Variable Neighborhood Search for Integrated Timetable Based Design of Railway Infrastructure

Igor Grujičić[1], Günther Raidl[1], Andreas Schöbel[2]

[1] *Institute of Computer Graphics and Algorithms*
Vienna University of Technology , Vienna, Austria

[2] *Institute of Transportation, Research Centre for Railway Engineering,*
Vienna University of Technology , Vienna, Austria

Abstract

In this paper we deal with the problem of building new or extending an existing railway infrastructure. The goal is to determine a minimum cost infrastructure fulfilling the requirements defined by an integrated timetable and the operation of the railway system. We first model this planning task as a combinatorial network optimization problem, capturing the essential aspects. We then present a metaheuristic solution method based on general variable neighborhood search that makes use of a dynamic programming procedure for realizing individual connections. Computational experiments indicate that the suggested approach is promising and the analysis of obtained results gives useful hints for future work in this area.

Keywords: Railway Infrastructure Design, Integrated Timetables, Dynamic Programming, Variable Neighborhood Search.

[1] Email: {grujicic|raidl}@ads.tuwien.ac.at
[2] Email: andreas.schoebel@tuwien.ac.at

1 Introduction

The design of new railway infrastructure is nowadays strongly guided by the prespecified integrated timetables that have been derived from expected traffic to be served [4]. Integrated timetables synchronize the traffic in major nodes (hubs, e.g., main railway stations in major cities) at regular time intervals, ensuring connectivity between different lines with minimum waiting times at those hubs, and allowing passengers to easily remember the regular departure and arrival times.

The Timetable Based Design of Railway Infrastructure (TTBDRI) problem can be summarized as follows: The aim is to extend an existing or to build a new railway infrastructure in such a way that all scheduled connections can be realized according to the given (integrated) timetable and costs are as low as possible. Implementing the concept of integrated timetables imposes major challenges and constraints, see e.g. [1]. Computational complexity of TTBDRI is discussed in [6].

In this paper we present a concrete combinatorial approach for modeling the basic problem which builds upon our former work [2]. It considers already existing railway infrastructure as well as extension possibilities in a fine-grained way. We then suggest a variable neighborhood search algorithm for approximately solving this problem, which makes use of a dynamic programming procedure for realizing individual connections locally optimal.

2 Problem Definition

We are given the following input data.

- An undirected graph $G = (V, E)$ represents the existing railway infrastructure plus all possible extensions. The node set V contains different types of nodes, first of all the following *infrastructure nodes* corresponding to real objects:
  - **track segment nodes** representing physical, simple track segments of a certain length, they always have at most degree two;
  - **switch nodes** representing classical switches, they have degree three (or possibly higher if more complex switches are modeled by single nodes);
  - **crossing nodes** representing crossings of two lines, which always have degree four;
  - **signal position nodes** representing signaling stations; they always have degree two.

  To model mutually exclusive alternatives for infrastructure extensions, we further use:

· **alternative nodes**, which have degree $k + 1$ for k mutually exclusive options.

Edges E represent the corresponding connections of the respective nodes. Parallel tracks are always represented by multiple paths.

- Let $V^{\mathrm{R}} \subseteq V$ be the set of signal position nodes. Paths starting and ending at such nodes and otherwise containing only nodes from $V \setminus V^{\mathrm{R}}$ are called (compound) routes. Once a compound route is reserved for a train, no other train is allowed to enter any part of this route before the train has left and the route is released again.
- Let the subgraph $G^0 = (V^0, E^0)$, with $V^0 \subseteq V$ and $E^0 \subset E$, correspond to the already existing infrastructure, and graph $G' = (V', E')$ with $V' = V \setminus V^0, E' = E \setminus E^0$ represent the infrastructure by which the existing infrastructure may be extended. Alternative nodes are considered to be part of V^0 iff one of the modeled options corresponds to an existing infrastructure, virtual nodes are part of V^0 if both adjacent nodes are also in V^0. All nodes $v \in V$ have associated costs $c_v \geq 0$ and lengths $l_v \geq 0$, with $c_v = 0$ for alternative nodes, virtual nodes, and all nodes in $v \in V^0$, $l_v = 0$ for signal position nodes, alternative nodes and virtual nodes.
- Set S represents the major railway stations considered in the integrated timetable. Each railway station $s \in S$ has associated a set of simple track segment nodes $V(s) \subset V$ corresponding to the tracks at the platforms for boarding/disembarking trains.
- Let $G^{\mathrm{D}} = (V, A)$ be the directed version of graph G, where we have for each edge $(u, v) \in E$ two corresponding oppositely directed arcs $(u, v), (v, u) \in A$.
- An integrated timetable specifies a set of connections $C = \{C_1, \ldots, C_{|C|}\}$ to be realized, where a connection $C_\ell \in C$ is a tuple (s_ℓ^{start}, s_ℓ^{end}, T_ℓ^{start}, T_ℓ^{end}, $G_\ell^{\mathrm{D}}, \mathit{train}_\ell, l_\ell$) with $s_\ell^{\mathrm{start}}, s_\ell^{\mathrm{end}} \in S$ being start and destination stations and T_ℓ^{start} and T_ℓ^{end} the times when the train may leave station s_ℓ^{start} and has to arrive at station s_ℓ^{end} latest, respectively. The connection has to be realized by a path in a given subgraph $G_\ell^{\mathrm{D}} = (V_\ell, A_\ell)$ with $V_\ell \subseteq V$ and $A_\ell \subseteq A$. It can safely be assumed that G_ℓ^{D} is acyclic. Finally, train_ℓ indicates the used train's ID. Typically, a train is used for a series of connections. Let $l(\mathit{train}_\ell)$ refer to the train's length.
- Values $m_{\ell,v} \geq 0$ indicate the maximum allowed average speed by which the train realizing connection $C_\ell \in C$ may go over node $v \in V_\ell$.

A solution consists of:

- A subgraph $G'' = (V'', E'')$ with $V'' \subset V$ and $E'' \subseteq E'$ indicating the infrastructure to be installed.
  Let $G^{\mathrm{e}} = (V^{\mathrm{e}}, E^{\mathrm{e}})$ represent the complete augmented infrastructure, i.e., $V^{\mathrm{e}} = V^0 \cup V''$ and $E^{\mathrm{e}} = E^0 \cup E''$.
- For each connection $C_\ell \in C$ a directed path $P_\ell \subseteq A_\ell$ starting at a node from $V(s_\ell^{\mathrm{start}})$, ending at a vertex from $V(s_\ell^{\mathrm{end}})$. Let $V(\mathrm{P}_\ell) \subseteq V_\ell$ be the set of all nodes on this path.
  Considering the signal position nodes V^{R} as separators, P_ℓ can be partitioned into the ordered list of compound routes $L_\ell = (P_{\ell,1}, \ldots, P_{\ell,\lambda_\ell})$ with corresponding node sets $V(P_{\ell,1}), \ldots, V(P_{\ell,\lambda_\ell})$. The length of route $P_{\ell,i}$, $i = 1, \ldots, \lambda_\ell$, is $l(P_{\ell,i}) = \sum_{v \in V(P_{\ell,i})} l_v$.
- For each (infrastructure) node $v \in V(P_\ell)$, $C_\ell \in C$, an actual (average) speed $\mathit{speed}_{\ell,v}$ that does not exceed the limit $m_{\ell,v}$. Consequently, the train takes time $T_{\ell,v} = l_v / \mathit{speed}_{\ell,v}$ for passing node v.
- For each route $P_{\ell,i}$, $i = 1, \ldots, \lambda_\ell$, $C_\ell \in C$, a reservation time slot $(T_{\ell,i}^{\mathrm{enter}}, T_{\ell,i}^{\mathrm{exit}})$ in which the train will be able to pass this route.

To be feasible, a solution must satisfy:

- For each connection $C_\ell \in C$: $\forall (u,v) \in P_\ell \to u, v \in V^{\mathrm{e}} \wedge (u,v) \in E^{\mathrm{e}}$, i.e., the infrastructure used in the chosen paths must exist or be installed.
- All constraints for realizing possible extensions (e.g., mutual exclusivity of some alternatives) must be adhered.
- The time slots of consecutive routes of a connection overlap by a certain safety margin.
- For each connection $C_\ell \in C$, the earliest start and latest arrival times T_ℓ^{start} and T_ℓ^{end} are adhered, respectively.
- At each time, each node $v \in V^{\mathrm{e}} \setminus V^{\mathrm{R}}$ (i.e., except signal position nodes) may only be part of at most one reserved route.
- If a train is used for two successive connections, its arrival node at the station's platform must be the same as the node where it leaves from later.

The objective is to find a feasible solution with minimum total costs $\sum_{v \in V''} c_v$.

3 Solution method

We apply a general variable neighborhood search (VNS) scheme with embedded variable neighborhood descent (VND) as proposed in [3]. A solution is represented by a permutation π of the connections C to be realized and the

VNS neighborhood structures are defined on it. Each generated candidate permutation is decoded into complete solution as defined above by a construction heuristic that considers each connection in the order given by the permutation and realizes it with local minimal costs by dynamic programming (DP).

The following section presents this DP, while Section 3.2 describes the VNS framework.

3.1 Dynamic programming

Let $C_\ell \in C$ be the (single) connection for which we want to find a feasible solution.

To cover the aspect that a train may possibly start from and end at different platforms we add start and end nodes σ and τ to the set V_ℓ of the connection C_ℓ, i.e., $V'_\ell = V_\ell \cup \{\sigma, \tau\}$. Furtheron, we define the extended arc set $A'_\ell = A_\ell \cup \{(\sigma, s) \mid \forall s \in V(s^{\text{start}}_\ell)\} \cup \{(s, \tau) \mid \forall s \in V(s^{\text{end}}_\ell)\}$.

For every node $v \in V_\ell \backslash V^{\text{R}}_\ell$ we calculate a minimum reservation time $\Delta T^{\min}_{\text{R},v}$, considering just predecessor and successor nodes, as: $\min\{l_u/m_u | (u, v) \in A'_\ell\} + \min\{l_u/m_u | (v, u) \in A'_\ell\} + l_v/m_v$.

For every node $v \in V_\ell \setminus V^{\text{R}}_\ell$ we define a set Y_v of time intervals in which it may be possible to reserve node v for the train to pass it. Every such time interval of Y_v has to have a duration of at least $\Delta T^{\min}_{\text{R},v}$.

Our DP stores labels $(c, T^{\text{start}}_{\text{R}}, T^{\text{end}}_{\text{R}}, t, \pi)$ for every reached node $v \in V'_\ell$, where:

(i) π represents the preceding node;
(ii) c represents the accumulated costs for the path from σ to v including c_v;
(iii) $T^{\text{start}}_{\text{R}}$ represents the earliest time from which the reservation of the node v may start;
(iv) $T^{\text{end}}_{\text{R}}$ represents the latest time until which the reservation of the node v may last;
(v) t represents the earliest arrival time at node v in $[T^{\text{start}}_{\text{R}}, T^{\text{end}}_{\text{R}}]$.

The initial label for node σ is: $(0, T^{\text{start}}_\ell, T^{\text{end}}_\ell, T^{\text{start}}_\ell, null)$.

The extend function $(c, T^{\text{start}}_{\text{R}}, T^{\text{end}}_{\text{R}}, t, \pi) \to (c', T^{\text{start}'}_{\text{R}}, T^{\text{end}'}_{\text{R}}, t', \pi')$ for considering as next step to go from node u to node v is: $c' = c + c_v$, $\pi' = u$ and for the calculation of the t', $T^{\text{start}'}_{\text{R}}$ and $T^{\text{end}'}_{\text{R}}$ we need to distinguish the following cases:

(i) when $v \in V(s^{\text{start}}_\ell)$, then for every $[T^{\text{low}}, T^{\text{up}}] \in Y_v \Rightarrow T^{\text{start}'}_{\text{R}} = T^{\text{low}}$, $T^{\text{end}'}_{\text{R}} = T^{\text{up}}$ and $t' = T^{\text{low}}$;

(ii) when $v \in V_\ell^{\mathrm{R}} \cup \{\tau\}$, then $T_{\mathrm{R}}^{\mathrm{start}\prime} = T_{\mathrm{R}}^{\mathrm{start}}$, $T_{\mathrm{R}}^{\mathrm{end}\prime} = T_{\mathrm{R}}^{\mathrm{end}}$ and $t' = t + l_u/m_u$;

(iii) when $v \in V_\ell \setminus (V(s_\ell^{\mathrm{start}}) \cup V_\ell^{\mathrm{R}} \cup \{\tau\})$, we distinguish the following:

- preceding node $u \in V_\ell^{\mathrm{R}}$, then for every $[T^{\mathrm{low}}, T^{\mathrm{up}}] \in Y_v$ having nonempty intersection with $[t, T_{\mathrm{R}}^{\mathrm{end}}] \Rightarrow T_{\mathrm{R}}^{\mathrm{start}\prime} = \begin{cases} t & \text{, if } T^{\mathrm{low}} \leq t \\ T^{\mathrm{low}} & \text{, otherwise} \end{cases}$, $T_{\mathrm{R}}^{\mathrm{end}\prime} = T^{\mathrm{up}}$ and $t' = T_{\mathrm{R}}^{\mathrm{start}\prime}$, nothing otherwise;
- preceding node $u \notin V_\ell^{\mathrm{R}}$, then for every $[T^{\mathrm{low}}, T^{\mathrm{up}}] \in Y_v$ having nonempty intersection with $[T_{\mathrm{R}}^{\mathrm{start}}, T_{\mathrm{R}}^{\mathrm{end}}] \Rightarrow [T_{\mathrm{R}}^{\mathrm{start}\prime}, T_{\mathrm{R}}^{\mathrm{end}\prime}] = [T^{\mathrm{low}}, T^{\mathrm{up}}] \cap [T_{\mathrm{R}}^{\mathrm{start}}, T_{\mathrm{R}}^{\mathrm{end}}]$ and $t' = \left(T_{\mathrm{R}}^{\mathrm{start}\prime} - T_{\mathrm{R}}^{\mathrm{start}}\right) + (t + l_u/m_u)$, nothing otherwise.

An extension is feasible iff the following two conditions hold. (a) The actual arrival time at node v has to be feasible, i.e, $t' \in [T_{\mathrm{R}}^{\mathrm{start}\prime}, T_{\mathrm{R}}^{\mathrm{end}\prime}]$. (b) A time exists at which the train can pass from the previous route to the current one. This is expressed as $T_{\mathrm{R}}^{\mathrm{start}\prime} \in [T_{\mathrm{W}}^{\mathrm{start}}, T_{\mathrm{W}}^{\mathrm{end}}]$ where $[T_{\mathrm{W}}^{\mathrm{start}}, T_{\mathrm{W}}^{\mathrm{end}}] = [t, T_{\mathrm{R}}^{\mathrm{end}}]$ of the last signal position node on the path from σ to v.

Labels that are dominated by others can be removed. A label $l = (c, T_{\mathrm{R}}^{\mathrm{start}}, T_{\mathrm{R}}^{\mathrm{end}}, t, \pi)$ dominates a label $l' = (c', T_{\mathrm{R}}^{\mathrm{start}\prime}, T_{\mathrm{R}}^{\mathrm{end}\prime}, t', \pi')$ iff the reservation time interval $[T_{\mathrm{R}}^{\mathrm{start}}, T_{\mathrm{R}}^{\mathrm{end}}]$ of label l contains the reservation time interval $[T_{\mathrm{R}}^{\mathrm{start}\prime}, T_{\mathrm{R}}^{\mathrm{end}\prime}]$ of label l' and $c \leq c'$ as well as $t \leq t'$ and at least one of these inequalities holds strictly.

Once we have reached end node τ an actual solution is obtained by going backwards towards start node σ. In every step we calculate the reservation time interval for the visited node and the speed used for traveling through it.

3.2 VNS framework

The VND considers the following classical permutation neighborhoods in a next-improvement manner in the given order: Adjacent pairwise exchange (Swap), Exchange, Forward-Shift, Forward-Shift-Block, Backward-Shift, Backward-Shift-Block, Double-Shift. For their explanations see e.g. [5].

Shaking in VNS: In every VNS iteration we first randomly choose between k-Swap and k-Exchange with equal probability. When k-Swap (k-Exchange) is selected, k random swap (exchange) moves are performed.

It might happen that the decoding of a permutation $\pi = (p_1, p_2, \ldots, p_i, \ldots, p_{|C|})$ to a feasible solution cannot be performed as some connection p_i cannot be realized by the DP procedure as previously realized connections block essential part. Such infeasible permutations are discarded, but even more importantly, all permutations starting with the same prefix $(p_1, p_2, \ldots, p_i)$ will also fail. In order to prevent an unnecessary investigation of permutations with already known infeasible prefixes in the further search, we store all these prefixes in a trie data structure and pre-check each candidate solution with it before calling the decoding procedure.

4 Experimental results

All experiments were carried out on an Intel Core i7-860 processor on 2.80GHz with 8GB of RAM. All algorithms have been implemented in C++. Tests are run on instances obtained from real scenarios in Austria as well as on artificially generated instances. We compare a restricted enumeration heuristic (REH) to the VNS.

In REH a solution is primarily built by following a greedy constructive approach. In each iteration, all not yet realized connections are tried for realization, and a feasible one increasing the total costs the least is adopted. In the case none of the remaining connections can be feasibly realized, the procedure backtracks to the previous position in the permutation and tries the next-best feasible choice.

The VNS as well as REH were terminated after 200 or 1000 seconds depending on $|C| \leq 10$ or $|C| > 10$, respectively, or, in case of the VNS, when 10 major iterations without improvement have been performed. Due to its stochastic nature, we ran the VNS 10 times on each instance and report here mean values.

Table 1 presents results of our tests. Listed are for each instance its name, the number of nodes and edges as well as the number of connections in timetable C, objective values, CPU-times t_{total}, and times needed by VNS to obtain the finally best solution t_{best}. Best results are printed bold.

Table 1: Results on small set of instances.

Instance				REH		VNS		
Name	$\|V\|$	$\|E\|$	$\|C\|$	Obj. Value*	t_{total} [s]	Obj. Value*	t_{best} [s]	t_{total} [s]
generated832168	108	118	4	0.25	0.26	**0**	12.20	12.20
generated433233	190	208	4	**0**	0.31	**0**	2.47	2.47
generated123948	153	166	8	62.6923	0.20	**0**	184.67	184.67
generated373145	132	146	8	–	200	**0.25**	120.23	200
generated491776	307	340	8	0.25	35.99	**0**	89.97	89.97
FB_scenario2	211	219	10	33.865	166.75	**0**	34.14	34.143
FB_scenario3	211	219	10	39.09	0.41	**5.25**	122.21	200
FB_scenario4	211	219	18	–	1000	**3.705**	186.63	1000
Average times					175.49		94.07	215.43

*Objective values are given in Millions of Euro.

Obtained results show that although on average the VNS requires more time than REH, it achieves better solutions on all test instances. In two cases the VNS has found solutions for instances where REH terminated without finding any feasible solution. Taken all together, we can conclude that the presented approach is efficient for smaller to mid-size instances. The fact that the VNS, but also REH, were not able to find a feasible solution even with prolonged execution times for instances where $|C| > 30$ tells us however, that further improvements are necessary.

5 Conclusions and future work

In this article we have presented a formal combinatorial optimization model for the integrated timetable-based design of railway infrastructure. We have then suggested a metaheuristic approach for approximately solving this problem, which consists of a VNS framework in which an exact dynamic programming procedure is embedded for realizing individual connections. Obtained results appear reasonable and encouraging but also indicate the need of further algorithmic improvements to solve more complex practical scenarios effectively.

In future work we aim at applying advanced hybrid metaheuristics but also exact techniques based on mathematical programming methods like column generation and Benders' decomposition.

References

[1] Caimi, G., Laumans, M., Schüpbach, K., Wörner, S. and Fuchsberger, M., *The periodic service intention as a conceptual framework for generating timetables with partial periodicity*, Transportation Planning and Technology, **34(4)** (2011), 323–339.

[2] Grujičić, I., Raidl, G., Schöbel, A., and Besau, G., *A Metaheuristic Approach for Integrated Timetable based Design of Railway Infrastructure*, in Proceedings of the 3rd International Conference on Road and Rail Infrastructure – CETRA 2014, Split, Croatia, (2014) 691 – 696.

[3] Hansen, P., and N. Mladenović, *An introduction to variable neighborhood search*, Voss, S., et al., editors, Meta-heuristics, Advances and trends in local search paradigms for optimization, Kluwer Academic Publishers, Dordrecht, (1999), 433–458.

[4] Lichtenegger, M., *Der integrierte Taktfahrplan: Abbildung und Konstruktion mit Hilfe der Graphentheorie, Minimierung der Realisierungskosten*, Eisenbahntechnische Rundschau, **40** (1991), 171–175.

[5] Reeves, C., *Landscapes, operators and heuristic search*, Annals of Operations Research, **86**(1999), 473 – 490.

[6] Schöbel, A., Raidl, G. R., Grujičić, I., Besau, G., and Schuster, G., *An Optimization Model for Integrated Timetable Based Design of Railway Infrastructure*, in Proceedings of the 5th International Seminar on Railway Operations Modelling and Analysis - RailCopenhagen 2013, Copenhagen, Denmark, (2013) 765–774.