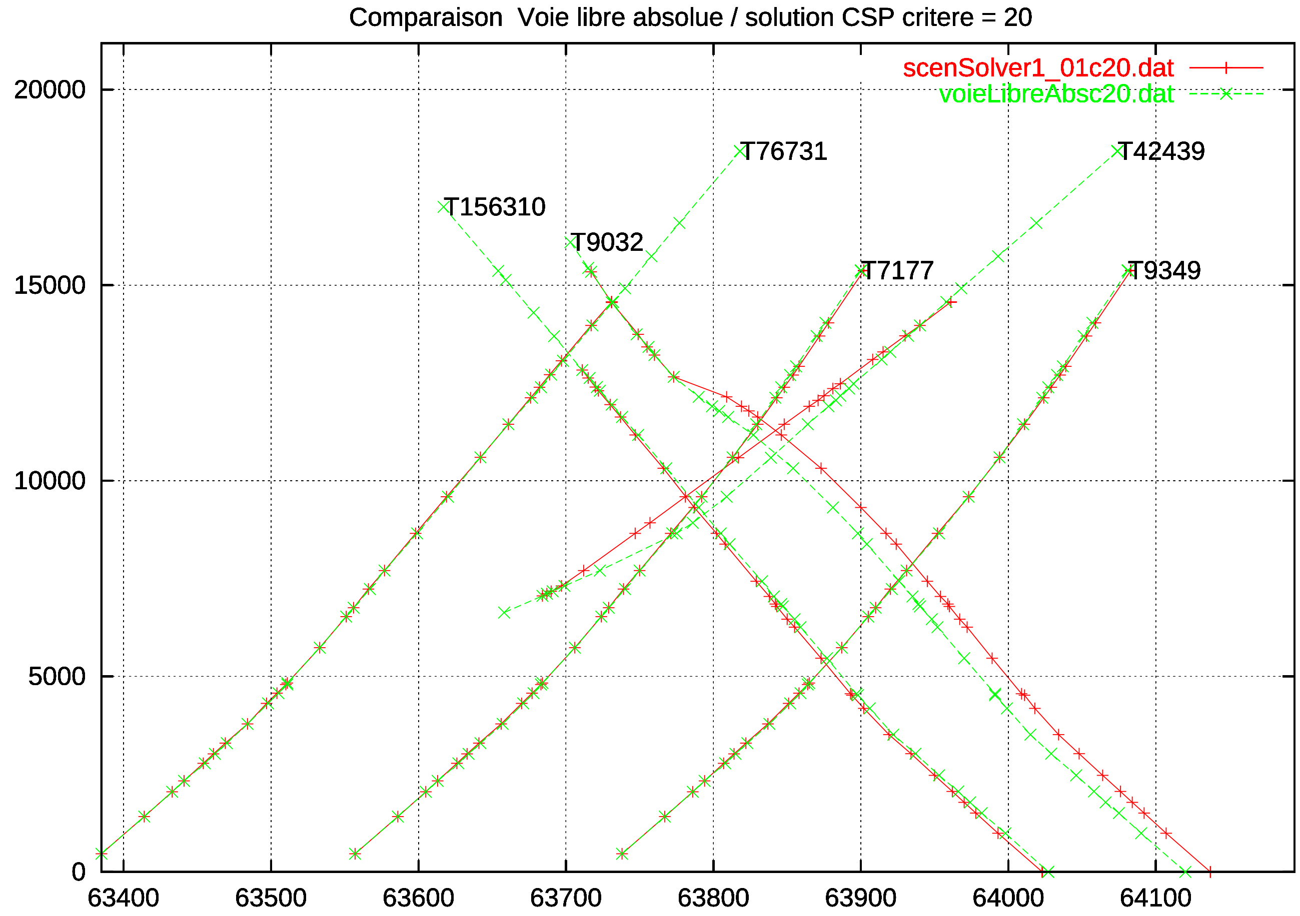
Time-distance graph of : - train runs with maximum speed

- train runs of the simulated scenario

sum of delays = 1210s

Results

Best solution (Temps CPU = 32s)



Time-distance graph of : - train runs with maximum speed

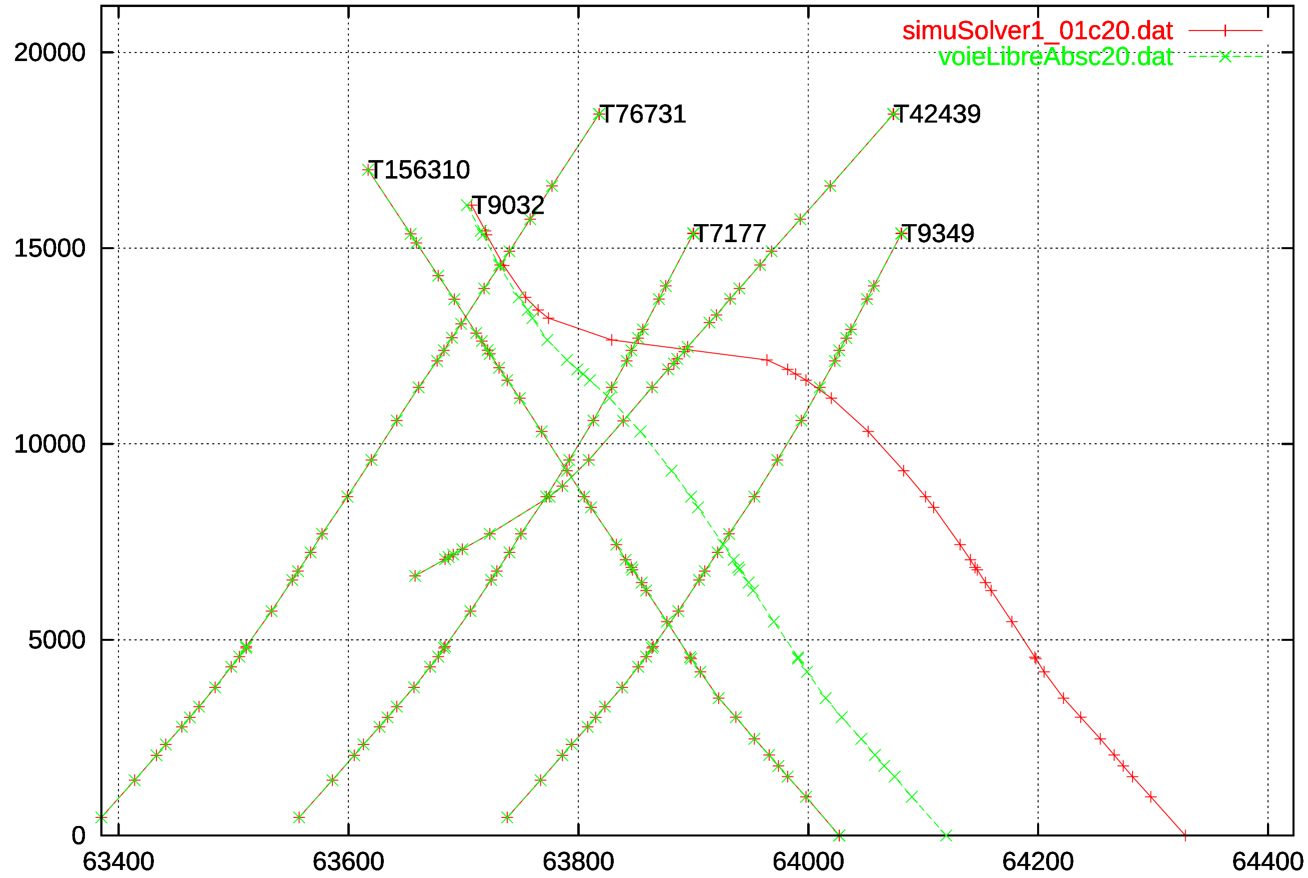
- train runs of the simulated scenario

sum of delays = 20s

Results

Simulation of the best solution

Comparaison Voie libre absolue / solution simulee CSP critere = 20



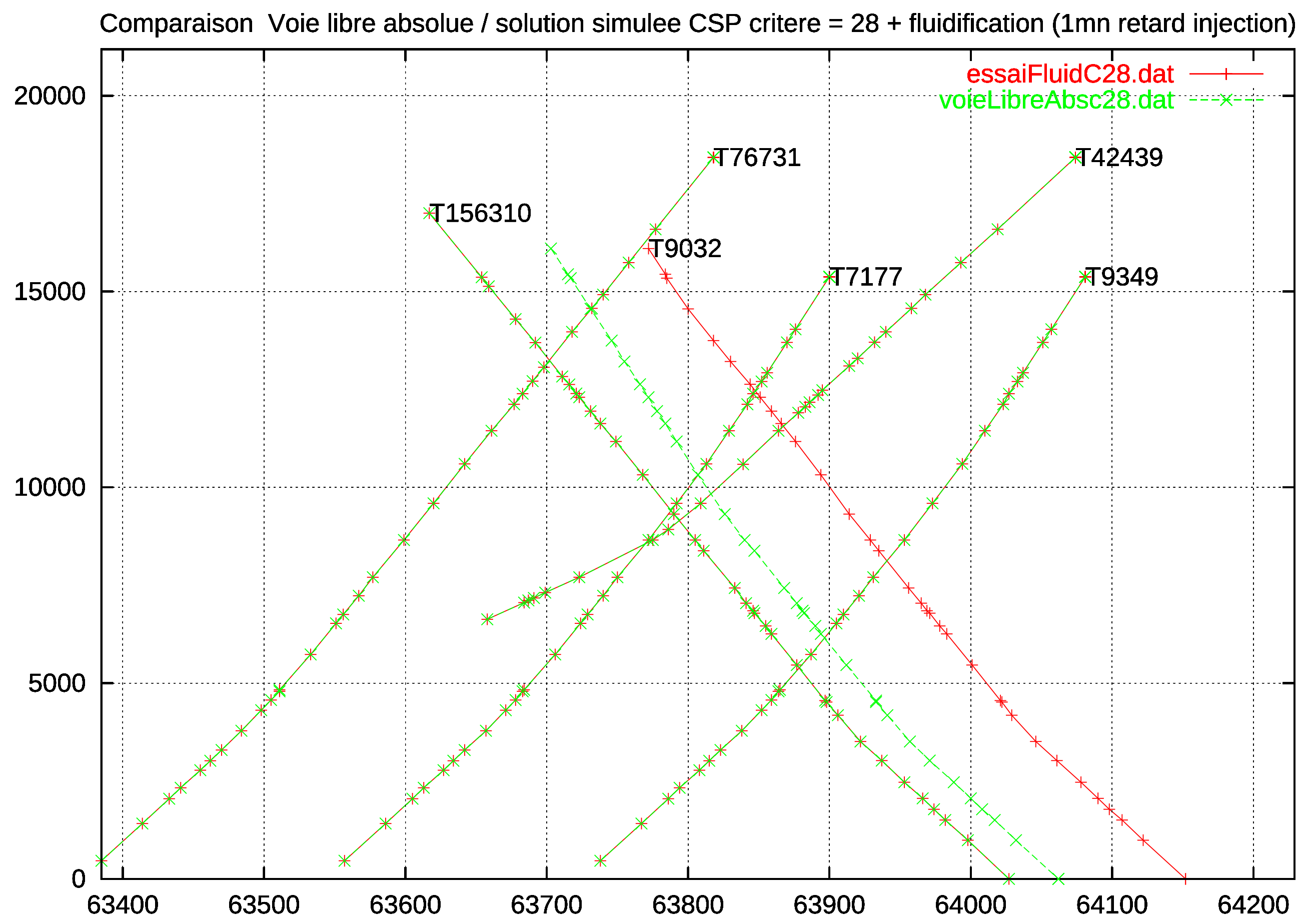
Time-distance graph of : - train runs with maximum speed

- train runs of the simulated scenario

sum of delays = 300s

Results

Best solution after relaxation of entrance constraint



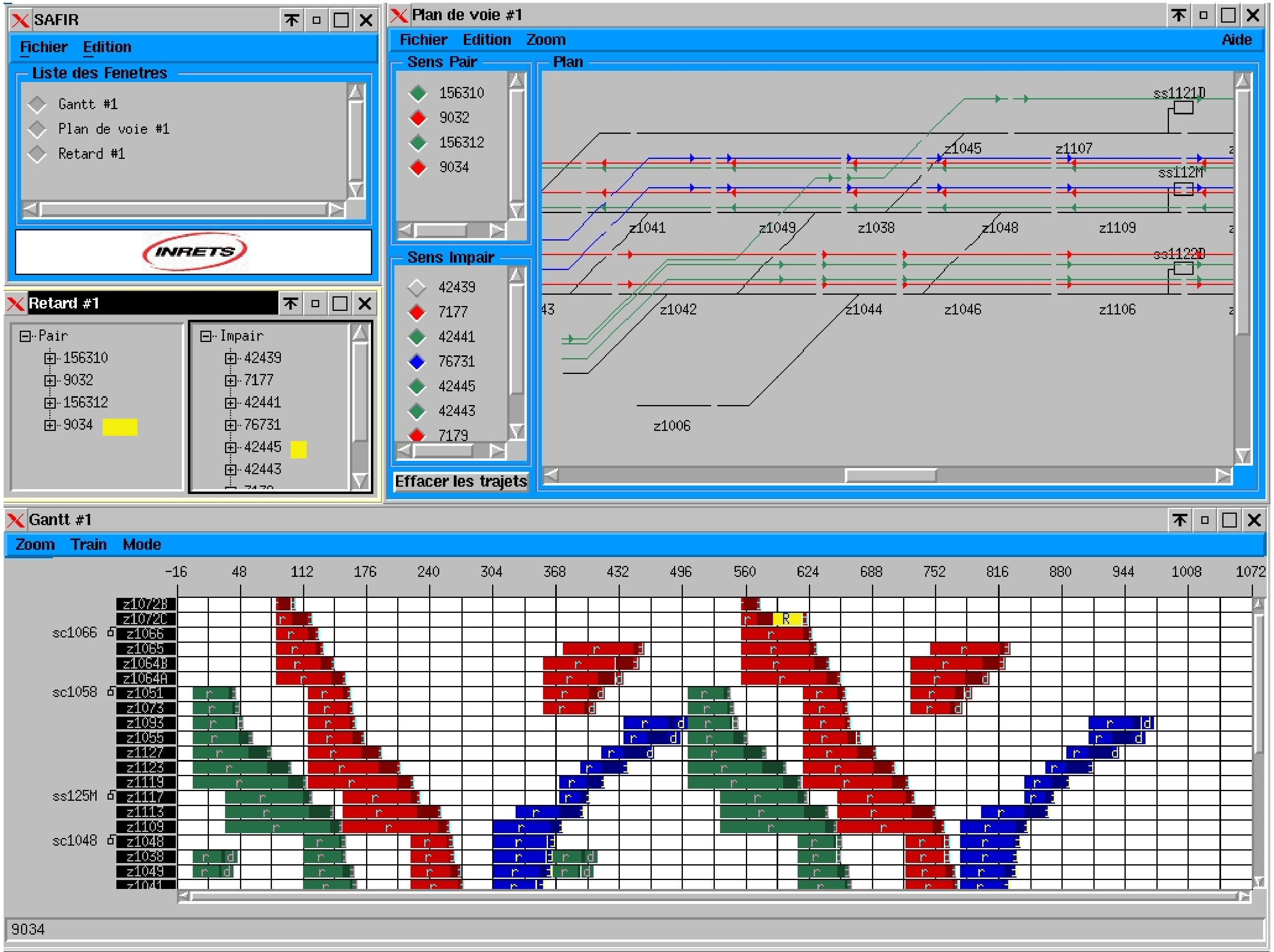
Time-distance graph of : - train runs with maximum speed

- train runs of the simulated scenario

sum of delays = 80s

Results

Graphic tools



Conclusions/ Issues

Conclusions :

☞ CP model :
- Input data : infrastructures, start/finish time detection
- Signalling system (number of aspects)
- No description of all possible conflicts

☞ Scenarios 6-10 trains : solutions with good ratio criterion value/CPU

Issues :

☞ Pre-processing to simplify the model

☞ Model acceleration/deceleration of trains

☞ Integrated CP-MIP approach

☞ Application to capacity analysis