Optimization and Synthesis of Railway Signalling Layout from Local Capacity Specifications

Bjørnar Luteberget1, Christian Johansen2, and Martin Steffen2

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 signalling components on a given track infrastructure to fulfill capacity specifications. The specifications and the optimization method are aimed towards the scope of signalling construction projects and their associated interlocking systems.

The main synthesis algorithm starts from an initial heuristic over-approximation of required signalling 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 signalling components, and (2) a derivative-free numerical optimization method using as cost function timing results given by a discrete event simulation engine.

Synthesizing all of the signalling layout might not always be appropriate in practice, and partial synthesis from an already valid design can be an 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.

1 Introduction

Signalling engineering, in the context of constructing of railway infrastructure, consists of setting up signals, train detectors, deraillers, and related equipment, and building a control 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 recommend using formal methods (of various kinds), and for higher safety integrity levels (SIL), they “highly recommend” them (cf. [7, 14, 2, 6]).

At the same time, the locations of signalling 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 signalling layout design which can cause operational scenarios to become infeasible or slow,
such as signal and detector placement, correct allocation and freeing of resources, track lengths, train lengths, etc. Capacity-related decisions in signalling border closely to the fields of timetable planning and the implementation of interlocking systems, and although tool support for verification of interlockings ([17, 10, 18]) and optimization of timetables ([15, 1, 21]) 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 [16]) signalling layout design still lacks appropriate modelling and analysis.

Consequently, railway construction projects usually rely on informal, vague, or even non-existent capacity specifications, and engineers need to make ad-hoc/manual analyses of how the layout and control system can provide this capacity. Systematic capacity analysis for railways is typically performed on the scale of national railway networks, using comprehensive timetables, focusing on delays, congestion, and only after a complete design is finished (cf. [21, 1, 9]). Large-scale capacity analysis in this style assumes railway signalling layouts as low-levels details which have already been correctly designed. We focus in this paper on specifying and fulfilling capacity measures that make sense on the scale of construction projects, typically a single or a few stations or railway lines.

In earlier work, we have developed and presented methods for both static [24, 25, 23] and dynamic [22] 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 specifications language [22] 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 signalling components, i.e. mainly signals and detectors, but also balises, derailleurs, 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 signalling 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 of an interactive tool, so that the engineer has the final say in making changes to the design, and can investigate how the changes influence the infrastructure and operational scenarios.

These methods are a step towards a railway signalling 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 are: (1) suggesting and demonstrating a novel specification-based design methodology for the layout of railway signalling 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 detectors together in an automated 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 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).

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 [28].
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).

Note here 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 is a complex task where an engineer balances a high number of diverse concerns.

Placing signals, detectors, and other components so that trains can be guided to their intended tracks and platforms safely and efficiently is the problem of railway signalling layout design.

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.
Fig. 3: Railway signalling layout design places a set of signalling components on a given track layout 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.

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 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).
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.

\[ v - v_0 \leq a \Delta t, \quad v_i^2 - v_j^2 \leq 2bs. \]

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 lifting 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 components in our work-flow for solving the railway signalling layout design problem:

1. **Track plan and capacity specification input**: track plans are graph-like structures with information about track lengths, boundary nodes, switches, and crossings, and we read this data from the railML format\(^3\). The capacity specifications are described in Section 3.1 below.

\(^3\) See [https://railml.org/](https://railml.org/)
2. **Initial design**: a heuristic algorithm is used to over-approximate the signalling components required to plan any set of movements on the infrastructure, if they are possible with the given track plan. See Section 3.2 below.

3. **Derived interlocking specification**: we rely on an automatic derivation of interlocking specifications from the layout, allowing only customizations which are global parameters (e.g. overlap policy). Such derivation algorithms have been described in the literature, see [31].

4. **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 solves the plans and optimizes the number of signals that are used. See Section 3.3 below.

5. **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 for how much improvement would be had by adding signals to it and repeating the optimization process. See Section 3.4 below.

6. **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.
3.1 Local Capacity Specifications

To capture typical performance and capacity requirements in construction projects, we define 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 $v_{\text{max}}$, a maximum acceleration $a$, and a maximum braking retardation $b$.
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 a overtaking requirement. See [22] for further examples. 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.

```
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 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

---

4 See input file format: https://luteberget.github.io/rollingdocs/usage.html
Guard every branch

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

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 [22] for details).

We encode 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 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 state and transitions using propositional logic. Using BMC for planning works by asserting 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.

The abstracted planning problem is encoded as a SAT instance by representing the occupancy status $o^i_{r_j}$ of each route $r_j$ in each state $i$: it can be free ($o^i_{r_j} =$ Free) or it can be occupied by a specific train ($o^i_{r_j} = t_k$). Each combination of route and train is represented by a Boolean variable, but we write constraints with $o^i_{r_j}$ as a variable from the set of trains. A dispatch plan is produced directly from the occupancy status $o^i_{r_j}$ of states by taking the difference between consecutive states and then dispatching any trains and routes which become active from one state to the next.
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.

Fig. 10: Propositional logic representation of constraints on the occupancy status \( o \) in the abstracted dispatch planning problem. Some details regarding overlaps, loops, and repetitions are left out for brevity, see [22].
A summary of constraints is shown in Fig. 10, and we have used well-known techniques for cardinality constraints (see [30]), non-Boolean value encodings (see [5]), and incremental solving (see [12, 13]) – further details in [22]. Interlocking features such as elementary routes, partial route release, flank protection, overlaps, overlap timeouts, and swinging overlaps, can be converted into this conflict-based problem representation for solving the abstract planning problem.

To find a subset of the signalling components from the initial design that is sufficient to successfully plan all the dispatches, we use the planning approach described above and add a set of signal usage Booleans $u$ indicating whether the signal is needed. The set of occupancy status Booleans $o$ 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:

$$\forall i \in \text{State} : \forall s \in \text{Signal} : \forall t \in \text{Train} : \neg u_s \Rightarrow$$
$$\bigvee \{ (o_i^t \neq t \land o_i^{t+1} = t) \mid \text{exit}(r) = s \} \Rightarrow$$
$$\bigvee \{ (o_i^t \neq t \land o_i^{t+1} = t) \mid \text{entry}(r) = s \}$$

Similar approaches are taken for other signalling component types.

Now we find the smallest set of signalling equipment which is sufficient to allow dispatching all scenarios. We use a simple technique to minimize number of signals: take the sum of $u$ variables as a unary-encoded number (see [5]) and solve SAT incrementally with a binary search on the upper bound on 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. However, timing constrains 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:

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

where $x$ 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 $t_{b+x}(d)$ is the time measure calculated by executing the dispatch plan $d$ by discrete event simulation on the infrastructure constructed by adding the signal and detector locations $x$ 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 \( x \) 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 used 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 \( x_0 \in D \), 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 \( v_i \in V \) and do a line search for the parameter \( \alpha \) giving the optimal point of \( x_i + \alpha v_i \). After updating \( x_i \) using each search vector, remove the search vector which yielded the highest \( \alpha \) and add instead the unit vector in the direction of \( x - x_0 \). 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 \( \alpha \) values for which \( x_i + \alpha v_i \) 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 [27, 8] for details.

![Fig. 11: 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.](image)
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 \(x\) 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

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

where \(l_a = 0\), \(l_b\) is the length of the track, and \(l_{\text{min}}\) 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. 11 for an example of signalling components being moved.

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

Discrete event simulation The time measure \(t\) is calculated by simulation on a fixed infrastructure, which is a well-established method in railway capacity research. We have developed a simple custom simulator which we will not describe in more detail here (see [29] for a methodological overview, and [19, 9, 20] for discrete events simulation for railway applications). Commercial railway simulation software can also be used instead of custom solutions.

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 [31].

4 Local Optimizations and Interactive Improvement

There are many reasons that a from-scratch synthesis can be unsuitable in practice. The main reason would be that the synthesis method itself is inadequate, for example if it fails to recognize a key concern that the design should be based upon, or if its calculation time prohibits practical use.

Even if the specifications successfully capture the capacity requirements, and the the synthesis algorithm in itself can adequately come up with designs with good capacity, there in practice often other constraints which can make a full from-scratch synthesis unsuitable. 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.

To increase the number of ways that our methods can be useful, we consider also each optimization step as described below as a possible incremental step.
Fig. 12: 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.

Towards a better design, which can be performed by a user interactively. Using a computer-assisted design program for railway, or a drafting program (such as AutoCAD) extended with semantic information about railway objects and rail network topology, the user gets suggestions for smaller changes to their design and can investigate how applying these changes affects the various scenarios.

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 the user accepts any of these changes, they 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 Related Work

Although the literature is comprehensive on railway engineering in general, the safety-critical implementation 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. [26] 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.

6 Conclusions and Future 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 rely on specifications that are 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.

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, which 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.

Although our method is capable of making good design choices in several simple 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.

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 largescale 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.
References


