Automated Drawing of Railway Schematics using Numerical Optimization in SAT

Bjørnar Luteberget, Koen Claessen, Christian Johansen

4 Dec 2019 – iFM'19

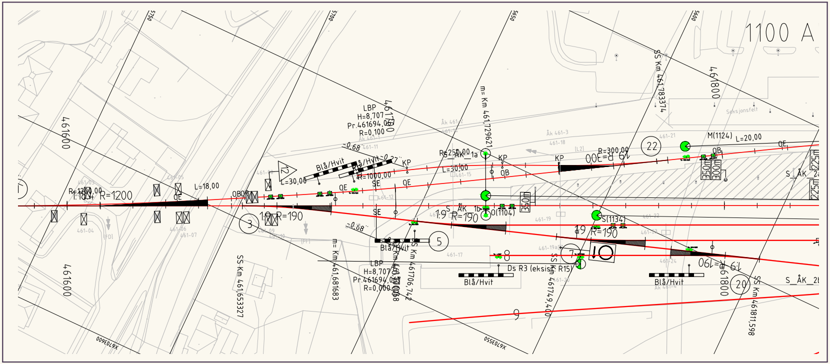
UiO : University of Oslo

RailCOMPLETE

CHALMERS | GÖTEBORGS UNIVERSITET

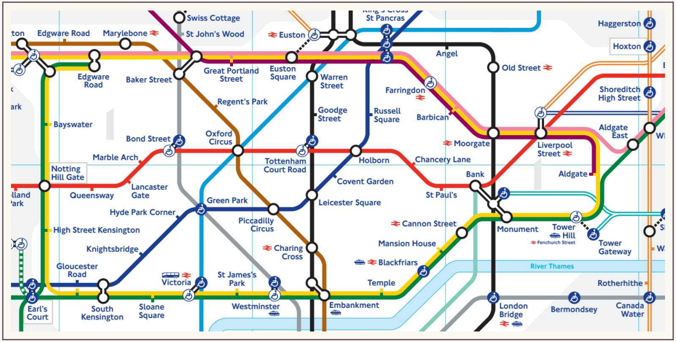
Schematic drawings

- Railway construction projects produce highly detailed blueprints.



Abstraction

A well-known railway schematic: passenger's metro maps.



Removing and compressing geographical information better conveys:

- topological structure (finding transfers), and
- sequential information along lines (finding your stop).

Bergen's topology:

Schematic drawings for various purposes

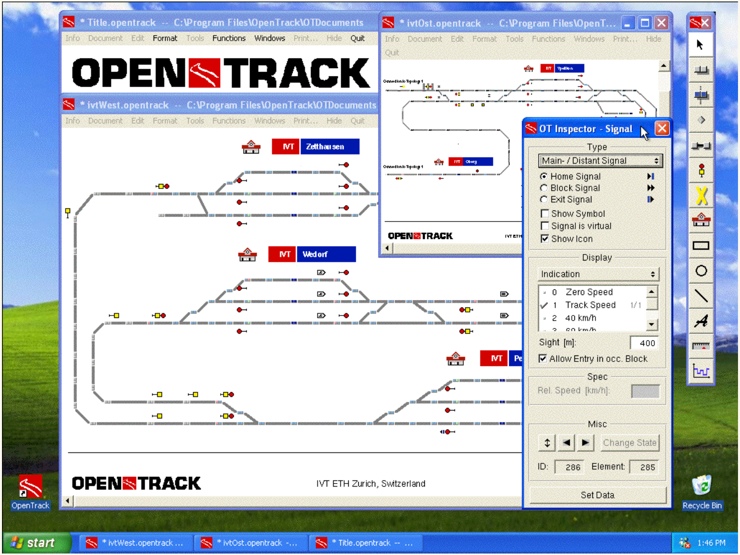
Back to railway engineering.

When a geographical/topological model is available, the following derived views are useful:

- Timetabling software
- Construction details
- Interlocking specifications
- Dispatcher’s screens

Use cases

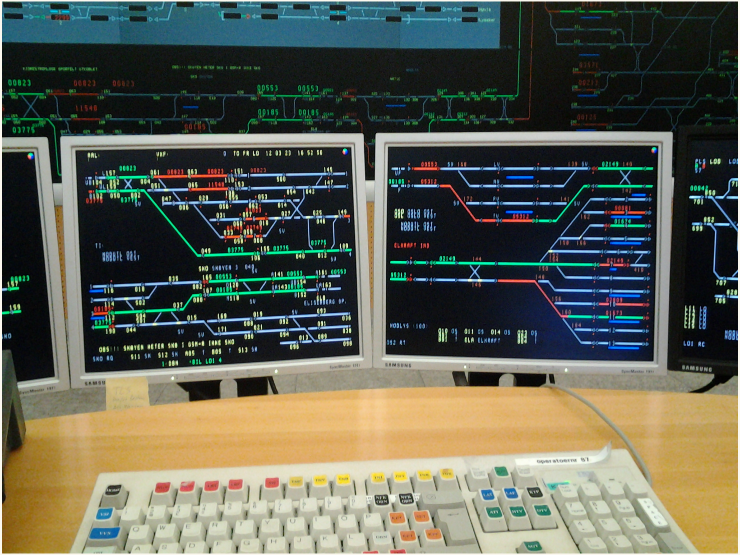
Timetable analysis and operations simulation



Source: OpenTrack

Use cases

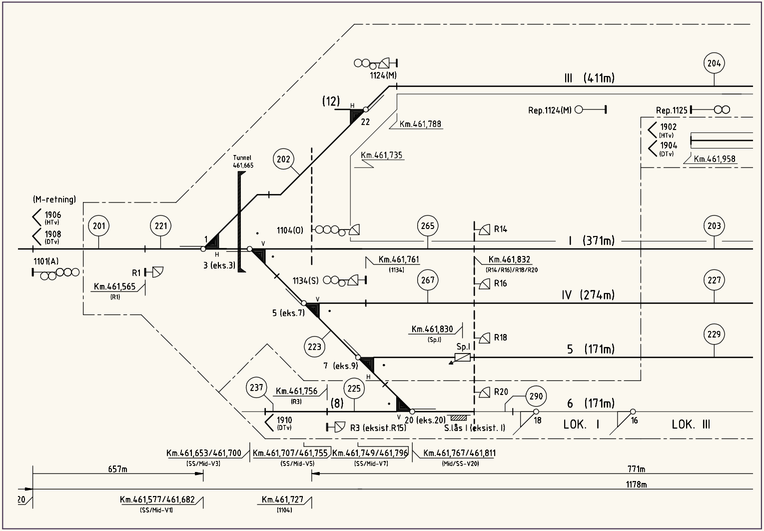
Dispatcher workstations



Source: Bane NOR control central

Use cases

Signalling/interlocking construction documentation



Source: Norconsult

Problem specification

- develop methods for producing schematic track plans ...
- ... suitable for infrastructure within a single corridor,
- each point can be mapped onto a linear axis.
- used in construction projects to reason about network topology and travelling lengths
- highways, railways, public transport systems.

Not suited for

- highly connected networks
- with many cycles

Why automate?

- During construction project planning, infrastructure is dynamic.
- Efficient drawing and updating of drawings helps coordinate documentation across 3D, 2D and schematic views.
- Low-effort transfer from geographical CAD drawings to other software requiring schematic presentation can allow:
  - Capacity analysis.
  - Dispatcher workstation simulation.
  - Visualization of interlocking behavior.

Related work

- Iterative and force-directed algorithms for gradually transforming a geographical network map.
  - S. Avelar. *Schematic Maps on Demand - Design, Modeling and Visualization*. PhD thesis, ETH Zürich, 2002
  - S. Cabello, M. de Berg, and M. J. van Kreveld. *Schematization of networks*. Comput. Geom., 30(3):223–228, 2005.
- Mixed-integer programming for finding exactly grid-structured and rigidly optimized solutions.
  - M. Nöllenburg and A. Wolff. *Drawing and labeling high-quality metro maps by mixed-integer programming*. IEEE Tr.Vis.Comput.Graph., 17(5):626–641, 2011.
  - O. Oke and S. Siddiqui. *Efficient automated schematic map drawing using multi- objective mixed integer programming*. Computers & OR, 61:1–17, 2015.
- Story-line algorithms.
  - van Dijk T.C., Lipp F., Markfelder P., Wolff A. (2018) *Computing Storyline Visualizations with Few Block Crossings*. Graph Drawing and Network Visualization. GD 2017, Springer.

Talk overview

1. **Formalize problem description for linear-schematic railway track plan.**
   - Infrastructure input structure
   - Drawing output structure
   - Drawing constraints
   - Drawing optimization criteria
2. Compare encoding approaches.
   - Naive SAT encoding
   - Cross-section-based SAT encoding
   - Levels-based SAT encoding modulo difference constraints

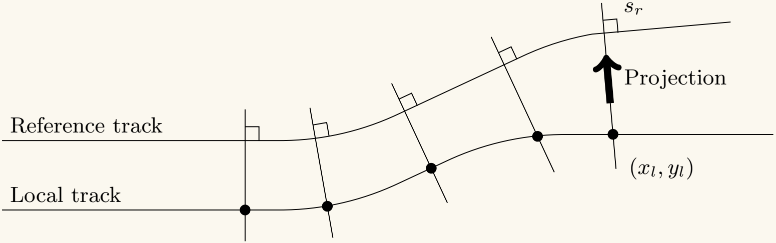
Preliminary definitions

Linear positioning system:

- Within a railway corridor, engineering convention assigns a scalar value to each point in the plane (*"mileage"*).

$$\text{linPos} : \mathbb{R}^2 \to \mathbb{R}$$

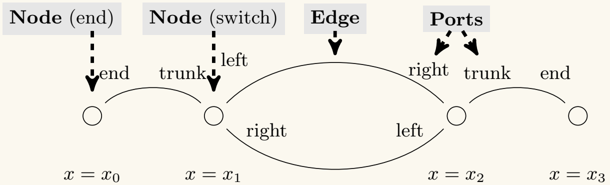
- Measures approximate travelling distance along the corridor.
- Usually implemented by mapping points to the arc-length parametrization of a curve in the corridor.



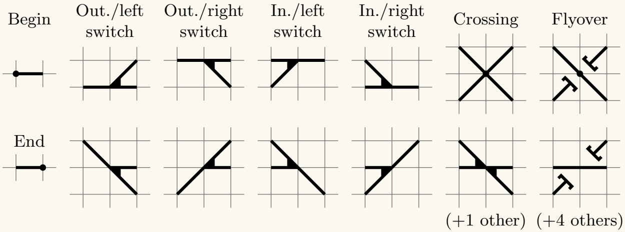
Preliminary definitions

Track network representation as a graph-like structure:

- Each node has linear position x.
- Each node has ports, and edges connect two ports.



- Node have a node class determining drawing and ports.



Problem definition

Linear schematic track drawing algorithm:

$$d : (N, E) \to L$$

- Nodes $N = \{n_i = (c_i, s_i)\}$, where $c_i \in C$ is a node class, and $s_i \in \mathbb{R}$.
- Edges $E = \{e_j = (n_a, p_a, n_b, p_b)\}$, where $n_a, n_b \in N$ are two nodes where $s_a < s_b$ and p_a, p_b are distinct, available ports.
- Lines $L = \{(e_j, l_j)\}$, where l_j is a polyline, representing the drawing of edge $e_j \in E$ by a sequence of points in $\mathbb{R}^2$,

$$\langle (x_1^j, y_1^j), (x_2^j, y_2^j), \ldots, (x_n^j, y_n^j) \rangle$$

Constraints (1/5): Legible nodes and edges

The most obvious drawing constraints:

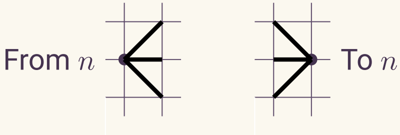
- The points at the end of the edges in L must uniquely identify the nodes.
- Edges must not cross or overlap.

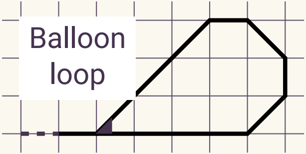
Constraints (2/5): Octilinearity

Octilinearity:
lines are horizontal, or diagonal at 45°.

This property contributes to the neat look of a schematic drawing, giving a clue that the drawing is not fully geometrically accurate.

When loops are present in the infrastructure, vertical lines may also be allowed, such as in the *balloon loop* used on many tram lines.



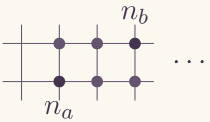


Constraints (3/5): Linear order

Linear order:
nodes are ordered left-to-right by linear position.

Gives a clear sense of sequence, useful for reasoning about train movements.

$s_a < s_b \Rightarrow$



$x_a \leq x_b$

Not a strict inequality in general, though along an edge

$$x_a + d \leq x_b,$$

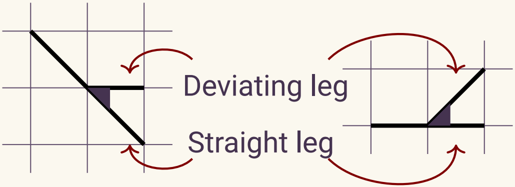
for some minimum distance d.

Constraints (4/5): Node shapes

Node shapes:
switches split the track into left and right leg.

Left and right are preserved so that the layout can be traced back to the geography.

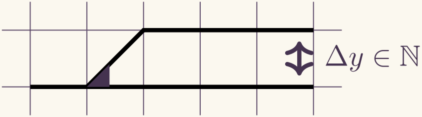
Since one of the legs of the switch is typically straight and the other is curved, it is also desirable to preserve the straight leg's direction relative to the trunk.



Constraints (5/5): Uniform vertical spacing

Uniform vertical spacing:
parallel tracks are typically required to be drawn at a specific distance from each other

x coordinates have no such restriction (but are often integer-valued to fulfill the octilinearity constraint)



Optimization criteria

To produce high quality drawings, we experimented with various optimization criteria.

The most important ones seem to be:

- Height / width of the drawing
- Number of bends

Other relevant criteria:

- Length of diagonal lines
- Non-default node shapes
- Number of edges that are only 1 unit long
- Vertical distances locally between adjacent edges
- Horizontal distances locally between closest nodes

Different criteria may be prioritized in different use cases.

Talk overview

1. Formalize problem description for linear-schematic railway track plan.
   - Infrastructure input structure
   - Drawing output structure
   - Drawing constraints
   - Drawing optimization criteria
2. **Compare encoding approaches.**
   - Naive SAT encoding
   - Cross-section-based SAT encoding
   - Levels-based SAT encoding modulo difference constraints

Naive SAT encoding

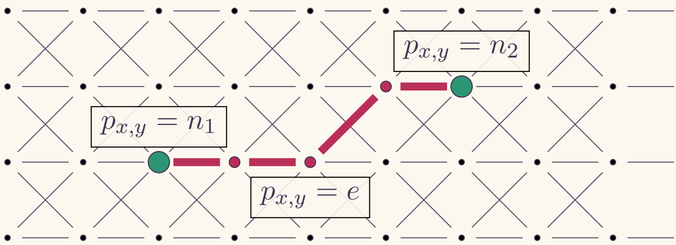
Problem **representation**:

- Consider a grid of width w and height h
- Each grid point (x, y) is either unused, represents a specific node, or an edge on the way from one node to another.

$$\forall x, y, \quad p_{x,y} : \{\text{unused}\} \cup \text{Node} \cup \text{Edge}$$

- Between grid points, horizontal or diagonal line segments

$$\forall x, y, \quad e_{x,y,\rightarrow}, \quad e_{x,y,\nearrow}, \quad e_{x,y,\searrow} \quad : \mathbb{B}$$



Constraints:

- (1): Legible nodes.

$$\forall n \in N : \text{exactlyOne}(\{p_{x,y} = n \mid (x,y) \in [0,w] \times [0,h]\})$$

- (1): Legible edges.
  - No unused edges:

$$e_{x,y,\rightarrow} \Rightarrow \neg(p_{x,y} = \text{unused}) \wedge \neg(p_{x+1,y} = \text{unused})$$

  - No crossing edges:

$$\neg e_{x,y,\searrow} \vee \neg e_{x,y+1,\nearrow}$$

  - Consistent edges:

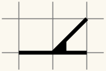
$$(p_{x,y} = n_1) \wedge e_{x,y,\rightarrow} \Rightarrow (p_{x+1,y} = n_2) \vee p_{x+1,y} = e$$

- (2): Octilinearity is implicit.
- (3): Node ordering:

$$p_{x_1,y} = n_i \Rightarrow \bigvee_{x_2 \geq x_1} p_{x_2,y} = n_{i+1}$$

- (4) Node shapes: with n_1 switch with right leg edge e to n_2:

$$\Bigg(p_{x,y} =$$



$$\Bigg) \Rightarrow e_{x,y,\rightarrow} \wedge (p_{x+1,y} = e \vee p_{x+1,y} = n_1)$$

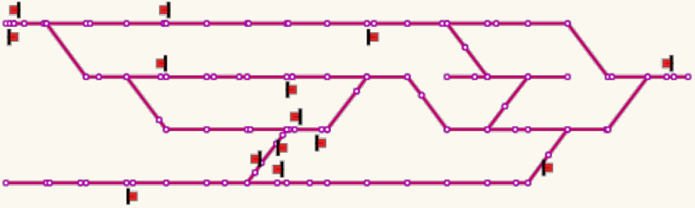
- (5) Uniform vertical spacing implicit in grid optimized for height.

Optimization using SAT

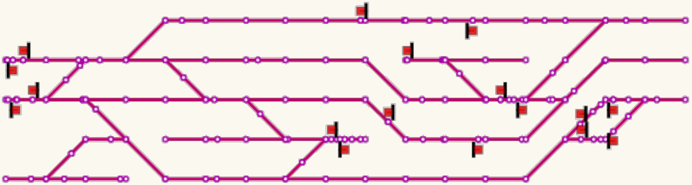
- Have a Boolean formula ϕ, including numbers represented as Booleans (e.g. binary or unary encoding).
- Want to optimize some number x.
- Solve the sequence of formulas
  $\phi \wedge (x < m_1)$,
  $\phi \wedge (x < m_2)$, $\ldots$,.
- Linear search: The sequence m_i goes down from some maximum value v: v, $v-1$, $v-2$, $v-3$, $\ldots$.
- Binary search: The sequence m_i in some interval $[0, v]$, bisect the interval until it contains a single number.

- Early experiments in making a visualization tool:

Eidsvoll:



Asker:



Naive encoding: Performance

However, performance was a problem.

Model	**Src.**	**Size**	**Direct/SAT**	
			hwb	size (v/c)
Eidsvoll	[19]	35	60.7	57k/153k
Arna	RC	57	294	167k/493k
Asker	[19]	64	T/O	104k/295k
Weert	[6]	102	T/O	304k/969k
5x10	T	228	T/O	2.8M/13M
5x20	T	478	T/O	2.8M/12M
10x5	T	203	T/O	3.0M/14M
20x5	T	403	T/O	3.0M/14M
10x10	T	453	T/O	2.6M/12M
15x15	T	1053	T/O	2.3M/10M

Possible improvements

Problems:

- The representation is large and two-dimensional.
- Cannot easily handle models with more than 50 nodes.

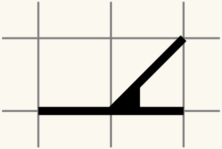
We can reduce the size of the representation by:

- Making better use of the horizontal order relation.
- Is there also an vertical ordering?

Making use of order relations

The vertical order of edges may be (mostly) derived from the topology.

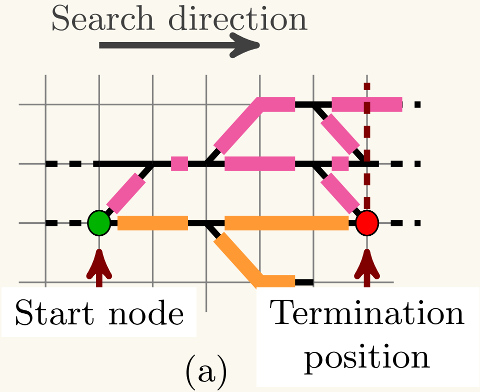
- Left is above right in outgoing switches



- Going in the facing direction of the switch, propagate the *above* and *below* nodes until any of the termination conditions happen:
  1. The above and below sets meet in a node.
  2. One of the sets has no more edges to follow.
- Union the results for all switches

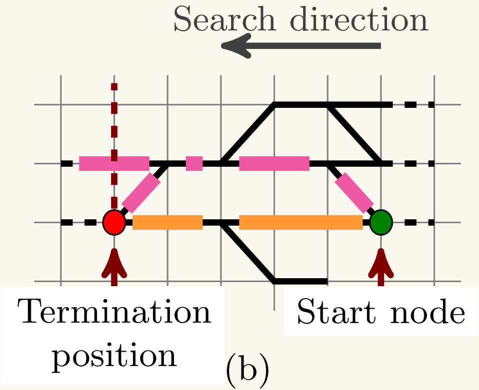
$$<_E = \bigcup_{n_i \in N} <_E^i$$

Vertical order relation



(a)

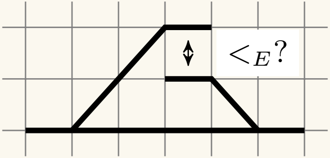
Vertical order relation



(b)

Vertical order relation

The vertical order relation is uniquely determined in many realistic cases, but may need additional user input in some cases.
The *clothes iron* example:

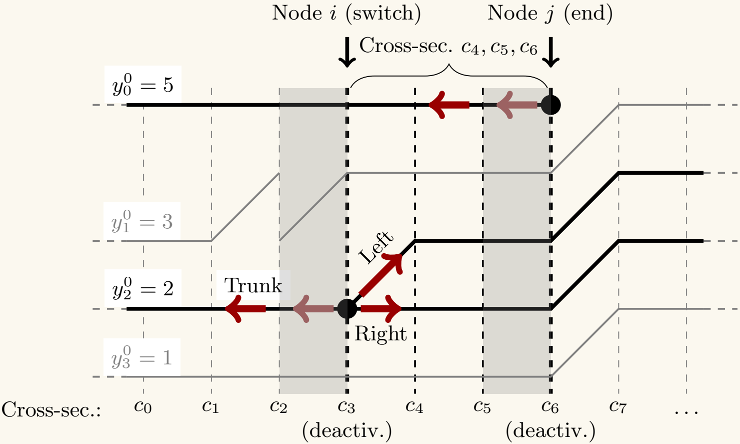


(c) Clothes iron

This may also be automated by using geometrical calculations.

Cross-section SAT encoding

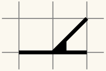
- Instead of representing all points on the grid, create vertical cross-sections of the drawing with unary-encoded integers for nodes' and edges' y values.



Representation:

- 3 cross-sections for consecutive nodes, activated $b_k \in \mathbb{B}$.
- A unary-encoded integer $y^k_{e_i} \in [0, Y]$ per edge active at cross-section k.
- Edge direction $d^k_{e_i} \in \{\text{Up}, \text{Straight}, \text{Down}\}$.
- The *ahead* variable $a^k_{e_i} \in \mathbb{B}$ indicates whether there is a propagated edge shape constraint.

Constraints:

- Node shape: for  :

$$\neg a^k_{e_i 1} \wedge d^k_{e_i 1} \neq \text{Up}$$

$$a^k_{e_{j1}} \wedge a^k_{e_{j2}}$$

$$\left(d^k_{e_i} = \text{Straight}\right) \Rightarrow \left(y^k_{e_i} = y^k_{e_{j2}} \wedge d^k_{e_{j2}} = \text{Straight} \wedge d^k_{e_{j1}} = \text{Left}\right)$$
$$\left(d^k_{e_i} = \text{Down}\right) \Rightarrow \left(y^k_{e_i} = y^k_{e_{j1}} \wedge d^k_{e_{j2}} = \text{Down} \wedge d^k_{e_{j1}} = \text{Straight}\right)$$

- Edge vertical order:

$$(e_i <_E e_j) \Rightarrow \bigwedge_{c_k} y^k_{e_i} \leq y^k_{e_j}$$

- Disabled cross-sections propagate all their values:

$$\neg b_k \Rightarrow \bigwedge_{e_i \in c_k} \left\{ y^k_{e_i} = y^{k+1}_{e_i} \wedge a^k_{e_i} = a^{k+1}_{e_i} \wedge d^k_{e_i} = d^{k+1}_{e_i} \right\}$$

- Enabled cross-sections require consistency between edge shapes and y values:

$$b_k \Rightarrow \bigwedge_{e_i \in c_k} \left\{ \left(\neg a^k_{e_i} \wedge d^{k+1}_{e_i} = \mathsf{Up} \right) \Rightarrow y^k_{e_i} + 1 = y^{k+1}_{e_i} \right\}$$

- Enabled cross-sections realize rightward-constrained *ahead* values a:

$$b_k \Rightarrow \bigwedge_{e_i \in c_k} \left\{ \left(a^k_{e_i} \Rightarrow y^k_{e_i} = y^{k+1}_{e_i} \right) \wedge \left(a^k_{e_i} \Rightarrow d^k_{e_i} = d^{k+1}_{e_i} \right) \wedge \neg a^{k+1}_{e_i} \right\}$$

Cross-section SAT encoding

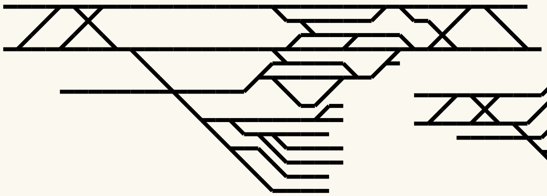
Advantages in the cross-section encoding:

- Node ordering is implicit in the encoding.
- Node coordinates are deltas (count of active cross-sections).
- Each cross-section has known edges.
- Edge locations are explicitly constrained.

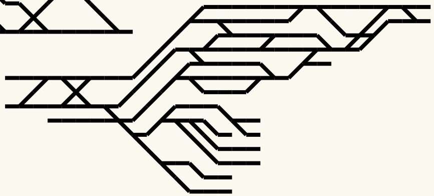
Cross-section encoding: Performance

Model	**Src.**	**Size**	**Direct/SAT**		**Cross-sec./SAT**			
			hwb	size (v/c)	hwb	hbw	bhw	size (v/c)
Eidsvoll	[19]	35	60.7	57k/153k	0.05	0.06	0.33	4.0k/28k
Arna	RC	57	294	167k/493k	0.26	0.65	1.06	11k/100k
Asker	[19]	64	T/O	104k/295k	0.61	1.02	0.87	14k/124k
Weert	[6]	102	T/O	304k/969k	0.72	19.3	21.4	29k/327k
5x10	T	228	T/O	2.8M/13M	5.83	7.48	8.08	46k/364k
5x20	T	478	T/O	2.8M/12M	279	299	T/O	265k/4.2M
10x5	T	203	T/O	3.0M/14M	0.52	0.59	1.08	20k/83k
20x5	T	403	T/O	3.0M/14M	1.95	2.50	3.36	44k/165k
10x10	T	453	T/O	2.6M/12M	21.9	22.4	40.7	96k/727k
15x15	T	1053	T/O	2.3M/10M	T/O	T/O	T/O	N/A

Comparisons: optimization criteria



Cross-sec./SAT, opt. height/bends

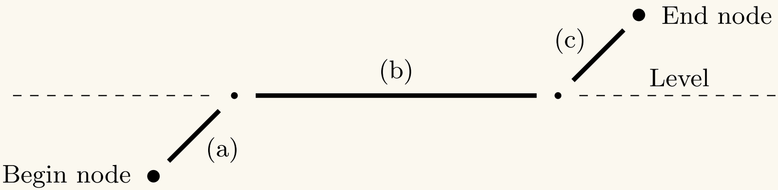


Cross-sec./SAT, opt. height/width

Levels-based SAT encoding

Another idea for simplification of the **representation**:

- Represent each edge by a single y coordinate, its *level*.



- The line is implicit from the nodes' coordinates and the edges y level.

Levels-based SAT encoding

- Also, represent node and edge y coordinates as unbounded integers and solve difference constraints on them, separately from the SAT problem (SAT modulo theories, SMT).

$$x - y \leq k$$

- We make each *level* distinct using $<_E$, but with an exception for 1-unit long edges ($q \in \mathbb{B}$):

$$e_a <_E e_b \Rightarrow \quad l_a \leq l_b, \quad (\neg q_a^{\text{up}} \wedge \neg q_b^{\text{down}} \wedge) \Rightarrow l_a + 1 \leq l_b$$

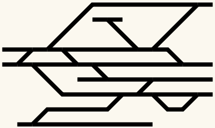
- Represent node x coordinates as deltas saturated at 2:

$$\Delta x_i \in \{0, 1, \geq 2\}$$

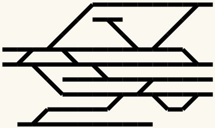
Levels encoding: Performance

Model	Src.	Size	Direct/SAT		Cross-sec./SAT				Levels/SAT	
			hwb	size (v/c)	hwb	hbw	bhw	size (v/c)	bhw	size (v/c)
Eidsvoll	[19]	35	60.7	57k/153k	0.05	0.06	0.33	4.0k/28k	0.02	2.3k/0.7k
Arna	RC	57	294	167k/493k	0.26	0.65	1.06	11k/100k	0.03	4.9k/1.3k
Asker	[19]	64	T/O	104k/295k	0.61	1.02	0.87	14k/124k	0.04	5.6k/2.0k
Weert	[6]	102	T/O	304k/969k	0.72	19.3	21.4	29k/327k	0.18	11k/4.0k
5x10	T	228	T/O	2.8M/13M	5.83	7.48	8.08	46k/364k	0.58	35k/2.7k
5x20	T	478	T/O	2.8M/12M	279	299	T/O	265k/4.2M	3.37	97k/7.7k
10x5	T	203	T/O	3.0M/14M	0.52	0.59	1.08	20k/83k	0.40	28k/2.0k
20x5	T	403	T/O	3.0M/14M	1.95	2.50	3.36	44k/165k	1.73	70k/4.0k
10x10	T	453	T/O	2.6M/12M	21.9	22.4	40.7	96k/727k	2.74	86k/5.5k
15x15	T	1053	T/O	2.3M/10M	T/O	T/O	T/O	N/A	22.7	255k/15k

Comparisons: Cross-sections vs. Levels



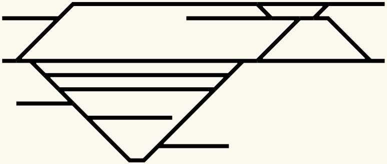
Levels/SAT



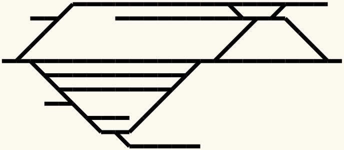
Cross-sec./SAT,
opt. height/bends

Comparisons: Cross-sections vs. Levels

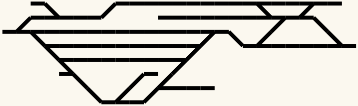
Model: Arna, imported from RailCOMPLETE CAD project



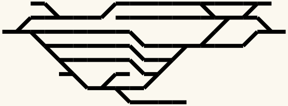
Levels/SAT



Cross-sec./SAT, opt. bends/width



Cross-sec./SAT, opt. height/bends



Cross-sec./SAT, opt. height/width

Conclusions

Conclusions:

- Incremental SAT solvers can draw railway schematics, managing several optimization criteria.
- The choice of encoding makes a significant difference for larger models.
- Better encodings required after finding a more structured solution space, where the order of symbols is hard-coded/implicit in the SAT problem.

Future work:

- Is the Levels formulation really NP complete?
- Industrial use would require symbol placement and interactive manipulation.