Runtime Performance Analysis of a MILP-Based Real-Time Railway Traffic Management Algorithm

1st Szilárd Aradi
Dept. of Control for Transportation and Vehicle Systems
Budapest University of Technology and Economics
Budapest, Hungary
aradi.szilard@kjk.bme.hu

2nd László Lindenmaier
Dept. of Control for Transportation and Vehicle Systems
Budapest University of Technology and Economics
Budapest, Hungary
lindenmaier.laszlo@kjk.bme.hu

3rd István Ferenc Lövétei
Dept. of Control for Transportation and Vehicle Systemss
Budapest University of Technology and Economics
Budapest, Hungary
lovetei.istvan@kjk.bme.hu

***Abstract*—The real-time railway traffic management problem occurs when the trains get off schedule due to different traffic perturbations. In this case, they must be rerouted, reordered, and rescheduled to resolve the possible conflicts. Nowadays, this problem is usually handled by human dispatchers. There are lots of algorithms aiming to support human dispatchers in making an optimal decision that minimizes delays. However, due to the real-time nature of the problem, the response time of these algorithms is crucial. In this paper, the runtime performance of a state-of-the-art mixed-integer linear programming model is analyzed in different solvers. The analysis is performed via Monte Carlo simulation, generating various realistic scenarios in an infrastructure model of a Hungarian railway control area.**

***Index Terms*—Rail transportation, Railway safety, Linear programming, Optimization, Runtime, Performance analysis**

I. Introduction

Railways have the lowest specific energy consumption and the highest passenger load factor compared to other means of road transport [1], [2]. Hence, there will be an increasing demand for passenger and freight rail services [3], [4]. This trend is also motivated by the global CO_2 emission reduction targets [5], [6]. The increasing demand requires a continuous improvement of the rail services. However, the rail infrastructure costs are significantly higher than for road transportation [7]. Therefore, traffic management systems (TMS) have an important role in intelligent railway transportation systems (IRTS) to increase capacity usage and efficiency [8], [9]. One of the biggest challenges of TMS, the real-time railway traffic management problem (rtRTMP), occurs when the trains suffer some primary delays and get off the original schedule due to different traffic disturbances [10]. The trains must be rerouted, reordered, and rescheduled to avoid possible conflicts. Although nowadays, this problem is usually handled by human dispatchers. There is much work toward automated real-time railway traffic management optimizing TMS considering different objectives such as passenger satisfaction or energy consumption [11], [12].

One of the commonly used optimization formulations of rtRTMP is mixed-integer linear programming (MILP). Pellegrini et al. proposed a MILP model in [13] that, besides resolving the conflicts with the consideration of the railway traffic regulations, minimizes the overall delays of the trains. A detailed analysis of the model's impact on the overall delays compared to the performance of a human dispatcher and the conventional FCFS (first-come-first-served) management strategy is presented in [14]. Besides the straightforward shape of the model, it can provide an optimal solution for the rtRTMP. In our previous work [15], we have also extended the problem with a safety-relevant regulation concept, i.e., the overlaps, considering the braking distance of the vehicles. However, since MILP is NP-complete in general, the response time of these algorithms can be critical in large control areas and complicated scenarios. The original model is extended with custom heuristics and valid inequalities in [16], [17] to reduce the number of feasible solutions and thus accelerate the optimization. Still, the runtime performance of MILP-based railway traffic management algorithms is critical to satisfying the requirements of real-time running. Rudan et al. analyzed the performance of a similar formulation combined with a model predictive framework in [18] and showed that the runtime also depends on the order of the constraints. In this paper, the runtime performance of the two MILP-based optimization algorithms in [13] and [15] are analyzed and compared with different configurations and in different commercial solvers. The evaluation is performed in a realistic simulation environment, generating different random scenarios and modeling a real Hungarian control area.

The rest of the paper is organized as follows. The formulation of the analyzed MILP models is described in Section II. The simulation environment, along with the evaluation methodology, are presented in Section III. The results are shown and discussed in Section IV. Finally, the conclusions are drawn with a brief outlook on the future works in Section V.

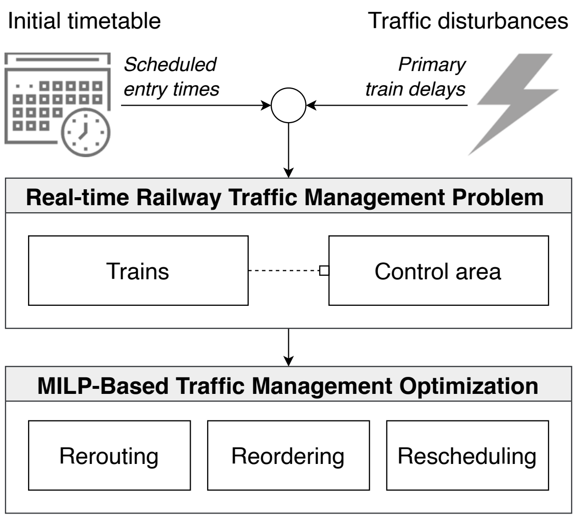

Fig. 1: The workflow of the traffic management algorithms.

II. Model formulation

In the case of nominal operations, the trains follow the initial timetable. However, the trains usually suffer some primary delays due to different traffic disturbances, leading to an rtRTMP, as illustrated in Fig. 1. First, the actual rtRTMP needs to be formulated by identifying the directly or indirectly affected trains and assigning them to the control area where the conflicts must be resolved. Then, some trains must be rerouted, reordered, and rescheduled, i.e., stop or slow down, to avoid collisions. The modeling of the rail network and trains and the MILP formulation of the analyzed algorithms are discussed in the following subsections.

A. Rail network and train model

The train conflicts must be resolved within a well-defined space of the rail network, the so-called control area. The analyzed algorithms rely on a multi-layer microscopic model of the rail infrastructure of the control area as in [19]. The lowest layer is a graph model representing the rail network, including detailed information, such as the rails' maximum velocity and length and the location of signals. Train schedules are given according to track circuits, in which the system automatically detects the presence of a train. In the second layer of the infrastructure model, the corresponding graph edges are clustered, constructing the track circuits. Finally, the series of track circuits between two consecutive signals are denoted as a block section—the block sections between the entry and exit locations of the control area form routes.

The trains are modeled to run with a constant velocity maximally allowed by a specific track circuit or what they are capable of. The acceleration and deceleration between the track circuits are neglected for simplicity. The trains are connected to the infrastructure by the location and time they intend to enter the control area. Their entry and exit locations define the routes available for them. A detailed explanation of the train and rail network models can be found in [19].

B. MILP formulation

A generic MILP problem is formulated as follows:

$$\min_x c^\top x \text{ with constraints } \begin{cases} A\,x \le b \\ A_{eq}\,x = b_{eq} \\ l_b \le x_s \le u_b \\ x_i \in \mathbb{Z},\ i = i_1, \dots, i_{n_i} \end{cases} \tag{1}$$

where n_i denotes the number of integer variables in the x optimization state space. The state space of the analyzed models consists of the x_t state of the individual $t = 1, \dots, n_t$ trains taking part in the rtRTMP. The x_t state of train t comprises of the vectors, including the

- $e_{t,r,tc}$ times when t enters the tc track circuit on route r,
- $d_{t,r,tc}$ delays assigned to t at tc track circuit on route r,
- $sRes_{t,tc}$ times when t starts to reserve the tc track circuit,
- $eRes_{t,tc}$ times when t releases the tc track circuit,
- $\alpha_{t,r}$ binary variables indicating if t travels on route r,
- $\beta_{t,t',tc}$ binary variables indicating if t uses tc before t'.

The model constraints are constructed considering the time and space coherence of the trains and the railway traffic regulations according to [13]. First, the time-related constraints are discussed in the following. Since the conflicts must be resolved in the control area, additional delays outside the control area are avoided by the constraint:

$$e_{t,r,tc_{t,in}} = (sched_t + \delta_t)\,\alpha_{t,r}\,, \tag{2}$$

where $tc_{t,in}$ and $sched_t$ denote the track circuit and time for t to enter the control area, considering its primary delay, δ_t. The binary route indicator, $\alpha_{t,r}$, has a value 1 if train t travels on route r. Then, the time schedule for other tc track circuits available for train t is given based on the run running time, considering the $dw_{t,tc}$ dwell time at tc

$$e_{t,r,tc} \ge e_{t,r,p_{r,tc}} + (run_{t,r,p_{r,tc}} + dw_{t,tc})\,\alpha_{t,r}\,, \tag{3}$$

where $p_{r,tc}$ denotes the track circuit preceding tc on route r. The delays assigned to train t at tc track circuits that are the last of their block section are expressed as

$$d_{t,r,tc} = e_{t,r,s_{r,tc}} - e_{t,r,tc} - (run_{t,r,tc} + dw_{t,tc})\,\alpha_{t,r}\,, \tag{4}$$

where $s_{r,tc}$ is the subsequent track circuit of tc along route r. Delaying trains at track circuits not having a signal at the end is not possible; hence, it is prevented as

$$d_{t,r,tc} = 0 \quad \text{if } bs_{r,tc} = bs_{r,s_{r,tc}}\,, \tag{5}$$

where $bs_{r,tc}$ denotes the block section of tc along route r. Then, the accumulated secondary delay assigned to train t is

$$D_t \ge \sum_{r \in R_t} e_{t,r,tc_{t,out}} + (run_{t,r,p_{r,tc}} + dw_{t,tc})\,\alpha_{t,r} - exit_t\,, \tag{6}$$

where $exit_t$ denotes the time when t is initially scheduled to exit the control area. The first term in (6) is the actual exit time computed from the entry times to the track circuit, $tc_{t,out}$, where t leaves the control area, considering the set of R_t routes available for t.

In the following, the constraints related to the capacity of the rail infrastructure and traffic regulations are explained. Train t starts to reserve all the tc track circuits belonging to the block section that it approaches before entering it as

$$sRes_{t,tc} = \sum_{r \in R_t} e_{t,r,ref_{r,tc}} - for \cdot \alpha_{t,r} \,, \tag{7}$$

where $ref_{r,tc}$ denotes the reference track circuit of tc, i.e., the first track circuit of $bs_{r,tc}$ block section, and for is the formation time, whichever is earlier the reservation starts. Similarly, the reservation is ended a while after the train leaves the track circuit, so

$$eRes_{t,tc} = \sum_{r \in R_t} e_{t,r,s_{r,tc}} + (cl_{t,r,tc} + rel)\, \alpha_{t,r} \,, \tag{8}$$

where $cl_{t,r,tc}$ and rel denote the clearing (i.e., the time until the part of the vehicle assembly occupies the track circuit after the train has already entered the subsequent one) and release time. According to railway traffic regulations, a track circuit can be reserved by at most one train at a time. The conflicting reservation of track circuit tc is prevented based on the precedence indicator between train t and t' as

$$\beta_{t,t',tc} + \beta_{t',t,tc} = 1 \tag{9}$$

$$sRes_{t,tc} \geq eRes_{t',tc} - M\, \beta_{t,t',tc} \tag{10}$$

$$sRes_{t',tc} \geq eRes_{t,tc} - M(1 - \beta_{t,t',tc}) \tag{11}$$

where M is a big constant. Constraint (9) imposes that the $\beta_{t,t',tc}$ precedence between train t and t' is mutually exclusive; therefore if t uses track circuit tc before t', $\beta_{t,t',tc} = 1$ and $\beta_{t',t,tc} = 0$, otherwise $\beta_{t,t',tc} = 0$ and $\beta_{t',t,tc} = 0$. While the disjoint constraints in (10) - (11) ensure that the train that later uses the track circuit cannot start reserving it until the other train releases it. Finally, the following constraint imposes that every train t uses exactly one route:

$$\sum_{r \in R_t} \alpha_{t,r} = 1 \,. \tag{12}$$

The objective of the optimization is to minimize the weighted secondary delays of the set of T trains as

$$\min \sum_{t \in T} w_t\, D_t \,, \tag{13}$$

where w_t denotes the weight assigned to train t.

In our previous work [15], we have extended the previous MILP formulation to incorporate an important, safety-critical traffic regulation concept, the overlaps. An overlap is the safety margin that must be proved to be clear beyond a stop signal to consider if the vehicle cannot stop before it due to increased braking distance. Therefore, the extended model keeps the track circuit beyond the stop signal reserved for the train approaching it until it is assumed to be able to stop comfortably. Since the model extension requires additional continuous and binary variables, the state space dimensionality is higher than in the original model. In this paper, besides evaluating the runtime performance of the original model, we assess the impact of the model extension.

III. Experimental Setup

The runtime performance of the two algorithms, i.e., the original and the extended one, is evaluated via Monte Carlo simulation generating independent 100 scenarios with feasible solutions. The scenarios rely on the infrastructure model of a real Hungarian control area, illustrated in Fig. 2. It consists of 89 track circuits, six platforms, nine entry/exit locations, and 107 feasible routes. Each random scenario involves ten trains affected by the rtRTMP, whose properties (e.g., scheduled entry time, desired velocity, etc.) are drawn from a uniform distribution with the parameters given in Table I. The parameters P_{dw} and P_{in} denote the probabilities used to simulate that a train has to dwell some time at one of the platforms and is already in the control area at the beginning of the scenario (i.e., initial train). These events follow a Bernoulli distribution with P_{dw} and P_{in}; hence if a randomly generated $r \sim \mathcal{U}(0,1)$ number is below P_{dw} or P_{in}, the corresponding train is simulated with a planned waiting time at one of the platforms on its route or starts the scenario already in one of the track circuits of the control area. The state space of the original and extended model defined by the number of trains, track circuits, and routes consists of 106920 and 127390 variables, respectively. The evaluation is performed on a Lenovo ThinkCentre PC (Intel Core i7-10700 2.9 GHz, 16 GB) in a Matlab development environment. Moreover, the results are compared across three different commonly used commercial solvers: FICO Xpress 8.14.2, Gurobi 10.0.1, and IBM ILOG CPLEX 12.1, which have dedicated Matlab interfaces. In the runtime measurement, the constraint generation is neglected to focus only on the performance of the models and solvers.

IV. Results

The results are evaluated along with three aspects. First, the average runtimes of the two models with the different solvers are computed considering all scenarios. In real-time applications, usually, there is a given response time available to provide a solution for the rtRTMP. Therefore, the optimality, i.e., the objective value, of the solutions is also examined with different limitations for the response time. Finally, the runtime performance sensitivity of the models with the different solvers to the structure of the constraint matrices is investigated. In the following, the results according to these aspects are detailed.

TABLE I: Train parameter settings

Parameter	Notation	Lower limit	Upper limit
Desired velocity	v_t [km/h]	50	160
Length of vehicle assembly	L_t [m]	100	300
Train priority	w_t [%]	0.1	100
Scheduled entry time	$sched_t$ [sec]	0	15
Waiting time	dw_t [sec]	0	180
Waiting probability	P_{dw} [%]	25	
Initial train probability	P_{in} [%]	50	

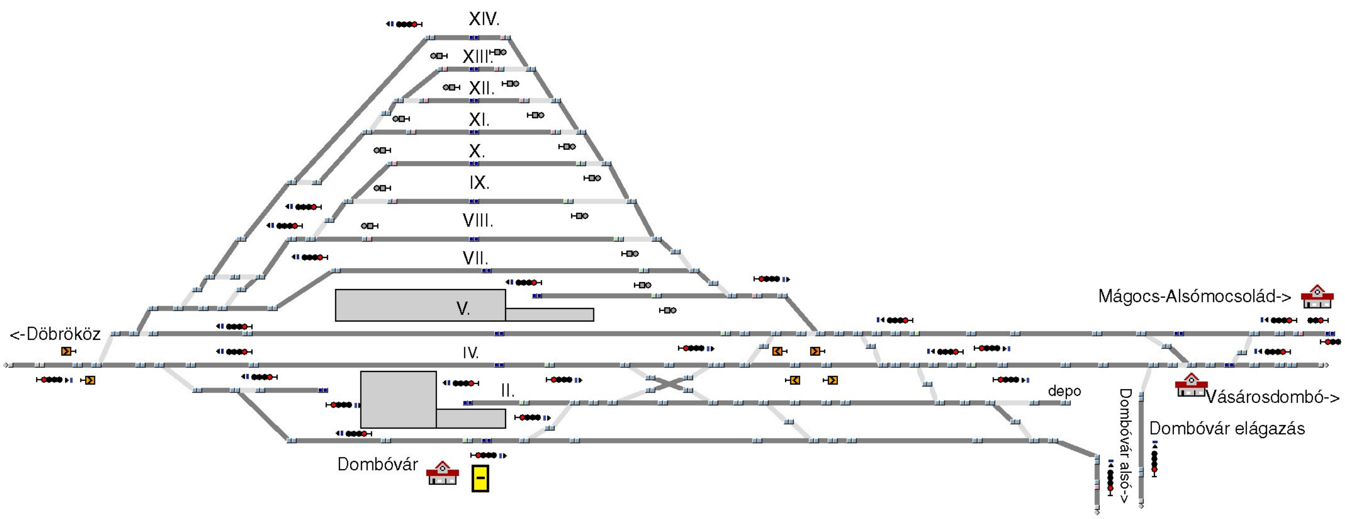

Fig. 2: Infrastructure model of the rail network used for evaluation

TABLE II: Relative runtime performance metrics of the models with different solvers (reference - bold)

Solvers	Original model [13]	Extended model [15]
FICO Xpress	$\mathbf{100.00}\%$	107.43%
Gurobi	47.25%	56.14%
IBM ILOG CPLEX	98.36%	125.84%

A. Average runtimes

The runtime performances of the two models in the three different solvers are given relative to the original model in FICO Xpress 8.14.2 computed as

$$P_{run} = \frac{1}{100}\sum_{i=1}^{100}\frac{rt_i}{\hat{rt}_i}\cdot 100\%\,, \tag{14}$$

where rt_i and $\hat{rt}_i$ denote the runtime of the evaluated model in the investigated solver and the original model in FICO Xpress 8.14.2 as a reference in the i-th scenario. The average runtime performance in (14) is computed after the runtimes of the different scenarios are normalized by the reference runtime to compensate for the different complexity of the scenarios. According to the evaluation results in Table II Gurobi significantly outperforms the other two solvers with 47.25% and 56.14% relative runtimes. FICO Xpress and IBM ILOG CPLEX show similar runtime performance for the original model with a 1.64% difference in favor of CPLEX. The runtime complexity of the extended model is clearly higher than the original model. However, the runtime overhead of the model extension varies in the different solvers. The lowest overhead is realized in Xpress with a 7.43% average runtime increase. While the average runtime of the extended model is 8.89% and 27.48% higher than in Gurobi and CPLEX than the original one.

B. Optimality within given response time

The average optimality of the solutions for the original and extended problems in the different solvers within a given response time is computed as

$$P_{opt} = \frac{1}{100}\sum_{i=1}^{100}\left(1+\frac{o_i-\hat{o}_i}{\frac{1}{100}\sum\limits_{i=1}^{100}\hat{o}_i)}\right)\cdot 100\,, \tag{15}$$

where o_i and $\hat{o}_i$ denote the objective values of the evaluated model in the examined solver and the globally optimal solution of the problem. The original and the extended models are evaluated separately relative to their corresponding references since their objective values may differ. In contrast with the runtime performance in (14), the objective value cannot be directly normalized by the reference since it can be zero. Therefore, the difference of o_i and $\hat{o}_i$ are normalized by the mean reference value in (15). The optimality performance metrics defined in (15) and the percentage of the scenarios solved optimally are shown in Table III with three different, 30-, 60-, and 90-second, available response times. Despite the different runtime performance metrics in Table II the solvers show similar performance regarding the optimality of the provided solution within the given response times. Gurobi is not affected by the runtime limitation, meaning that every scenario is solver optimally both in the original and the extended model. Only two scenarios of the original problem could not be solved optimally within the lowest 30-second available response time by FICO and IBM ILOG CPLEX. Moreover, even in these cases, they provide a solution close to the global optimum, increasing the average objective value by only 0.11% and 0.03%. The results are similar in the extended model, but Xpress slightly outperforms CPLEX with a 0.08% lower average objective value. In the case of higher response times, only CPLEX is affected by the limitation. It cannot solve one of the one hundred scenarios formed according to

TABLE III: Relative optimality within given response times (P_{opt} / percentage of scenarios with global optimum)

Solvers	Original model [13]	Extended model [15]
Max response time = 30 sec		
FICO Xpress	100.11/98%	100.02/99%
Gurobi	100.00/100%	100.00/100%
IBM ILOG CPLEX	100.03/98%	100.10/97%
Max response time = 60 sec		
FICO Xpress	100.00/100%	100.00/100%
Gurobi	100.00/100%	100.00/100%
IBM ILOG CPLEX	100.02/99%	100.02/99%
Max response time = 90 sec		
FICO Xpress	100.00/100%	100.00/100%
Gurobi	100.00/100%	100.00/100%
IBM ILOG CPLEX	100.02/99%	100.00/100%

the original model, neither in 60 nor 90 seconds. However, its extended form is solved optimally within 90 seconds.

C. Sensitivity to the order of the constraints

Finally, the sensitivity of the two models in the three different solvers to the structure of the constraint matrices is investigated as in [18]. According to [18], the most beneficial is the block-angular structure for the A and A_{eq} constraint matrices regarding the runtime performance. Each row of the matrices represents a constraint. Therefore, we build up the constraints considering the order of the trains and the track circuits to obtain the block-angular structure. Then, the rows of the constraint matrices are permuted randomly separately in the one hundred scenarios. The runtimes of the original and extended models in the different solvers under the influence of randomly ordered constraints are illustrated over the runtimes with block-angular structure in Fig. 3. The runtimes of the two models (original - left, extended - right) and the three solvers (FICO - first, Gurobi - second, IBM - third) are shown separately on a semilog scale due to visibility with the curves representing the reference values on the logarithmic scale. Fig. 3 confirms the results of Section IV-B, showing that most of the scenarios are solved within 30 seconds by all solvers. It can be seen that the random permutation of the constraints does not have a significant impact on the runtimes on average, except for the extended model in Fico Xpress. In that case, the runtimes with the permuted constraints are higher than with the block-angular structure, deviating from the reference runtime curve. This tendency can be particularly seen in more complex scenarios with higher reference runtimes. Although some scenarios take more time to be solved in the other solvers, others get faster. Therefore, they do not show such a sensitivity to the order of the constraints as Fico Xpress for the extended model.

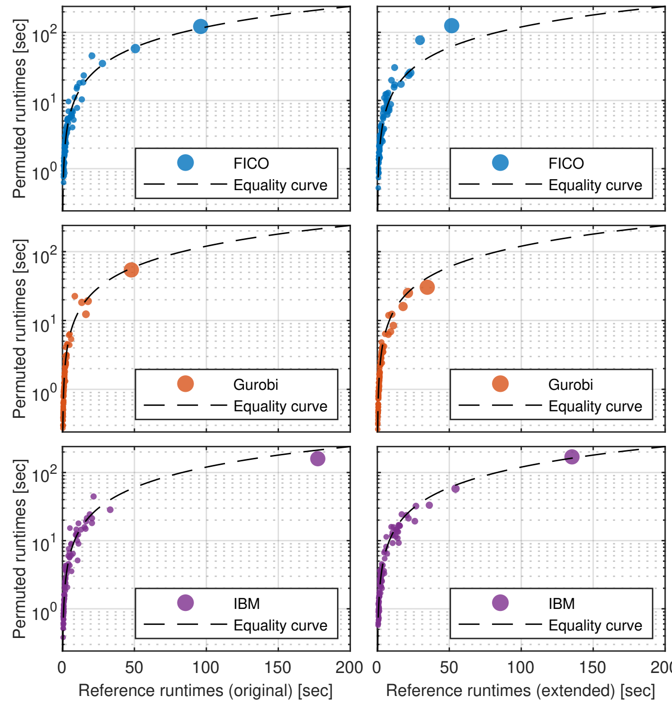

Fig. 3: The workflow of the traffic management algorithms.

V. Conclusions and Future Works

According to the runtime performance analysis, we found Gurobi the fastest of the evaluated commercial solvers to solve the MILP formulation of the real-time railway traffic management problem. However, the other two solvers, Fico Xpress and IBM ILOG CPLEX are also suitable for the problem, providing a feasible and, in most cases, optimal solution within the response times available for human dispatchers. In the analysis, we found that the model extension proposed in our previous work [15] does not cause a significant overhead compared to the original model in [13], except in CPLEX, where the average runtime increase is about 27.5%. Therefore, the extended model considering the safety-critical overlaps can also be applied in a real-time traffic management system. Finally, the models did not show a high sensitivity to the orders of the constraints, except the extended model in Fico Xpress, which has a mediate runtime increase due to the random reordering of the constraint matrices.

It is important to note that these commercial software are expensive, so it is worthwhile for decision-makers to know which one performs best for the problems relevant to them. On the other hand, the proper formalization of the problems to be solved is still important, as all solvers can benefit from it. That is why we would like to decrease the complexity of the evaluated models in our future research, using common decomposition techniques to meet the requirements of real-time applicability in more complex scenarios as well.

Acknowledgment

The research was supported in part by the European Union within the framework of the National Laboratory for Autonomous Systems. (RRF-2.3.1-21-2022-00002).

This work was partially supported by the János Bolyai Research Scholarship of the Hungarian Academy of Sciences.

References

[1] T. Skrucany, M. Kendra, M. Skorupa, J. Grencik, and T. Figlus, "Comparison of chosen environmental aspects in individual road transport and railway passenger transport," *Procedia Engineering*, vol. 192, pp. 806–811, 2017, 12th international scientific conference of young scientists on sustainable, modern and safe transport. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1877705817326863

[2] S. Ribeiro, M. Figueroa, F. Creutzig, C. Dubeux, J. Hupe, and S. Kobayashi, *Global Energy Assessment - Toward a Sustainable Future. Chapter 9 - Energy End-Use: Transport*, 06 2012.

[3] D. M. Z. Islam, R. Jackson, T. H. Zunder, and A. Burgess, "Assessing the impact of the 2011 eu transport white paper-a rail freight demand forecast up to 2050 for the eu27," *European Transport Research Review*, vol. 7, no. 3, pp. 1–9, 2015.

[4] C. Filippi, G. Guastaroba, L. Peirano, and M. G. Speranza, "Trends in passenger transport optimisation," *International Transactions in Operational Research*, 2023.

[5] V. Chaturvedi and S. H. Kim, "Long term energy and emission implications of a global shift to electricity-based public rail transportation system," *Energy Policy*, vol. 81, pp. 176–185, 2015.

[6] A. García-Olivares, J. Solé, and O. Osychenko, "Transportation in a 100% renewable energy system," *Energy Conversion and Management*, vol. 158, pp. 266–285, 2018. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0196890417312050

[7] J. Dulac, "Global land transport infrastructure requirements," *Paris: International Energy Agency*, vol. 20, pp. 1–54, 2013.

[8] K. N. Qureshi and A. H. Abdullah, "A survey on intelligent transportation systems," *Middle-East Journal of Scientific Research*, vol. 15, no. 5, pp. 629–642, 2013.

[9] D. Tokody, G. Schuster, and J. Papp, "Study of how to implement an intelligent railway system in hungary," in *2015 IEEE 13th International Symposium on Intelligent Systems and Informatics (SISY)*, 2015, pp. 199–204.

[10] F. Corman, "Real-time railway traffic management: Dispatching in complex large and busy railway networks," Ph.D. dissertation, Delft University of Technology, 2010.

[11] F. Corman and L. Meng, "A review of online dynamic models and algorithms for railway traffic management," *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 3, pp. 1274–1284, 2015.

[12] V. Cacchiani, D. Huisman, M. Kidd, L. Kroon, P. Toth, L. Veelenturf, and J. Wagenaar, "An overview of recovery models and algorithms for real-time railway rescheduling," *Transportation Research Part B: Methodological*, vol. 63, pp. 15–37, 2014.

[13] P. Pellegrini, G. Marlière, and J. Rodriguez, "Real Time Railway Traffic Management Modeling Track-Circuits," in *12th Workshop on Algorithmic Approaches for Transportation Modelling, Optimization, and Systems*, ser. OpenAccess Series in Informatics (OASIcs), D. Delling and L. Liberti, Eds., vol. 25. Dagstuhl, Germany: Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2012, pp. 23–34. [Online]. Available: http://drops.dagstuhl.de/opus/volltexte/2012/3700

[14] P. Pellegrini, G. Marlière, and J. Rodriguez, "A detailed analysis of the actual impact of real-time railway traffic management optimization," *Journal of Rail Transport Planning & Management*, vol. 6, no. 1, pp. 13–31, 2016. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2210970616300026

[15] L. Lindenmaier, S. Aradi, T. Bécsi, and I. F. Lövétei, "Milp-based optimization of the extended real- time railway traffic management problem," in *2022 IEEE 16th International Symposium on Applied Computational Intelligence and Informatics (SACI)*, 2022, pp. 000 105–000 110.

[16] P. Pellegrini, G. Marlière, R. Pesenti, and J. Rodriguez, "Recife-milp: An effective milp-based heuristic for the real-time railway traffic management problem," *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 5, pp. 2609–2619, 2015.

[17] P. Pellegrini, R. Pesenti, and J. Rodriguez, "Efficient train re-routing and rescheduling: Valid inequalities and reformulation of recife-milp," *Transportation Research Part B: Methodological*, vol. 120, pp. 33–48, 2019. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0191261518307276

[18] J. Rudan, B. Kersbergen, T. van den Boom, and K. Hangos, "Performance analysis of milp based model predictive control algorithms for dynamic railway scheduling," in *2013 European Control Conference (ECC)*, 2013, pp. 4562–4567.

[19] L. Lindenmaier, I. F. Lövétei, G. Lukács, and S. Aradi, "Infrastructure modeling and optimization to solve real-time railway traffic management problems," *Periodica Polytechnica Transportation Engineering*, vol. 49, no. 3, pp. 270–282, 2021.