Synthesis of Railway Signaling Layout from Local Capacity Specifications*

Bjørnar Luteberget[1], Christian Johansen[2], and Martin Steffen[2]

[1] Railcomplete AS, Oslo, Norway
[2] Department of Informatics, University of Oslo, Norway
bjlut@railcomplete.no,{cristi,msteffen}@ifi.uio.no

Abstract. We present an optimization-based synthesis method for laying out railway signaling components on a given track infrastructure to fulfill capacity specifications. The specifications and the optimization method are designed to be suitable for the scope of signaling construction projects and their associated interlocking systems, but can be adapted to related problems in, e.g., highway, tram, or airport runway designs.
The main synthesis algorithm starts from an initial heuristic over-approximation of required signaling components and iterates towards better designs using two main optimization techniques: (1) global simultaneous planning of all operational scenarios using incremental SAT-based optimization to eliminate redundant signaling components, and (2) a derivative-free numerical optimization method using as cost function timing results given by a discrete event simulation engine, applied on all the plans from (1).
Synthesizing all of the signaling layout might not always be appropriate in practice, and partial synthesis from an already valid design is a more practical alternative. In consequence, we focus also on the usefulness of the individual optimization steps: SAT-based planning is used to suggest removal of redundant signaling components, whereas numerical optimization of timing results is used to suggest moving signaling components around on the layout, or adding new components. Such changes are suggested to railway engineers using an interactive tool where they can investigate the consequences of applying the various optimizations.

Keywords: railway signaling · capacity · on-the-fly synthesis · incremental SAT · interactive · derivative-free numerical optimization · discrete event simulation

1 Introduction

Signaling engineering for railway infrastructure consists of setting up signals, train detectors, derailers, and related equipment, and then building a control

* The first author was partially supported by the project *RailCons – Automated Methods and Tools for Ensuring Consistency of Railway Designs*, with number 248714 funded by the Norwegian Research Council and Railcomplete AS.

system called the *interlocking* which ensures that all train movements happen in a safe sequence. Comprehensive regulations and processes have been put in place to ensure the safety of such systems, and standards and authorities "highly recommend" using formal methods (of various kinds) for higher safety integrity levels like SIL4 (cf. [7, 12, 2, 6]).

The precise locations of signaling components on the railway tracks can have crucial impact on the capacity of the railway, i.e., its ability to handle intended operational scenarios in a timely manner. Many details of the signaling layout design can cause operational scenarios to become infeasible or slow, s.a.: signal and detector placement, correct allocation and freeing of resources, track lengths, train lengths, etc. Capacity-related decisions in signaling are closely related to the fields of timetable planning and the implementation of interlocking systems, and although tool support for verification of interlockings ([15, 10, 16]) and optimization of timetables ([13, 1, 19]) has been thoroughly investigated and developed since the beginnings of computer science (for example, the maximum flow problem was originally formulated to estimate railway network capacity, see [14]) signaling layout design still lacks appropriate modeling and analysis tools.

Consequently, railway construction projects usually rely on informal, vague, or non-existent capacity specifications, and engineers need to make ad-hoc/manual analyses of how the layout and control system can provide the required capacity. Systematic capacity analysis for railways is typically performed on the scale of national networks, using comprehensive timetables, focusing on delays, congestion, and only after a complete design is finished (cf. [19, 1, 9]). Large-scale capacity analysis thus assumes railway signaling layouts as low-level details which have already been correctly designed. In contrast, we focus in this paper on specifying and fulfilling capacity measures that make sense in the setting of construction projects, typically for a single or a few stations or railway lines.

In earlier work, we have developed methods for both static [23, 24, 22] and dynamic [21] analysis of railway designs and developed tools which run fast enough to be used for immediate feedback in an interactive design process. We have also developed a verification system and a capacity specification language [21] for construction projects, which verifies properties such as running time, train frequency, overtaking and crossing. Building on this verification work, we present in this paper an optimization method where signaling components, i.e., mainly signals and detectors, but also balises, derailers, and catch points, can be moved or removed from the design to improve capacity.

We show how our SAT-based planning procedure can be extended to find redundant signaling equipment, and how a simulator can be extended to move signaling equipment around using continuous-domain mathematical optimization methods and discrete event simulation. With the use of a heuristic initial design algorithm, the optimization procedures can be applied even if the user has not yet supplied any working signaling design, and in this way we get a synthesis algorithm. If a working design is already in place, our method suggests possible design improvements to the user in an interactive style, so that the engineer has the final say in making changes to the design, and can investigate

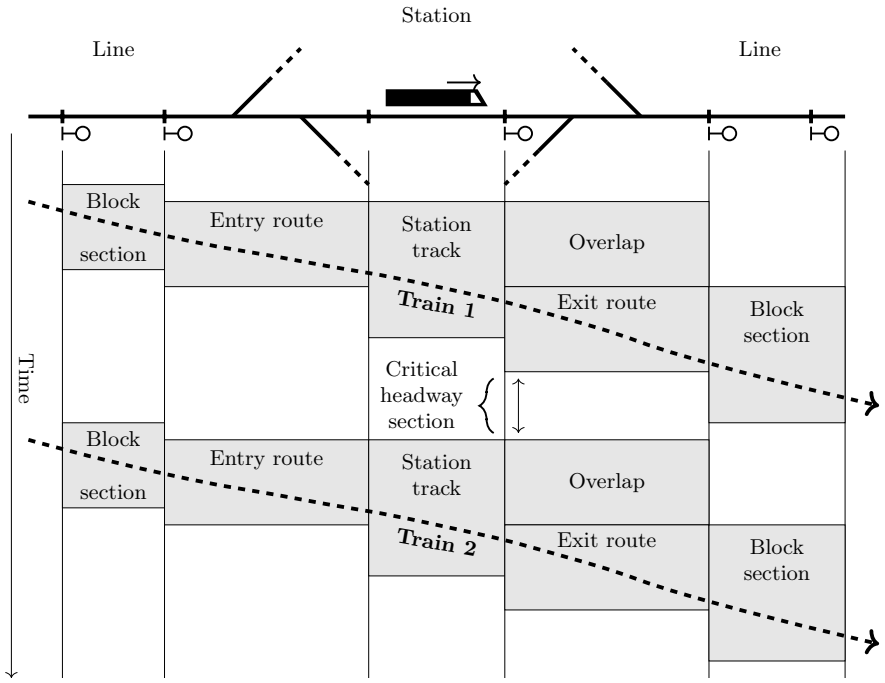

Fig. 1: Blocking time diagram showing two (non-stopping) trains traveling from a line blocking section into a station and back onto a line blocking section. Dashed lines indicate train locations and velocity, and gray boxes indicate the lengths and times of sections exclusively allocated to the trains. Figure adapted from [27].

how the changes influence the infrastructure and operational scenarios. Thus, our method can consider some signals fixed, i.e., part of the design, while there rest are amenable to optimization.

These methods are a step towards a railway signaling engineering methodology based on explicit specifications, and using analysis and verification tools every step along the way, which we believe can improve decision-making.

The main contributions of this paper thus are: (1) defining and demonstrating a novel specification-based design methodology for automating the layout of railway signaling components, (2) extending existing planning and simulation methods to make changes in the designs which improve their quality with respect to given specifications, and (3) showing how incremental optimization and partial synthesis can be used in specification-based design through an interactive tool.

2 Background

The basic safety principles used in most railways around the world are based on dividing railway lines into *fixed blocking sections*, and use signals and train

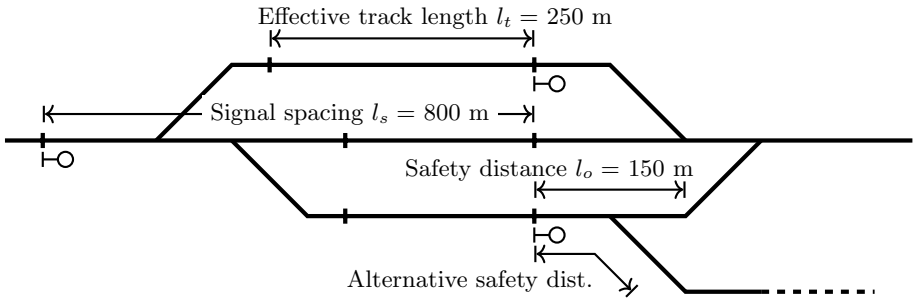

Fig. 2: A schematic track plan, a key artifact in designing the signalling system in a route-based interlocking system. The plan is annotated with signalling components and distances between locations relevant for interlocking safety requirements.

detectors together in an electronic interlocking system which prevents one train from entering a blocking section before it has been cleared by the previous train.

The block section principle directly impacts the maximum frequency of trains, and consequently the *capacity* of the railway, through the interplay between train parameters (length, acceleration, and braking power), track layout (how many tracks are available at which stations), and the location of signalling equipment. The topic of this paper is how to design this infrastructure, specifically how to choose the number and locations of signals and detectors to optimize capacity.

There are two main design methods for deciding signal and detector locations, which have different application areas. The first method is the *blocking time diagram* where a single track on a railway line, or a single path through a railway station, is presented on the horizontal axis, and consecutive trains traveling the same path are plotted with the blocking time of each section shown as rectangles stretching out on the vertical time axis (see Fig. 1).

The second design method is to use a schematic track plan showing the topology of tracks and the locations of signals, detectors, and other signalling system components. The schematic plan is not geographically accurate (for the sake of readability) but is annotated with traveling lengths between relevant locations, such as from one signal to the next signal or detector. This plan is used in the design of *route-based interlocking systems* to make assessments of the effective lengths of station tracks, safety distances from a signal to other tracks (so-called *overlaps*), and more (see Fig. 2).

Observe how the blocking time diagram and the schematic plan provide views in different dimensions: the blocking time diagram provides continuous time and a single spatial dimension but does not treat different choices of path, while the schematic track plan shows all paths at once, but does not directly show how a train would travel in time. The latter concerns *schedulability*, while the former concerns *timing*. For detailed signalling design, the decisions that impact the interaction between these two analysis domains are a complex task where an engineer balances a number of diverse concerns.

2.1 Railway Signalling Layout Design

We define the *railway signalling layout design* problem as follows: given a track plan, and a set of intended operational scenarios, decide on a set of signalling components (signals, detectors, etc.) and their locations, such that it is possible to implement a safe interlocking control system with which the specified operational scenarios can be dispatched efficiently (see example in Fig. 3).

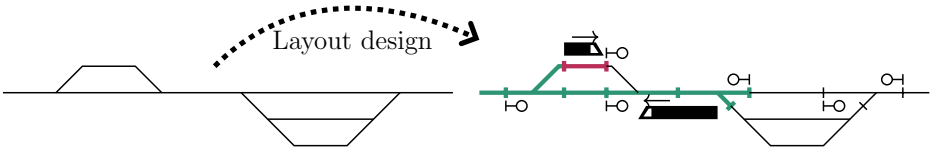

Fig. 3: Railway signalling layout design places a set of signalling components *(as on the right)* on a given track layout *(as on the left)* to ensure that a set of capacity specifications can be fulfilled by dispatching trains in some way.

The main constraints imposed on a signalling design can be classified into four main categories:

1. **Physical infrastructure**: all the trains are guided by the rails and can only travel where the rails guide them. The space that trains move on is a graph with linear connections between nodes.
2. **Allocation of resources**: railway signals are connected to a control system called the interlocking, which ensures mutual exclusion of trains by reading from detectors and ensuring that signals can only signal movement authority when it is safe to do so. This entails that one can only allocate and free resources in certain groupings (see example in Fig. 4).
3. **Limited communication**: the most obvious way to improve capacity on an existing railway line is to install more signals to more finely subdivide the allocation of space so that trains can be traveling more closely on the line. However, since the train driver always has to be able to stop the train within the limits of the currently given length of movement authority, putting

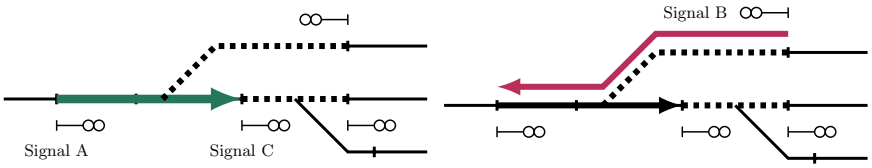

Fig. 4: Allocation and freeing of resources can only be done within the limits of what information the control system can send and receive. In the left figure, a train traveling from Signal A must travel at least until Signal C, and all resources in this path must be allocated and in a safe state before the train can proceed from A. In the right figure, no train can proceed from Signal B because parts of the path require the same resources, meaning elementary routes are conflicting and cannot be used simultaneously.

signals too close together will lower the speed that the train can travel with. This means that there is a limit to how many signals one can install before the capacity starts to decrease because of this (see Fig. 5).

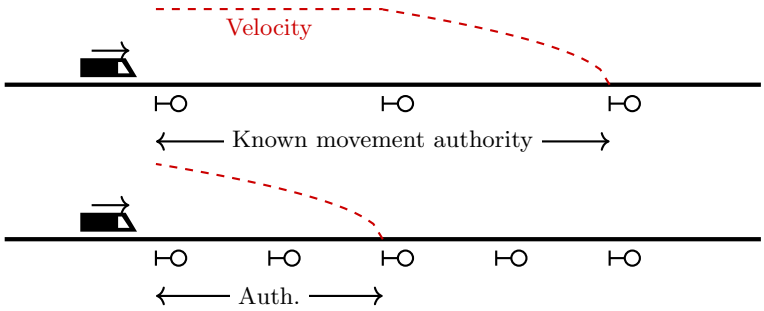

Fig. 5: Signal information only carries across two signals (so-called *distant signals*).

4. **Laws of motion**: when a train is given a movement authority, this authority has a limited length and a limited maximum velocity. The driver must choose when to accelerate and brake to stay within the given authority.

In the methods for optimization and synthesis proposed below, we assume that the above constraints are absolute. In practice, engineers have subtle workarounds for each of these constraints whenever the situation requires a non-standard solution. Physical infrastructure (1) can often be modified by taking a step back in the planning process and re-evaluating the track layout together with track engineers. Allocation of resources (2) can be overcome by designing certain movements to be performed as shunting movements, i.e., a second-grade class of movement authority with lower safety requirements. Limited communication (3) can also be overcome by increasing the number of different aspects that the signals can communicate, or by using cab signalling, giving additional communication between the interlocking system and the train driver. The ETCS Level 2 system currently being implemented in many European countries is capable of signalling any number of routes simultaneously through digital radio communication, effectively removing the infrastructure-to-driver communication restriction. Finally, the laws of motion (4) cannot be overcome in themselves, but increasing the requirements for vehicles' acceleration and braking power may improve a layout design's expected performance.

3 Method

The following list is a summary of the components in our work-flow for solving the railway signalling layout design problem automatically and incrementally:

1. **Track plan and capacity specification input**: Track plans are graph-like structures with information about track lengths, boundary nodes, switches,

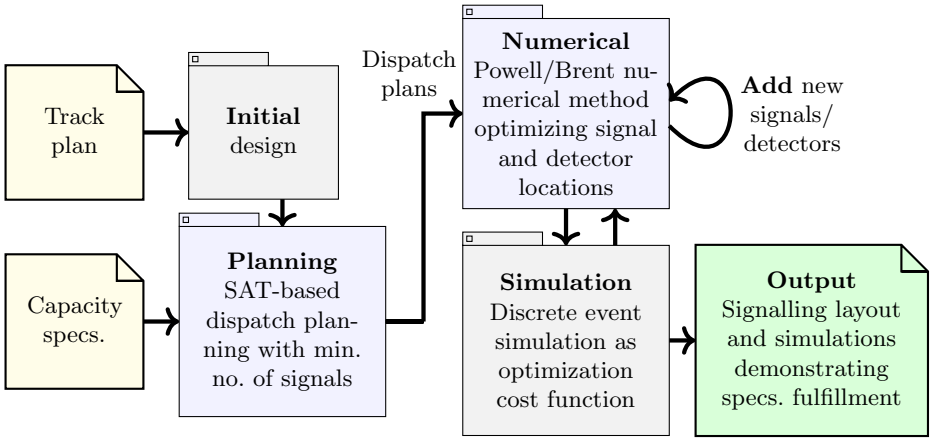

Fig. 6: Synthesis process overview. Track plan and capacity specifications are given as input, and together with an initial design based on a heuristic algorithm they are given to the SAT-based planner for simultaneous dispatch planning of all usage scenarios. A numerical method takes the dispatch plans and adjusts the locations and number of signals and detectors until no better result from simulation is achieved.

and crossings, and are read from the railML format[3]. We use our method from [21] for local capacity specifications in SAT, summarized in Section 3.1.

2. **Initial design**: We propose in Section 3.2 a heuristic algorithm to over-approximate the signaling components required to plan the set of all possible movements on the given track plan. This forms our initial maximal design.
3. **Planning optimization**: Ignoring all timing aspects, we calculate the smallest set of signals and detectors that are able to dispatch all of the scenarios described in the local capacity specifications. This is done by solving a planning problem where all scenarios are planned simultaneously. An incremental SAT solver derives the plans and optimizes the number of signals that are used. This extends our work from [21], and is detailed in Section 3.3.
4. **Numerical optimization**: A measure for the performance of the design is calculated by dispatching all of the planned ways to realize the performance specifications and measuring the difference between the required time and the simulated time. This measure is used as a goal function for a meta-heuristic numerical optimization algorithm for moving the signals around, and when this algorithm converges, each track is tested using Discrete Event **Simulation** for how much improvement would be obtained by adding signals to it and repeating the optimization process. See Section 3.4 below.
5. **Output**: After the process is done, the user is left with a design and a set of dispatch plans and simulated train movements which describe how the capacity requirements are fulfilled by this design.

[3] See `https://railml.org/`

The overall work-flow of our method is thought to be automatic, without manual intervention, unless the user wants to define some signals fixed, which would then be considered part of the track plan input. For this, our synthesis must be incremental, and integrated in the engineers' design tool, offering formal methods automation without requiring any prior knowledge.

3.1 Local Capacity Specifications

To capture typical performance and capacity requirements in construction projects, we have defined in [21] an **operational scenario** $S = (V, M, C)$ as follows:

1. A set of **vehicle types** V, each defined by a length l, a maximum velocity $v_{\max}$, a maximum acceleration a, and a maximum braking deceleration b.
2. A set of **movements** M, each defined by a vehicle type and an ordered sequence of visits. Each visit q is a set of alternative locations $\{l_i\}$ and an optional minimum dwelling time t_d.
3. A set of **timing constraints** C, which are two visits q_a, q_b, and an optional numerical constraint t_c on the minimum time between visit q_a and q_b. The two visits can come from different movements. If the time constraint t_c is omitted, the visits are only required to be ordered, so that $t_{q_a} < t_{q_b}$.

We give here only a simple example of an overtaking requirement. See [21] for further examples[4]. Overtaking as an operational scenario means that two trains traveling in the same direction can be reordered. For example, we specify a passenger train traveling from `b1` to `b2`, and a goods train with the same visits. Timing constraints ensure that the passenger train enters first while the goods train exits first. (Fig. 9 or Fig. 3 contain tracts where this can be performed.)

```
movement passengertrain { visit #p_in [b1]; visit #p_out [b2] }
movement goodstrain { visit #g_in [b1]; visit #g_out [b2] }
timing p_in < g_in; timing g_out < p_out
```

Specifications of this kind can be used to express requirements on running time, train frequency, overtaking, crossing, and similar scenarios which are relevant in railway construction projects. Since we typically only need to refer to locations such as model boundaries and loading/unloading locations, these specifications are not tied to a specific design, and can often be re-used even when the design of the station changes drastically.

3.2 Initial Design

When starting from an empty set of signalling components, most operational scenarios are not possible to even dispatch, because the railway interlocking safety principles require detectors and signals to have control over movements for safety purposes. Instead of searching for signalling components to add to

[4] See complete format: `https://luteberget.github.io/rollingdocs/usage.html`

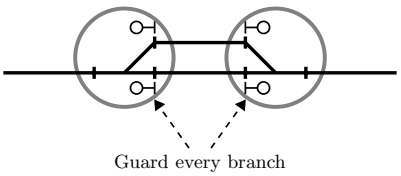

Fig. 7: *Initial design:* put signals in place before every trailing switch, i.e. where tracks join together.

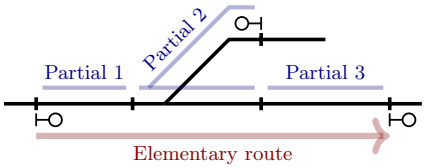

Fig. 8: The planning abstraction of the train dispatch allocates a set of partial routes to each train. Elementary routes are sets of partial routes which must always be allocated together.

the design to allow dispatching to happen, we start the synthesis procedure by heuristically over-approximating the components required to perform dispatch. We insert a signal and a detector in front of every trailing switch, and at a set of specified lengths corresponding to the choices of length of safety zone. We also insert a detector in front of every facing switch. See Figure 7. If more than one train is required on the same track for overtaking or crossing, we can also choose to insert signals at multiples of the trains' lengths. When there are several paths of the specified length leading to a trailing switch, we put signals and detectors at all the relevant locations. This design aims to allow all possible dispatches and we rely on the next stage of the synthesis to remove redundant equipment.

3.3 SAT-based dispatch planning

The operational scenarios of the local capacity specifications describe train movements only declaratively, so the first step to analyzing concrete states of the system is to solve a planning problem which gives us a set of dispatch plans, i.e., determining sequences of trains and elementary routes which make the trains end up visiting locations according to the movements specification.

Instead of using a constraint solver system (e.g. SMT solvers) to solve for route dispatching and train dynamics simultaneously, we have chosen to separate the *abstracted planning problem* (i.e. selecting elementary routes to dispatch) from the physical constraints of train dynamics. This choice was made for performance and extensibility reasons (see [21, Sec.III] for details).

We use the encoding from [21, Sec.III(B)] of an instance of the abstracted planning problem into an instance of the Boolean satisfiability problem (SAT, see [4] for an overview of SAT techniques). We consider the problem as a model checking problem, and use the technique of bounded model checking (BMC) [3] to unroll the transition relation of the system for a number of steps k, expressing states and transitions using propositional logic. We thus assert the existence of a plan, so that when the corresponding SAT instance is satisfiable, it proves the fulfillment of the performance requirements and gives an example plan for it. When unsatisfiable, we are ensured that there is no plan within the number of k steps. Interlocking features such as elementary routes, partial route release, flank

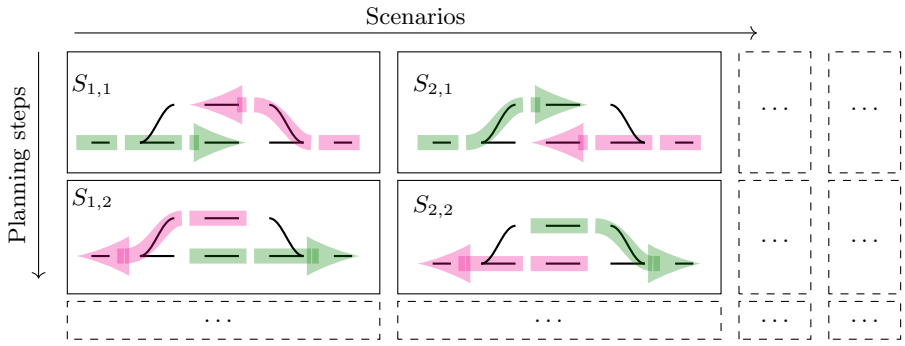

Fig. 9: The planning matrix consists of the occupation status of a set of partial routes for each state required for dispatch planning, and for each scenario in the local capacity requirements. The top left cells show an example dispatch of a crossing movement where green areas show track segments which are currently occupied by a train going from left to right, while the pink areas show track segments which are currently occupied by a train going from right to left.

protection, overlaps, overlap timeouts, and swinging overlaps, can be converted into our representation for solving the abstract planning problem.

To find a subset of the signaling components from the initial design that is sufficient to successfully plan all the dispatches, we extend the planning approach described above by adding a set of signal usage Booleans u indicating whether the signal is needed. The set of occupancy status Booleans o_r^i (for route r in state i, taking values either *Free* or a train t) is repeated once for each operational scenario, resulting in a SAT instance with parallel execution of each scenario on copies of the same infrastructure (see Fig. 9). We link the signal usage status u to each copy of the state so that the signal is marked as needed if it is used independently of other signals:

$$\begin{aligned}\forall i \in \text{State} : \forall s \in \text{Signal} : \forall t \in \text{Train} : \quad & \neg u_s \Rightarrow \\ & \bigvee \left\{ \left(o_r^i \neq t \wedge o_r^{i+1} = t\right) \mid \text{exit}(r) = s \right\} \Rightarrow \\ & \bigvee \left\{ \left(o_r^i \neq t \wedge o_r^{i+1} = t\right) \mid \text{entry}(r) = s \right\} \ .\end{aligned}$$

Similar approaches are taken for other signaling component types.

Now we find the smallest set of signaling equipment which is sufficient to allow dispatching all scenarios. We minimize the number of signals by: taking the sum of u variables as a unary-encoded number (see [5]) and then solving SAT incrementally with a binary search on the upper bound of the sum.

3.4 Numerical optimization

When we have a design where dispatching is possible, we have fulfilled the discrete part of the dispatch plan. Timing constraints, however might not yet be

fulfilled, and we might also want to improve on the total execution time of the various dispatch plans. To improve on the basic design found by the planner, we solve a numerical optimization problem with a cost function f defined as a weighted sum of dispatch timing measures:

$$f_b(\boldsymbol{x}) = \sum_s w_s \left(\frac{1}{n_s} \sum_d t_{b+\boldsymbol{x}}(d) \right),$$

where $\boldsymbol{x}$ is a vector with components representing the location of each signal and detector, s indexes operational scenarios from the set of capacity specifications, w_s is weight assigned to the operational scenario, d indexes the set of n_s alternative dispatch plans derived by the planning algorithm for each operational scenario, and $t_{b+\boldsymbol{x}}(d)$ is the time measure calculated by executing the dispatch plan d using the discrete event simulation component (described in Section 3.5) on an infrastructure constructed by adding the signal and detector locations $\boldsymbol{x}$ to the base track plan infrastructure b.

We define two basic operations for optimizing the timing performance of a signalling layout:

1. Searching for the optimal signalling component locations $\boldsymbol{x}$ for a fixed set of components located on a fixed set of tracks in a fixed order using Powell's method and Brent's method of derivative-free numerical optimization.
2. Adding a new signal or detector to any track.

Powell's method and Brent's method. Since we use simulation to measure the cost of a design, we do not have an expression for the derivative of the cost function f_b, and this function is not even guaranteed to be continuous. Even so, it is possible to use numerical methods for local optimization without taking derivatives. We use Brent's method for minimization in the single-parameter case, with the generalization to multivariate functions by Powell's method.

Powell's method works as follows: given a domain $D \subset \mathbb{R}^n$, an initial point $\boldsymbol{x}_0 \in D$, and a cost function $f : D \to \mathbb{R}$, create a set of search vectors V initially containing each of the unit vectors aligned with each axis of $\mathbb{R}^n$. Iterate through the search vectors $\boldsymbol{v}_i \in V$ and do a line search for the parameter α giving the optimal point of $\boldsymbol{x}_{i+1} = f(\boldsymbol{x}_i + \alpha \boldsymbol{v}_i)$. After updating $\boldsymbol{x}$ using each search vector, remove the search vector which yielded the highest α and add instead the unit vector in the direction of $\boldsymbol{x} - \boldsymbol{x}_0$. See [8] for details.

Brent's method for optimization is used for the line search sub-routine in Powell's method. It takes a range of α values for which $\boldsymbol{x}_i + \alpha \boldsymbol{v}_i$ is inside D, and does a robust line search which finds a local minimum even for non-smooth and discontinuous functions. The method keeps a set of the three best points seen so far and fits a quadratic polynomial with the three best function values as parameters (called *inverse quadratic interpolation*). If the predicted optimum by the quadratic fit falls within an expected range, it used as the new best guess, otherwise the method falls back to golden-section search. See [26, 8] for details.

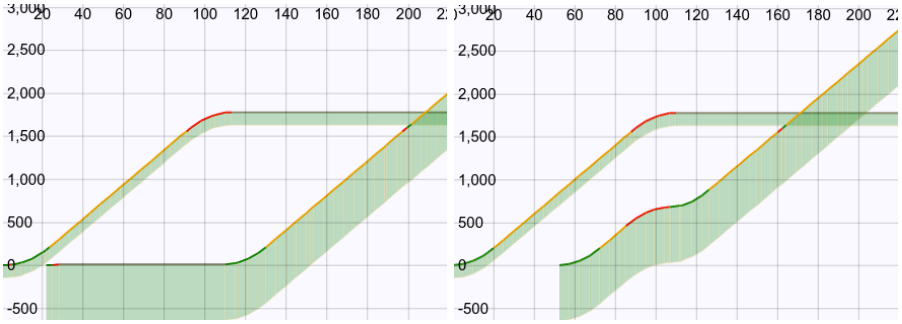

Fig. 10: Partial screen capture from our interactive design tool showing before (left) and after (right) improving signal and detector locations for a two-track station on an overtaking scenario. Note that the time axis is horizontal in this example. A signal at $x \approx 0$ m is moved to $x \approx 700$ m so that the overtaking train is unblocked at an earlier time, lowering the overall time taken to perform the operation.

To simplify the use of the numerical algorithms, we map each signalling component's position to an intrinsic coordinate in the interval $[0, 1]$, so that the vector $\boldsymbol{x}$ keeps within $D = [0, 1]^n$. For a component with position p relative to the start of its track, if the component is the only component on a track, we define its intrinsic coordinate as

$$x = \frac{p - (l_a + l_{\min})}{(l_b - l_{\min}) - (l_a + l_{\min})},$$

where $l_a = 0$, l_b is the length of the track, and $l_{\min}$ is the minimum spacing between components. When there are several components on the same track, we convert the coordinates by processing the components in order of increasing p, and adjusting l_a to correspond to the location of the previous component on the track. In this way the whole of $[0, 1]^n$ represents valid component positions and we do not have to apply constraints to the search space by other methods.

See Fig. 10 for an example of signalling compoments being moved.

Adding new components. When the above optimization has converged for a fixed set of components $\boldsymbol{x}$, we iterate over each track (and each direction), adding a new component and including its dimensions in $\boldsymbol{x}$, re-running optimization, and see which track, if any, most benefits from adding a signal or detector.

3.5 Discrete event simulation

The time measure used in the optimization loop (of Section 3.4) is calculated by simulation on a fixed infrastructure, which is a well-established method in railway capacity research. For this we use the custom simulator which we developed in [21, Sec.III], not described here, (see [28] for a methodological overview, and [17, 9, 18] for discrete events simulation for railway applications). Commercial railway simulation software can also be used instead of custom solutions.

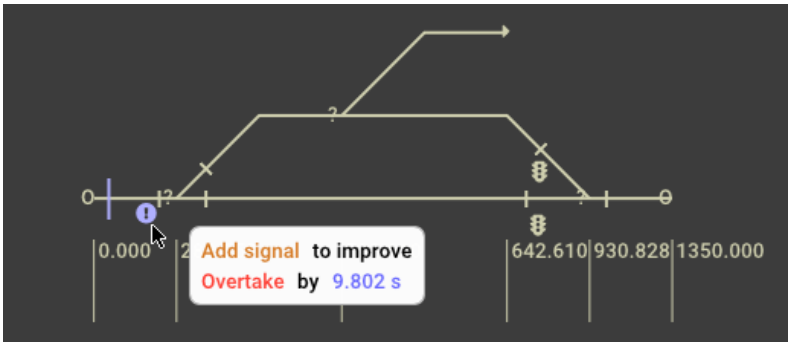

Fig. 11: Partial screen capture from our interactive design tool showing suggestions for design improvement to the user, inspired by integrated development environments used for programming. The individual optimization steps run their calculations as a background process, showing an information symbol where the algorithm is able to provide an improvement over the current design. The user can decide to implement it or to dismiss this change and similar changes from future suggestions.

We also use an automated derivation procedure for interlocking specifications to adjust the behavior of the control system after making changes in the infrastructure, similar to the procedure described in [29].

4 Local Optimizations and Interactive Improvement

In practice, synthesis from-scratch may well be ill-suited. The principle reason for this is the incompleteness of our synthesis method, which implies several inadequacies including, e.g., failing to recognize key concerns the design should be based upon, or if its calculation time prohibits practical use. But even if the specification successfully captures the capacity requirements, and the synthesis algorithm can come up with designs with good capacity, there are in practice often other constraints which can make a full from-scratch synthesis ill-suited. For example, in upgrade construction projects, it might be more useful to search for and suggest small changes which would be the most effective remedies for bottlenecks in a station's capacity. In fact, in such interactive verification and synthesis situations like ours, incompleteness is not a concern since we know that the problem is too difficult for automation and we only aim for the formal tool to provide help to the human. In that case we are mainly interested in the *correctness* of the method, i.e., the help that it provides should be useful help and not spurious suggestions; whereas incompleteness only means that there are some solutions that the tool cannot find, thus becoming the responsibility of the human. So we instead strive for good coverage of the solution space.

Our method and tool[5] can be used in several ways, i.e.: we consider each optimization step as described below as a possible incremental step towards a

[5] Usage details of our tool can be found on the project's web page: `https://www.mn.uio.no/ifi/english/research/projects/railcons/index.html#Tools`

better design, which can be performed by a user interactively. Using a computer-assisted design program for railway (s.a. RailComplete) with semantic information about railway objects and rail network topology, the user gets suggestions for small changes to their design and can investigate how applying these changes affects the various scenarios (e.g., see Fig. 11).

Local optimization steps suggested to the user are the following:

- **Redundant equipment**: if removing a single object from the drawing can still be made to satisfy all local capacity requirements, the program suggests that the object is redundant. This class of suggestions is based on the SAT-based component minimization technique described above.
- **Local move of equipment**: if moving a single object or a set of nearby objects can improve the overall capacity measure on the station, the program suggests moving the object (or set of objects). This class of suggestions is based on the numerical timing optimization technique described above.
- **Adding equipment**: if adding a single piece of equipment (and performing local moves of equipment afterwards) can improve timing, the program suggests this to the user. This class of suggestions is based on the numerical timing optimization technique described above.

When accepting any of these changes, a user can investigate how the dispatch plans and the timings change. The tool meanwhile calculates new suggestions based on the new layout. We have developed a prototype tool which can calculate and suggest such changes to a user while they are editing their layout, and we are currently starting testing of this tool in an industrial setting together with railway engineers to investigate how useful such suggestions are, and how often they can be used compared to a from-scratch synthesis.

5 Conclusions, Related and Further Work

We have presented a method for partially or fully automating signalling layout design using SAT-based planning and discrete event simulation. The automation of verification, optimization, and synthesis relies on specifications tailored to the relevant scope, and we hope that this is a step on the way to integrating explicit formal specifications into the layout design process. More details can be found in the PhD thesis of the first author [20, Chap.4].

Our planning algorithm uses fixed blocks, so it handles conventional lamp signalling and the European standard ERTMS/ETCS Level 2, while handling Level 3 (which uses moving block) would require changes to the planning algorithm.

The simulation paradigm is imperative, progressing by calculating train trajectories forward in time. This makes the overall synthesis easily extensible with timing-related details, such as engine and braking power models, resistance models, operational regulations, automatic train control systems, etc., which do not impact the applicability of the dispatch plan but impact the timing performance.

For the local incremental operations above we consider useful running times to be under one second, so the method can be used integrated inside engineers' design tools, offering instant feedback.

5.1 Related works

Although the literature is comprehensive on the safety-critical implementation of railway interlockings and operational analysis of large-scale railway networks, the signalling layout problem in itself has little coverage. We are only aware of the following works: Mao et al. [25] presented a genetic algorithm solution to signal placement, but the method is limited to the one-dimensional railway line, and does not handle signal placements inside stations/interlockings. Dillmann and Hähnle [11] describe a heuristic algorithm for upgrading German conventional signalling systems to an ETCS system, aiming to replicate the behavior and capacity of the existing system.

5.2 Further work

Although our method is capable of making good design choices in several industrial standard models, we are aware of several limitations. Firstly, the method is not complete – we cannot guarantee finding an optimum because of the following: (1) the initial design does not guarantee maximum possible schedulability, (2) although the global simultaneous planning is exact in finding the smallest subset of the initial plan which can dispatch the operational scenarios, this set might not be the optimal starting point for timing optimization, and (3) the cost that we use for numerical optimization can have multiple local optima, especially when summing the score for competing operational scenarios, in which case the method described above is not guaranteed to find the global optimum.

However, incomplete methods are often very useful in practice, and for us it remains to thoroughly test how much gains our formal automation brings to the engineers. We also need to evaluate empirically the quality of the resulting signal placement as a crucial factor for industrial adoption.

We have also identified the following concerns for scalability of the method: (1) the specification language is practical to use for passing tracks, junctions, and medium-sized terminal stations, but on large-sized terminals and larger-scale analysis across multiple stations, the language is not easy to use because it specifies single movements separately, (2) optimizing the number of detectors in the SAT problem requires quantifying over all paths, which will cause scaling problems on larger track plans with many path choices, and (3) the algorithm for adding new signals to improve performance is naive, and will be expensive for track plans with a large number of tracks.

However, for such large-scale analysis it is already common to use commercial tools like OpenTrack[6] or LUKS[7], whereas our method is meant to be used on smaller scales as in the design phase, aiming to help the engineer to reduce the amount of errors the commercial tools would later find.

[6] "OpenTrack: Simulation of railway networks" 2018. `http://www.opentrack.ch/`

[7] "LUKS: Analysis of lines and junctions" 2018. `http://www.via-con.de/development/luks`

References

1. Abril, M., Barber, F., Ingolotti, L., Salido, M., Tormos, P., Lova, A.: An assessment of railway capacity. Transportation Research Part E: Logistics and Transportation Review 44(5), 774 – 806 (2008), https://doi.org/10.1016/j.tre.2007.04.001
2. Basile, D., ter Beek, M.H., Fantechi, A., Gnesi, S., Mazzanti, F., Piattino, A., Trentini, D., Ferrari, A.: On the industrial uptake of formal methods in the railway domain — A survey with stakeholders. In: Furia, C.A., Winter, K. (eds.) Integrated Formal Methods IFM 2018. Lecture Notes in Computer Science, vol. 11023, pp. 20–29. Springer-Verlag (2018), https://doi.org/10.1007/978-3-319-98938-9_2
3. Biere, A., Cimatti, A., Clarke, E.M., Strichman, O., Zhu, Y.: Bounded model checking. Advances in computers 58(11), 117–148 (2003), https://doi.org/10.1016/S0065-2458(03)58003-2
4. Biere, A., Heule, M., van Maaren, H., Walsh, T. (eds.): Handbook of Satisfiability, Frontiers in Artificial Intelligence and Applications, vol. 185. IOS Press (2009)
5. Björk, M.: Successful SAT encoding techniques. Journal on Satisfiability, Boolean Modeling and Computation 7(4), 189–201 (2011), https://satassociation.org/jsat/index.php/jsat/article/view/153/118
6. Borälv, A., Stålmarck, G.: Formal verification in railways. In: Hinchey, M.G., Bowen, J.P. (eds.) Industrial-Strength Formal Methods in Practice, pp. 329–350. Springer (1999), https://doi.org/10.1007/978-1-4471-0523-7_15
7. Boulanger, J.L.: CENELEC 50128 and IEC 62279 Standards. Wiley-ISTE (Mar 2015)
8. Brent, R.P.: Algorithms for minimization without derivatives. Dover Publications, Mineola, N.Y (2002)
9. Büker, T., Seybold, B.: Stochastic modelling of delay propagation in large networks. Journal of Rail Transport Planning & Management 2(1-2), 34–50 (2012), https://doi.org/10.1016/j.jrtpm.2012.10.001
10. Cimatti, A., Corvino, R., Lazzaro, A., Narasamdya, I., Rizzo, T., Roveri, M., Sanseviero, A., Tchaltsev, A.: Formal verification and validation of ERTMS industrial railway train spacing system. In: Madhusudan, P., Seshia, S.A. (eds.) 24th International Conference on Computer Aided Verification (CAV). Lecture Notes in Computer Science, vol. 7358, pp. 378–393. Springer-Verlag (2012), https://doi.org/10.1007/978-3-642-31424-7_29
11. Dillmann, S., Hähnle, R.: Automated planning of ETCS tracks. In: Proceedings of Reliability, Safety, and Security of Railway Systems (RSSRail 2019). Lecture Notes in Computer Science, vol. 11495, pp. 79–90. Springer-Verlag (2019)
12. Fantechi, A.: Twenty-five years of formal methods and railways: What next? In: Counsell, S., Núñez, M. (eds.) Software Engineering and Formal Methods (SEFM). Lecture Notes in Computer Science, vol. 8368, pp. 167–183. Springer (2013), https://doi.org/10.1007/978-3-319-05032-4_13
13. Hansen, I.A., Pachl, J.: Railway Timetabling and Operations. Eurailpress (2014)
14. Harris, T., Ross, F.S.: Fundamentals of a method for evaluating rail net capacities. Tech. Rep. RM-1573, Rand Corp. (1955)
15. Hartonas-Garmhausen, V., Campos, S.V.A., Cimatti, A., Clarke, E.M., Giunchiglia, F.: Verification of a safety-critical railway interlocking system with real-time constraints. Sci. Comput. Program. 36(1), 53–64 (2000), https://doi.org/10.1016/S0167-6423(99)00016-7
16. Haxthausen, A.E., Peleska, J., Kinder, S.: A formal approach for the construction and verification of railway control systems. Formal Asp. Comput. 23(2), 191–219 (2011), https://doi.org/10.1007/s00165-009-0143-6

17. Hürlimann, D.: Objektorientierte Modellierung von Infrastrukturelementen und Betriebsvorgängen im Eisenbahnwesen. Ph.D. thesis, ETH Zurich (2002), https://www.research-collection.ethz.ch/handle/20.500.11850/47957
18. Kamburjan, E., Hähnle, R., Schön, S.: Formal modeling and analysis of railway operations with active objects. Sci. Comput. Program. 166, 167–193 (2018), https://doi.org/10.1016/j.scico.2018.07.001
19. Landex, A.: Methods to estimate railway capacity and passenger delays. Ph.D. thesis, Technical University of Denmark (DTU) (2008), http://orbit.dtu.dk/en/publications/id(f5578206-74c3-4c94-ba0d-43f7da82bf95).html
20. Luteberget, B.: Automated Reasoning for Planning Railway Infrastructure. Ph.D. thesis, Faculty of Mathematics and Natural Sciences, University of Oslo (2019)
21. Luteberget, B., Claessen, K., Johansen, C.: Design-time railway capacity verification using SAT modulo discrete event simulation. In: Bjørner, N., Gurfinkel, A. (eds.) Formal Methods in Computer Aided Design (FMCAD). pp. 1–9. IEEE (2018), https://doi.org/10.23919/FMCAD.2018.8603003
22. Luteberget, B., Johansen, C.: Efficient verification of railway infrastructure designs against standard regulations. Formal Methods in System Design 52(1), 1–32 (2018), https://doi.org/10.1007/s10703-017-0281-z
23. Luteberget, B., Johansen, C., Feyling, C., Steffen, M.: Rule-based incremental verification tools applied to railway designs and regulations. In: Fitzgerald, J., Heitmeyer, C., Gnesi, S., Philippou, A. (eds.) Formal Methods (FM 2016). Lecture Notes in Computer Science, vol. 9995, pp. 772–778. Springer-Verlag (2016), http://dx.doi.org/10.1007/978-3-319-48989-6_49
24. Luteberget, B., Johansen, C., Steffen, M.: Rule-based consistency checking of railway infrastructure designs. In: 12th International Conference on Integrated Formal Methods (iFM). LNCS, vol. 9681, pp. 491–507. Springer (2016), http://dx.doi.org/10.1007/978-3-319-33693-0_31
25. Mao, B., Liu, J., Ding, Y., Liu, H., Ho, T.K.: Signalling layout for fixed-block railway lines with real-coded genetic algorithms. Hong Kong Institution of Engineers, Transactions 13(1), 35–40 (2006), https://eprints.qut.edu.au/38260/
26. Nocedal, J., Wright, S.J.: Numerical Optimization. Springer, second edn. (2006)
27. Pachl, J.: Railway Operation and Control. VTD Rail Publishing (2015)
28. Robinson, S.: Simulation: The Practice of Model Development and Use. John Wiley & Sons, Inc., USA (2004)
29. Vu, L.H., Haxthausen, A.E., Peleska, J.: A domain-specific language for railway interlocking systems. In: Schnieder, E., Tarnai, G. (eds.) Proceedings of the 10th Symposium on Formal Methods for Automation and Safety in Railway and Automotive Systems, (FORMS/FORMAT). pp. 200–209. TU Braunschweig (2014)