





Design Tasks and Their Complexity for the European Train Control System with Hybrid Train Detection

Stefan Engels , Tom Peham , Judith Przigoda, Nils Przigoda , and Robert Wille ,

Abstract

Railway networks have become increasingly important in recent times, especially in moving freight and public transportation from road traffic and planes to more environmentally friendly trains. Since expanding the global railway network is time- and resource-consuming, maximizing the rail capacity of the existing infrastructure is desirable. However, simply running more trains is infeasible as certain constraints enforced by the train control system must be satisfied. The capacity of a network depends (amongst others) on the distance between trains allowed by this safety system. While most signaling systems rely on fixed blocks defined by costly hardware, new specifications provided by Level 2 with Hybrid Train Detection of the European Train Control System (ETCS L2 HTD), formerly known as ETCS Hybrid Level 3, allow the usage of virtual subsections. This additional degree of freedom allows for shorter train following times and, thus, more trains on existing railway tracks.

On the other hand, new design tasks arise on which automated methods might be helpful for designers of modern railway networks. However, although first approaches exist that solve design problems arising within ETCS L2 HTD, neither formal descriptions nor results on the computational complexity of the corresponding design tasks exist. In this paper, we fill this gap by providing a formal description of design tasks for ETCS L2 HTD and proof that these tasks are NP-complete or NP-hard, respectively. By that, we are providing a solid basis for the future development of methods to solve those tasks, which will be integrated into the *Munich Train Control Toolkit* available open-source on GitHub at <https://github.com/cda-tum/mtct>.

Index Terms

Train control systems, hybrid train detection, design automation, formalization, complexity

S. Engels and T. Peham are with the Technical University of Munich, TUM School of Computation, Information and Technology, Chair for Design Automation, 80333 Munich, Germany.

E-mail: {stefan.engels, tom.peham}@tum.de

J. Przigoda and N. Przigoda are with Siemens Mobility GmbH, 38126 Braunschweig, Germany.

E-mail: {judith.przigoda, nils.przigoda}@siemens.com

R. Wille is with the Technical University of Munich, TUM School of Computation, Information and Technology, Chair for Design Automation, 80333 Munich, Germany, and also with the Software Competence Center Hagenberg GmbH (SCCH), 4232 Hagenberg, Austria.

E-mail: robert.wille@tum.de

I. INTRODUCTION

Railway networks are an important part of both public transportation and logistics, e.g., to reduce carbon emissions compared to cars and planes. Unlike road traffic, trains have relatively long braking distances. Hence, trains cannot operate on sight due to safety constraints. Moreover, switches have to be safely set for trains to turn. Because of that, a control system is needed for efficient and safe rail traffic. Signaling systems are essential to allow or deny a train to use a specific track part. For this, many national train control systems have been developed. The European Union Agency for Railways has listed about 40 systems alone in Europe [1]. In Germany, for example, *PZB 90* (intermittent train protection) and *LZB* (continuous train control) are widely used [2]. For reasons of interoperability, these national control systems are unified in the *European Train Control System* (ETCS) [3], [4]. It is specified in the *Control Command and Signalling Technical Support Instrument* (CCS TSI) [5], in particular the set of specifications listed in its appendix A [6]. Even for metro lines, a radio-based system analog to ETCS (Level 2) is being implemented, namely the *Communication Based Train Control* (CBTC) [7], [8]. Similarly, major standardized systems have been implemented in China (*Chinese Train Control System*, CTCS) and North America (*Positive Train Control*, PTC) [9].

A. European Train Control System

Currently, most railway systems in the world rely on block signaling, where the whole railway network is divided into blocks equipped with means of train detection to determine whether a given block is currently occupied by a train. Within a block, at most, there can only be one train at a time [11]. This paper focuses on ETCS, even though the model and key components likely carry over to other modern control systems. ETCS can be implemented in various levels [3], [4]. Previously, the specifications distinguished three levels, however with the CCS TSI 2023 [5] Level 2 and Level 3 have been combined.

Level 1 (L1) relies on signals and hardware on the tracks (e.g., Eurobalises and Euroloops), which transmit position and signaling information at discrete points (Eurobalise) or predefined intervals (Euroloop). The trackside fully handles the detection of block occupations (e.g., using axle counters), see Figure 1a, in which the following train slows down because it receives the moving authority into TTD2 only at the next Eurobalise.

In *Level 2* (L2), a permanent GSM-R connection between the train and the control system exists. By doing so, the train can continuously receive the maximal permitted speed. Signals are now optional but are often included as a fallback system. The train receives its moving authority through the GSM-R connection and is no longer bound to the discrete positions of Eurobalises.

At the same time, train location and integrity can be managed at different parts of the control system. Classically, reporting block sections as free works as in L1. However, due to the difference mentioned above in transmitting the moving authority, the trains in Figure 1b can follow each other more closely.

The block section borders are equipped with *Trackside Train Detection* (TTD) systems, e.g., axle counters, called TTD sections. Thus far, those blocks have often been defined by geography and economic considerations or as a trade-off between efforts to install axle counters and possible benefits. Because of this, the length of the corresponding TTD sections ranges from some meters (e.g., around turnouts) or some hundred meters (e.g., at

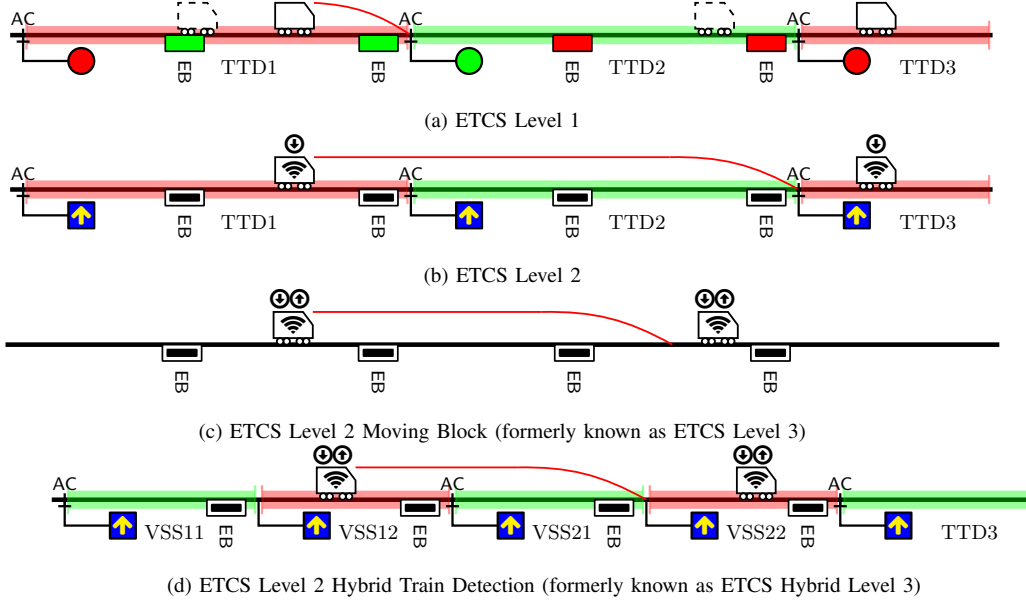


Fig. 1: Schematic drawings of various ETCS levels by Engels et al. [10] published under a CC BY 4.0 license

train stations) to several kilometers (e.g., in remote areas). Of course, this significantly affects the efficiency of the underlying railway networks.

With the introduction of *Moving Block*, formerly known as ETCS Level 3, those principles have changed for the first time since the 19th century. The train reports the exact position itself, and no trackside detection is necessary. For this, trains must be equipped with an integrity system that detects if a train is still complete. Only in this case can it safely release parts of the track. Rather than relying on fixed blocks, trains can follow at their braking distances since no hardware predefines any blocks, see Figure 1c. However, in practice, such systems impose new challenges [12].

Hence, Level 2 can also be used with a compromise of both classical block signaling and moving blocks. In *Level 2 Hybrid Train Detection* (L2 HTD), formerly known as Hybrid Level 3, TTD sections that already exist are divided into smaller *Virtual Subsections* (VSS), which do not require physical axle counters anymore and, hence, allow for a much higher degree of freedom in the utilization of existing railway networks, see Figure 1d. Independently of trackside hardware, the occupation of these virtual subsections is observed by the position and integrity data obtained from the trains [13].

B. Related Work

This additional degree of freedom allows for shorter train following times and, thus, more trains on existing railway tracks. At the same time, new design tasks arise that require automated methods to be helpful for designers of modern railway networks. However, thus far, design automation methods for ETCS, on the one hand, mainly focus on blocks characterized by hardware (e.g., axle counters). Because of this, they use general measures of capacity, energy efficiency, or economic efficiency independently from a specific schedule to propose a signaling

layout [14], [15], [16], [17], [18], [19], [20], [21], [22]. Thus, the respective solution is expected to work well on various schedules. On the other hand, train routing was considered separately, assuming the block layout had already been decided upon. Approaches are mainly separated in macroscopic routing [23] and track allocation on a microscopic level, i.e., considering the constraints imposed by the control system and sections. Track allocation was considered in general [24], [25], [26] and specifically focusing on railway stations [27], [28], [29].

This research has been extended to future signaling systems, such as ETCS L2 HTD and Moving Block control systems. Headways on such railway lines are different to former theory leading to adaptations in the blocking-time theory [30]. Besides that, recent research by Schlechte et al. optimizes time tables on dedicated moving block systems [31], which was later extended to use lazy constraints to speed up computation [32], [33]. However, implementing a moving block system in practice is significantly harder than using virtual subsections, in which case only the design but not the control logic changes [12].

While (almost) all of these methods can also be applied for ETCS L2 HTD, because virtual subsections and TTD sections have essentially the same consequences in final operation (assuming no system fails), they are not utilizing the additional degree of freedom introduced by virtual subsections. Using ETCS L2 HTD, changing a previous block layout defined by TTD sections even *after* building the network is now possible. In particular, it might be reasonable to consider a desired schedule directly because changing a block layout with a new schedule is more manageable than before. However, corresponding joint considerations of infrastructure planning and scheduling are just at the beginning.

For ETCS L2 HTD, specifically, the contributions to the “Hybrid ERTMS/ETCS Level 3 Case Study” provided an step in this direction [34]. Here, the main goal was to detect potential ambiguities in the specification to fix them in later versions. Submitted papers do this using iUML-B, Event-B, Formal B, Electrum, SysML/KAOS, and SPIN [34]. However, while these formalizations provided initial (non-ambiguous) interpretations of the ETCS L2 HTD, no precise design tasks, e.g., how to generate new layouts, can be derived from that. To the best of our knowledge, only research proposed in [35], [36], [10], [37] explicitly considered the automatic design of hybrid train detection systems based on specific timetable requests. However, even those papers specified the design tasks at a relatively high level, and neither formal descriptions nor results on the computational complexity of those tasks have been provided.

Overall, all related work on developing design methods for ETCS L2 HTD has been conducted without a clear definition nor a formal understanding of the complexity of the underlying task. As a consequence, we currently have heuristics that usually generate non-optimal¹ results, on the one hand, [36] and exact solutions which significantly suffer from huge run-times and severely limited scalability on the other [35], [10], [37], but do not know whether this is caused by insufficient methodology or complexity reasons. Similar scheduling problems (e.g., [24]) have been proven to be NP-hard by using the alternative graph introduced in [38]. At the same time, we are not aware of formal complexity results on the design tasks considered in [35], [36], [10], [37].

¹with respect to some objective describing the operational outcome, see also Section III

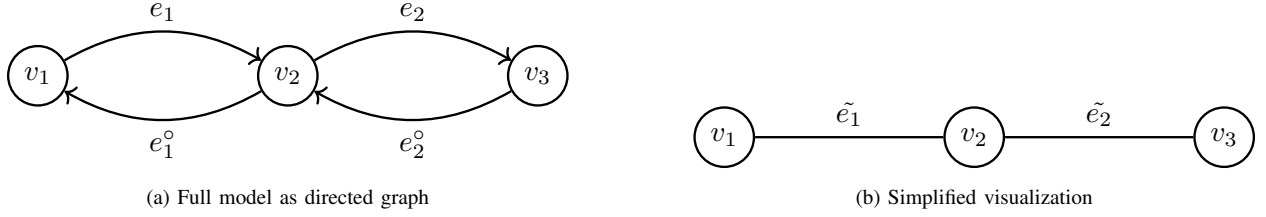


Fig. 2: Graphs representing a single track

In this paper, we aim to shed light on this. To this end, we first introduce a graph model and review the relevant concepts of ETCS L2 HTD on this in Section II. Afterward, we discuss three essential design tasks resulting from the new degrees of freedom and provide clear, i.e., formal, problem definitions and examples in Section III. This includes proofs showing that these new design tasks remain complex, i.e., are NP-complete or NP-hard, respectively (even if some restrictions are added). By this, we provide a proper and valuable theoretical foundation (both on the problem description and its complexity) for upcoming solution-oriented research in this area. Arising methods for the automated design of such systems will be made available within the *Munich Train Control Toolkit* (MTCT) hosted on GitHub (<https://github.com/cda-tum/mtct>).

II. PRELIMINARIES

In this section, we provide a basic definition of railway networks using graph terminology, which, step by step, is extended to cover the relevant properties. Within this, we define trains, their routes, and components related to the train control system. Based on that, in Section III we formalize the design tasks, which might arise in the context of ETCS L2 HTD.

A. Basic Definition of Railway Networks

In the following, we derive a definition for a railway network and valid movements on it that provide the basis for formulating corresponding design and verification tasks. For this, we introduce a directed graph $G = (V, E)$ together with further properties and constraints. In a straightforward and abstract view, edges E represent tracks that connect to the nodes V representing certain *points of interests* of the network (e.g., turnouts, crossings, signals, and stations). While railway tracks do not have a direction, a notion of direction will be important when introducing trains to prevent corner cases that are cumbersome to exclude in an undirected setting. Unless special operational constraints limit the direction on a track, we add edges for both directions; see Figure 2a representing a single track on which trains can move from left to right and from right to left. However, as said, edges (v_1, v_2) and (v_2, v_1) , for example, represent the same track segment. For edge $e = (u, v) \in E$, we denote its reverse edge $e^\circ = (v, u)$ if existent. In particular, e and e° represent the same track segment and $(e^\circ)^\circ = e$.

Remark II.1. To simplify notation and visualization, we let \tilde{e} denote both e and e° and visualize this by an undirected edge. Hence, the graph in Figure 2a simplifies to Figure 2b. Whenever we use an undirected edge \tilde{e} , it should be clear from the context to which of the two directed edges we refer. While this notation is rather imprecise,

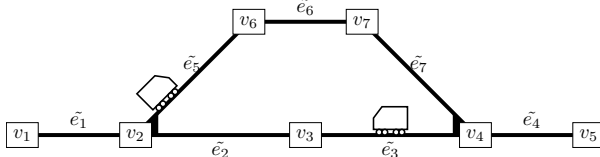


Fig. 3: Siding as a graph

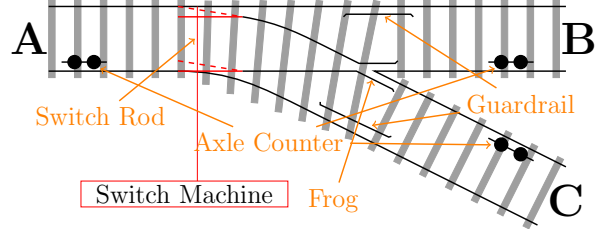


Fig. 4: Simple turnout

we believe it will improve the readability of some concepts as long as we bear in mind that they represent two edges simultaneously.

Of course, an edge $e \in E$ has a natural length, e.g., in meters, which we denote by a mapping $\text{len}: E \rightarrow \mathbb{R}_{>0}$. Note that $\text{len}(e) = \text{len}(e^\circ)$ since they represent the same track segment.

Example II.2. Fig. 3 shows a simple railway layout with a siding. Because of this, it is possible that two trains driving in opposite directions can pass each other. For example, the train driving from v_1 to v_5 can use the turnout at v_2 via v_6 and v_7 and come back to the main track using turnout v_4 and, then, drive towards v_5 . The second train, driving from v_5 to v_1 , can drive via v_4 , v_3 , and v_2 to its final destination v_1 . In this example, nodes v_2 and v_4 represent simple turnouts.

Turnouts usually do not allow trains to arbitrarily switch between the conjoining tracks. Instead, the possible ways over the turnout depend (besides others), e.g., on the direction the train is coming from. More precisely, Figure 4 shows a schematic drawing of a simple turnout as a standard right-hand switch with all its details. A train coming from the left-hand side (marked with an **A**) can take two ways: either to **B** or **C**. In contrast, a train coming from the right-hand side (either from **C** or **B**) can take only one way, namely to **A**. These restrictions on movements are not yet incorporated within the described graph framework. Hence, we introduce a successor function $s_v: \delta^{in}(v) \rightarrow \mathcal{P}(\delta^{out}(v))$ for every vertex $v \in V$, where $\delta^{in}(v)$ and $\delta^{out}(v)$ are the sets of edges entering and leaving v respectively. Then, $s_v(e_{in})$ represents the possible successor edges when entering v from a certain edge $e_{in} \in \delta^{in}(v)$.

Example II.3. In the case of Example II.2, the successor functions $\{s_v\}_{v \in V}$ are defined by

$$s_{v_3}(e) = \begin{cases} \{\tilde{e}_3\}, & \text{if } e = \tilde{e}_2 \\ \{\tilde{e}_2\}, & \text{if } e = \tilde{e}_3 \end{cases} \quad s_{v_7}(e) = \begin{cases} \{\tilde{e}_7\}, & \text{if } e = \tilde{e}_6 \\ \{\tilde{e}_6\}, & \text{if } e = \tilde{e}_7 \end{cases} \quad (1)$$

$$s_{v_6}(e) = \begin{cases} \{\tilde{e}_6\}, & \text{if } e = \tilde{e}_5 \\ \{\tilde{e}_5\}, & \text{if } e = \tilde{e}_6 \end{cases} \quad s_{v_1}(\tilde{e}_1) = s_{v_5}(\tilde{e}_4) = \emptyset \quad (2)$$

for the nodes corresponding to straight tracks without any turnouts and

$$s_{v_2}(e) = \begin{cases} \{\tilde{e}_2, \tilde{e}_5\}, & \text{if } e = \tilde{e}_1 \\ \{\tilde{e}_1\}, & \text{if } e \in \{\tilde{e}_2, \tilde{e}_5\} \end{cases} \quad s_{v_4}(e) = \begin{cases} \{\tilde{e}_3, \tilde{e}_7\}, & \text{if } e = \tilde{e}_4 \\ \{\tilde{e}_4\}, & \text{if } e \in \{\tilde{e}_3, \tilde{e}_7\} \end{cases} \quad (3)$$

for the functions corresponding to turnouts.

In practice, a considered railway network will be connected to neighboring networks that are out of scope for the respective design tasks. Hence, trains will enter and exit the network under consideration at these borders.

Definition II.4 (Border Vertices). *The set of border vertices is denoted by $\mathcal{B} \subseteq V$ and all $v \in \mathcal{B}$ fulfill $|\Gamma(v)| = 1$, where $\Gamma(\cdot)$ denotes the neighboring vertices, i.e., they cannot be in the interior of the network. It is assumed that trains can only enter and exit the network with predefined headway times $h: \mathcal{B} \rightarrow \mathbb{R}_{\geq 0}$.*

Example II.5. *In the case of Example II.2, $\mathcal{B} = \{v_1, v_5\}$ could hold with $h(v_1) = h(v_5) = 120s$.*

Using the above, we can now define a simple railway network as follows:

Definition II.6 (Railway Network). *A railway network $N = (G, \text{len}, \{s_v\}_{v \in V})$ is defined by*

- *a directed graph $G = (V, E)$ with vertices V being the set of points of interest and edges E being the railway tracks between the aforementioned points of interest,*
- *a mapping $\text{len}: E \rightarrow \mathbb{R}_{>0}$ denoting the length of each edge such that $\text{len}(e) = \text{len}(e^\circ)$ for every pair of edges $e, e^\circ \in E$,*
- *a family of mappings $\{s_v\}_{v \in V}$, where $s_v: \delta^{\text{in}}(v) \rightarrow \mathcal{P}(\delta^{\text{out}}(v))$ represents the valid movements over v , and*
- *border vertices $\mathcal{B} \subseteq V$ together with headway times $h: \mathcal{B} \rightarrow \mathbb{R}_{\geq 0}$.*

However, railway networks do not only consist of turnouts where just three tracks are connected. For example, they might also cross each other. Reasons for this might be double junctions, where simple *crossings* occur, as depicted in Figure 5 or *slip switches* as depicted in Figure 6. The underlying graph structure is the same; however, the respective successor functions differ. In general, tracks that cross each other can be modeled as shown in Figure 7, where node v_c corresponds to the crossing point.

In the case of a simple *crossing*, e.g., where the green and yellow lines cross each other in Figure 5, a train can move from v_1 (v_2) via v_c to v_3 (v_4) or vice versa, but it is not possible that a train moves from v_1 via v_c to v_2 or v_4 . To model this movement restriction, the corresponding successor function s_{v_c} is given by

$$s_{v_c}(\tilde{e}_1) = \{\tilde{e}_3\}, \quad s_{v_c}(\tilde{e}_2) = \{\tilde{e}_4\}, \quad (4)$$

$$s_{v_c}(\tilde{e}_3) = \{\tilde{e}_1\} \quad \text{and} \quad s_{v_c}(\tilde{e}_4) = \{\tilde{e}_2\}. \quad (5)$$

On the other hand, *double slip switches* as, e.g., shown in Fig. 6 are quite common in the entry to bigger stations as their usage saves place, which is normally very limited in such. Their switch machines make changing the track

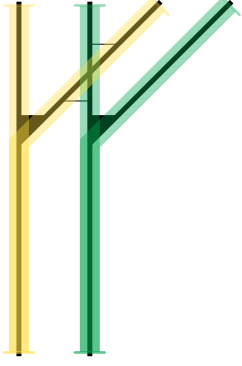


Fig. 5: Double junction, or level junction



Fig. 6: Double slip switch at Munich Central Station by Láczy [39] published under a CC BY-SA 2.0 license

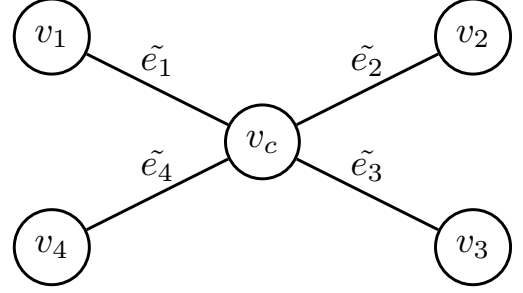


Fig. 7: Graph of crossing

from one side to the other possible. In this case, the successor function at v_c changes to

$$s_{v_c}(e) = \begin{cases} \{\tilde{e}_2, \tilde{e}_3\}, & \text{if } e \in \{\tilde{e}_1, \tilde{e}_4\} \\ \{\tilde{e}_1, \tilde{e}_4\}, & \text{if } e \in \{\tilde{e}_2, \tilde{e}_3\} \end{cases}. \quad (6)$$

Likewise, a *single slip switch* looks similar to double slip switches but allows fewer movements, i.e., for 2 out of 4 entry points, there is no possibility to change the track. More precisely, we can model this using

$$s_{v_c}(\tilde{e}_1) = \{\tilde{e}_2, \tilde{e}_3\}, \quad s_{v_c}(\tilde{e}_3) = \{\tilde{e}_1\}, \quad (7)$$

$$s_{v_c}(\tilde{e}_2) = \{\tilde{e}_1, \tilde{e}_4\} \quad \text{and} \quad s_{v_c}(\tilde{e}_4) = \{\tilde{e}_2\}. \quad (8)$$

These examples demonstrate that any real-world railway network can be modeled using Definition II.6. Even if further not-mentioned points of interest occur, they can be represented analog to the above examples.

Note that the above definition is somewhat similar to the double-vertex graph introduced by Montigel [40] even though the successor functions are included in the double-vertices. At the same time, the choice to use these successor functions is similar to how valid train movements are modeled in [41]. Because of how their specific MIP model is constructed, they do not require directed edges, which we believe is vital in our solution-independent description. Hence, the necessity to define train schedules, valid movements, and the possibility to add new sections differs from their work.

In principle, it is not important which of the above infrastructure models is used. We chose successor functions because we find this graph model to be more intuitive for the purpose of this work. The upcoming design tasks could easily also be formulated on, e.g., Montigel's double-vertex graph model[40].

Using the above definition, we can also formally define valid track sequences, which could be traversed by a train and, hence, establish a connection between the edges of the graph and its successor functions.

Definition II.7 (Track Sequence). *For some $k \in \mathbb{N}$, a track sequence is a k -tuple $(e_1, \dots, e_k) \in E^k$ ($e_i = (v_i, w_i)$) such that $v_{i+1} = w_i$ for all $i = 1, \dots, k-1$. The track sequence is valid if, and only if, $e_{i+1} \in s_{w_i}(e_i)$ for all $i = 1, \dots, k-1$.*

Example II.8. *Consider again the network depicted in Figure 3. Assume that all edges e_i are facing right and e_i° are facing left. Then, (e_2°, e_5, e_6) is a track sequence since they form a path in the graph. However, the movement from e_2° to e_5 is forbidden by the successor function describing the logic of turnout v_2 . Hence, the track sequence is not valid. On the other hand, (e_1, e_5, e_6) is a valid track sequence.*

Finally, observe that the above description does not model reality up to every detail. For example, the defined model cannot currently model dead-end stations. For this, trains must be allowed to turn around on certain edges after a predefined waiting time. However, we believe we have covered the most essential properties described above. Hence, it is reasonable to develop further work on the presented formulation first and, if they turn out promising, add further constraints afterward.

B. Trains and Their Schedules

Until now, we have focused on modeling the railway network without considering the trains and their movements. In this section, we define trains as well as their movements and schedules.

Definition II.9 (Train). *A train $\text{tr} := (l^{(\text{tr})}, v_{\max}^{(\text{tr})}, a^{(\text{tr})}, d^{(\text{tr})}, \text{tim}^{(\text{tr})})$ is a tuple, where*

- $l^{(\text{tr})} \in \mathbb{R}_{>0}$ defines its length and,
- $v_{\max}^{(\text{tr})}$ defines its maximal possible speed,
- $a^{(\text{tr})}$ and $d^{(\text{tr})}$ denote the maximal acceleration and deceleration respectively, and
- $\text{tim}^{(\text{tr})} \in \{\text{true}, \text{false}\}$ denotes if tr is equipped with train integrity monitoring (TIM).

Train lengths do not necessarily coincide with the length of edges within a railway network $N = (G, \text{len}, \{s_v\}_{v \in V})$. In particular, trains might only be located on parts of an edge. For this, we denote a *track interval* $\text{iv} := (e, \lambda, \mu) \in E \times [0, 1] \times [0, 1]$ such that $\lambda \leq \mu$, where λ denotes the start and μ the end of the section in relative terms.

Definition II.10 (Track Range). *A track range $\text{rg} := (s_{\text{in}}, \text{iv}_1, \dots, \text{iv}_k, s_{\text{out}})$ is then given by a series of track intervals $\text{iv}_i = (e_i, \lambda_i, \mu_i)$ together with $s_{\text{in}}, s_{\text{out}} \geq 0$. In this context, s_{in} and s_{out} denote the parts of the train that are outside of the network at the rear (incoming vertex) and front (outgoing vertex), respectively. The following conditions must hold by definition:*

- 1) (e_1, \dots, e_k) is a valid track sequence according to Definition II.7,
- 2) $\lambda_i = 0$ for every $1 < i \leq k$,
- 3) $\mu_i = 1$ for every $1 \leq i < k$,
- 4) $s_{\text{in}} \neq 0 \Rightarrow e_1 = (u, v)$ for some $u \in \mathcal{B}$ and $\lambda_1 = 0$, and,
- 5) $s_{\text{out}} \neq 0 \Rightarrow e_k = (u, v)$ for some $v \in \mathcal{B}$ and $\mu_k = 1$.

Naturally, the lengths of the edges carry over to track intervals and ranges by setting $\text{len}(\text{iv}) := (\mu - \lambda) \cdot \text{len}(e)$ and

$$\text{len}(\text{rg}) := s_{in} + s_{out} + \sum_{i=1}^k \text{len}(\text{iv}_i) = s_{in} + s_{out} + \sum_{i=1}^k (\mu_i - \lambda_i) \cdot \text{len}(e_i). \quad (9)$$

If $k \geq 2$, then we can also write

$$\text{len}(\text{rg}) = s_{in} + (1 - \lambda_1) \cdot \text{len}(e_1) + \sum_{i=2}^{k-1} \text{len}(e_i) + \mu_k \cdot \text{len}(e_k) + s_{out}. \quad (10)$$

At this point, it would be intuitive to define the position of a train at a given time as a track range rg with $\text{len}(\text{rg}) = l^{(\text{tr})}$. However, it is convenient to use a slightly different definition, which later makes it easier to properly define separation trains (or headways) of different trains with respect to each other. Note that the entire distance needed to brake has to be already cleared from other trains. Hence, one could say that a train does not only occupy the track parts on which it is physically present, but also all of its braking distance ahead.

Definition II.11 (Train Route and Occupation). *For a given train tr , a train route $R_{[a,b]}^{(\text{tr})} := \left(\text{rg}_t^{(\text{tr})} \right)_{t \in [a,b]}$ is a series of track ranges for every time $t \in [a,b]$, such that $\text{len}(\text{rg}_t^{(\text{tr})}) = l^{(\text{tr})} + \text{bd}_t^{(\text{tr})}$, where $\text{bd}_t^{(\text{tr})}$ denotes the distance tr requires to come to a full stop if it were to start decelerating at time t . In this context $\text{rg}_t^{(\text{tr})}$ is bound by the trains rear end on the one and its moving authority on the other side.*

Given some track range $\text{rg} := (s_{in}, (e_1, \lambda_1, \mu_1), \dots, (e_k, \lambda_k, \mu_k), s_{out})$, we denote

$$\text{occ}(\text{rg}) := \{e_1, \dots, e_k\} \quad (11)$$

to be the occupied track segments. Naturally

$$\text{occ}_t^{(\text{tr})} := \text{occ}(\text{rg}_t^{(\text{tr})}) \quad (12)$$

corresponds to the segments occupied by train tr at time $t \in [a,b]$.

As of now, trains can theoretically move and jump around a railway network arbitrarily because the different time steps are not linked with each other. Consider two track ranges $\text{rg}^{(1)} = (s_{in}^{(1)}, (e_1^{(1)}, \lambda_1^{(1)}, \mu_1^{(1)}), \dots, (e_l^{(1)}, \lambda_l^{(1)}, \mu_l^{(1)}), s_{out}^{(1)})$ and $\text{rg}^{(2)} = (s_{in}^{(2)}, (e_1^{(2)}, \lambda_1^{(2)}, \mu_1^{(2)}), \dots, (e_k^{(2)}, \lambda_k^{(2)}, \mu_k^{(2)}), s_{out}^{(2)})$. They are said to *intersect (in order)* if there exists an $s \in \{1, 2, \dots, \min\{l, k\}\}$ such that $e_{l-s+i}^{(1)} = e_i^{(2)}$ for all $i \in \{1, \dots, s\}$, i.e., the end of $\text{rg}^{(1)}$ and start of $\text{rg}^{(2)}$ overlap. We then define

$$\text{dist}(\text{rg}^{(1)}, \text{rg}^{(2)}) := \underbrace{(s_{in}^{(1)} - s_{in}^{(2)})}_{\text{distance outside network}} + \overbrace{\sum_{i=1}^{l-s} (\mu_i^{(1)} - \lambda_i^{(1)}) \text{len}(e_i^{(1)})}^{\text{length of non-overlapping edges of } \text{rg}^{(1)}} + \underbrace{(\lambda_1^{(2)} - \lambda_{l-s+1}^{(1)}) \text{len}(e_{l-s+1}^{(1)})}_{\text{length on overlapping edge}} \quad (13)$$

to be their *distance*. Using this, we can characterize the varying speed of a train at time t by the following definition.

Definition II.12 (Train Speed). *A train $\text{tr} := \left(l^{(\text{tr})}, v_{max}^{(\text{tr})}, \left(\text{rg}_t^{(\text{tr})} \right)_{t \geq 0} \right)$ is said to respect its maximal speed if for every $t \geq 0$ and $0 < \varepsilon < \frac{l^{(\text{tr})}}{v_{max}^{(\text{tr})}}$ it holds that $\text{rg}_t^{(\text{tr})}$ and $\text{rg}_{t+\varepsilon}^{(\text{tr})}$ intersect in order with*

$$\text{dist}(\text{rg}_t^{(\text{tr})}, \text{rg}_{t+\varepsilon}^{(\text{tr})}) \leq \varepsilon \cdot v_{max}^{(\text{tr})}. \quad (14)$$

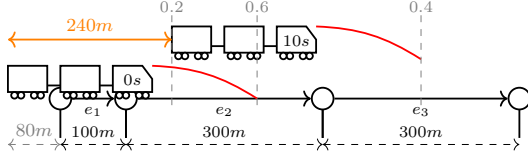


Fig. 8: A train tr at two overlapping time steps t and $t + \varepsilon$

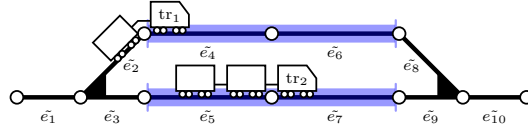


Fig. 9: Example train station

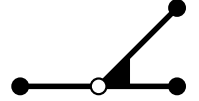


Fig. 10: Turnout

Its speed at time t is then given by

$$v_t^{(tr)} := \frac{d}{d\varepsilon} \left[\text{dist} \left(\text{rg}_t^{(tr)}, \text{rg}_{t+\varepsilon}^{(tr)} \right) \right]_{\varepsilon=0} = \lim_{\varepsilon \rightarrow 0} \frac{\text{dist} \left(\text{rg}_t^{(tr)}, \text{rg}_{t+\varepsilon}^{(tr)} \right)}{\varepsilon}. \quad (15)$$

Moreover, the change in speed is bound by

$$-d^{(tr)} \leq \frac{d}{dt} \left(v_t^{(tr)} \right) \leq a^{(tr)}. \quad (16)$$

A train respects its braking distance if for every $t \geq 0$

$$\text{bd}_t^{(tr)} \geq \frac{v_t^{(tr)}}{2 \cdot d^{(tr)}}. \quad (17)$$

By definition, $t \in \mathbb{R}$, so that train velocity and occupation are defined continuously. When solving design tasks proposed in Section III, time might or might not have to be discretized depending on the applied algorithm. However, this does not affect the problem description itself, which assumes continuous time and is independent of proposed solving strategies.

Of course, further details of train movements can be modeled within this framework as additional constraints. For example, the train's acceleration and deceleration properties might depend on its current speed and/or the underlying track's gradient. Hence, braking curves defined in the official specifications can also be considered [42] by altering Equations (16) and (17) respectively. Usually, trains should not face red signals when planning their movements. Such a requirement could, e.g., be included as further constraints on the speed to disallow stopping on an open line.

Example II.13. Consider Figure 8 with an 210m-long train occupying $(80, (e_1, 0, 1), (e_2, 0, 0.6), 0)$, which moved to $(0, (e_2, 0.2, 1), (e_3, 0, 0.4))$ within 10 seconds (we have $\text{bd}_{0s}^{(tr)} = \text{bd}_{10s}^{(tr)} = 150m$). The distance $\text{dist} \left(\text{rg}_{0s}^{(tr)}, \text{rg}_{10s}^{(tr)} \right)$ is given by $\underbrace{(80m - 0m)}_{s_{in}^{(0s)} - s_{in}^{(10s)}} + \underbrace{(1 - 0)}_{\mu_1^{(0s)} - \lambda_1^{(0s)}} \cdot \underbrace{100m}_{\text{len}(e_1)} + \underbrace{(0.2 - 0)}_{\lambda_1^{(10s)} - \lambda_2^{(0s)}} \cdot \underbrace{300m}_{\text{len}(e_2)} = 240m$. Hence, we can approximate its speed by $\frac{240m}{10s} = 24 \frac{m}{s}$.

Of course, a train does not drive anywhere on a network but has a fixed set of stations to visit or possibly even a fixed schedule. We denote a *train station* to be a subset of edges $S \subseteq E$. A train tr is said to be *in* station S at time t if $\text{occ}_t^{(tr)} \subseteq S$. It *stops* in S if additionally $v_t^{(tr)} = 0$. For example, a station can include all tracks of a certain real-world train station. If necessary, we can also restrict it to only cover a subset, e.g., if some platforms should be used solely by regional and others only by long-distance trains.

Example II.14. Consider Figure 9, a station $S = \{\tilde{e}_4, \tilde{e}_5, \tilde{e}_6, \tilde{e}_7\}$, and the two trains depicted. Here, tr_1 occupies \tilde{e}_2 and \tilde{e}_4 . Since $\tilde{e}_2 \notin S$, tr_1 is not in the train station. On the other hand, tr_2 occupies $\tilde{e}_5 \in S$ and $\tilde{e}_7 \in S$ and, hence, is in the train station.

Combining the above and adding time constraints, we are now ready to define train schedules.

Definition II.15 (Timetable Requests). Let tr_1, \dots, tr_m be m trains on a railway network N . For every train tr_i , a timetable request consists of

- an entry node $v_{in}^{(tr_i)}$ together with a desired arrival interval $[\underline{t}_{in}^{(tr_i)}, \bar{t}_{in}^{(tr_i)}]$,
- an exit node $v_{out}^{(tr_i)}$ together with a desired departure interval $[\underline{t}_{out}^{(tr_i)}, \bar{t}_{out}^{(tr_i)}]$, as well as,
- a list of m_i stations $S_1^{(tr_i)}, \dots, S_{m_i}^{(tr_i)} \subseteq E$ together with
 - desired arrival intervals $[\underline{\alpha}_1^{(tr_i)}, \bar{\alpha}_1^{(tr_i)}], \dots, [\underline{\alpha}_{m_i}^{(tr_i)}, \bar{\alpha}_{m_i}^{(tr_i)}]$,
 - desired departure intervals $[\underline{\delta}_1^{(tr_i)}, \bar{\delta}_1^{(tr_i)}], \dots, [\underline{\delta}_{m_i}^{(tr_i)}, \bar{\delta}_{m_i}^{(tr_i)}]$, and,
 - minimal stopping times $\Delta t_1^{(tr_i)}, \dots, \Delta t_{m_i}^{(tr_i)} \geq 0$.

A train tr_i respects its timetable request if,

- it enters the network at node $v_{in}^{(tr_i)}$ at some time $t_{in} \in [\underline{t}_{in}^{(tr_i)}, \bar{t}_{in}^{(tr_i)}]$, i.e., $s_{in}^t = \text{len}(tr_i)$ and $e_{in} \in \text{occ}_{t+dt}^{(tr_i)}$ for all $0 < dt < \varepsilon$ for some $\varepsilon > 0$, where $e_{in} \in \delta^{out}(v_{in}^{(tr_i)})$,
- it leaves the network at node $v_{out}^{(tr_i)}$ at some time $t_{out} \in [\underline{t}_{out}^{(tr_i)}, \bar{t}_{out}^{(tr_i)}]$, i.e., $s_{out}^t = \text{len}(tr_i)$ and $e_{out} \in \text{occ}_{t-dt}^{(tr_i)}$ for all $0 < dt < \varepsilon$ for some $\varepsilon > 0$, where $e_{out} \in \delta^{in}(v_{out}^{(tr_i)})$, and,
- for every $j \in \{1, \dots, m_i\}$, there exists an interval $[\underline{t}_j^{(tr_i)}, \bar{t}_j^{(tr_i)}]$ with $\bar{t}_j^{(tr_i)} - \underline{t}_j^{(tr_i)} \geq \Delta t_j^{(tr_i)}$, $\underline{t}_j^{(tr_i)} \in [\underline{\alpha}_j^{(tr_i)}, \bar{\alpha}_j^{(tr_i)}]$, and $\bar{t}_j^{(tr_i)} \in [\underline{\delta}_j^{(tr_i)}, \bar{\delta}_j^{(tr_i)}]$, such that tr_i stops in $S_j^{(tr_i)}$ for every $t \in [\underline{t}_j^{(tr_i)}, \bar{t}_j^{(tr_i)}]$.

C. Sections and Virtual Subsections in Railway Networks

As mentioned, trains cannot drive independently from each other, and running into each other is an unwanted situation. For this, a control system ensures that a block is occupied by at most one train, which still needs to be defined formally. To this end, *Trackside Train Detection* (TTD) systems, e.g., axle counters, are used to collect corresponding information on the occupation of blocks. Accordingly, corresponding blocks are frequently also called TTD sections.

Unfortunately, an edge $e \in E$ of the graph definition from above does not match a TTD section. A TTD section starts and ends with, e.g., axle counters. Three axle counters are usually used for a turnout: one for each of the three tracks that meet at the turnout (see also Figure 4). In this case, the switch's three-track segments (edges) belong to the same block. To model such TTD sections, we introduce another helping function on the vertices, also called border function, $C_V: V \rightarrow \{0 = \text{no border}, 2 = \text{TTD-Border}\}$, which indicates if a certain vertex is equipped with trackside train detection and, thus, corresponds to a border between two TTD sections. A simple turnout would typically correspond to one central node of value 0 and three outer nodes of value 2, in which case the respective edges will have a rather small length corresponding to the distance between the turnout's center and the axle counters. This is depicted in Figure 10: A black filled node \bullet visualizes a vertex which represents a

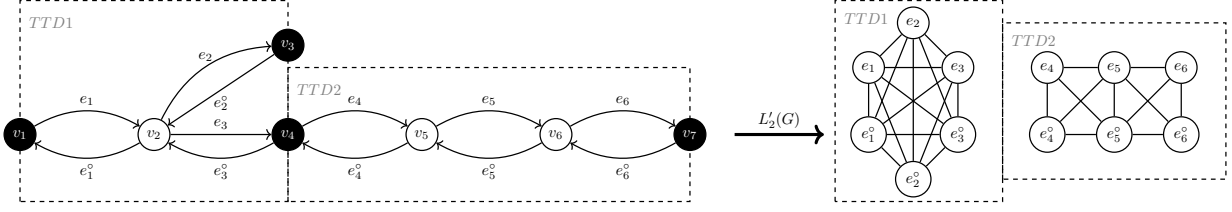


Fig. 11: TTD sections

TTD-Border, i.e., C_V evaluates to $2 = \text{TTD-Border}$. A white filled circle \circ visualizes a vertex which represents no TTD-Border, i.e., C_V evaluates to $0 = \text{no border}$. Abstractly, we refer to a vertex $v \in V$ with $C_V(v) = k$ to be a border of level k . Further, we define by $L'_k(G) := (V'_k, E'_k)$ an *undirected* graph interchanging the role of vertices and edges and additionally ignoring borders of level $\geq k$.

Formally, we define $V'_k := E$ and

$$E'_k := \{\{e_1, e_2\} \subseteq V'_k : e_2 = e_1^\circ \text{ or } \tilde{e}_1 \cap \tilde{e}_2 \cap C_K \neq \emptyset\}, \quad (18)$$

where $C_K := \{v \in V : C_V(v) < k\}$ are the nodes with border value less than k . Now, if two nodes within $L'_2(G)$ are connected, then their corresponding track segments are not separated by any TTD; hence, they fall within one TTD section. In particular, if we set

$$TTDs := \{T_1, \dots, T_m\} = CC(L'_2(G)), \quad (19)$$

where $CC(\cdot)$ denotes the connected components, then T_1, \dots, T_m correspond exactly to the TTD sections. Edges within a section $T_i \in TTDs$ are not separated by any TTD, whereas a TTD section $T_i \in T$ is separated from any other TTD section $T_j \in TTDs - \{T_i\}$ by corresponding borders. Naturally, this provides a partition of the edges of the original graph G .

Example II.16. Consider a graph with seven vertices as depicted in Figure 11. Vertices v_1 to v_4 represent a simple turnout as in Figure 4 connected to a straight track up to v_7 . As before, \bullet corresponds to a TTD border and \circ to no border. Figure 11 shows how the network transforms into $L'_2(G)$, which consists of two connected components. The vertices of each connected component (i.e., edges in the original graph) compose one TTD section. The corresponding sections are marked in both graphs in Figure 11.

With the introduction of hybrid train detection, sections do not have to be separated by TTDs. In particular, this allows splitting existing TTD sections into virtual subsections (VSS). To also model these, we add a VSS-Border element to the border function as follows: $C_V : V \rightarrow \{0 = \text{no border}, 1 = \text{VSS-Border}, 2 = \text{TTD-Border}\}$. In figures, we plot VSS borders using \oslash . Similarly to Equation (19) the VSS sections V_1, \dots, V_n are given by the connected components of $L'_1(G)$, i.e.,

$$\{V_1, \dots, V_n\} := CC(L'_1(G)). \quad (20)$$

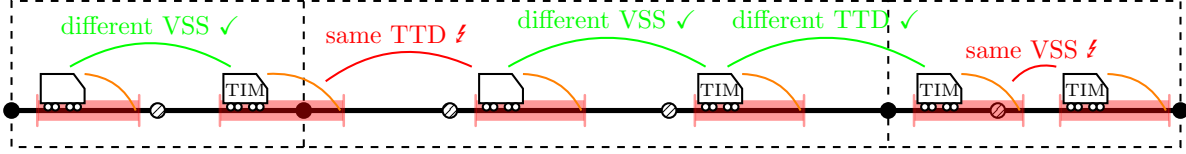


Fig. 12: VSS condition

A key concept in train control is the division of the tracks into multiple blocks. A block is only allowed to be occupied by at most one train. In the context of ETCS L2 HTD, these blocks are usually given by the VSS sections that function essentially analog to their TTD counterparts.

The concrete control logic depends on if a train is equipped with *train integrity monitoring* (TIM) or not. A train with TIM can safely reports its rear end, because it is determined with sufficient certainty that the train is still complete and has not lost any wagons. Because of this a VSS section can be safely reported as free without the need of any TTD hardware. Hence, any other train is permitted to enter this VSS that has just been cleared. On the other hand, a train without TIM can only safely report its front position. Thus, a VSS remains occupied until the corresponding TTD section is cleared by the trackside hardware. These cases are depicted in Figure 12, in which only 4 out of 6 trains are equipped with TIM.

Hence, the respective control condition can be formalized as follows:

Definition II.17 (VSS Condition). *Let V_1, \dots, V_n be the VSS-sections as defined in Equation (20) and tr_1, \dots, tr_m be the trains operating on the respective railway network. The train control system ensures the following condition for every $i \in \{1, \dots, n\}$, $j \in \{1, \dots, m\}$, $k \in \{1, \dots, m\} - \{j\}$, and $t \geq 0$:*

$$V_i \cap \text{occ}_t^{(tr_j)} \neq \emptyset \Rightarrow V_i \cap \text{occ}_t^{(tr_k)} = \emptyset. \quad (21)$$

Furthermore, let T_1, \dots, T_l be the TTD-sections as defined in Equation (19). For any $h \in \{1, \dots, l\}$, $j \in \{1, \dots, m\}$, and $t \geq 0$ with $T_h \cap \text{occ}_t^{(tr_j)} \neq \emptyset$ let

$$\hat{t}_h^{(tr_j)}(t) := \max \left\{ 0, \sup \left\{ \tau : 0 \leq \tau \leq t, T_h \cap \text{occ}_\tau^{(tr_j)} = \emptyset \right\} \right\}. \quad (22)$$

Using the above definition, for every $j \in \{1, \dots, m\}$ with $\text{tim}^{(tr_j)} = \text{false}$ as well as $k \in \{1, \dots, m\} - \{j\}$, $h \in \{1, \dots, l\}$, $i \in \{1, \dots, n\}$, $t \geq 0$ and $\tau \in [\hat{t}_h^{(tr_j)}(t), t]$ it has to additionally hold that

$$T_h \cap V_i \cap \text{occ}_\tau^{(tr_j)} \neq \emptyset \Rightarrow V_i \cap \text{occ}_t^{(tr_k)} = \emptyset. \quad (23)$$

Intuitively, in Definition II.17, Equation (21) ensures that only one VSS is occupied at any given time. $\hat{t}_h^{(tr_j)}(t)$ is the time at which tr_j entered (began to occupy) T_h . Equation (23) simply states that if a train without TIM occupies a TTD section T_h any VSS within T_h previously occupied by that train is not yet cleared.

Because the braking distance was already included in each train's occupation (see Definition II.11), the conditions in Definition II.17 ensure the safe distance required between following trains.

Remark II.18. *If all trains are equipped with TIM, only Equation (21) remains in Definition II.17. Using an ETCS L2 control with fixed TTD sections and no VSS, Definition II.17, again, simplifies to Equation (21) with T_1, \dots, T_h (TTD-sections) instead of V_1, \dots, V_n (VSS-sections), i.e., $T_i \cap \text{occ}_t^{(tr_j)} \neq \emptyset \Rightarrow T_i \cap \text{occ}_t^{(tr_k)} = \emptyset$.*

Overall, given a railway network N , a certain number of trains, and a list of timetable requests, the following properties are desired whenever considering a design task:

- Every train respects its maximal speed and braking distance (Definition II.12)
- Every train respects its timetable request (Definition II.15).
- The VSS condition is satisfied (Definition II.17).

Again, note that one can, in theory, model the behavior of the control system in as much detail as desired. For example, interlockings or time needed for turnouts to change their position could be considered using additional constraints. Nevertheless, we believe the level of detail provided in this paper suffices for defining relevant design tasks.

III. DESIGN TASKS AND THEIR COMPLEXITY

In this section, we define different design tasks that might occur in ETCS L2 HTD systems and could be valuable to automate. They are formulated independently from whichever solution method might be chosen to solve them. In Section III-A, we formulate a respective verification tasks. While the verification is arguably more of a theoretical nature, it provides the basis for the design tasks proposed in Sections III-B and III-C. Especially the optimization tasks in Section III-C, which use the added flexibility by HTD control systems to use specific timetable requests instead of general schedule-independent KPIs, are both, of practical interest and not yet well studied.

The proposed design tasks in the following subsections are formulated to help infrastructure planners to find optimal signaling layouts in terms of enabling the best operational outcome based on specific timetable requests. They do not yet consider robustness with respect to delays. For classical signaling systems capacity was already considered under disturbed conditions leading to real-time rescheduling problems [43]. Recently such rescheduling problems have also been considered for modern (moving block) control systems [44]. At the same time, previous work focuses on minimizing the effect of delays on a fixed signaling layout. An interesting extension to the presented design tasks would be to also consider stochastic distributions on expected delays. In that case one could aim to optimize the VSS layout to minimize the negative effects of expected rescheduling during operation. At the time of writing, the proposed design tasks are not well studied even in the deterministic case. Hence, for now, we focus on providing efficient solutions for the proposed design tasks. Once reasonable methods exist, adding expected stochastic delays would be a natural extension.

A. Verification of Timetable Requests on ETCS L2 HTD Layouts

Note that a desired timetable request might be impossible on a particular railway network since only one train can be located on each TTD/VSS section, a restricted track speed, or a limited train speed.

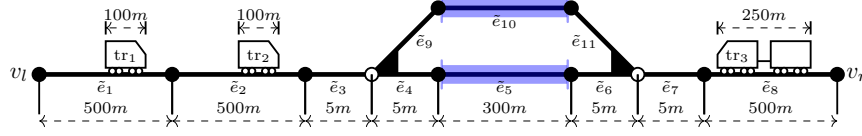


Fig. 13: Infeasible verification task

VERIFICATION OF TIMETABLE REQUESTS (TR-VER)

Input: A railway network $N = (G, \text{len}, \{s_v\}_{v \in V})$; border function $C_V: V \rightarrow \{0, 1, 2\}$; trains $\text{tr}_1, \dots, \text{tr}_m$; a list of timetable requests for tr_1 to tr_m .

Task: Do there exist train routes $R^{(\text{tr}_1)}, \dots, R^{(\text{tr}_m)}$ such that the VSS condition is fulfilled and all trains respect their speed and timetable requests? If yes, provide a certificate.

Example III.1. Consider the train network in Figure 13 and three trains tr_1 to tr_3 . As before, \bullet corresponds to a TTD border and \circ to no border. Also, we are given one real-world train station $S := \{\tilde{e}_5, \tilde{e}_{10}\}$, which is marked in blue. The timetable requests are specified as follows:

tr_1 : Entering at v_l at $t = 120$ (i.e., within $[120, 120]$), leaving at v_r at $t = 645$ (i.e., within $[645, 645]$), and stopping in S from $t = 240$ to $t = 300$ (i.e., arriving within $[120, 240]$, leaving within $[300, 645]$, and stopping for a minimal time of $\Delta t = 60$).

tr_2 : Entering at v_l at $t = 0$, leaving at v_r at $t = 420$, and stopping in S from $t = 120$ to $t = 300$.

tr_3 : Entering at v_r at $t = 0$, leaving at v_l at $t = 420$, and stopping in S from $t = 180$ to $t = 300$.

where time is measured in seconds. Observe that the station S only consists of two blocks; hence, a maximum of two trains can stop at a time. However, the timetable requests require all three trains to be stopped at the station at time $t = 270$, for example. Hence, there are no train routes such that the VSS condition is fulfilled, the trains respect their timetable requests, and the verification task returns infeasibility.

Observe that the routing on a fixed VSS layout might differ slightly from systems using trackside-oriented train detection, see also discussion around Definition II.17. This is because trains equipped with *train integrity monitoring* (TIM) can report VSS sections free independently of their corresponding TTD sections. Any train can follow closely. At the same time, trains without TIM need TTD hardware (e.g., axle counters) to report a section as free. Hence, a train can only follow if a non-TIM train has cleared an entire TTD section.

If all trains are equipped with TIM, the situation simplifies. In that case, there is no logical difference between TTD and VSS sections, and the block layout is fixed for the verification tasks defined above. Hence, TR-VER is essentially equivalent to the task on classical layouts in this (simplified) case. Scheduling under such classical signaling systems is known to be NP-hard in general [38], [24]. In particular, it is reasonable to assume that TR-VER has similar complexity. At the same time, there are minor differences to the scheduling problems considered so that the proof does not carry over formally. While the following complexity result is not unexpected, we formally include them for reasons of completeness. Moreover, it serves as a basis for the following design tasks in Sections III-B and III-C, which are tailored to ETCS L2 HTD.

Theorem III.2. *TR-VER is NP-complete even if all routes and exact stopping times in train stations are fixed and the railway network is planar.*

Proof. Given a certificate, we can check feasibility using the definitions in polynomial time; hence, TR-VER is in NP. To show NP-completeness, we reduce Monotone 3-SAT- $(\leq 2, \leq 2)$ to TR-VER. Monotone 3-SAT- $(\leq 2, \leq 2)$ are all SAT instances in which every clause contains three literals (all of which contain only unnegated or only negated variables), and every variable appears in clauses at most twice unnegated and twice negated. Monotone 3-SAT- $(\leq 2, \leq 2)$ is known to be NP-complete [45].

For reduction, we are given an instance of Monotone 3-SAT- $(\leq 2, \leq 2)$ on variables x_1, \dots, x_n defined by clauses $\mathcal{C} = \{C_1, \dots, C_m\}$, each consisting of three literals. For every clause, we add two vertices $c_{in}^{(j)}$ and $c_{out}^{(j)}$ for $j = 1, \dots, m$. Moreover, we add nodes x_0 and x_{n+1} . W.l.o.g., assume that clauses C_1, \dots, C_k contain only nonnegated literals and clauses C_{k+1}, \dots, C_m contain only negated literals. Then we add edges $(c_{in}^{(j)}, x_0)$ and $(x_{n+1}, c_{out}^{(j)})$ for every $1 \leq j \leq k$ as well as $(c_{in}^{(j)}, x_{n+1})$ and $(x_0, c_{out}^{(j)})$ for every $k+1 \leq j \leq m$. For every literal, we add vertices $x_j^{(1)}, x_j^{(2l)}, x_j^{(2)}, x_j^{(2r)}$, and $x_j^{(3)}$ for every $1 \leq j \leq n$. We connect them using edges $(x_j^{(1)}, x_j^{(2r)}), (x_j^{(2r)}, x_j^{(3)}), (x_j^{(3)}, x_j^{(2l)}), (x_j^{(2l)}, x_j^{(1)}), (x_j^{(1)}, x_j^{(2)}), (x_j^{(2)}, x_j^{(3)}), (x_j^{(3)}, x_j^{(2)}),$ and $(x_j^{(2)}, x_j^{(1)})$. Finally, we add vertices $x_{j,j+1}^{(l)}$ and $x_{j,j+1}^{(r)}$ for every $0 \leq j \leq n$ to connect the aforementioned segments. This is done using edges $(x_j^{(3)}, x_{j,j+1}^{(r)}), (x_{j,j+1}^{(r)}, x_{j+1}^{(1)}), (x_{j+1}^{(1)}, x_{j,j+1}^{(l)}),$ and $(x_{j,j+1}^{(l)}, x_j^{(2)})$. W.l.o.g., we can assume that all edges have length 1 and the successor functions $s_v(e) := \delta^{out}(v)$ do not restrict any movement. Moreover, we define one station

$$S := \bigcup_{j=1}^m \left\{ (x_j^{(1)}, x_j^{(2)}), (x_j^{(2)}, x_j^{(3)}), (x_j^{(3)}, x_j^{(2)}), (x_j^{(2)}, x_j^{(1)}) \right\}. \quad (24)$$

The resulting network is planar and consists of

$$2m + 2 + 5n + 2(n + 1) = 2m + 7n + 4 = \mathcal{O}(n + m) \quad (25)$$

nodes and

$$2m + 8n + 4(n + 1) = 2m + 12n + 4 = \mathcal{O}(n + m) \quad (26)$$

edges. In particular, the problem size has no exponential blow-up, and the reduction is polynomial.

As an example consider the monotone 3-SAT- $(\leq 2, \leq 2)$ instance $\bigwedge_{i=1}^3 C_i$ with

$$C_1 = x_1 \vee x_3 \vee x_4, \quad C_2 = x_2 \vee x_3 \vee x_4, \quad C_3 = \neg x_1 \vee \neg x_2 \vee \neg x_4. \quad (27)$$

Figure 14 shows the graph resulting from Equation 27 together with respective trains and routes, which we add in the next proof step. Moreover, the station S is marked in orange.

Finally, we add one train tr_i (of length 1 and with maximal speed 1) per clause C_i for every $1 \leq i \leq m$. For $1 \leq i \leq k$, the trains use the following edges

- 1) $c_{in}^{(i)} \rightarrow x_0 \rightarrow x_{0,1}^{(r)}$
- 2) For every $j = 1, 2, \dots, n$ they traverse
 - a) If $x_j \in C_i$, then $x_{j-1,j}^{(r)} \rightarrow x_j^{(1)} \rightarrow x_j^{(2)} \rightarrow x_j^{(3)} \rightarrow x_{j,j+1}^{(r)}$

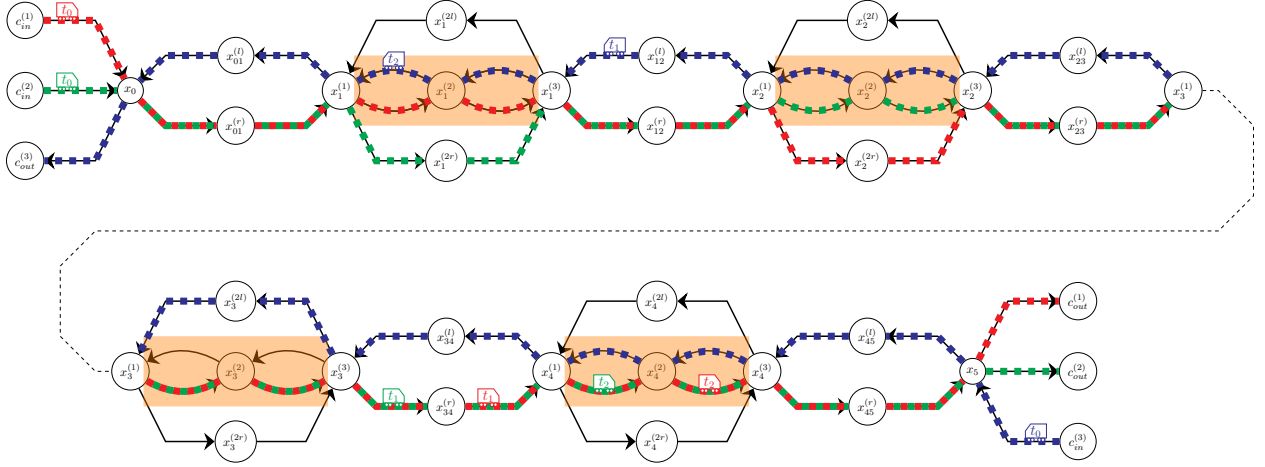


Fig. 14: Reduction from Monotone 3-SAT-($\leq 2, \leq 2$) to TR-VER

b) If $x_j \notin C_i$, then $x_{j-1,j}^{(r)} \rightarrow x_j^{(1)} \rightarrow x_j^{(2r)} \rightarrow x_j^{(3)} \rightarrow x_{j,j+1}^{(r)}$

3) Finally, $x_{n,n+1}^{(r)} \rightarrow x_{n+1} \rightarrow c_{out}^{(i)}$

For $k+1 \leq i \leq m$, trains travel in the opposite direction, i.e.,

1) $c_{in}^{(i)} \rightarrow x_{n+1} \rightarrow x_{n,n+1}^{(l)}$

2) For every $j = n, n-1, \dots, 1$ they traverse

a) If $x_j \in C_i$, then $x_{j,j+1}^{(l)} \rightarrow x_j^{(3)} \rightarrow x_j^{(2)} \rightarrow x_j^{(1)} \rightarrow x_{j-1,j}^{(l)}$

b) If $x_j \notin C_i$, then $x_{j,j+1}^{(l)} \rightarrow x_j^{(3)} \rightarrow x_j^{(2l)} \rightarrow x_j^{(1)} \rightarrow x_{j-1,j}^{(l)}$

3) Finally, $x_{0,1}^{(l)} \rightarrow x_0 \rightarrow c_{out}^{(i)}$

Every train's route consists of $2 + 4n + 2 = 4(n+1) =: T_n$ edges. They are marked in Figure 14 accordingly.

All trains shall enter at time $t = 0$ at their respective entry node $c_{in}^{(i)}$. They required to stop in S from time $t = m \cdot T_n + 4$ to $t = m \cdot T_n + 6$, and shall leave the network at time $t = m \cdot T_n + 6 + m \cdot T_n = 2mT_n + 6$ at their respective exit node $c_{out}^{(i)}$. Since $2mT_n + 6 = \mathcal{O}(m \cdot n)$, all specified times are still polynomial in the input size.

Assume that the given monotone 3-SAT-($\leq 2, \leq 2$) instance is feasible. From any particular solution, define $A := \{j : x_j \text{ is true}\}$ and $A^C := \{1, \dots, m\} - A = \{j : x_j \text{ is false}\}$. For $1 \leq i \leq k$, define $\tilde{x}(i) := \min\{j : j \in A \text{ and } x_j \in C_i\}$. For $k+1 \leq i \leq m$, we have $\tilde{x}(i) := \min\{j : j \in A^C \text{ and } \neg x_j \in C_i\}$. The order in which trains traverse shared edges is not predefined; hence, w.l.o.g. assume that $\tilde{x}(i) \geq \tilde{x}(i+1)$ for every $1 \leq i \leq k-1$ and $\tilde{x}(i) \leq \tilde{x}(i+1)$ for every $k+1 \leq i \leq m-1$. Otherwise, we can rename the trains at this point. For $1 \leq i \leq m$, we let tr_i stop on $(c_{in}^{(1)}, x_0)$ until $t = (i-1) \cdot T_n + 1$. Afterwards, trains $i = 1, \dots, k$ travel at maximal speed all the way to $(x_{\tilde{x}(i)-1, \tilde{x}(i)}^{(r)}, x_{\tilde{x}(i)}^{(1)})$ or $(x_{\tilde{x}(i)-1}^{(r)}, x_{\tilde{x}(i)-1, \tilde{x}(i)}^{(r)})$ if the previous edge is already occupied. Because a literal appears in only two clauses, it cannot happen that both edges are already occupied. Similarly, for trains traveling in the reverse direction ($i = k+1, \dots, m$), travel at maximal speed all the way to $(x_{\tilde{x}(i), \tilde{x}(i)+1}^{(l)}, x_{\tilde{x}(i)}^{(3)})$ or $(x_{\tilde{x}(i)+1}^{(1)}, x_{\tilde{x}(i), \tilde{x}(i)+1}^{(l)})$ if the previous edge is already occupied. Again, because a literal appears in only two clauses, it cannot happen that both edges are already occupied. All trains will arrive latest at time $t = i \cdot T_n$. At time $t = m \cdot T_n$, they can

all traverse two edges into stop S within at most four time steps. Finally, they can proceed in the same order to the respective exit vertices in, again, at most $m \cdot T_n$ time steps and wait on their last edge without affecting other trains. There are no conflicts because of the chosen order and feasibility of the monotone 3-SAT- $(\leq 2, \leq 2)$. In particular, this constitutes a feasible solution to the corresponding TR-VER instance. In Figure 14, the trains are depicted at $t_0 = 1$, $t_1 = m \cdot T_n$, and $t_2 = m \cdot T_n + 5$. This setting corresponds to setting x_1 to false and x_4 to true.

On the other hand, if we are given a feasible timing of the corresponding TR-VER instance, we check the positions at time $t = m \cdot T_n + 5$. We set x_j to be true if $(x_j^{(1)}, x_j^{(2)})$ or $(x_j^{(2)}, x_j^{(3)})$ is occupied and to false if $(x_j^{(3)}, x_j^{(2)})$ or $(x_j^{(2)}, x_j^{(1)})$ is occupied. All other variables are chosen arbitrarily. Note that a variable cannot be set to true and false because then trains would collide on the opposing tracks. Hence, this constitutes a feasible solution to the corresponding monotone 3-SAT- $(\leq 2, \leq 2)$ instance by design.

Combining the above, we have argued that a monotone 3-SAT- $(\leq 2, \leq 2)$ instance is feasible if, and only if, the corresponding TR-VER instance is. Hence, we have polynomially reduced Monotone 3-SAT- $(\leq 2, \leq 2)$ to TR-VER, proving the latter's NP-completeness. \square

Observe that no VSS is created in a verification setting. Instead, the layout is fixed. Hence, the above results essentially carry over to fixed-block systems such as classical ETCS L2.

B. Generation of VSS Layouts

As mentioned before, the potential of hybrid train detection is that virtual subsections can be added without requiring expensive track-side detection hardware. In particular, what adding VSS sections means within our framework needs to be formalized. Namely, they correspond to inserting a VSS border vertex to separate a given edge into two or rather split both directional edges at the same position as follows.

Definition III.3 (VSS Addition Operator). *For an edge $e_\star = (v_1, v_2) \in E$ and a scalar $\rho \in (0, 1)$, the VSS addition operator $\psi_{e_\star}^\rho$ takes as an input a railway network $N = (G, \text{len}, \{s_v\}_{v \in V})$, a border function C_V , and a set of stations \mathcal{S} . It then adds a VSS border $\rho \cdot \text{len}(e_\star)$ away from v_1 . More formally, it outputs a new network $N' = (G', \text{len}', \{s'_v\}_{v \in V'})$, a border function $C_{V'}$, and a set of stations \mathcal{S}' as follows:*

The vertices are expanded by a new vertex denoting a new VSS border, i.e.,

$$V' = V \cup \{v_{\text{new}}\} \quad C_{V'}(v) = \begin{cases} C_V(v), & \text{if } v \in V \\ 1, & \text{if } v = v_{\text{new}} \end{cases}. \quad (28)$$

Edges e_\star and possibly e_\star° are split at the new vertex into $e_1 := (v_1, v_{\text{new}})$ and $e_2 := (v_{\text{new}}, v_2)$ plus possibly their reverse edges, i.e., using

$$E^{\Delta^-} := \{e_\star, e_\star^\circ\} \cap E \quad E^{\Delta^+} := \begin{cases} \{e_1, e_2\}, & \text{if } e_\star^\circ \notin E \\ \{e_1, e_2, e_1^\circ, e_2^\circ\}, & \text{if } e_\star^\circ \in E \end{cases} \quad (29)$$

we have

$$E' = (E - E^{\Delta^-}) \cup E^{\Delta^+}. \quad (30)$$

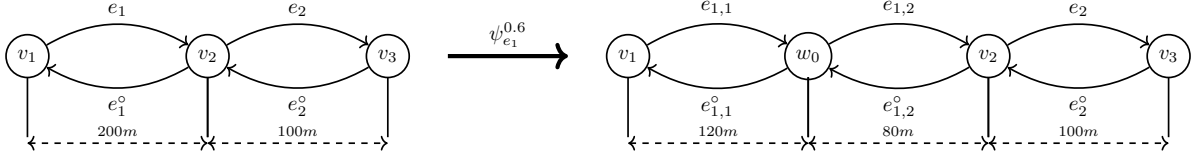


Fig. 15: VSS addition operator

The lengths are updated according to parameter ρ , i.e.,

$$\text{len}'(e) = \begin{cases} \text{len}(e), & \text{if } e \in E \\ \rho \text{len}(e_\star), & \text{if } e \in \{e_1, e_1^\circ\} \\ (1 - \rho) \text{len}(e_\star), & \text{if } e \in \{e_2, e_2^\circ\} \end{cases} \quad (31)$$

The successor functions $\{s'_v\}_{v \in V'}$ are updated accordingly to represent the same movements in practice, i.e., using $E_1 := \{e \in \delta_E^{\text{in}}(v_1) : e_\star \in s_{v_1}(e)\}$ and $E_2 := \{e \in \delta_E^{\text{in}}(v_2) : e_\star^\circ \in s_{v_2}(e)\}$ we have

$$s'_v = s_v \text{ for all } v \in V' - \{v_1, v_2, v_{\text{new}}\} \quad s'_{v_{\text{new}}}(e) = \begin{cases} \{e_2\}, & \text{if } e = e_1 \\ \{e_1^\circ\}, & \text{if } e = e_2^\circ \end{cases} \quad (32)$$

$$s'_{v_1}(e) = \begin{cases} (s_{v_1}(e) - \{e_\star\}) \cup \{e_1\}, & \text{if } e \in E_1 \\ s_{v_1}(e_1^\circ), & \text{if } e = e_1^\circ \\ s_{v_1}(e), & \text{otherwise} \end{cases} \quad s'_{v_2}(e) = \begin{cases} (s_{v_2}(e) - \{e_\star\}) \cup \{e_2^\circ\}, & \text{if } e \in E_2 \\ s_{v_2}(e_\star), & \text{if } e = e_2 \\ s_{v_2}(e), & \text{otherwise} \end{cases} \quad (33)$$

If a station includes e_\star or its reverse, it is updated accordingly, i.e., using $f(e_\star) := \{e_1, e_2\}$, $f(e_\star^\circ) := \{e_1^\circ, e_2^\circ\}$, and

$$\Delta^-(S) := \{e_\star, e_\star^\circ\} \cap S \quad \Delta^+(S) := \bigcup_{e \in \Delta^-(S)} f(e) \quad (34)$$

we have

$$\mathcal{S}' = \{(S - \Delta^-(S)) \cup \Delta^+(S) : S \in \mathcal{S}\}. \quad (35)$$

We denote by Ψ the set of all VSS addition operators.

Example III.4. Consider the graph depicted in Figure 15. Assume we want to separate the track between v_1 and v_2 into two VSS sections corresponding to adding a VSS border, say w_0 . The first section has a length of $0.6 \cdot 200m = 120m$, the latter of $80m$. The transformation is shown in Figure 15 and corresponds to the operator $\psi_{e_1}^{0.6}$, which is equivalent to $\psi_{e_1^\circ}^{0.4}$.

Some elements (e.g., turnouts) require special treatment and classical train separation. In our formulation, this can be achieved by marking some of the TTD sections as *unbreakable*. It is then forbidden to apply the VSS addition operator on an edge within any unbreakable TTD section. For example, any TTD section containing a turnout could be marked in such a way. By doing this, the control logic around switches would not change, while at the same time allowing optimization using VSS on all permitted tracks (only).

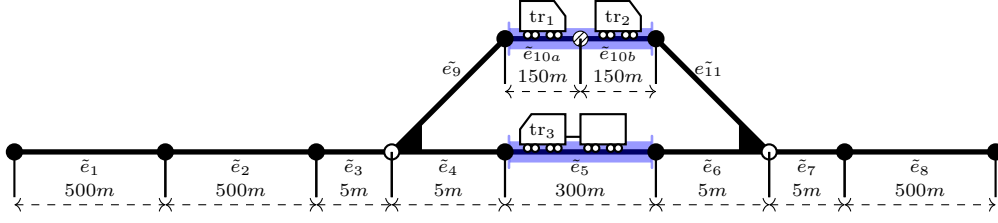


Fig. 16: Generation of VSS layout ($t = 300$)

By leaving the TTDs fixed and only adding VSS sections, we can change the block structure at a low cost. Because of this, it might be possible to change the virtual subsections when a new train schedule should be implemented. Hence, we can somewhat “overfit” to the current timetable requests without relying solely on schedule-independent KPIs (e.g., headway). This leads to the following design task of generating VSS layouts.

GENERATION OF VSS LAYOUT (VSS-GEN)

Input: A railway network $N = (G, \text{len}, \{s_v\}_{v \in V})$; border function $C_V: V \rightarrow \{0, 1, 2\}$; trains $\text{tr}_1, \dots, \text{tr}_m$; a list of timetable requests for tr_1 to tr_m .

Task: Determine a minimal number of VSS addition operators $\psi^{(1)}, \dots, \psi^{(k)} \in \Psi$ (and train routes $R^{(\text{tr}_1)}, \dots, R^{(\text{tr}_m)}$) such that on $\psi^{(k)} \circ \dots \circ \psi^{(1)}(\cdot)$ the VSS condition is fulfilled and all trains respect their speed and timetable requests or output that no such operators exist.

If possible, we try to keep the number of additional virtual subsections small to keep it tractable for the control system during operation. In particular, we do not require a hard optimum because one section more or less will likely not pose any additional costs. On the other hand, it might be interesting to still have a minimal number of subsections to make given timetable requests realizable. In a later stage, one could then add more subsections to improve robustness; if the generated VSS layout from this design task is minimal, there might be more space to add subsections to improve robustness.

Example III.5. Again, consider the setting from Example III.1 and Figure 13. If we separate \tilde{e}_{10} into two virtual subsections by including a VSS border (\odot), the train station S_0 has three blocks. In fact, we can now route the trains as shown in Figure 16, i.e., in the end, trains tr_1 and tr_2 can occupy the newly created subsections and train tr_3 the other track. Hence, by adding one VSS border, we have made the previously infeasible timetable requests realizable. Note that we could also have added additional subsections, e.g., split track \tilde{e}_5 and have tr_3 occupy two blocks in the end. However, as said, our objective is to add rather few VSS borders, and one additional VSS border suffices to make the timetable requests realizable.

Since VSS-GEN is not a decision problem, NP-completeness is not defined in this context. However, NP-hardness can be shown by relating it to one of the previously defined verification tasks, where we use the more general definition that a problem is NP-hard if it can be reduced from some NP-complete problem, hence, not requiring the NP-hard problem to be of decision type [46].

Theorem III.6. *VSS-GEN is NP-hard even if all routes and exact stopping times in train stations are fixed and the railway network is planar.*

Proof. Consider an instance of TR-VER. Assume we are given an oracle to solve any instance of VSS-GEN. The given TR-VER instance constitutes valid input to VSS-GEN. Let k^* be the number of VSS addition operators used in an optimal solution of VSS-GEN on the respective input or let $k^* = \infty$ if no feasible solution exists. Then, the solution to the given TR-VER instance is “yes” if, and only if, $k^* = 0$. In particular, we have polynomially reduced TR-VER to VSS-GEN. The claim follows directly from Theorem III.2. \square

Remark III.7. *Instead of adding additional vertices in a continuous setting using VSS addition operators, we might also consider a discretized version, which might be easier to handle computationally. In this setting, we would require the railway network N to consist of all nodes that could possibly become a VSS border. Adding a VSS border is then equivalent to changing $C_V(v)$ from 0 to 1 for some $v \in V$ with $|\Gamma(v)| \leq 2$, where Γ denotes the neighborhood (ignoring the direction of edges).*

C. Schedule Optimization Using the Potential of VSS

In practice, minimizing VSS sections might not be of great interest. Instead, one might want to optimize the operational outcome, e.g., by minimizing travel times. Intuitively, one might think that this is an (computationally) easy problem, because the “best” layout is creating very small VSS (e.g., every meter or even less). In that case, trains would basically operate as if a moving block control system were implemented. However, according to industry experts, such an implementation would not be feasible because every virtual block has to be specified within the interlocking and the *Radio Block Center* (RBC). These components must fulfill the *safety integrity level* (SIL) 4 and are time-consuming and costly to develop and maintain. Because many such systems were developed and approved for classical control systems, the address space for block sections has limited capacity.

Thus, in practice, there is a significant upper limit on how many VSS can be defined, which should be considered within the planning process. Hence, the question arises on strategically placing these VSS to optimize the operational outcome. In particular, a proposed solution of placing a VSS every meter is likely not feasible.

SCHEDULE OPTIMIZATION (VSS-OPT)

Input: A railway network $N = (G, \text{len}, \{s_v\}_{v \in V})$; border function $C_V: V \rightarrow \{0, 1, 2\}$; trains $\text{tr}_1, \dots, \text{tr}_m$; a list of timetable requests for tr_1 to tr_m ; an upper bound K_{\max} .

Task: Determine VSS addition operators $\psi^{(1)}, \dots, \psi^{(k)} \in \Psi$ (and train routes $R^{(\text{tr}_1)}, \dots, R^{(\text{tr}_m)}$) such that $k \leq K_{\max}$, on $\psi^{(k)} \circ \dots \circ \psi^{(1)}(\cdot)$ the VSS condition is fulfilled and all trains respect their speed and timetable requests or output that no such operators exist. Among all feasible solutions, the one with minimal travel times is returned. As an objective, one might, e.g., minimize the (weighted) sum of travel times or the longest travel time.

Another operational objective might be to maximize capacity. Assume, e.g., one plans a mixed railway network on which passenger trains should operate, adhering to certain scheduling constraints. On the other hand, there are additional requests for freight trains running through the network. In this setting, one might want to maximize the

number of freight trains without negatively affecting the timetable requests of passenger trains by using the full potential of VSS sections.

CAPACITY OPTIMIZATION (VSS-CAP)

Input: A railway network $N = (G, \text{len}, \{s_v\}_{v \in V})$; border function $C_V: V \rightarrow \{0, 1, 2\}$; trains $\text{tr}_1, \dots, \text{tr}_{m_1}$ and $\tilde{\text{tr}}_1, \dots, \tilde{\text{tr}}_{m_2}$; a list of timetable requests for every train; an upper bound K_{\max} .

Task: Determine VSS addition operators $\psi^{(1)}, \dots, \psi^{(k)} \in \Psi$, a subset $\mathcal{R} \subseteq \{1, \dots, m_2\}$, train routes $R^{(\text{tr}_1)}, \dots, R^{(\text{tr}_m)}$ and $R^{(\tilde{\text{tr}}_j)}$ ($j \in \mathcal{R}$) such that $k \leq K_{\max}$, on $\psi^{(k)} \circ \dots \circ \psi^{(1)}(\cdot)$ the VSS condition is fulfilled and all routed trains respect their speed and timetable requests or output that no such operators exist. Among all feasible solutions, the one maximizing $|\mathcal{R}|$ (i.e., the number of routed optional trains).

Theorem III.8. *VSS-OPT and VSS-CAP are NP-hard even if all (potential) routes and exact stopping times in train stations are fixed, and the railway network is planar.*

Proof. Consider an instance of TR-VER. Assume we are given an oracle to solve any instance of VSS-OPT. The given TR-VER instance constitutes valid input to VSS-OPT. Let t^* be the optimal travel time objective or $t^* = \infty$ if no feasible solution exists. Observe that it is easy to deduce a theoretical upper bound on t^* in the feasible case, say \bar{t} , from the latest possible exit times $\bar{t}_{out}^{tr_1}, \dots, \bar{t}_{out}^{tr_m}$, e.g., $\bar{t} = \bar{t}_{out}^{tr_1} + \dots + \bar{t}_{out}^{tr_m}$ if the objective is given by the sum of travel times. Then, the solution to the given TR-VER instance is “yes” if, and only if, $t^* \neq \infty$, i.e., $t^* \leq \bar{t}$.

The given TR-VER instance also constitutes valid input to VSS-CAP. Let $r^* = |\mathcal{R}^*|$ be the maximal number of optional trains $\tilde{\text{tr}}_1, \dots, \tilde{\text{tr}}_{m_2}$ that are routed or $r^* = -\infty$ if the problem is infeasible even for $\mathcal{R} = \emptyset$. Then, the solution to the given TR-VER instance is “yes” if, and only if, $r^* \neq \infty$, i.e., $r^* \leq m_2$.

In particular, we have polynomially reduced TR-VER to VSS-OPT and VSS-CAP. The claim follows directly from Theorem III.2. \square

IV. CONCLUSIONS

In this paper, we formalized important properties of ETCS L2 HTD and corresponding design tasks. Moreover, we argued that the framework could eventually also model more realistic train movements as additional constraints by stating some examples of possible extensions in Section II. While the verification tasks in Section III-A are analog to older level systems (since the block layout is fixed), the two latter tasks (Sections III-B and III-C) make use of the new freedom provided by virtual subsections. Furthermore, we proved that these tasks are NP-hard (or NP-complete, respectively). By this, we provided a theoretical foundation for further developing corresponding design methods for ETCS L2 HTD. These methods will be integrated into the Munich Train Control Toolkit (MTCT) hosted on GitHub (<https://github.com/cda-tum/mtct>).

The results confirm that previously proposed heuristics and exact methods (presented, e.g., in [35], [36]), which only generate non-optimal results or have a severely limited scalability, respectively, do not suffer from insufficient methodology but complexity reasons. Overall, for the first time, a solid basis for developing (ideally scalable) design automation solutions incorporating the freedom allowed by virtual subsections within ETCS L2 HTD is

provided. These methods might be a valuable addition to the already existing work on fixed-block control systems summarized in Section I. It will also be interesting to see how these methods tailored to hybrid train detection systems perform compared to previously designed methods that do not consider the specific timetable requests for generating signaling layouts.

REFERENCES

- [1] O. Gemine, A. Hougardy, E. Lepailleur, ERTMS unit: Assignment of values to ETCS variables, Tech. Rep. ERA_ERTMS_040001, European Union Agency for Railways, version 1.33 (2023).
URL <https://www.era.europa.eu/system/files/2023-02/ETCSvariablesandvalues.pdf>
- [2] U. Maschek, Sicherung des Schienenverkehrs, Springer-Verlag GmbH, 2018.
URL <https://doi.org/10.1007/978-3-658-22878-1>
- [3] European Commission, ETCS levels and modes, Website (2024).
URL https://transport.ec.europa.eu/transport-modes/rail/ertms/how-does-it-work/etcs-levels-and-modes_en
- [4] L. Schnieder, European Train Control System (ETCS), Springer Berlin Heidelberg, 2021.
URL <https://doi.org/10.1007/978-3-662-62878-2>
- [5] European Commission, Commission implementing regulation (EU) 2023/1695 of 10 august 2023 on the technical specification for interoperability relating to the control-command and signalling subsystems of the rail system in the European Union and repealing regulation (EU) 2016/919, in: Official Journal of the European Union, Vol. 66 of L 222, 2023, pp. 380–560.
URL https://eur-lex.europa.eu/eli/reg_impl/2023/1695/oj
- [6] European Union Agency for Railways, Mandatory specifications (ETCS B4 R1, RMR: GSM-R B1 MR1 + FRMCS B0, ATO B1 R1) (2023).
URL <https://www.era.europa.eu/era-folder/1-ccs-tsi-appendix-mandatory-specifications-etcs-b4-r1-rmr-gsm-r-b1-mr1-frmcs-b0-ato-b1>
- [7] L. Schnieder, Communications-Based Train Control (CBTC), Springer Berlin Heidelberg, 2021.
URL <https://doi.org/10.1007/978-3-662-62876-8>
- [8] Stadtwerke München GmbH, Digitalisierung der Münchner U-Bahn, Press Release (Aug. 2019).
URL <https://www.mvg.de/ueber/presse-print/pressemeldungen/2019/august/2019-08-02-cbtc-bahnsteigtueren.html>
- [9] J. Pachl, Railway Signalling Principles: Edition 2.0, Universitätsbibliothek Braunschweig, 2021.
URL <https://doi.org/10.24355/dbbs.084-202110181429-0>
- [10] S. Engels, T. Peham, R. Wille, A symbolic design method for ETCS Hybrid Level 3 at different degrees of accuracy, in: 23rd Symposium on Algorithmic Approaches for Transportation Modelling, Optimization, and Systems (ATMOS 2023), Vol. 115, Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2023.
URL <https://doi.org/10.4230/OASICS.ATMOS.2023.6>
- [11] J. Pachl, Besonderheiten ausländischer Eisenbahnbetriebsverfahren, 2019.
URL <https://doi.org/10.1007/978-3-658-23853-7>
- [12] M. Bartholomeus, L. Arenas, R. Treydel, F. Hausmann, N. Geduhn, A. Bossy, ERTMS Hybrid Level 3, SIGNAL + DRAHT (110) 1+2/2018 (2018) 15–22.
URL https://www.eurailpress.de/fileadmin/user_upload/SD_1_2-2018_Bartholomaeus_ua.pdf
- [13] EEIG ERTMS Users Group, ERTMS/ETCS hybrid train detection, Tech. Rep. 16E042, version 1E (2022).
URL https://ertms.be/wp-content/uploads/2023/06/16E0421F_HTD.pdf
- [14] D. Gill, C. Goodman, Computer-based optimisation techniques for mass transit railway signalling design, IEE Proceedings B Electric Power Applications 139 (3) (1992) 261.
URL <https://doi.org/10.1049/ip-b.1992.0031>
- [15] C. Chang, D. Du, Improved optimisation method using genetic algorithms for mass transit signalling block-layout design, IEE Proceedings - Electric Power Applications 145 (3) (1998) 266.
URL <https://doi.org/10.1049/ip-epa:19981733>

- [16] C. Chang, D. Du, Further improvement of optimisation method for mass transit signalling block-layout design using differential evolution, *IEE Proceedings - Electric Power Applications* 146 (5) (1999) 559.
URL <https://doi.org/10.1049/ip-epa:19990223>
- [17] B. Ke, N. Chen, Signalling blocklayout and strategy of train operation for saving energy in mass rapid transit systems, *IEE Proceedings - Electric Power Applications* 152 (2) (2005) 129.
URL <https://doi.org/10.1049/ip-epa:20045188>
- [18] B.-R. Ke, M.-C. Chen, C.-L. Lin, Block-layout design using MAX-MIN ant system for saving energy on mass rapid transit systems, *IEEE Transactions on Intelligent Transportation Systems* 10 (2) (2009) 226–235.
URL <https://doi.org/10.1109/TITS.2009.2018324>
- [19] S. Dillmann, R. Hähnle, Automated planning of ETCS tracks, in: *Reliability, Safety, and Security of Railway Systems. Modelling, Analysis, Verification, and Certification*, Springer International Publishing, 2019, pp. 79–90.
URL https://doi.org/10.1007/978-3-030-18744-6_5
- [20] B. Luteberget, C. Johansen, M. Steffen, Synthesis of railway signaling layout from local capacity specifications, in: *Lecture Notes in Computer Science*, Springer International Publishing, 2019, pp. 121–137.
URL https://doi.org/10.1007/978-3-030-30942-8_9
- [21] V. Vignali, F. Cuppi, C. Lantieri, N. Dimola, T. Galasso, L. Rapagnà, A methodology for the design of sections block length on ETCS L2 railway networks, *Journal of Rail Transport Planning & Management* 13 (2020) 100160.
URL <https://doi.org/10.1016/j.jrtpm.2019.100160>
- [22] W. Sun, D. Newton, S. Cox, Signalling the layout -automatic design of the optimum ETCS L2 track sections (Apr. 2020).
URL https://www.researchgate.net/publication/340442986_Signalling_the_Layout_-_Automatic_Design_of_the_Optimum_ETCS_L2_Track_Sections
- [23] J.-W. Goossens, S. van Hoesel, L. Kroon, On solving multi-type railway line planning problems, *European Journal of Operational Research* 168 (2) (2006) 403–424.
URL <https://doi.org/10.1016/j.ejor.2004.04.036>
- [24] A. D'Ariano, D. Pacciarelli, M. Pranzo, A branch and bound algorithm for scheduling trains in a railway network, *European Journal of Operational Research* 183 (2) (2007) 643–657.
URL <https://doi.org/10.1016/j.ejor.2006.10.034>
- [25] R. Borndörfer, T. Schlechte, Solving railway track allocation problems, in: *Operations Research Proceedings*, Springer Berlin Heidelberg, 2008, pp. 117–122.
URL https://doi.org/10.1007/978-3-540-77903-2_18
- [26] R. M. Lusby, J. Larsen, M. Ehrgott, D. Ryan, Railway track allocation: models and methods, *OR Spectrum* 33 (4) (2009) 843–883.
URL <https://doi.org/10.1007/s00291-009-0189-0>
- [27] M. Carey, S. Carville, Scheduling and platforming trains at busy complex stations, *Transportation Research Part A: Policy and Practice* 37 (3) (2003) 195–224.
URL [https://doi.org/10.1016/S0965-8564\(02\)00012-5](https://doi.org/10.1016/S0965-8564(02)00012-5)
- [28] R. M. Lusby, J. Larsen, M. Ehrgott, D. M. Ryan, A set packing inspired method for real-time junction train routing, *Computers & Operations Research* 40 (3) (2013) 713–724.
URL <https://doi.org/10.1016/j.cor.2011.12.004>
- [29] Y. Wang, R. Song, S. He, Z. Song, Train routing and track allocation optimization model of multi-station high-speed railway hub, *Sustainability* 14 (12) (2022) 7292.
URL <https://doi.org/10.3390/su14127292>
- [30] T. Büker, T. Graffagnino, E. Hennig, A. Kuckelberg, Enhancement of blocking-time theory to represent future interlocking architectures, in: A. Peterson, M. Joborn, M. Bohlin (Eds.), *RailNorrköping. 8th International Conference on Railway Operations Modelling and Analysis (ICROMA)*, Vol. 69, Linköping Electronic Conference Proceedings, 2019, pp. 219–240.
URL https://ep.liu.se/konferensartikel.aspx?series=ecp&issue=69&Article_No=15
- [31] T. Schlechte, R. Borndörfer, J. Denißen, S. Heller, T. Klug, M. Küpper, N. Lindner, M. Reuther, A. Söhlke, W. Steadman, Timetable optimization for a moving block system, *Journal of Rail Transport Planning & Management* 22 (2022) 100315.
URL <https://doi.org/10.1016/j.jrtpm.2022.100315>

- [32] T. Klug, M. Reuther, T. Schlechte, Does laziness pay off? - a lazy-constraint approach to timetabling, in: M. D’Emidio, N. Lindner (Eds.), 22nd Symposium on Algorithmic Approaches for Transportation Modelling, Optimization, and Systems (ATMOS 2022), Vol. 106, Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2022.
URL <https://doi.org/10.4230/OASIcs.ATMOS.2022.11>
- [33] S. Engels, R. Wille, Comparing lazy constraint selection strategies in train routing with moving block control (2024). [arXiv:2405.18977](https://arxiv.org/abs/2405.18977), doi:10.48550/ARXIV.2405.18977.
- [34] M. Butler, A. Raschke, T. S. Hoang, K. Reichl (Eds.), Abstract State Machines, Alloy, B, TLA, VDM, and Z, Springer International Publishing, 2018.
URL <https://doi.org/10.1007/978-3-319-91271-4>
- [35] R. Wille, T. Peham, J. Przigoda, N. Przigoda, Towards automatic design and verification for Level 3 of the European Train Control System, in: 2021 Design, Automation & Test in Europe Conference & Exhibition (DATE), IEEE, 2021.
URL <https://doi.org/10.23919/date51398.2021.9473935>
- [36] T. Peham, J. Przigoda, N. Przigoda, R. Wille, Optimal railway routing using virtual subsections, in: Reliability, Safety, and Security of Railway Systems. Modelling, Analysis, Verification, and Certification, Springer International Publishing, 2022, pp. 63–79.
URL https://doi.org/10.1007/978-3-031-05814-1_5
- [37] S. Engels, R. Wille, Late breaking results: Iterative design automation for train control with hybrid train detection, in: 2024 Design, Automation & Test in Europe Conference & Exhibition (DATE), 2024, pp. 1–2.
URL <https://doi.org/10.23919/DATE58400.2024.10546590>
- [38] A. Mascis, D. Pacciarelli, Job-shop scheduling with blocking and no-wait constraints, European Journal of Operational Research 143 (3) (2002) 498–517.
URL [https://doi.org/10.1016/S0377-2217\(01\)00338-1](https://doi.org/10.1016/S0377-2217(01)00338-1)
- [39] B. Lácay, Weichen, Flickr (Dec. 2005).
URL <https://www.flickr.com/photos/dustpuppy/78871005/in/photostream/>
- [40] M. Montigel, Formal representation of track topologies by double vertex graphs, Computers in railways III 2 (1992) 359–370.
- [41] Q. K. Dang, T. Bourdeaud’huy, K. Mesghouni, A. Toguyéni, Low-level modeling for optimal train routing and scheduling in busy railway stations (2019).
URL <https://doi.org/10.5281/ZENODO.3455633>
- [42] European Union Agency for Railways, Introduction to ETCS braking curves, Tech. Rep. ERA_ERTMS_040026, version 1.5 (2020).
URL <https://www.era.europa.eu/system/files/2022-11/IntroductiontoETCSbrakingcurves.pdf>
- [43] R. M. Goverde, F. Corman, A. D’Ariano, Railway line capacity consumption of different railway signalling systems under scheduled and disturbed conditions, Journal of Rail Transport Planning & Management 3 (3) (2013) 78–94.
URL <https://doi.org/10.1016/j.jrtpm.2013.12.001>
- [44] N. D. Versluis, E. Quaglietta, R. M. Goverde, P. Pellegrini, J. Rodriguez, Real-time railway traffic management under moving-block signalling: A literature review and research agenda, Transportation Research Part C: Emerging Technologies 158 (2024) 104438.
URL <https://doi.org/10.1016/j.trc.2023.104438>
- [45] A. Darmann, J. Döcker, On simplified NP-complete variants of monotone 3-Sat, Discrete Applied Mathematics 292 (2021).
URL <https://doi.org/10.1016/j.dam.2020.12.010>
- [46] J. V. Leeuwen, Handbook of Theoretical Computer Science: Algorithms and Complexity, Vol. A, Elsevier Science, 1990.