CASPT 2015

Extraction of Significant Conflicts in Periodic Timetabling

Michael Kümmling · Peter Großmann · Jens Opitz · Reyk Weiß · Karl Nachtigall

Abstract The development of the program system TAKT for automatic calculation of periodic railway passenger timetables and its application on large-scale and complex strategic timetable studies of the future German railway network gave promising results. However, these practical applications also exhibited one deficit of TAKT and the used methods: It is nearly impossible to understand or retrace the gained timetables. This paper describes a technique for extracting significant conflicts from the timetabling problem, allowing further understanding of the results and precise measurements. The method performs well even on large real-world timetabling problem, as first computational results show.

Keywords timetabling · conflict extraction · periodic event scheduling problem

1 Introduction

The total rail transport amount is increasing further, but there is only a limited budget available for maintenance and development of railway infrastructure. As infrastructural measures require a long preliminary lead time from first ideas to realisation, versatile methods for early but comprehensive assessment of railway infrastructure are needed. Creating strategic timetables in a long-term perspective has become an important method for achieving this task (refer to Weigand (2012)).

The German railway network has a polycentric structure, requiring a sophisticated coordination and synchronisation of railway passenger services. Integrated regular-interval timetables (IRIT) are the ideal approach (refer e.g.

M. Kümmling · P. Großmann · J. Opitz · R. Weiß · K. Nachtigall
Chair of Traffic Flow Science, Technische Universität Dresden, 01062 Dresden, Germany

M. Kümmling
Tel.: +49 351 463 36741
E-mail: michael.kuemmling@tu-dresden.de

to Lichtenegger (1990)), but this simple structure is usually not accomplishable, as it imposes strict requirements on running times and station layouts. Yet, it is still desired to realise as many parts as possible in a not fully integrated regular-interval timetable (see FGSV (2001)).

Automatic timetabling is an efficient instrument in strategic timetabling. During the last years the automatic timetabling software TAKT was developed at the Chair of Traffic Flow Science at TU Dresden. It incorporates several models and methods for automatic railway timetable generation (see Opitz (2009)). In this paper, we focus on periodic timetables for passenger rail networks. Refer to Opitz (2009) and Kümmling et al (2013) for an overview of TAKT's functionality for passenger rail networks.

To develop a reliable evaluation of infrastructure measures, close justifications and explanations are required for any deviations from past or desired timetables and any special solutions. Nevertheless, TAKT does not comply with this at all, as the automatic resolving of conflicts is a "black box". This paper describes methods for detecting significant conflicts within the periodic timetabling problem which are reason to notable deviations from original constraints.

In Section 2 we provide the preliminaries for periodic event scheduling. Local conflicts are introduced in Section 3 as well as an algorithm for extracting them. Significant conflicts and their extraction are specified in Section 4. And finally, In Section 5, we present computational results of a first application of the described methods. We conclude the work in Section 6 and give a further scientific outlook.

2 Periodic Timetabling

In the last 15 years, the Periodic Event Scheduling Problem (PESP) established as one of the most suitable problem formulations for regular-interval timetabling. It is introduced by Serafini and Ukovich (1989). The related periodic event network (PEN) permits flexible representation of almost all periodic timetables' constraints. For instance, PESP and its application to railway fixed-interval timetabling are discussed in detail by Nachtigall (1998) and Opitz (2009).

The train network, which is the base of the timetabling problem, contains routes $\mathcal{L}$ running on a railway network with stations $\mathcal{S}$. Each Route $L \in \mathcal{L}$ serves a specified sequence of stations $S \in \mathcal{S}$. All constraints are modelled into an PEN. Its nodes in $\mathcal{V}$ represent arrival events $(L, arr, S) \in \mathcal{V}$ and departure events $(L, dep, S) \in \mathcal{V}$. The timetable $\mathbf{T} \in \mathbb{Z}^{|V|}$ assigns to every event $i \in \mathcal{V}$ a potential $T_i \in \mathbb{Z}$, $0 \le T_i < t_T$. In a periodic timetable with period $t_T \in \mathbb{N}^+$ the event happens periodically at all times $T_i + zt_T$, $z \in \mathbb{Z}$.

The network's arcs $a \in \mathcal{A}$: $i \to j$ are basically time consuming processes. All arcs are mapped to periodic intervals $[t_{min,a}, t_{max,a}]_{t_T}$ by the mapping $m(a)$. A periodic interval $[t_{min,a}, t_{max,a}]_{t_T}$ denotes the interval from $t_{min,a}$ to $t_{max,a}$ and all possible repetitions, shifted by zt_T.

A timetable **T** is considered valid if and only if

$$\forall a \in \mathcal{A}\colon \exists z_a \in \mathbb{Z}\colon t_{min,a} \leq T_j - T_i - z_a t_T \leq t_{max,a}. \quad (1)$$

The lower slack y_a is the deviance of the actual processing time from the lower bound such that

$$0 \leq y_a = T_j - T_i - z_a t_T - t_{min,a} < t_T. \quad (2)$$

The periodic event scheduling problem (PESP) is the decision, whether there exists any valid timetable for a given PEN $\mathcal{N} = (\mathcal{V}, \mathcal{A}, m, t_T)$. For feasible problems, a timetable can be calculated.

This universal model allows the modelling of running time, stopping time, headway and transfer time. For instance, trains are encoded as alternating sequences of running activities $(L, dep, S) \to (L, arr, S')$ and stops $(L, arr, S) \to (L, dep, S)$. Headway between different routes include both safety headway representing the permitted minimum headway and also evenly distributed headway of different trains running partly on the same railway line. Transfer times include several of different requirements: vehicle transfers, staff transfers and passenger transfers.

The *satisfiability problem* (SAT) consists of a *propositional formula* F and is the decision problem whether it exists an *interpretation* (or assignment) J from the set of propositional formulas to the set $\{0, 1\}$ of truth values such that J assigns 1 to F. In such case, J is called *model* for F ($J \models F$) and F is said to be *satisfiable*. SAT is NP-complete (Cook (1971)).

It is well-known, that each propositional formula can be transformed into a semantically equivalent formula in *conjunctive normal form (CNF)*, where a formula $F = c_1 \wedge \ldots \wedge c_m$ is in CNF if it is a conjunction of m clauses, a *clause* $c = L_1 \vee \ldots \vee L_n$ is a disjunction of n literals, and a *literal* L is either a propositional variable p or its negation $\neg p$. Most modern SAT solvers accept SAT instances in CNF. For more details about SAT we refer to the literature (Biere et al (2009)).

Encoding PESP to SAT enables the usage of efficient SAT solvers for solving PESP. We refer to Großmann et al (2012) for further reading. The task outlined in Section 1 results in a high number of usually conflict-ridden constraints. This means that initially formulated PENs are usually not feasible (e.g. see Opitz (2009)). Therefore, automatic conflict resolution becomes the most important pillar of automatic timetable computation. We refer to Großmann et al (2015) for further information on automatic conflict resolution in TAKT. Conflict resolution results in relaxation of arcs, meaning raising the upper bound $t_{max,a}$ or decreasing the lower bound $t_{min,a}$ of an arc a.

3 Extracting Local Conflicts

For infeasible PENs there exists a subset of the network's constraints, which still leaves the instance in an infeasible state. In real-world settings such a conflict helps understanding the automatically calculated timetable. Großmann (2012) defines local conflicts and a method for extracting local conflicts.

Definition 1 (Local Conflict) Let S be a set of arcs, called remainder, and $\mathcal{D}$ be a set of sets of arcs, such that $\mathcal{A} = S \cup \bigcup_{A \in \mathcal{D}} A$ for a PEN $\mathcal{N} = (\mathcal{V}, \mathcal{A}, m, t_T)$ with

$$\begin{aligned} &\forall A \in \mathcal{D} : A \cap S = \emptyset \text{ and} \\ &\forall A_1, A_2 \in \mathcal{D}, A_1 \neq A_2 : A_1 \cap A_2 = \emptyset. \end{aligned} \tag{3}$$

In addition, let $\mathcal{Z} \subseteq \mathcal{A}$ and $\mathcal{D}' \subseteq \mathcal{D}$ such that $\mathcal{Z} = S \cup \bigcup_{A \in \mathcal{D}'} A$.
Then a PEN $\mathcal{C} = (\mathcal{V}, \mathcal{Z}, m, t_T)$ is a *local conflict* if and only if $\mathcal{C}$ is infeasible and

$$\forall A \in \mathcal{D}' : (\mathcal{V}, \mathcal{Z} \setminus A, m, t_T) \text{ is feasible.}$$

The definition means that a local conflict is minimal with respect to the number of arc sets in $\mathcal{D}$ and the infeasible property. Each set of arcs $A \in \mathcal{D}$, represents a certain property. For example, a set of arcs could represent a certain route. Hence, removing this set of arcs would eliminate this route out of the network. The remainder S often represents the set of arcs, such that they predefine certain potentials in conjunction with the constraint mapping m. All sets of arcs in $\{S\} \cup \mathcal{D}$ shall be pairwise disjoint, which is stated in (3).

It exists a research field in SAT, which tries to extract minimally unsatisfiable subformulas. Thus, it reduces the amount of clauses by maintaining unsatisfiability. This can be hooked up to a higher level by minimizing sets of clauses. This will be stated in the following definition (Nadel (2010); Ryvchin and Strichman (2011)).

Definition 2 (High-Level Minimally Unsatisfiable Subformula) Let R be a set of clauses, called remainder, $\mathcal{G}$ be a set of sets of clauses and $F = R \wedge (\bigwedge_{G \in \mathcal{G}} G) \in \mathcal{L}(\mathcal{R})$ be in CNF with

$$\begin{aligned} &\forall G \in \mathcal{G} : G \cap R = \emptyset \\ &\forall G_1, G_2 \in \mathcal{G}, G_1 \neq G_2 : G_1 \cap G_2 = \emptyset. \end{aligned} \tag{4}$$

Then $M = R \wedge (\bigwedge_{G \in \mathcal{G}'} G) \subseteq F$ with $\mathcal{G}' \subseteq \mathcal{G}$ is a *high-level minimally unsatisfiable subformula* (HLMUS) if and only if M is unsatisfiable and $\forall G \in \mathcal{G}' : M \setminus G$ is satisfiable.

In real-world instances each $G \in \mathcal{G}$ represents a certain property. In the used SAT encoding, a constraint is represented as a set of clauses. Hence, minimizing the unsatisfiable core in such an instance means minimizing the number of sets of arcs in $\mathcal{G}$ respectively the domain given constraints, while maintaining the domain dependent given remainder R. As in case of local conflicts, in (4) is stated that all sets of clauses belonging to F are disjoint.

It has been shown in recent research (Nadel (2010); Ryvchin and Strichman (2011); Marques-Silva and Lynce (2011)) respectively the experimental results that it exists efficient HLMUS extractors, which perform well on a lot of instances[1].

The constraints of a Periodic Event Network can be encoded as clauses of the corresponding SAT instance. In PESP terms, a constraint $m(a)$ with respect to an arc a can be encoded via $enc\_con(a, m(a))$ (cf. Großmann et al (2012)). On the one hand, each set of arcs will be a set of corresponding encoded PESP constraints respectively its arcs $A \in \mathcal{D}$ as chosen in Definition 1. On the other hand, each constraint of all arcs in S must be always part of the conflict and must be encoded into the remainder R.

Definition 3 (Local Conflict Extraction) Let $\mathcal{N} = (\mathcal{V}, \mathcal{A}, m, t_T)$ be a PEN as in Definition 1,

$$R := \bigwedge_{n \in \mathcal{V}} enc(n) \wedge \bigwedge_{a \in S} enc\_con(a, m(a))$$

$$\mathcal{G} := \Big\{ \bigwedge_{a \in A} enc\_con(a, m(a)) \mid \forall E \in \mathcal{D} \Big\}$$

be the sets of Definition 2 and M be a corresponding HLMUS. Then $\mathcal{C} = (\mathcal{V}, \mathcal{Z}, m, t_T)$ with

$$\mathcal{Z} := \{a \in \mathcal{A} \mid \forall G = \bigwedge_{a \in A} enc\_con(a, m(a)) \in \mathcal{G}, A \in \mathcal{D} : G \subseteq M\} \cup S$$

is an extracted local conflict of $\mathcal{N}$.

Now, we can clearly extract an HLMUS by giving a PEN. For further reading like proofs of the soundness and completeness and a complexity classification we refer to Großmann (2012). Consecutive extraction of several conflicts is achieved by removing all arcs of the found conflicts from the PEN. This allows for extraction of multiple conflicts without the need to resolve the conflicts found before. Yet, not every conflict can be found during such cycle as some arcs may be part of several conflicts. In this situation, the latter conflicts may become feasible by the removal of common arcs and cannot be extracted without solving the first conflict.

Conflicts are considered unresolvable, if they cannot be resolved, even if all arcs are relaxed by the maximum possible value $t_T - 1$. To test for any unresolvable conflicts, all relaxable arcs are removed from the PEN. If this reduced network is infeasible, there exists an unresolvable conflict. To extract all unresolvable local conflicts, the previously described method has to be applied on the accordingly reduced PEN.

4 Extraction of Significant Conflicts

Conflict extraction within a large and complex PEN will likely result in a large number of conflicts. Yet, most of these conflicts can be resolved by a low

[1] `http://www.satcompetition.org/2011/`

amount of relaxation minutes per arc as application on real-world timetabling problems has shown (see Großmann (2012)). As small relaxations mean only slight deviations from original constraints and their impact on travelling times is negligible, only a selection of conflicts which have more impact is of interest for further analysis as described in Kümmling (2014). Therefore, a criterion is needed to get a much smaller selection of significant conflicts.

So far, the only known universal method to evaluate conflicts in terms of this paper is to measure the effects of the conflict's resolution, meaning the needed relaxation minutes. Provided there is a certain relaxation still acceptable, this relaxation can be applied to all relaxable arcs within the PEN. After application of the acceptable relaxation (threshold), all remaining conflicts cannot be solved by an acceptable, small relaxation. Such conflicts are considered significant.

For each arc an individual threshold can be defined. Due to the high number of relaxable arcs, suitable thresholds have to be assigned to different kinds of time consuming processes represented by the arcs: stops, transfers, symmetry and intervals between trains.

Minimum safe headways depend on the train's running times. As PESP does not support dependencies between different constraints, we generally consider headway and running time constraints not relaxable.

The relaxation of stopping times serves different purposes in practice:

1. passing of trains with different average speed and meeting of trains on one-track lines
2. establish connections between different trains
3. synchronisation of different trains to achieve regular intervals on all lines

Passing of passenger trains by other passenger trains is usually discouraged, but is necessary on several lines. Meeting of trains on single-track lines obviously cannot be avoided, when the single-track section exceeds a specific length. Not all stations are suitable for passing or meeting of trains. The detection of suitable passing loops is possible by inspecting several features of the used microscopic infrastructure data. The method is based on the principles described in Pöhle et al (2012), for further reading we refer to Weiß et al (2015). Stopping times in stations not suitable for passing are considered not being relaxable, equalling to a threshold of 0.

The importance of connections depends highly on passenger flows. As described in Kümmling (2013), yet no method is known for reliably assigning passenger flows to trains and connections in a rather complex network without any timetable. Nevertheless, it is possible to identify reasonable transfers which possibly allow shorter travel times or additional journeys (see Kümmling (2013)). Additionally, usually trains wait only for connections to trains of the same or higher level within the hierarchy (usually regional trains vs. long-distance trains).

Extending stopping times for synchronisation of trains is especially important in the German railway network, as there are many line sections, which are served by different routes. But it is still desired, that all lines on a line section add up to a fairly regular service. Figure 1 shows an excerpt of the German long-distance train network for illustration. Relaxing stops for synchronisation

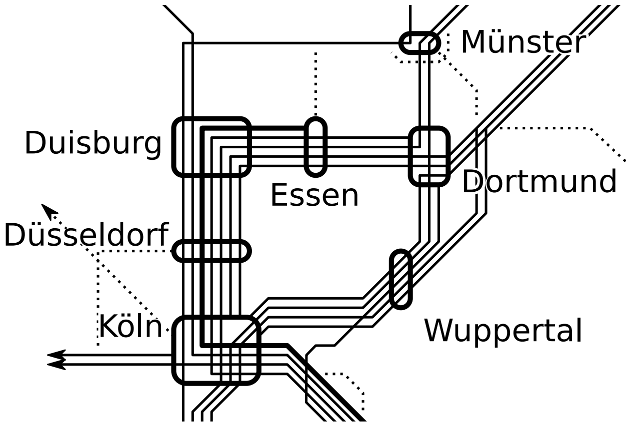

Fig. 1 excerpt of the German long-distance train network in the year 2013

or establishment of connections does not coercively require meeting or passing of trains. Nevertheless, we generally restrict relaxation of stopping arcs to stations suitable for passing of trains. Extending stopping times on other stations would lead to longer blockage of the only available station track and would obstruct other railway traffic. Connections and synchronisation mostly both happen in major division nodes. Hence we join both cases. We recommend separating stops into not relaxable, relaxable only for meeting/passing of trains and relaxable for connections or synchronisation. Suitable thresholds are yet to be found.

Despite the difficulties in automatically establishing sets of important connections, the German railway transport agencies impose large sets of strongly desired connections. These are modelled as transfer arcs. The threshold for these arcs conforms to the acceptable interchange waiting time. Symmetry and regular intervals between different routes are desired to achieve an easily memorable train service.

In Order to extract significant local conflicts, the following steps have to be conducted:

1. check whether PEN is resolvable
2. if network is unresolvable, extract unresolvable local conflicts and return
   if network is resolvable, relax each relaxable arc by individual threshold
3. extract all local conflicts from modified PEN

Narrowing down the number of conflicts to a limited set of significant conflicts induces the possibility to manually analyse the extracted conflicts and retrieve one or more causes for each single conflict. This allows either justifying the gained results or modify the used input data and the defined constraints.

The latter one is a common scenario, as faults in the microscopic infrastructure data, the detailed train data or the extensive number of additional requirements and connections happen quite often and cannot easily be avoided.

5 Application

For first tests, a simplified version of the above described methods was applied on a long-term study for a periodic timetable for the German passenger rail transport network. The network includes all periodic long-distance passenger trains and about 80 per cent of the most important regional and local trains. The associated PEN compromises 85258 arcs. A large number of stopping arcs in stations not suitable for passing/meeting of trains were automatically set unrelaxable by TAKT as described in Section 4. Additionally, further stopping arcs corresponding to stops in intermediary stations were set unrelaxable, as passing of passenger trains by other ones is usually discouraged, especially "out in the sticks" at small intermediary stations. All other arcs got one uniform threshold assigned. The described methods to distinguish different sets of arcs were not implemented yet.

Subsequently, the chosen threshold was increased by steps of one minute, as long as there were any significant conflicts. Table 1 shows the development of number of significant conflicts for an example. The computations were conducted on an Intel Xeon CPU E5-2690 based server with 256 GB RAM, but used not more than one core and 500 MB RAM. The SAT solver Glucose 4.0[2] is used for the initial solution of the SAT problem, the HLMUS solver MoUsSaka 1.16[3] is used for extraction of MUS.

The conflicts having the largest impact in conflict resolution are found in a few minutes. Compared to the computation time of at least two hours required for automatic conflict resolution of the whole network, this allows a fast extraction of the worst conflicts with respect to needed relaxation minutes.

The conflict size was measured by the number of enclosed arcs. Yet, there are no tools for analysis of conflicts and therefore, a reasonable size allowing an understanding of the conflict is arguable. Even though conflicts of all sizes allow locating the trains and connections causing the conflict. The conflict size fluctuation is likely reasoned by the problem that some conflicts may be masked by others, as described at the end of Section 3. Still, conflict sizes up to 300 arcs allow to narrow the analysis down considerably compared to the network's size of 85258 arcs.

6 Conclusions

The described method succeeds to extract significant problems even from large real-world timetabling problems in reasonable computation times. As

[2] http://www.labri.fr/perso/lsimon/glucose/
[3] http://www-pr.informatik.uni-tuebingen.de/?site=forschung/sat/SApperloT

Table 1 computational results for German passenger rail transport network

threshold in min	conflicts	time in min	min size	av. size	max size
0	≥ 165	> 140	≤ 3		≥ 202
1	≥ 120	> 140	≤ 3		≥ 587
2	≥ 79	> 140	≤ 3		≥ 196
3	65	140	3	49.59	1112
4	43	114	3	27.47	241
5	≥ 26	> 140	≤ 5		≥ 300
6	19	51	5	42.95	256
7	10	30	5	26.80	112
8	9	25	6	52.67	271
9	8	20	6	50.63	268
10	5	11	6	18.40	38
11	2	11	10	17.00	24
12	2	7	13	27.50	42
13	2	10	15	132.50	250
14	1	5	13	13.00	13
15	1	7	21	21.00	21
16	1	5	21	21.00	21
17	1	5	52	52.00	52
18	0	1	-	-	-
19	0	1	-	-	-
20	0	1	-	-	-

we only extract local conflicts, which are PENs too, automatic analysis and visualisation of the found conflicts may improve the evaluation and justification process further, especially for practitioners with low experience in PESP based timetabling.

The possibility to use different thresholds per arc type is yet to be implemented. Suitable thresholds have to be developed in close collaboration with practitioners to reach industrial application. Still, the presented criteria for significant conflicts are quite simple and do not consider many important factors in timetable evaluations. Especially the number of trainsets needed to run a route has a high economic impact. Further research is needed, to expand the presented method by such criteria.

References

Biere A, Heule M, van Maaren H, Walsh T (eds) (2009) Handbook of Satisfiability. IOS Press

Cook SA (1971) The complexity of theorem-proving procedures. In: Proceedings of the third annual ACM symposium on Theory of computing, ACM, New York, STOC '71, pp 151–158

FGSV (ed) (2001) Merkblatt zum Integralen Taktfahrplan. No. 137 in FGSV-Merkblatt

Großmann P (2012) Extracting and resolving local conflicts in periodic event networks. diploma thesis, Technische Universität Dresden

Großmann P, Hölldobler S, Manthey N, Nachtigall K, Opitz J, Steinke P (2012) Solving periodic event scheduling problems with SAT. In: Advanced Research in Applied Artificial Intelligence – 25th International Conference on Industrial Engineering and Other Applications of Applied Intelligent Systems, Springer, Lecture Notes in Computer Science, vol 7345, pp 166–175

Großmann P, Opitz J, Weiß R, Kümmling M (2015) On resolving infeasible periodic event networks. In: Conference on Advanced Systems in Public Transport, Rotterdam, (under review)

Kümmling M (2013) Umlegungsbasierte Fahrplanoptimierung von Personenverkehrstaktfahrplänen. research paper, Technische Universität Dresden

Kümmling M (2014) Erkennung und Auswertung von signifikanten Konflikten in der Taktfahrplanung. diploma thesis, Technische Universität Dresden

Kümmling M, Großmann P, Nachtigall K, Opitz J, Weiß R (2013) The state-of-the-art realization of automatic railway timetable computation. In: Proceedings of the 3rd International Conference on Models and Technologies for Intelligent Transportation Systems 2013, Dresden, pp 397–405

Lichtenegger M (1990) Der Taktfahrplan – Abbildung und Konstruktion mit Hilfe der Graphentheorie, Minimierung der Realisierungskosten. phd thesis, Technische Universität Graz

Marques-Silva J, Lynce I (2011) On improving MUS extraction algorithms. In: Sakallah KA, Simon L (eds) SAT, Springer, Lecture Notes in Computer Science, vol 6695, pp 159–173

Nachtigall K (1998) Periodic network optimization and fixed interval timetable. habilitation thesis, Universität Hildesheim

Nadel A (2010) Boosting minimal unsatisfiable core extraction. In: Bloem R, Sharygina N (eds) FMCAD, IEEE, pp 221–229

Opitz J (2009) Automatische Erzeugung und Optimierung von Taktfahrplänen in Schienenverkehrsnetzen. Gabler Verlag, also: PhD thesis, Technische Universität Dresden

Pöhle D, Weiß R, Opitz J, Nachtigall K (2012) Neuartiges automatisches Routing in realen Eisenbahnnetzen mit dem Programmsystem TAKT. Eisenbahntechnische Rundschau 61(10):58–65

Ryvchin V, Strichman O (2011) Faster extraction of high-level minimal unsatisfiable cores. In: Sakallah KA, Simon L (eds) SAT, Springer, Lecture Notes in Computer Science, vol 6695, pp 174–187

Serafini P, Ukovich W (1989) A model for periodic scheduling problems. SIAM Journal on Discrete Mathematics 2(4):550–581

Weigand W (2012) Von der Angebotsplanung über den Langfristfahrplan zur Weiterentwicklung der Infrastrukturd. Eisenbahntechnische Rundschau 61(7+8):18–25

Weiß R, Opitz J, Großmann P, Kümmling M (2015) Automatische Nutzung von Überholgleisen bei der optimalen Planung von Güterverkehrstrassen. Eisenbahntechnische Rundschau (under review)