Train platforming with scenario based robustness: An exact biobjective method

Kamenga, Franck[1,2], Leloup, Leonard[1], Houzel, Bertrand[1] and Merabet, Boubekeur[1]

[1]SNCF Réseau, Direction Générale Ile-de-France, DPF Solutions, France
[2]Université Lille-Nord-de-France, IFSTTAR COSYS/ESTAS, France

Abstract

Railway stations are a key element in the management of railway capacity because of the constraints associated with their platforms and the complex track system allowing trains to access them. More precisely train platforming problem (TPP) consists in routing trains through a station with respect of safety and commercial requirements. Otherwise, routing has to cope with usual disturbances such as delays or track closures. Yet, various forms of residual capacity ensure easy recovering. The residual capacity defines two main recovery robustness indicators: the respect of buffer times in order to deal with frequent delays, the assignment of back-up routes in order to deal with track closure. As those two aspects are generally contradictory, this paper proposes a bi-objective approach. It allows exploring compromise solutions. We extend an integer linear programming formulation already implemented in the decision-aiding tool OpenGOV developed at SNCF Réseau. This algorithm leads to experimentation with OpenGOV framework. We tackle real instances of traffic at main French stations: Paris Gare de Lyon, Paris Est and Lyon Part-Dieu. We obtain Pareto fronts in a reasonable time. The algorithm gives many relevant solutions that could not be obtained with a simple stable set formulation.

Keywords: Robust optimization; Train Routing; Train Platforming; Recovery Robustness

1 Introduction

Railway capacity is a crucial issue for busy networks like many European ones. A good management of the capacity maximizes the usage rate of the infrastructure, thus it increases its rentability. Stations and complex junctions strongly affect railway operation [12]. Infrastructure managers need to plan traffic in those junctions by routing trains with a given timetable, it is the train platforming problem (TPP). Routing trains through stations is of major interest when it is necessary to accommodate a dense traffic on a limited infrastructure. Therefore, computer-aided train platforming is relevant for the management of saturated junctions. The problem is first and foremost a response to a strategic requirement, since it is useful for infrastructure sizing. In this way, it is possible to evaluate the gain in flow rate that new railway station facilities make. However, the TPP is mostly a tactical issue, as it primarily reflects planning for track allocation. It follows timetabling and checks its feasibility.

Many works show that the resolution of the TPP is tackled well thanks to combinatorial optimization. A first kind of approach uses scheduling models [11] [10]. Besides, many are based on conflicts graphs. Those specify incompatibilities with a a given timetable. In order to provide a conflict-free allocation, a colouring model can be considered [1]. Significant theoretical and experimental results were given by a stable set formulation solved by linear programming [5].

As the management of traffic in busy stations is a major lever for the quality of railway operations, the need for rationalization of resources is growing. Therefore, SNCF Réseau the main French infrastructure manager developed the software OpenGOV, a decisionaiding tool to deal with railway capacity in stations. It incorporates an optimization engine which solves the train platforming problem. OpenGOV is deployed at a twenties of scheduling services in French main stations.

Furthermore, recent railway planing contributions show increasing interest in robustness. It is a crucial point in train platforming, as it is a primordial requirement. Indeed, a routing must be adapted to the various disturbances. A platforming must bring resiliency in railway operation. It is harmful that a single incident has serious consequences (delays, train cancellations). The first framework is that of small delays. Domino effect is avoided to ease recovery. The second framework concerns heavier problems (signal failures, hardware failures, etc.) requiring a modification of the routing. Thus, some routings are perceived as more robust by operators. On the one hand, they consider that the routing must be stable to delays. On the other hand, it must allow by limited changes an adaptation to disturbed situations frequently encountered. This paper tackles these two approaches for robust train platforming. In order to cope with these two aspects, we propose an algorithm based on multiobjective combinatorial optimization.

2 Problem description

2.1 The train platforming problem

The train platforming problem can be described as followed. Assuming a station track configuration, train timetables as well as origins and destination, we would like to know if all trains can pass through the station without conflicts and respect possible operational requirements.

The layout of the station gives a set of routes R. We introduce a set of trains T. For a train $t \in T$, the set of feasible its assignations to a route is denoted by V_t. $V = \cup_{t \in T}$ is the set of all assignations. Every assignation $v \in V$ has an utility ω_v which attests to the importance of the assignment of the train on the given route. The weights take account of preferred platforms for particular service. For an assignation $v \in V$ we denote $t(v) \in T$ the train routed. As trains schedules are given, we can determine if the assignation v is compatible to the allocation of another assignation v^0.

A conflicts graph [13] for the TPP is an undirected graph denoted $G = (V,E)$, whose:

- nodes V are the set of assignations, with a strictly positive integer weight ω_v.
- edges represent the incompatibilities between assignations, there is an edge for acouple of conflicted allocations as well as assignments of a same train

Thus the TPP comes down to the search of a maximum stable set (or node packing) in the conflicts graph. We can notice that nodes V_t associated to a same train $t \in T$ in the conflicts graph constitute a clique. These cliques partition the set of vertices.

2.2 Robustness issues

It is believed that platforming will be more robust if it minimizes the risk of traffic disruption with unstable schedules. Otherwise, backup platforms facilitates recovery schemes under disturbances. They are particularly relevant to cope with temporary track closures. If a train is assigned to a route, an alternative route which uses a neighbour platform must be free to have a backup.

On one hand, most hypothetical conflicts are removed to limit delay propagation. Thus the planning process includes buffer times. This leads to routings which spreads the trains over the entire infrastructure. On the other hand, to cope with temporary track closures there must be large time slots when a part of the station is free. Thus, these two visions of residual capacity are contradictory. The present paper deals with these two approaches to elaborate compromises.

3 Related works

3.1 Concepts in Robustness

As robust planning provides disturbance-resistant solutions, scenario-based approaches are quite relevant. A set of scenarios S is given for a problem, it gathers the most representative situations. The occurrence of a scenario can modify costs or profits, so the objective function. However, this case is marginal in railway planning. But the decision space is often specific to a scenario. And yet a strict robust solution must be feasible for all scenarios. This very conservative approach leads to sub-optimal solutions.

In order to seek for a compromise with optimality light robustness [8] allows the solution to be infeasible in some scenarios and guarantees a threshold value for the objective function. A such problem consists in minimizing penalties due to the violation of constraints made to ensure feasibility in disturbed situations. Besides that, recovery robustness [9] takes into account the adaptation process during operations. Indeed, replanning can be done if an infeasibility occurs. In this approach a solution gets feasible if a recovery algorithm exists for every scenario. This algorithm builds a feasible alternative from the current solution. Nevertheless, the recovery problem is often more complex than the original one.

3.2 Robustness in train platforming: a state of art

Many contributions deal with delay-resilience in train platforming. Buffer times are common solution in order to obtain a feasible platforming in case of late. Indeed, timetables are given with tolerance intervals the model must cope with. Thus small delays that often occurs should be absorbed [7]. The buffers indicate extra time intervals that should separate two trains using a same track section. A platforming remains proper if the buffer times are enough to guarantee a stability in the timetable. Alberto Caprara and al. [4] utilizes traffic conflicts at platforms and switch areas to obtain a delay propagation network. Then an integer program includes this network and the necessary buffers. Other formulations based on scheduling models [6], [12] allow deviations from the initial schedule as well as modifying order of the trains.

Such sophisticated formulations can be avoided if the detailed delay propagation is neglected. It is enough to add buffer times at every conflicts to take account of timetable tolerances. We should note that this method gives greater buffer times but is much easier to implement. Many works introduce a penalty for assignations which do not respect buffers. In the node-packing formulation, the conflicts graph is enriched with penalty edges denoted A which link couples of assignations violating buffer times recommendation. As a penalty is activated when both assignations in a the couple (v,v^0) $\in A$ are chosen, a quadratic objective function is natural. Though, a convenient linearization can be performed [5] with variables which indicates if a not recommended couple of assignations is chosen. Alberto Caprara and al. deal with these additional variables thanks to a pricing and the separation of the constraints (4). Gabrio Caimi and al. [2] implement a local search algorithm to minimize the conflicts. While Alberto Caprara and al. [5] and Gabrio Caimi and al. [2] implement penalties linear in the time interval separating two conflicted trains, Thijs Dewilde and al. [7] use a multiplicative inverse function. This refinement often brings more delay resistant solutions.

Alberto Caprara and al. [3] introduce back-up platforms in the node-packing formulation. These platforms belong to the neighbourhood of the assigned path and must be free during the whole occupancy time. Nevertheless the model does not require a route which leads to the back-up platform to be free.

4 Backup platforming algorithm with buffer times

4.1 Bi-objective Formulation

We denote $G_R = (V,E \cup A)$ an enriched conflicts graph. The weight of a penalty edge $a \in A$ is denoted c_a. We introduce the binary decision variables x_v for every node $v \in V$ where $x_v = 1$ if the assignation is chosen and $x_v = 0$ otherwise. In addition to ordinary stable set constraints (Equation 2), the formulation is strengthened with few clique constraints (Equation 3). Theses cliques are the assignations to a same train. For a penalty edge $a \in A$ we denote y_a a binary variable equal to 1 if its both nodes are picked and 0 otherwise. Then, constraints in the Equation 4 are stated. A first objective f_{TR} (Equation 1) gives the total penalty of buffer times violation.

For a node $v = (t,r) \in V$ we denote $N(v)$ assignations of the train t which uses a nearby platform. In other words $N(v)$ represents potential backup routing or a the neighbourhood of v. We introduce the binary decision variable π_v for the assignment to a backup platform. $\pi_v = 1$ when the node v is picked as a backup and $\pi_v = 0$ otherwise. As a backup route must belong to neighbourhood of the initial route, the formulation contains the Equation 5. It ensures that a train can have a backup platform only if it is allocated to an initial route. Moreover, a train can obviously have only one backup route (Equation 6). In order to guarantee the backup route to be free, we introduce the Equation 7. A second objective function f_{BR} (Equation 1) corresponds to the weights of the trains assigned without backup route.

We denote **Ω** the expected performance of the platforming. It can be the weight of the maximum stable set obtained without considering any robustness aspects or a lower value. The equation 8 imposes a minimal size for the solution.

$$\min\left(f_{TR}=\sum_{a\in A}c_a y_a\,,\ f_{BR}=\sum_{v\in V}\omega_v\,(x_v-\pi_v)\right) \tag{1}$$

$$x_v+x_{v'}\leq 1 \quad \forall(v,v')\in E \tag{2}$$

$$\sum_{v\in V_t}x_v\leq 1 \quad t\in T \tag{3}$$

$$y_a\geq x_v+x_{v'}-1 \quad a=(v,v')\in A \tag{4}$$

$$\sum_{v'\in V_{t(v)}\setminus N(v)}\pi_{v'}+x_v\leq 1 \quad v\in V \tag{5}$$

$$\sum_{v\in N(v)}\pi_{v'}\leq x_v \quad v\in T \tag{6}$$

$$x_v+\pi_{v'}\leq 1 \quad \forall(v,v')\in E:t(v)\neq t(v') \tag{7}$$

$$\sum_{v\in V}\omega_v x_v\geq\Omega \tag{8}$$

$$x_v,y_a,\pi_v\in\{0,1\} \quad \forall\, v\in V,a\in A \tag{9}$$

The backup robust train platforming problem with buffer times (BTR-TPP) is the search of a solution of the program (2-9) which optimize the objectives (1).

4.2 Robustness proprieties

We can notice that two trains can have the same backup platform. Nevertheless, this aspect has a low impact to recovery. Indeed, in this paragraph we would like to study recovery with a BTR-TPP solution when a platform gets temporary closed. Let us assume that all assigned trains have a back-up platform. Then, the trains which occupies this platform are rerouted thanks to the backup solution. What guarantees that the routing is feasible?

We can consider a station whose layout is given in the figure 1. Two trains are assigned to the platform C with the green route. The first train arrives at 08:17 and leaves the station at 08:30, the second train arrives at 08:35 and leaves at 08:50. A potential conflict occurs during the train 1 departure and the train 2 arrival. But the train 1 leaves the conflict zone a minute and a half after its departure time and the train 2 enters in the zone a minute and a half before its arrival time. Therefore, the initial routing is feasible. The train 1 is assigned to the backup platform E and the train 2 to the platform D. They both use the purple route, which gives a wider conflict zone using the switches 2, 3 and 4. Here train 1 leaves the conflict zone 3 minutes after its departure time and the train 2 enters in the zone 3 minutes before its arrival time. There should be a 6 minutes headway between the two trains, to have a feasible backup routing.

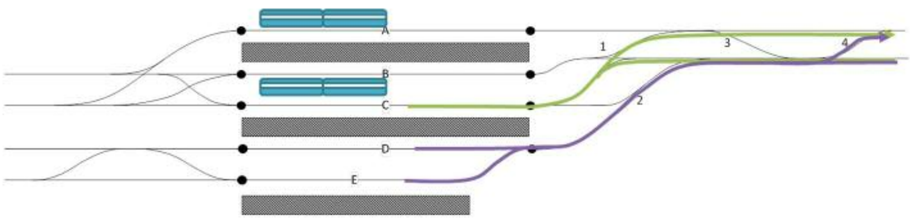


Figure 1: Two trains are assigned to the green route and have the purple backup routes.

The infeasibility is due to an asymmetric layout of the station given in the example. And yet, there are many stations where this problem does not exist. Therefore, we must focus on the nature of traffic conflicts. First of all, a platform track is occupied by train a while after its departure time (the moment it starts to move). Then if a following train wants to get in the same track, it will have to wait a release time. This forms the reoccupation conflict, whose headway times also called reoccupation times depends of the relative directions. Conflicts can also occur in the switch area. But sometimes the conflict zones are not adjacent to platform tracks. Trip times from and to the station are taken into account. Whatever the positions of conflict zones, headway times are specified between arrival or departure times.

Definition 1. The layout of a station has a short switch area if for every platform reoccupation times for given directions and rolling stock features are grater than headway times in the part of the switch area leading to the platform.

Definition 2. The layout of a station is said to be symmetric if reoccupation times do not depend of platforms.

Lemma 3. *In a symmetric station with a short switch area, if two trains t_1 and t_2 can be assigned to the same platform then there are no conflicts between any assignment of the two trains.* $\forall v_1 \in V_{t1}, v_2 \in V_{t2}, (v_1, v_2) \notin E$.

Proof. As the trains t_1 and t_2 can occupy the same platform their schedules respect their platform headway times. As the station is symmetric they also respect headway times for all platforms. Moreover, the switch area is short, thus headway times in the switch area are fulfilled too. Hence, the safety problem brings no conflicts between assignations of the trains t_1 and t_2. □

Theorem 4. *In a symmetric station with a short switch area, a BTR-TPP solution, where all assigned trains have a back-up platform, is recovery robust for the temporary closure of any platform.*

Proof. We denote (x^*, π^*, y^*) the solution of the BTR-TPP. For a given platform p, we consider its temporary closure. Let T_p^* be the set of trains assigned to this platform during its closure. We consider the following set of of nodes $\rho = \{v \in V_t : t \in T \setminus T_p^*, x_v^* = 1\} \cup \{v \in V_t : t \in T_p^*, \pi_v^* = 1\}$. This assignment re-routes the trains of T_p^* to their backup platform. We have to check if ρ is feasible. Obviously thanks to the constraints (7) there are no conflicts between backup assignations and the primary ones. As the trains in T_p^* share the same platform, the lemma 3 ensures no conflicts between backup assignations.

□

Remark 5. If the station has a short switch area and the neighbourhood of all assignations only contains routes with lower reoccupation times then we have the result of the Theorem 4.

4.3 Search of efficient solutions

In order to ease the study, solutions are represented in the criterion space (Buffer times penalties, Back up routes penalties). For reader's convenience, a solution (x,y,π) can be denoted ξ. Thus, a routing ξ can be represented with the point $(f_{TR}(\xi), f_{BR}(\xi))$. We denote X the set of feasible solutions in the formulation.

In order to get efficient solutions and obtain a Pareto front Ξ in the criterion space, we use iterative methods based on epsilon-constraint scalarization. First, we have to compute a lexicographic minimization to obtain a first point ξ_1 in the Pareto front, with the best value on f_{BR}. An other lexicographic optimization is computed as well to get a solution ξ_{TR} with the best value on f_{TR}. As the weights $(\omega_v)_{v\in V}$ are non-zero positive integers, we consider the non-zero positive integer $\varepsilon = gcd((\omega_v)_{v\in V})$. ε gives the minimal variation of f_{BR}.

Then, the idea is to perform the lexicographic minimization:

$$\min_{\xi \in X : f_{BR}(\xi) \leq f_{BR}{}^{M}} \left(f_{TR}(\xi), f_{BR}(\xi)\right) \tag{10}$$

The upper bound $f_{BR}{}^{M}$ is initialized at $f_{BR}(\xi_1)$. Its value is increased by ε at each iteration. The algorithm is stopped once the solution ξ_{TR} is reached. Thus, $f_{BR}(\xi_{TR})\varepsilon - f_{BR}(\xi_1)$ iterations are performed. But, the size of the Pareto front can be lower. If the algorithm started from the solution ξ_{TR} and the lower bound $f_{BR}{}^{M}$ were decreasing (like epsilon-constraint method), there would be $|\Xi|+1$ iterations. Nevertheless, a solution would not be feasible for the next iteration. The optimization would be initialized from scratch. In order to cope with this issue, we propose a greedy search. Experiments uses also this variant for comparison.

5 Experiments

5.1 Cases studied

A first set of instances is based on Paris Gare de Lyon station. It is a terminal station with 22 tracks and 140 routes modeled. The station hosts regional services, as well as national and international high-speed railway companies. Instances are based on a regular week day in 2017 containing 251 trains. We have the microscopic layout and simulations giving occupancy times and travel times. Thanks to these, we deduce that it is a short switch area station. Otherwise, Paris Gare de Lyon is divided into two zones in which all platform tracks have the same reoccupation proprieties. For backup platforming, neighbourhood of the assignations only contains routes with platforms in the same zone. This is relevant according to operative dispatching rules.

The second case study is Paris Est station. It is a 29 tracks terminal station with 312 routes. More than half of the traffic is dedicated to regional and suburban services. The rest of the trains make intercity and international links. Instances are based on a regular week day in 2017 containing 307 trains. It is also a short switch area station. Its platforms are divided into three areas with a given reoccupation time.

Experiments consider at last, the Lyon Part-Dieu area (Figure 2). This 11 tracks through station is a major railway hub. It is the first connecting station in France. Indeed, it structures regional traffic and concentrates a significant part of high speed services. The station is modeled with 66 routes and has a short switch area. Reoccupation times are the same for all tracks. We consider a regular week day scheduled for 2017 which includes 517 trains. We notice that it is one of the few places in France that suffers from saturation.

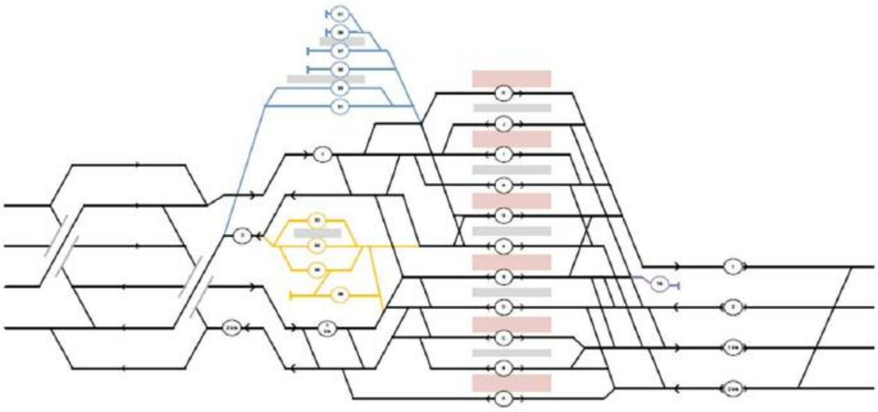


Figure 2: Layout of Lyon Part-Dieu station

Table 1: List of instances

Instance	Station	Size of tolerance intervals (minutes)	Nodes $\|V\|$	Number of edges $\|E\|$	Buffer Time edges $\|A\|$
PLY2L	Paris Gare de Lyon	2	15 276	15 557 078	1 897 963
PLY3L	Paris Gare de Lyon	3	15 276	15 557 078	2 590 720
PLY4L	Paris Gare de Lyon	4	15 276	15 557 078	2 759 805
PE1L	Paris Est	1	27 961	15 254 976	610 199
PE2L	Paris Est	2	27 961	15 254 976	1 098 358
PE3L	Paris Est	3	27 961	15 254 976	1 252 128
LYD2L	Lyon Part-Dieu	2	17 060	15 671 578	3 656 701
LYD3L	Lyon Part-Dieu	3	17 060	15 671 578	6 241 523
LYD3L	Lyon Part-Dieu	4	17 060	15 671 578	8 337 278

Instances are made considering different tolerance intervals. These tolerances are supposed uniform and symmetric. So that if the size of the interval is 4 minutes, an event will be able to happen 2 minutes before or after the scheduled time.

5.2 Implementation

The experiments are carried out with the OpenGOV software. OpenGOV integrates a resolution of the safety problem and an user interface. Thus a pre-treatment is performed to give a conflicts graph. We implement then the ε increasing version of the BTR-TPP algorithm as well as the ε decreasing one. These algorithms are coded in Java and run with CPLEX version 12.6. Experiments are run on an Intel(R) Xeon(R) CPU E3-1220 v3

3.10GHz, 8GB RAM, 64 bits operating system.

5.3 Results

Instance	ε increasing version			ε decreasing version		
	Number of iterations	Points in the Pareto front	Comput. Time (s)	Number of iterations	Points in the Pareto front	Comput. Time (s)
PLY2L	42	28	4843	29	28	7365
PLY3L	49	26	5198	27	26	6621
PLY4L	36	21	4354	22	21	6206
PE1L	31	15	3289	16	15	1724
PE2L	32	17	3720	18	17	2018
PE3L	36	14	3837	15	14	1815
LYD2L	63	13	8006	14	13	4104
LYD3L	47	13	5917	14	13	4686
LYD4L	50	12	6453	13	12	5199

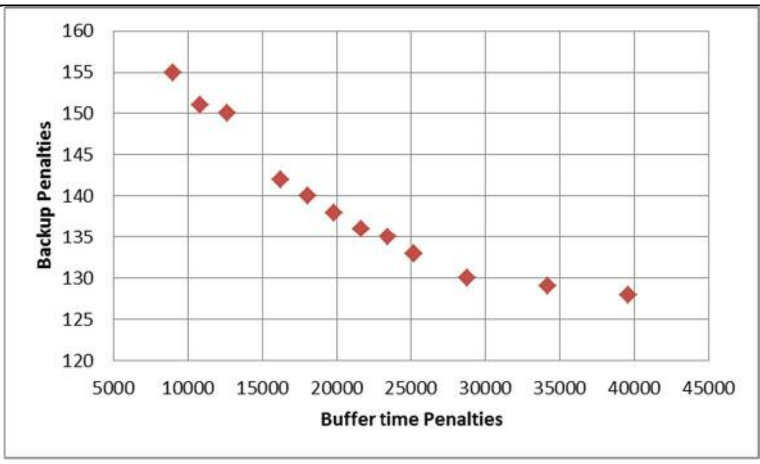


Figure 3: Instance LYD3L Pareto front

6 Conclusion

This paper explains conflicting aspects of robustness in platforming perceived by railway operators and suggests a biobjective algorithm which gives compromises. The programming method aims at maximizing two kinds of residual capacity to ensure to ensure easy recovery: buffer times and back-up routings. An integer linear program which extends a stable set formulation is implemented in the OpenGOV software.

The model is applied on nine instances based on real traffic data at three main french stations. The algorithm gives many relevant solutions that could not be obtained with a simple stable set formulation.

In future research, the model will be extended to take into account more disturbances scenarios and price of robustness. Otherwise, we will focus on preference modeling for a deployment in railway stations.

Literature

[1] BILLIONNET, Alain: Using integer programming to solve the train-platforming problem. In: *Transportation Science* 37 (2003), Nr. 2, S. 213–222

[2] CAIMI, Gabrio; BURKOLTER, Dan; HERRMANN, Thomas: Finding delay-tolerant train routings through stations. In: *Operations Research Proceedings 2004* Springer, 2005, S. 136–143

[3] CAPRARA, Alberto; GALLI, Laura; KROON, Leo; MARÓTI, Gábor; TOTH, Paolo: Robust train routing and online re-scheduling. In: *OASIcs-OpenAccess Series in Informatics* Bd. 14 Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2010

[4] CAPRARA, Alberto; GALLI, Laura; STILLER, Sebastian; TOTH, Paolo: Recoveryro bust platforming by network buffering. In: *RailZurich2009*, 2009

[5] CAPRARA, Alberto; GALLI, Laura; TOTH, Paolo: Solution of the train platforming problem. In: *Transportation Science* 45 (2011), Nr. 2, S. 246–257

[6] CAREY, Malachy; LOCKWOOD, David: A model, algorithms and strategy for train pathing. In: *Journal of the Operational Research Society* 46 (1995), Nr. 8, S. 988–1005

[7] DEWILDE, Thijs; SELS, Peter; CATTRYSSE, Dirk; VANSTEENWEGEN, Pieter: Improving the robustness in railway station areas. In: *European Journal of Operational Research* 235 (2014), Nr. 1, S. 276–286

[8] FISCHETTI, Matteo; MONACI, Michele: Light robustness. In: *Robust and online large-scale optimization*. Springer, 2009, S. 61–84

[9] LIEBCHEN, Christian; LÜBBECKE, Marco; MÖHRING, Rolf; STILLER, Sebastian: The concept of recoverable robustness, linear programming recovery, and railway applications. In: *Robust and online large-scale optimization*. Springer, 2009, S. 1–27

[10] LUSBY, Richard M.; LARSEN, Jesper; RYAN, David M.; EHRGOTT, Matthias: Routing trains through railway junctions: A new set-packing approach. In: *Transportation Science* 45 (2011), Nr. 2, S. 228–245

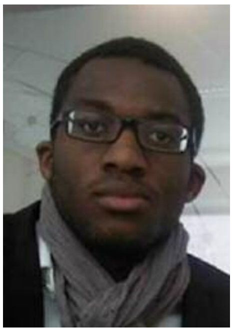
[11] RODRIGUEZ, Joaquín: A constraint programming model for real-time train scheduling at junctions. In: *Transportation Research Part B: Methodological* 41 (2007), Nr. 2, S. 231–245

[12] SELS, Peter; VANSTEENWEGEN, Pieter; DEWILDE, Thijs; CATTRYSSE, Dirk; WAQUET, Bertrand; JOUBERT, Antoine: The train platforming problem: The infrastructure

management company perspective. In: *Transportation Research Part B: Methodological* 61 (2014), S. 55–72

[13] ZWANEVELD, Peter J.; KROON, Leo G.; ROMEIJN, H E.; SALOMON, Marc; DAUZERE-PERES, Stephane; VAN HOESEL, Stan P.; AMBERGEN, Harrie W.: Routing trains through railway stations: Model formulation and algorithms. In: *Transportation science* 30 (1996), Nr. 3, S. 181–194

Authors



Kamenga, Franck

He is a Ph.D. student in Operations Research under the supervision of Joaquin Rodriguez at IFSTTAR, Université Lille Nord de France, Lille, France, since 2016. His research is carried out in collaboration with DPF Solutions, a department of SNCF Réseau which develops software tools for railway operations. He studies robust scheduling and its application to the train unit shunting problem.

Leloup, Leonard

He received a M.Sc. in Operations Research at Telecom ParisTech, Université Paris Saclay, France, in 2016. He is an engineer at DPF Solutions, a department of SNCF Réseau which develops software tools for railway operations. He takes part in the development of OpenGOV, a software support for train platforming.

His areas of interest are traffic management and timetabling.