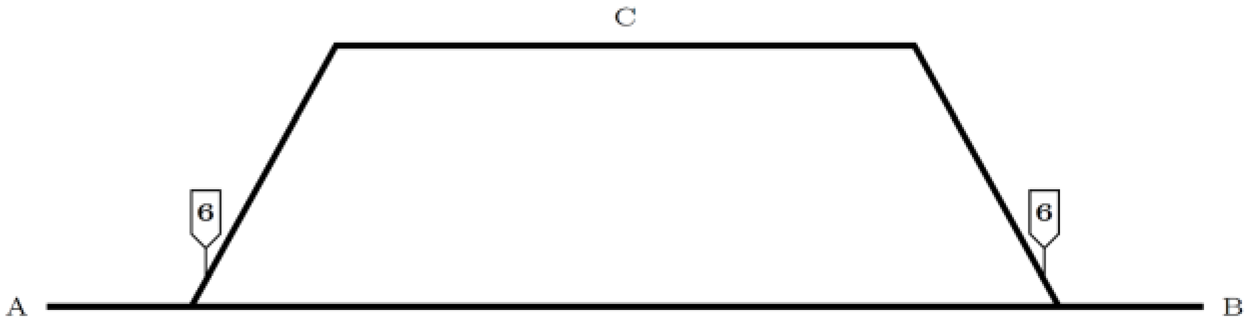
Optimal Railway Routing Using Virtual Subsections

Tom Peham[1], Judith Przigoda[2],
Nils Przigoda[2], and Robert Wille[1,3]

[1]Technical University of Munich, Germany
[2]Siemens Mobility GmbH, Braunschweig, Germany
[3]Software Competence Center Hagenberg, Austria

https://www.cda.cit.tum.de/

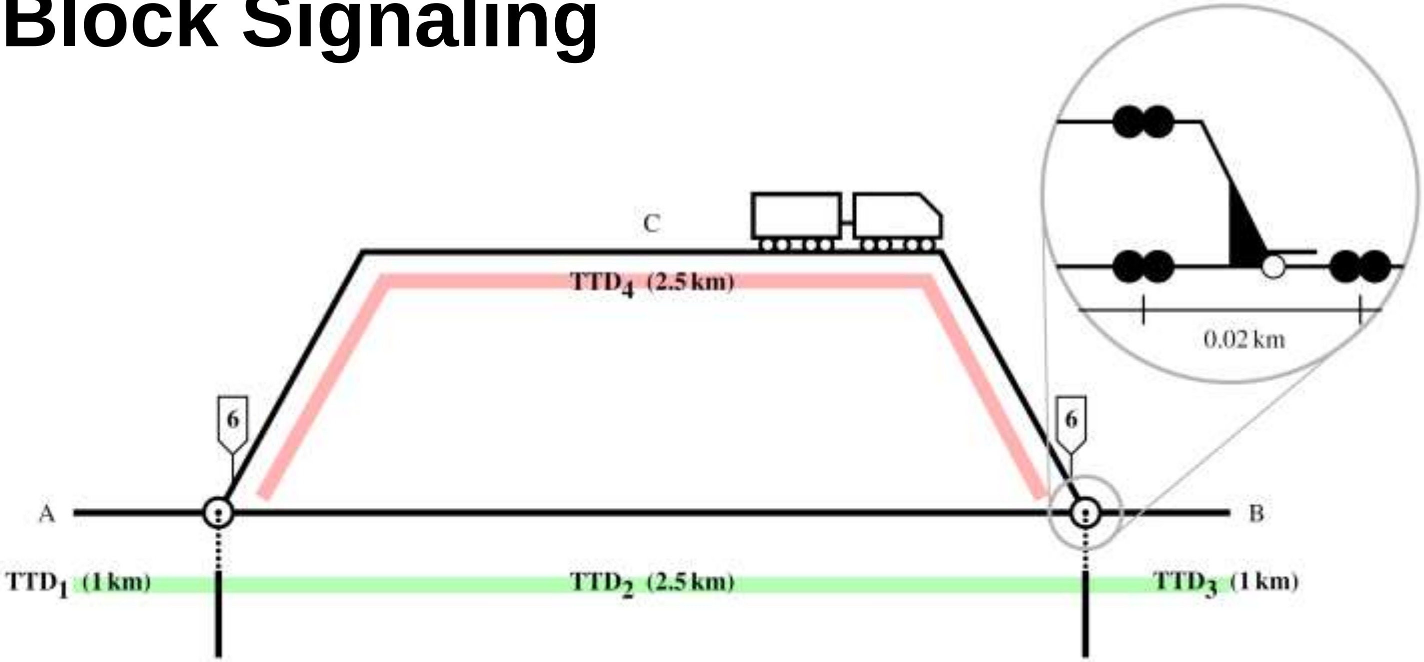
RAILWAY SIGNALING



- Prevent trains from running into each other
- Check, whether schedule is realistic/possible
- Fundamental principle today: **Block signaling**

Train	Start	Goal	Speed[km/h]	Length[m]	Departure Time	Arrival Time
1	A	B	180	400	0:00	0:04:30
2	B	A	120	700	0:00	0:04:00
3	A	C	120	100	0:01	0:03:00
4	B	A	180	250	0:01	0:05:00

Block Signaling

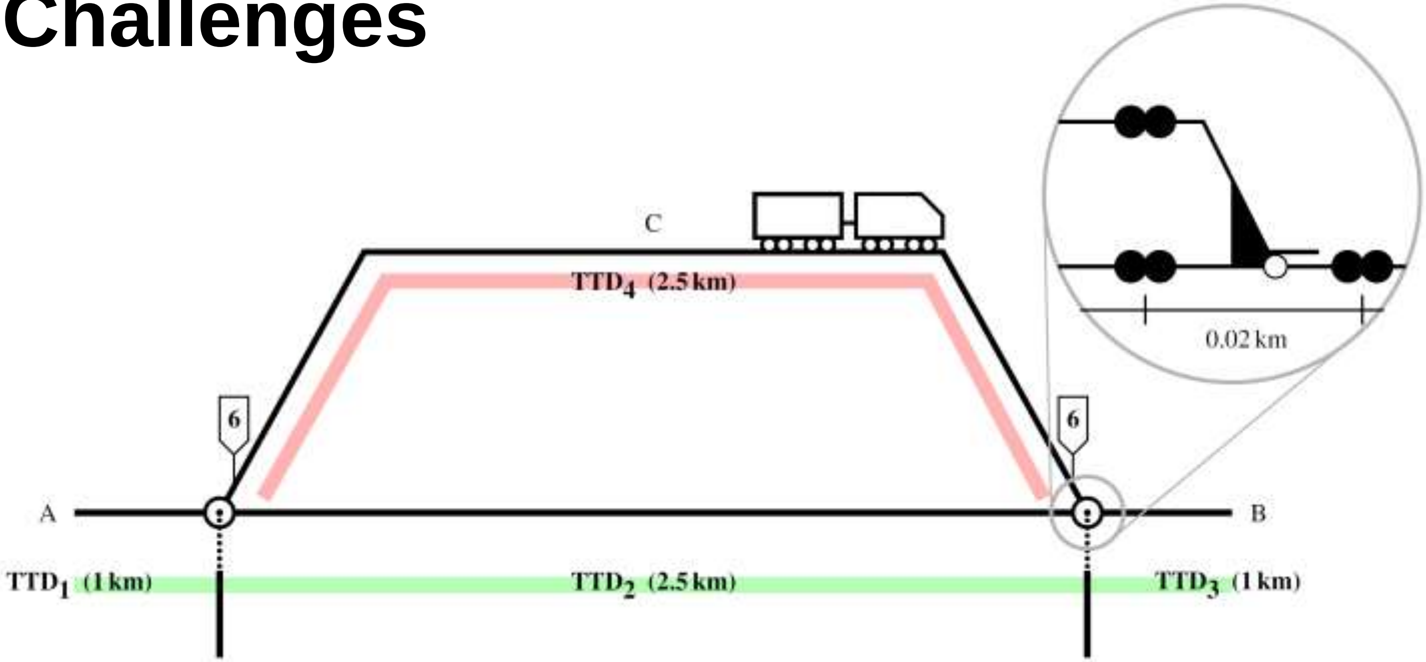


- Railway network divided into blocks
- At most one train is allowed to occupy a block at any given time

- Requires a **Trackside Train Detection System (TTD)**, e.g., axle counters

Train	Start	Goal	Speed[km/h]	Length[m]	Departure Time	Arrival Time
1	A	B	180	400	0:00	0:04:30
2	B	A	120	700	0:00	0:04:00
3	A	C	120	100	0:01	0:03:00
4	B	A	180	250	0:01	0:05:00

Challenges



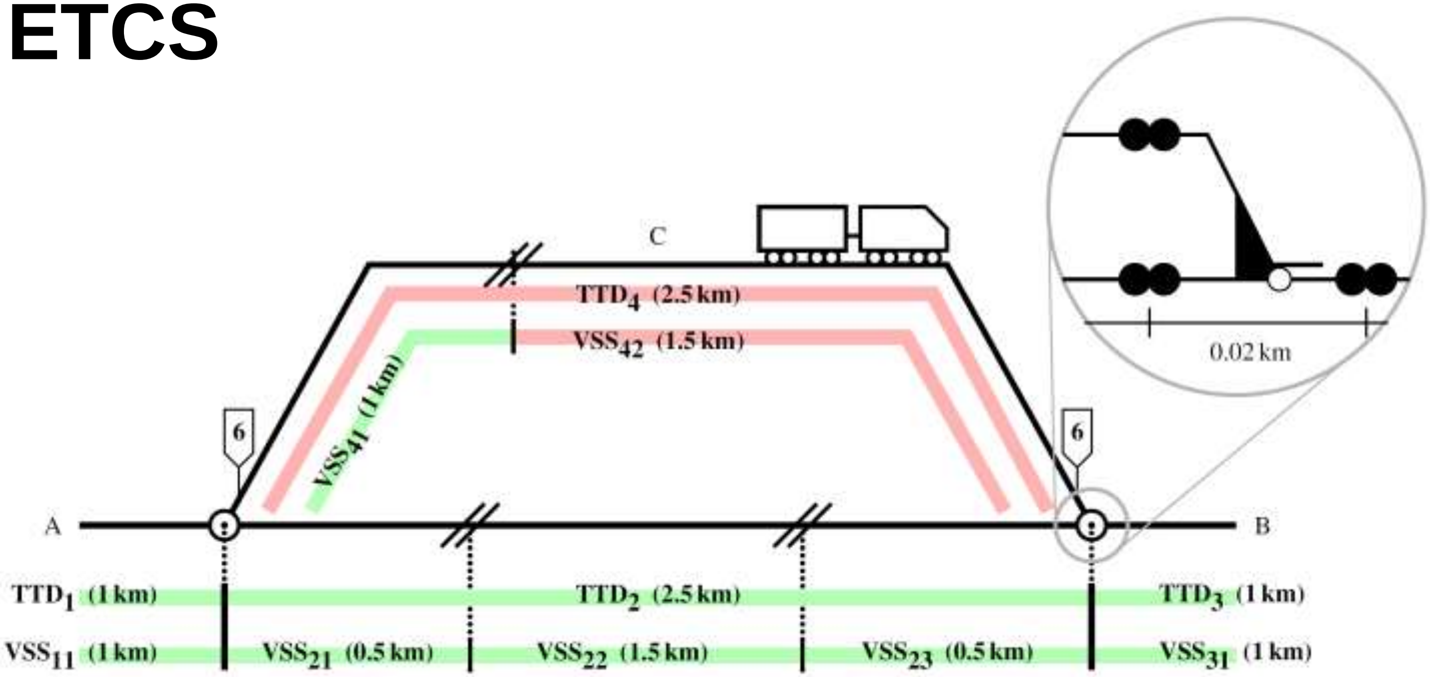
- TDDs usually defined by trade-offs
- Lengths varies between some meters and several kilometers

➔ **Affecting efficiency of the network**

➔ **Rather static**

Train	Start	Goal	Speed[km/h]	Length[m]	Departure Time	Arrival Time
1	A	B	180	400	0:00	0:04:30
2	B	A	120	700	0:00	0:04:00
3	A	C	120	100	0:01	0:03:00
4	B	A	180	250	0:01	0:05:00

ETCS



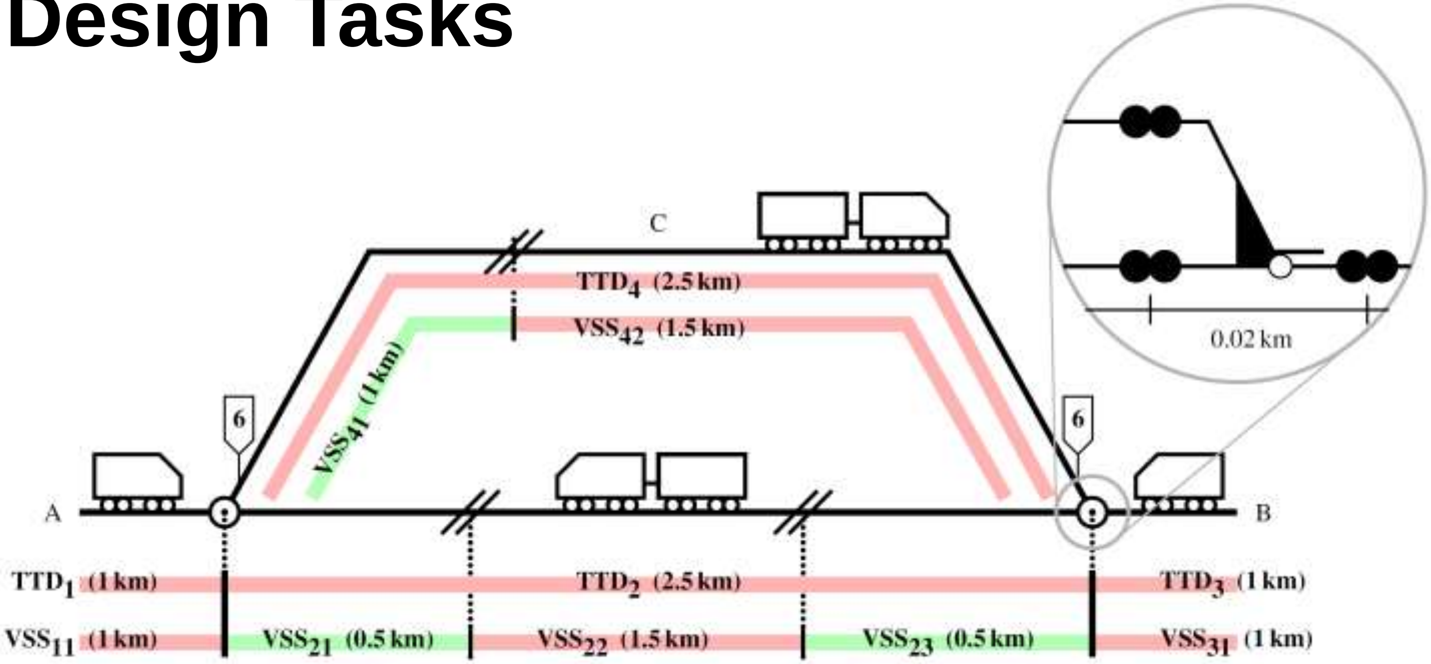
- European Train Control System (here: Hybrid Level 3)
- Allows **Virtual Subsections (VSSs)**

➔ **Do not require physical axle counters anymore**

➔ **Allow for a higher degree of freedom**

Train	Start	Goal	Speed[km/h]	Length[m]	Departure Time	Arrival Time
1	A	B	180	400	0:00	0:04:30
2	B	A	120	700	0:00	0:04:00
3	A	C	120	100	0:01	0:03:00
4	B	A	180	250	0:01	0:05:00

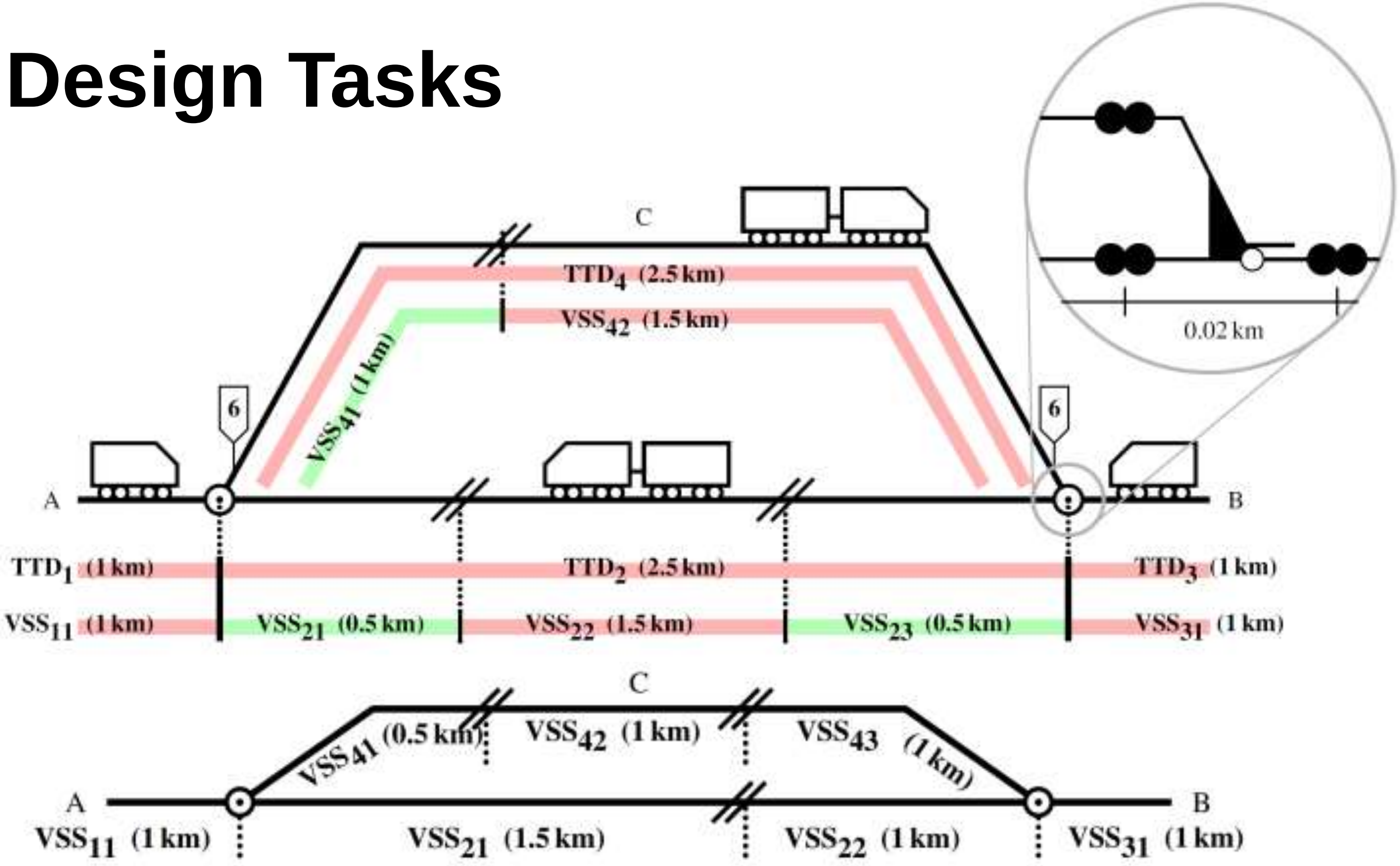
Design Tasks



- Given:
  - Layout
  - Desired Schedule
- Verification of Train Schedules on ETCS Layouts
- Generation of VSS Layouts
- Schedule Optimization Using the Potential of VSS

Train	Start	Goal	Speed[km/h]	Length[m]	Departure Time	Arrival Time
1	A	B	180	400	0:00	0:04:30
2	B	A	120	700	0:00	0:04:00
3	A	C	120	100	0:01	0:03:00
4	B	A	180	250	0:01	0:05:00

Design Tasks



- Given:
  - Layout
  - Desired Schedule
- Verification of Train Schedules on ETCS Layouts
- Generation of VSS Layouts
- Schedule Optimization Using the Potential of VSS
- **But**
  - **Highly non-trivial tasks**
  - **Thus far, mainly rely on manual labor**

Train	Start	Goal	Speed[km/h]	Length[m]	Departure Time	Arrival Time
1	A	B	180	400	0:00	0:03:30
2	B	A	120	700	0:00	0:02:30
3	A	C	120	100	0:01	0:02:30
4	B	A	180	250	0:01	0:03:30

Generation of VSS Layouts – Previous Approach

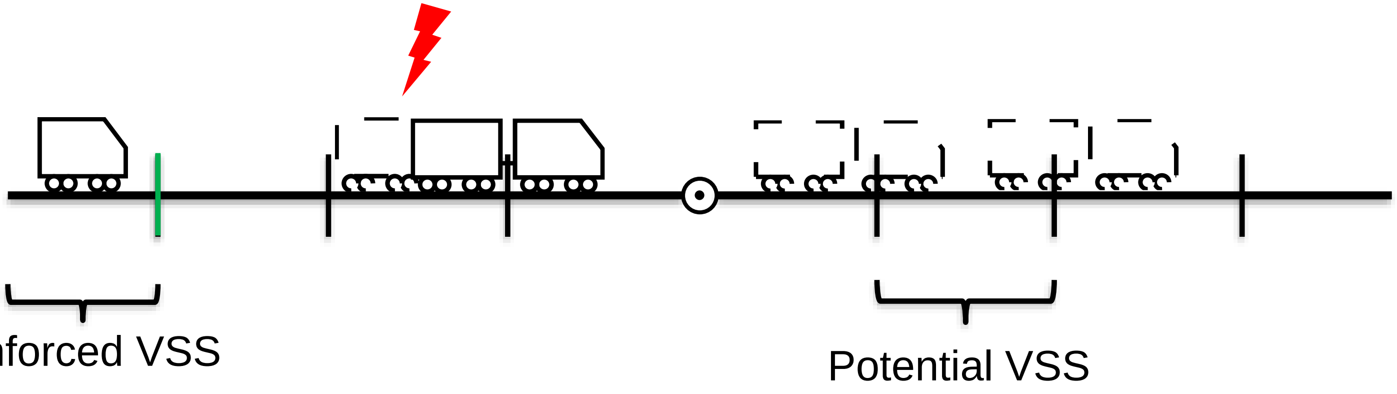
Encode problem as a Boolean formula

Placement:

$$\bigvee_{c \in chains(l^*_{tr})} \left(\bigwedge_{e \in c} occupies^{t_i}_{tr,e} \bigwedge_{f \in E \setminus c} \neg occupies^{t_i}_{tr,f} \right)$$

Movement:

$$occupies^{t_i}_{tr,e} \implies \bigvee_{f \in reachable(e,tr)} occupies^{t_{i+1}}_{tr,f}$$



Block Signaling Constraint:

$$\left(occupies^{t_i}_{tr_1,e} \wedge occupies^{t_i}_{tr_2,f}\right) \implies \bigvee_{v \in \mathrm{between}(e,f)} border_v$$

Collision Constraint

$$occupies^{t_i}_{tr_1,e} \wedge occupies^{t_{i+1}}_{tr_1,f} \implies \bigwedge_{g \in \mathrm{paths}(e,f,tr_1)} \neg occupies^{t_i}_{tr_2,g} \wedge \neg occupies^{t_{i+1}}_{tr_2,g}$$

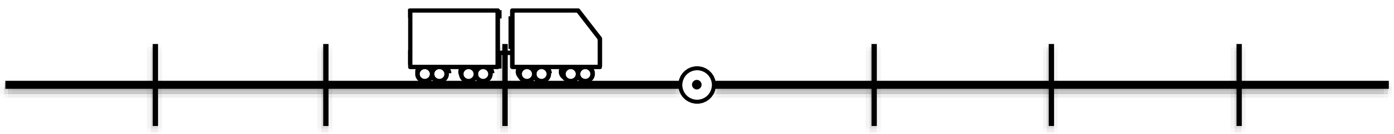
SAT Solver tries to find optimal satisfying assignment

Issues with Discretization – Infeasible Configurations

- Bad choice of discretization of space in relation to discretization of time:

Timestep: $10\ s$
Segment length: $1000\ m$

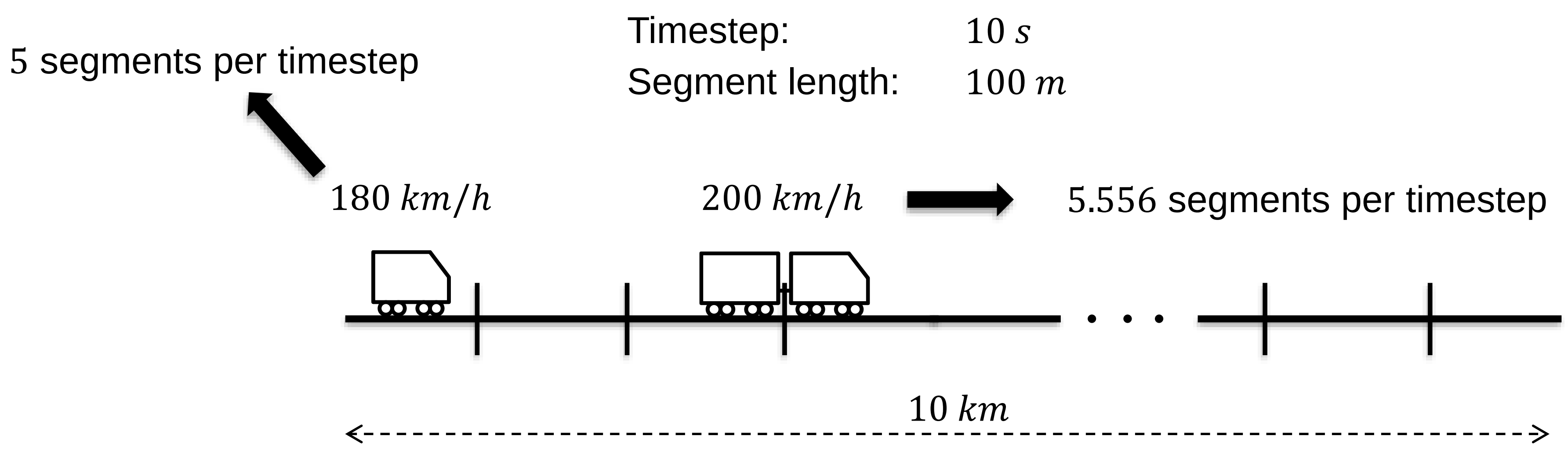
$200\ km/h$ ➡ 0.556 segments per timestep



- Rounding down leads to speed of 0!
- Rounding up (to 1) corresponds to a real-world speed of $360\ km/h$

Issues with Discretization – Rounding Errors

- Optimal solutions in discrete space are not actually optimal solutions

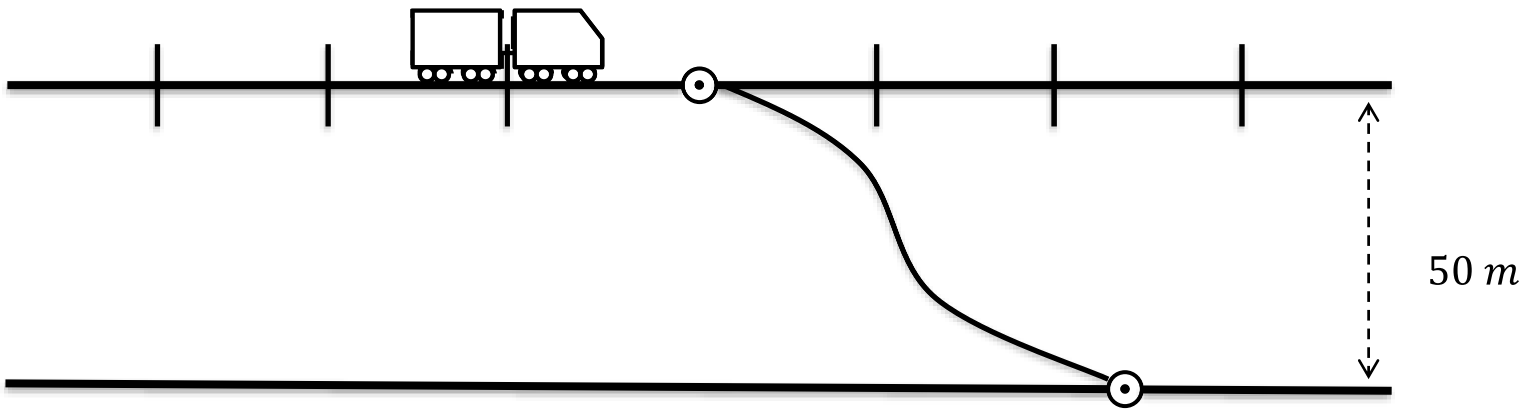


- Optimal solution in real-world: 18 time steps
- Optimal solution in discrete space: 20 time steps
  - + suboptimal placement of VSS

Issues with Discretization - Oversimplifications

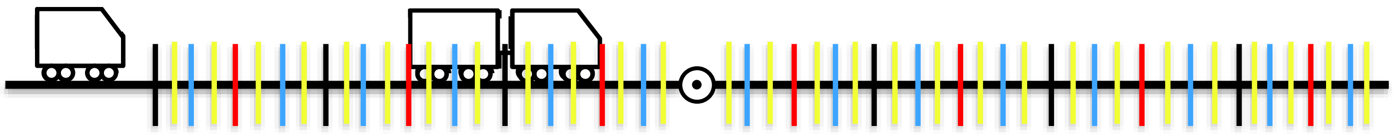
- Parts of a railway network cannot be modelled with a coarse spatial resolution

Timestep: $10\ s$
Segment length: $100\ m$



Solution 1 – Smaller Segments

- Avoid stated issues by employing a fine resolution



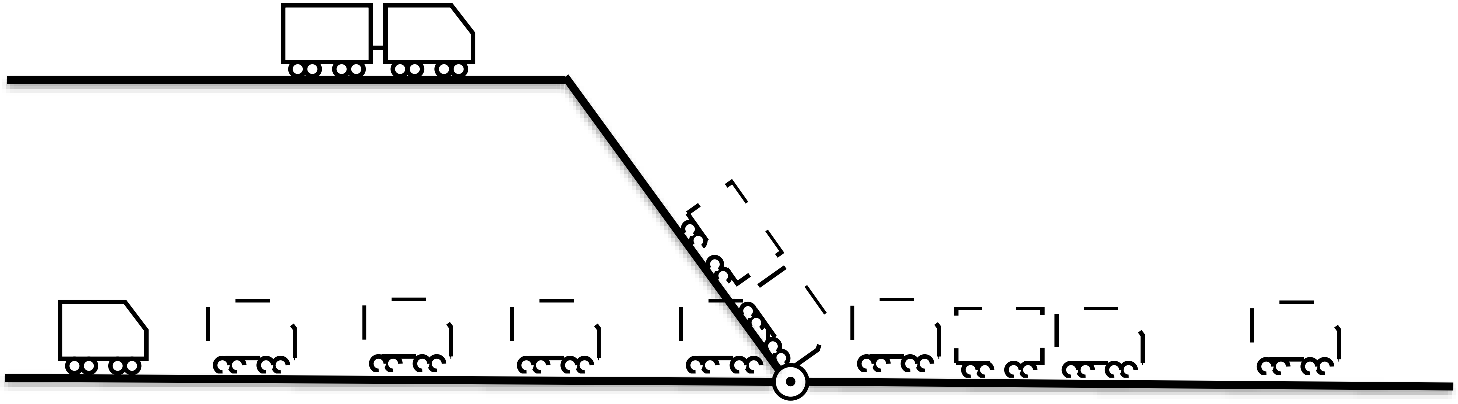
- The number of constraints grows with $O(n^3)$ in the number of segments
  - Even for trivial examples this leads to enormous SAT formulations

- Requires experimentation or expertise
  - Not completely automatic

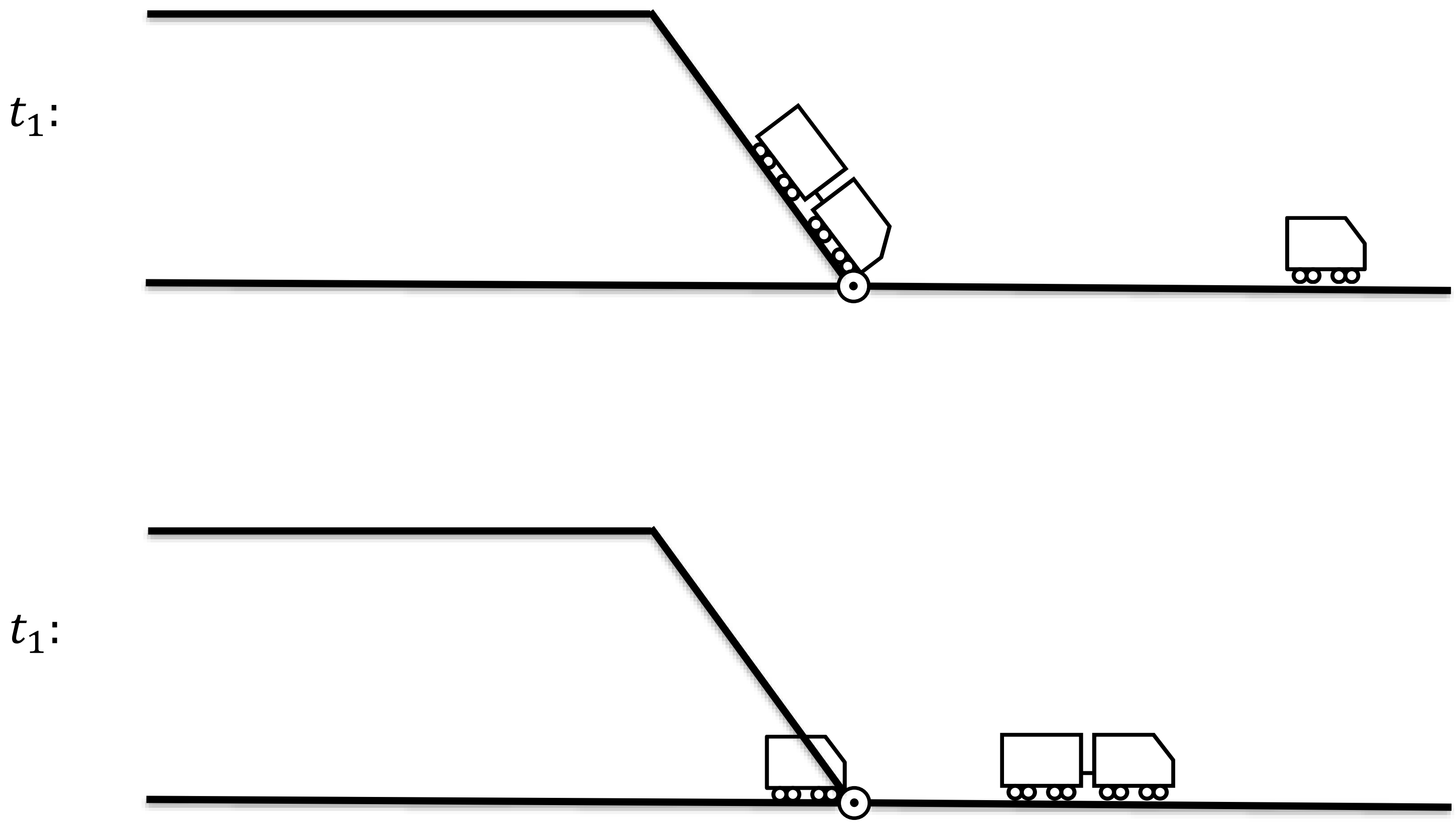
Solution 2 - Novel A*-Search Approach

- State s: Position of all trains on the network and incomplete VSS layout

- Cost $g(s)$: Time elapsed to reach state s
- Heuristic $h(s)$: Time until all trains reach their goal from s if no collisions occurred
- Estimate of true cost: $f(s) = g(s) + h(s)$
- Next states: All possible positions reachable by trains within one timestep
- How to handle this large state space?
  - Ignore irrelevant states

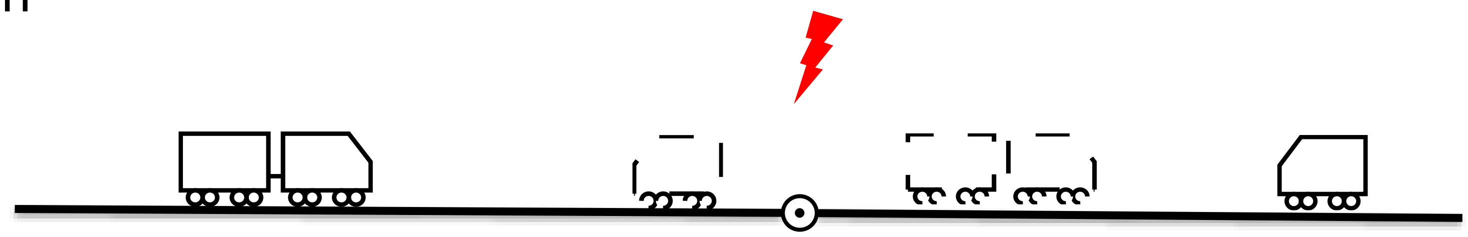


Computing Next States

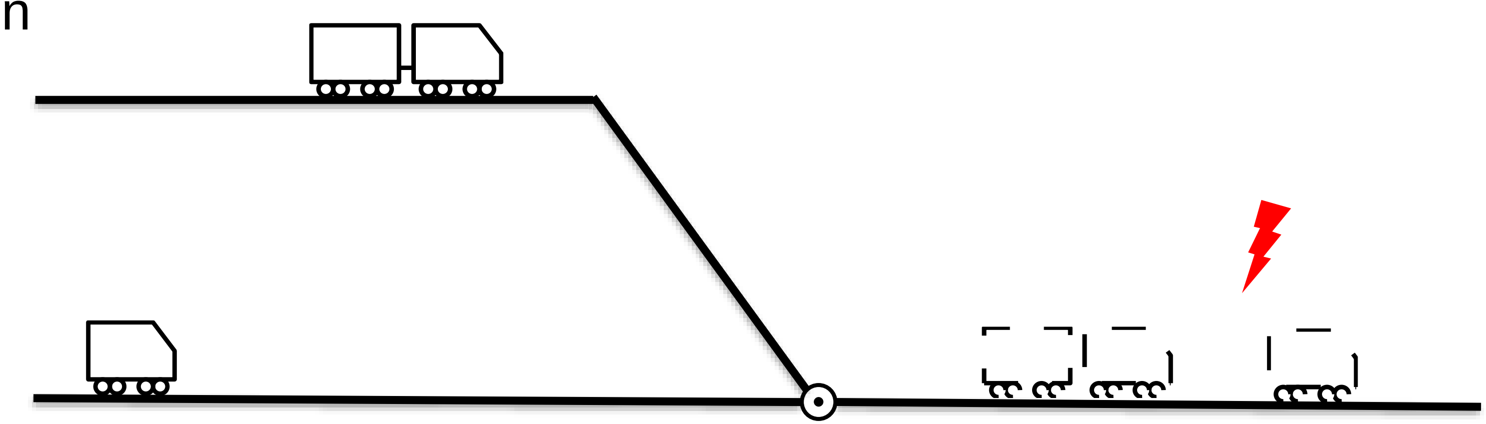


Handling Collisions

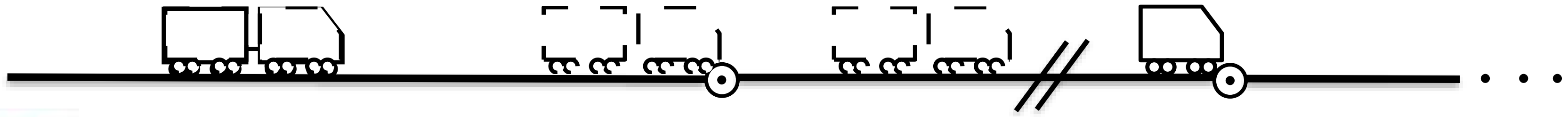
- Head-on Collision
  - Invalid State



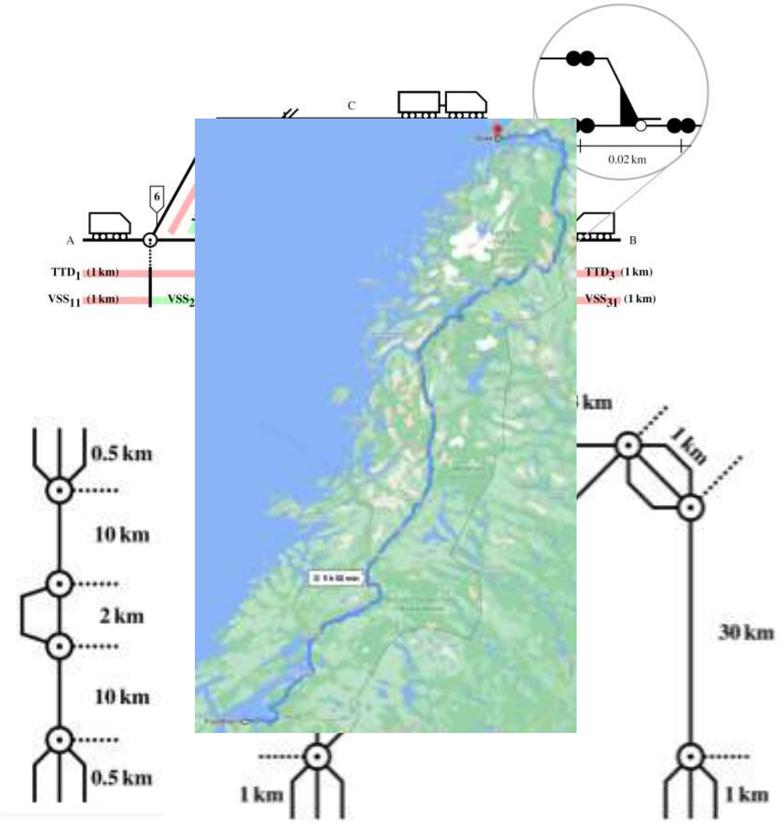
- Overlap Collision
  - Invalid State



- Rear-end Collision
  - Resolved by new VSS

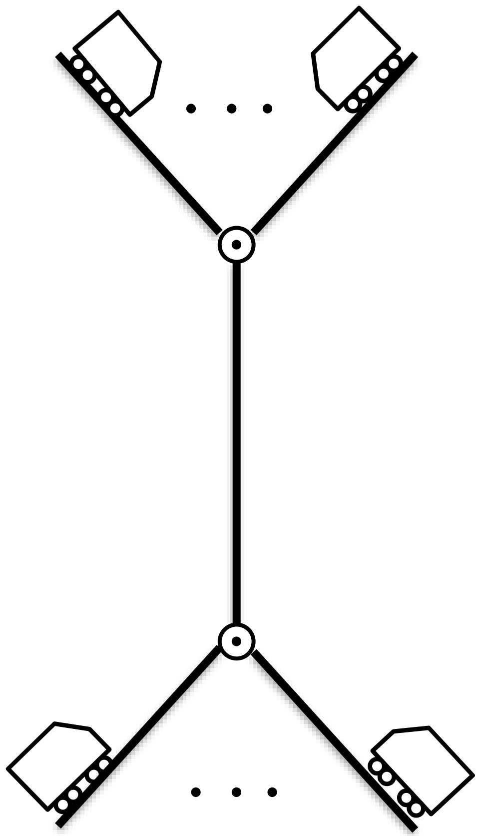


Application and Case Studies



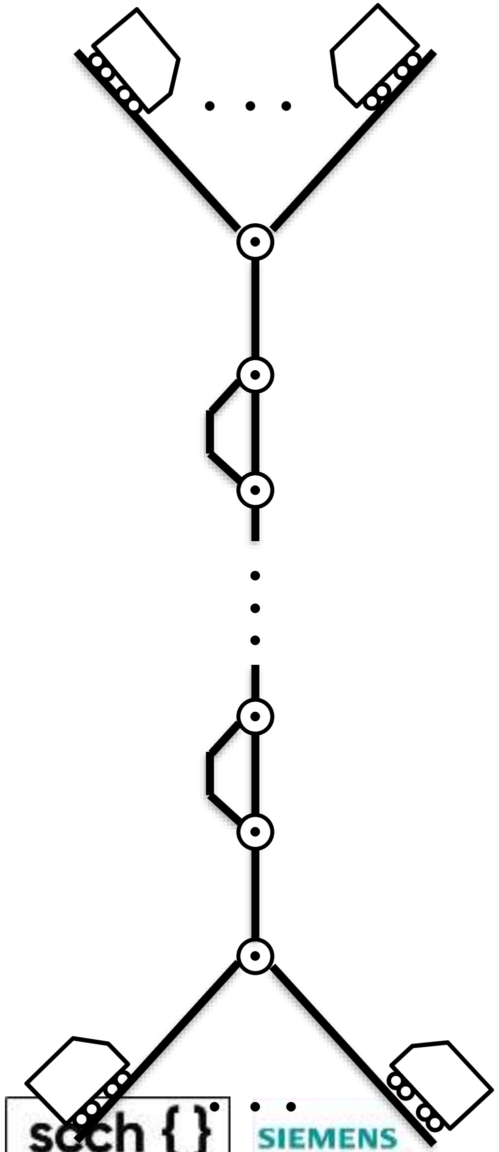
Method	Configuration r_s [m]	t_{max}	TTD/VSS	Time Steps	$\sum t$	Runtime [s]
Running Example (with 4 trains an total travel length of 7 km)						
SAT (discretized)	500	11	5	7	23	0.1
A* Search	-		9	7	21	< 0.1
Simple Example (with 4 trains and total travel length of 27 km)						
SAT (discretized)	500	20	14	15	53	29.2
A* Search	-		26	15	50	< 0.1
Complex Example (with 6 trains and total travel length of 148 km)						
SAT (discretized)	1000	18	25	16	71	124.9
A* Search	-		42	14	58	138.3
Nordlandsbanen (with 3 trains and total travel length of 819.6 km)						
SAT (discretized)	1000	140	-	-	-	> 3600
A* Search	-		519	135	286	45.713

Application and Case Studies



Method	Configuration r_s [m]	Configuration t_{max}	TTD/VSS	Time Steps	$\sum t$	Runtime [s]
Bottleneck (with 4 trains and total travel length of 10 km)						
SAT (discretized)	1000	20	13	18	60	0.6
	500	20	13	18	60	2.3
	100	20	16	15	54	84.9
	50	20	16	15	54	777.9
	50	15	16	15	54	866.5
A* Search	-		39	15	50	< 0.1
Bottleneck (with 10 trains and total travel lengh of 2.6 km)						
SAT (discretized)	1000	20	*Unsatisfiable*			1185.9
	1000	30	-	-	-	> 3600
	100	15	-	-	-	> 3600
A* Search	-		30	12	65	11.1
Bottleneck (with 12 trains and total travel length of 3 km)						
SAT (discretized)	1000	20	*Unsatisfiable*			1275.1
A* Search	-		34	15	92	371.0

Application and Case Studies

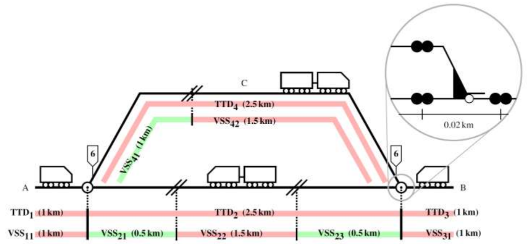


Method	Configuration r_s [m]	t_{max}	TTD/VSS	Time Steps	$\sum t$	Runtime [s]
Bidirectional (with 6 trains and total travel length of 14.6 km)						
SAT (discretized)	1000	30	16	30	124	50.6
	500	30	18	21	112	698.2
	100	30	-	-	-	$>$ 3600
	100	23	-	-	-	$>$ 3600
A* Search	-		53	22	105	1.6
Train Station (with 6 trains and total travel length of 7.1 km)						
SAT (discretized)	1000	30	19	9	39	1.1
	500	30	19	9	39	1.1
	100	30	31	21	114	64.1
	50	30	31	22	117	1381.3
A* Search	-		58	22	110	17.7
Train Station (with 8 trains and total travel length of 7.3 km)						
SAT (discretized)	1000	30	21	11	59	9.6
	500	30	21	11	59	9.6
	100	30	-	-	-	$>$ 3600
	100	23	33	23	159	564.1
A* Search	-		*Out of Memory*			-

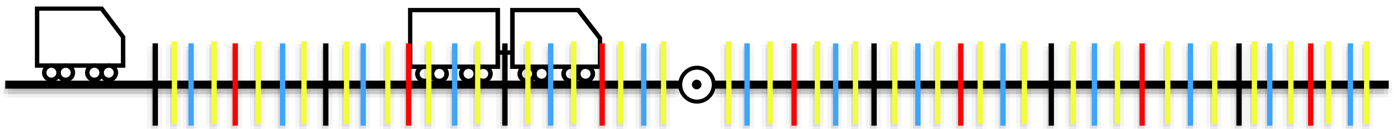
Conclusions

- Exploiting the potential and degree of freedom offered by ETCS Level 3

Train	Start	Goal	Speed[km/h]	Length[m]	Departure Time	Arrival Time
1	A	B	180	400	0:00	0:04:30
2	B	A	120	700	0:00	0:04:00
3	A	C	120	100	0:01	0:03:00
4	B	A	180	250	0:01	0:05:00



- Initial solution (utilizing satisfiability solvers) by discretizing time and space
  - Issues with discretization



- Novel A* Search solution
  - No discretization needed → VSS created on the fly
  - Scales much better

- Future Work: Scale to more realistic scenarios with heuristic solution
- **More at https://www.cda.cit.tum.de/research/etcs/**

TUM scch { } software competence center hagenberg SIEMENS Ingenuity for life