Applicability of rescheduling strategies in tram networks

Daniel Lückerath[1,2], Oliver Ullrich[1], Ewald Speckenmeyer[1]
[1]Institut für Informatik, Universität zu Köln
[2]Institut für Nachrichtentechnik, Fachhochschule Köln
lueckerath@informatik.uni-koeln.de

Highly utilized tram networks, where multiple lines share tracks and stations, are inevitably affected by disturbances during daily operation. While consequences of small, local perturbations may be counteracted by schedule characteristics, e.g. robustness, long lasting disturbances have to be addressed by dispatchers via schedule adjustments.
Several methods for the identification and assessment of different rescheduling actions have been proposed. However, most of these methods have only been applied in railway networks. Therefore, in this paper we compare different rescheduling strategies and assess their applicability in tram networks.
This paper begins with a description of possible rescheduling actions and the requirements and limitations to rescheduling strategies in tram networks. Different strategies for railway networks are then described and compared in regard to their applicability in tram networks.

1 Introduction

Tram networks are frequently affected by disturbances, many of which are inevitable. Especially highly utilized networks, where several lines share tracks and stations, are prone to perturbations.

While the consequences of small, local disturbances can be counteracted by considering schedule characteristics like robustness during the planning stage, long lasting perturbations call for direct intervention by dispatchers during daily operation, e.g. via schedule adjustments.

The method of adjusting a schedule during daily operation is called rescheduling. It aims for the generation of transitional schedules in order to cope with the impacts of major perturbations. To generate those temporary schedules dispatchers may change dwell times of vehicles at stations, travel times between stations or even whole routes of vehicles through the network. Because the schedule adjustments are carried out during daily operation, rescheduling is time-critical, i.e. dispatchers have only a limited time frame to devise and evaluate possible schedule adjustments without causing further disturbances. At the same time the resulting schedule should also be robust and allow for reinstatement of the original time table after the perturbation subsided.

Thus, there is a need for methods and tools that help network operators decide which rescheduling actions to take and how these actions will affect tram operation. To address this issue we want to broaden the scope of our project *Computer Aided Traffic Scheduling (CATS)*. Up until now we developed simulation and optimization methods to generate and evaluate robust time tables, which adhere to transport planning requirements (see [7, 12]). We now want to employ our simulation software during daily operation, utilizing the underlying parallelization framework (see [11]) to provide dispatchers with a tool set for quick evaluation of different rescheduling actions.

Several rescheduling strategies have been proposed (see [1, 2, 3, 4, 5, 6, 8, 9, 10, 13]). However, most of them have only been applied to railway networks. Because tram and railway networks differ in multiple aspects, e.g. schedule density or network size, rescheduling strategies that work in railway networks may not be applicable in tram networks.

Thus, in this paper we conduct our initial comparison of different rescheduling strategies for railway networks and assess their applicability in tram networks.

The paper continues with a description of the requirements and limitations to rescheduling strategies for tram networks (section 2). We then present and evaluate some rescheduling strategies for railway networks (section 3), which are afterwards compared for their applicability in tram networks (section 4). The paper closes with a short summary of lessons learned and some remarks on further research (section 5).

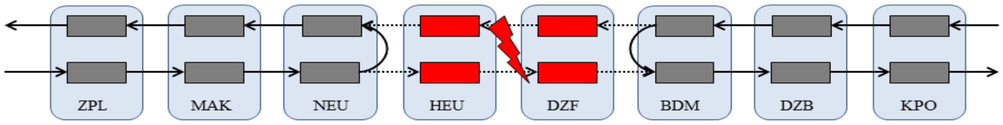

Figure 1. Example for route separation. Small gray/red rectangles depict platforms, while big blue rectangles with rounded edges show how platforms are joined into stations. Dotted arrows indicate the regular route

2 Requirements and limitations to rescheduling strategies for tram networks

To assess the applicability of existing railway rescheduling strategies we identify requirements and limitations to tram rescheduling strategies, which result from differences in network and schedule design between both systems.

Besides the mere number of stations and the length of tracks, differences in network design relate to the size of stations, track redundancy and the number of track switches that join different parts of the network. While stations in railway networks often consist of multiple platforms which may be approached by vehicles from different directions, most stations in tram networks consist only of two platforms (one for each direction). Therefore, re-routing of vehicles within stations, i.e. dynamic platform assignment, is not applicable for most tram networks. A similar observation can be made for tracks: In highly utilized areas of railway networks often multiple tracks for the same direction exist, e.g. one track for commuter trains and another for freight trains or long distance trains which do not stop at every station. In tram networks, on the other hand, there exists at most one track for each direction, which makes it impossible for vehicles to overtake a slower, damaged vehicle. Lastly, in most tram networks the number of locations where trams can switch from one route to another is small, limiting the degree to which redirection (e.g. due to a blocked route) is practicable.

In regard to commuter trams/trains both systems employ periodic time tables, but schedules for tram networks usually have a smaller tact interval than schedules for railway networks (e.g. ten minutes versus 60 minutes). Thus, the safety distance between successive vehicles is tighter in tram systems, resulting in shorter dwell times at stations and smaller time frames for dispatchers to devise schedule adjustments. Should the employed rescheduling action take longer than the safety distance, the follow-up vehicle will also be affected by the disruption and further rescheduling actions have to be undertaken.

As a result of these limitations the following rescheduling actions seem most applicable in tram systems:

I. Separation of a line route into two partial routes in order to avoid blocked tracks or stations. For example the route of Cologne's line 9 could be separated by turning vehicles around at station Deutz/Messe (BDM) for the west-bound variant and at station Neumarkt (NEU) for the east-bound variant, thus omitting hypothetically blocked stations Deutzer Freiheit (DZF) and Heumarkt (HEU) (see figure 1).

II. Shortening of routes at stations where vehicles can turn around. This is a special case of route separation, where only one partial route is serviced.

III. Redirection of vehicles at track switches where different lines from varying directions meet, e.g. Cologne's line 12 could be redirected at station Zülpicher Platz (ZPL) and travel along the route of line 9 to station Neumarkt (see figure 2).

IV. Adjusting arrival and departure times at relevant locations in the network (e.g. platforms or track switches) to adjust the tram order at the (next) joining track switch.

In addition to providing these actions, a feasible approach must also be able to handle realistic problem instances in acceptable time. As a reference point for this we choose Cologne's tram network of 2001, which consists of 528 platforms and 58 track switches connected via 584 tracks. 15 lines with 182 line routes are served by 178 vehicles which execute 2,814 trips per operational day. At most inner city platforms this results in a safety distance of two minutes, limiting the available computational time.

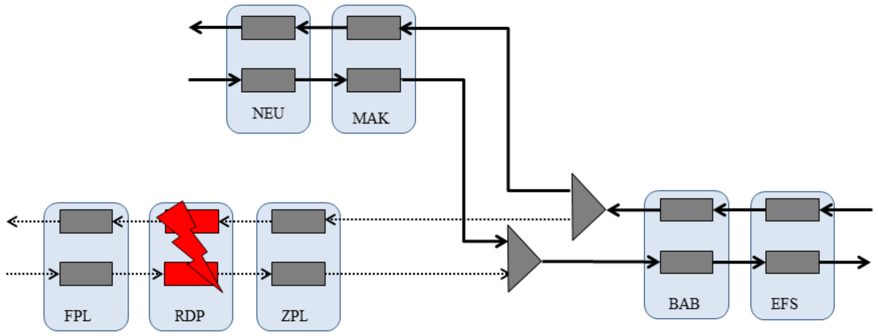

Figure 2. Example for vehicle redirection. Triangles depict track switches, small gray/red rectangles platforms and big blue rectangles with rounded edges show stations. Dotted arrows indicate the regular route

3 Rescheduling in railway systems

There exist several different approaches to rescheduling (see [1, 2, 3, 4, 5, 6, 8, 9, 10, 13]), some even considering bimodal traffic systems (see e.g. [5, 6, 13]). Because an exhaustive review of all approaches would go beyond the scope of this paper, we will only review those which seem most promising for applicability within tram networks and consider only one traffic system.

3.1 D'Ariano and Pranzo

D'Ariano and Pranzo in [4] describe an extension to the real-time dispatching system *ROMA (Railway traffic Optimization by Means of Alternative Graphs)* for short-term prediction of railway traffic under strong disturbances. Their objective is to evaluate the effects of rescheduling actions for a given time horizon. Because of the complexity of the problem they decompose the time period under examination into smaller, tractable time intervals which are solved in cascade. The output of each subproblem (i.e. position and speed of the vehicles at the end of the corresponding time interval) is used as input constraint for the subsequent time interval.

To model the railway network the authors use stations, signals and block sections (a track segment between two signals which may host at most one vehicle at a time). The movement of all vehicles during a given time horizon is defined by the schedule, which specifies, for each train, planned arrival/passing times at relevant locations along its route (e.g. stations or track switches).

Within each tractable time interval D'Ariano and Pranzo solve the following three problems:

1. Finding a feasible route for each vehicle without using already occupied tracks.
2. Scheduling train orders and exact arrival/departure times at stations as well as at relevant locations in the network (e.g. track switches).
3. Ensuring a minimum safety distance between vehicles while maintaining acceptable speed profiles.

To solve problem 1 the applied software module checks if there are different possible routes for a vehicle to use and whether those are already occupied or not. If no feasible route can be assigned external support by the dispatcher is requested.

Problem 2 is formulated as a job shop scheduling problem using the alternative graph formulation (see [3]). This formulation requires that a route for each vehicle is given (i.e. problem 1 is solved) and traversing times for tracks are known in advance. If conflicts between trains arise, a passing order must be defined. This is done by either using a *Branch and Bound* algorithm or a simple *First Come First Served* dispatching rule.

To solve problem 3 ROMA checks the compatibility of the schedule with the current vehicle dynamics and signal states and if necessary adjusts the vehicle speed profiles. These two steps are performed until a feasible schedule with acceptable speed profiles is obtained.

By solving those three problems the resulting application can be used to perform all rescheduling actions from section 2. The solution to problem 1 corresponds to performing rescheduling action I, II or III, while defining the passing order and scheduling the exact arrival and departure times at stations (i.e. solving problem 2) corresponds to rescheduling action IV. Finally, changing the speed profiles while solving problem 3, indirectly corresponds to performing rescheduling action IV.

D'Ariano and Pranzo apply the resulting software to the route Utrecht – Den Bosch in the Dutch railway network. This area consists of 191 block sections and 21 platforms. The employed time table has a tact interval of 60 minutes and up to 40 vehicles are scheduled. Depending on the length of the time horizon for which traffic predictions are done, ROMA takes between 14 seconds (one hour time horizon) and 787 seconds (nine hours) to solve the problem. It has to be noted, that the runtime does not increase linear with the time horizon. For example with a time horizon of two hours the application needs 503 seconds to solve the problem, which is too long for dense tram schedules, where the safety distance between vehicles is only two minutes. Thus, the method is either only applicable for small networks with few vehicles or its runtime has to be decreased, e.g. through solving the problem in parallel for separated parts of the network.

3.2 Corman et al.

Corman et al. in [2] present a solution strategy for the *Bi-objective conflict detection and resolution (BCDR)* problem, which deals with finding a set of non-dominated schedules that minimize train delay as well as the number of missed connections.

The authors formulate the BCDR problem as an alternative graph and determine the *Pareto front* (i.e. the maximal set of non-dominated solutions) by iteratively solving the *conflict detection and resolution (CDR)* problem for different sets of enforced train connections. To solve the CDR problem the authors employ the Branch and Bound algorithm described in [3], which minimizes train delay.

As D'Ariano and Pranzo in [4] the authors assume that the traveling time of vehicles can be determined in advance, i.e. trains travel at their scheduled speed whenever possible, recovering small delays by using buffer times inserted in the time table. In addition they use the same approach as D'Ariano and Pranzo to model the railway network and the schedule.

To approximate the Pareto front Corman et al. describe two algorithms, named *Add* and *Remove*. *Add* starts with the schedule solution that maintains none of the possible train connections and generates new solutions by adding enforced connections. *Remove* on the other hand starts with the schedule solution that maintains all train connections and generates new solutions by removing connections.

Because it solves the CDR problem the resulting application is at least able to perform rescheduling action IV from section 2. Given the information contained in [2] and [3] the software seems not to be able to adjust the routes of vehicles (i.e. performing rescheduling actions I – III).

While preserving connections may be crucial in railway networks with long tact intervals, this objective is less crucial in tram networks with dense schedules where passengers do not have to wait long for the next vehicle. However, as soon as connections between the tram network and other traffic systems (e.g. long distance train) have to be considered this once again becomes crucial.

Like D'Ariano and Pranzo, Corman et al. conduct computational experiments based on the Dutch railway network area around Utrecht, employing a time table with a tact interval of 60 minutes, which schedules up to 80 trains. Depending on the number of enforced connections and the severity of inserted perturbations, algorithm *Add* solves the problem in 166 to 309 seconds, while algorithm *Remove* takes between 283 and 705 seconds. Like the approach of D'Ariano and Pranzo this seems to be too long for dense tram schedules, but could very well be accelerated by parallelizing the implementation.

3.3 Törnquist Krasemann

Törnquist Krasemann in [9] proposes a greedy algorithm for rescheduling during daily operation which generates good-enough feasible schedules, independent of the underlying disturbance scenario, within 30 seconds. The algorithm is a complement to a previous approach (see [8, 10]) which formulates the rescheduling problem as a mixed integer linear program, but is not able to find good solutions for some disturbance scenarios and a time horizon longer than 60 minutes within acceptable time. Furthermore, resulting from a recent analysis of the infrastructure under examination the new algorithm includes the possibility to consider routing of vehicles within stations.

To model the railway network Törnquist Krasemann divides it into line sections and station sections. Each section has up to n parallel tracks and line sections

Rescheduling action	Approach		
	D'Ariano/Pranzo	Corman et al.	Törnquist Krasemann
I	✓	×	×
II	✓	×	×
III	✓	×	?
IV	✓	✓	✓
Runtime	14 s – 787 s	166 s – 705 s	30 s

Table 1. Summary of possible rescheduling actions and runtimes for the compared approaches

are sequences of one or several consecutive blocks. Unlike D'Ariano/Pranzo and Corman et al. blocks are not modeled explicitly but rather through the adherence to safety distance constraints between vehicles traveling in the same direction. A schedule is defined as sequences of consecutive events, which define points in time at which a specific vehicle is planned to occupy a certain section.

To solve the rescheduling problem the greedy algorithm uses depth-first search to build up a tree with events as nodes. With each new level of the tree a successor to the previous event is chosen. In addition each node holds an estimation of the disturbance consequences of the partial solution. After the first branch of the tree is completed a feasible schedule is obtained and the remaining computational time, up to a predefined time limit, is used to improve the solution, by backtracking to nodes which provide more promising disturbance estimations.

By modeling sections with up to n parallel tracks the resulting application may be able to at least partially perform rescheduling action III from section 2. However, given the information from [9] it is not clear if the developed method is able to handle a change of the destination station, which is the only viable rescheduling action in tram networks regarding changes in routes.

Finally, because the developed algorithm schedules the occupation of each section, it is able to perform rescheduling action IV.

For computational experiments Törnquist Krasemann uses data from the Norrköping traffic district in Sweden. The sub network is composed of 28 stations, 15 double-tracked sections and 17 single tracked sections. All tracks are bi-directional and all but one station have between two and 14 tracks. The author conducts experiments using 20 different scenarios, a maximum tolerated computational time of 30 seconds and a time horizon of 90 minutes, during which 46 to 51 vehicles are scheduled.

To evaluate the solution quality of the algorithm, the scenarios are also solved with CPLEX using a modified version of the formulation proposed in [8, 10]. While the greedy algorithm finds a first feasible solution in less than one second and very often finds other solutions within the first few seconds, in many of the examined scenarios the solution quality of the greedy algorithm cannot be evaluated because CPLEX does not find a solution within 24 hours. A reduction of the time horizon from 90 to 60 minutes (and subsequently reducing the number of scheduled trains) resolves this issue for most scenarios.

While in eight out of 20 scenarios the greedy algorithm finds a solution with an objective function value equal to the optimal solution obtained by CPLEX, in another eight scenarios the objective function values differ significantly. However, the difference between objective function values gives little information about the applicability of the time table, because the schedules may differ due to the way they are constructed. To evaluate the applicability of the schedules during daily operation they should be simulated.

The main appeal of the application lies in its short runtime even with a time horizon larger than 60 minutes, which could make it very promising for use in tram networks. This is especially true if the implementation is parallelized to explore several branches of the tree simultaneously, as suggested by Törnquist Krasemann.

4 Comparison of rescheduling strategies

Substantial evaluation of the rescheduling approaches of D'Ariano/Pranzo, Corman et al. and Törnquist Krasemann allows for the conclusion that they are all applicable to tram networks to a certain degree. (For a summary see table 1)

While the approach of D'Ariano and Pranzo can be used to perform all rescheduling actions from section 2, its runtime increases too fast for applicability in tram networks with dense schedules. Solving the problem in parallel for separated parts of the network or for exploration of different solution alternatives simultaneously might resolve this issue.

The approach of Corman et al. seems to be able to perform rescheduling action IV only, and it is the only one we have examined that considers more than one optimization goal. However, minimizing the number of missed connections is not as crucial in case of tram networks as in case of railway networks. Nonetheless their approach can be adapted to incorporate other optimization goals, e.g. satisfying given sets of transport planning requirements (see [12]). As with the approach of D'Ariano and Pranzo, the runtime may become problematic when applying the approach to tram networks. A parallel version of the Branch and Bound algorithm may remedy this problem.

Although it seems to be able to perform rescheduling action IV only, the approach by Törnquist Krasemann is very appealing because of its very short runtime of 30 seconds regardless of the underlying disturbance scenario. Furthermore, it can be easily parallelized by completing different branches of the tree simultaneously. However, the applicability of the obtained solutions remains open. Firstly, because for time horizons longer than 60 minutes only a few reference solutions were obtained using CPLEX. Secondly, because no dynamic system influences were considered, i.e. the schedules were not simulated or tested in the real system.

5 Conclusions and future work

In this paper we described, evaluated and compared different rescheduling strategies for railway networks in regard to their applicability to tram networks. While our first theoretical examination of the described strategies indicates that all of them seem to be applicable to tram networks to a certain degree, it also shows a couple of problems that have to be addressed: Based on the information available to us it is not clear if the approaches of Corman et al. and Törnquist Krasemann can be used to perform rescheduling actions I, II or III. In addition, the runtime for the approaches of D'Ariano/Pranzo and Corman et al. seems to be too long for applicability in tram networks with dense schedules, making it necessary to look into the possibility of parallelizing the implementations.

To assess the applicability of different rescheduling strategies more thoroughly, we want to implement a tool which allows us to manually apply schedule adjustments and gives instant visual assessment of the expected consequences. Based on the data obtained by using this tool we plan to implement an optimization module that incorporates promising rescheduling strategies and is tightly connected to our existing simulation module (see [7, 12]). We hope this will allow for developing strategies specialized on tram networks.

6 Acknowledgements

This work was partially supported by *Rhein Energie Stiftung Jugend Beruf Wissenschaft* under grant number W-10-2-002. We thank the other members of our work group Patrick Kuckertz and Bert Randerath.

7 References

[1] Adenso-Díaz, B.; Oliva González, M.; González-Torre, P.: *On-line timetable re-scheduling in regional train services.* Transportation Research Part B: Methodological, Vol. 33, 1999, pp. 387-398.

[2] Corman, F.; D'Ariano, A.; Pacciarelli, D.; Pranzo, M.: *Bi-objective conflict detection and resolution in railway traffic management.* Transportation Research Part C: Emerging Technologies, Vol. 20, 2012, pp. 79-94.

[3] D'Ariano, A.; Pacciarelli, D.; Pranzo, M.: *A branch and bound algorithm for scheduling trains in a railway network.* European Journal of Operational Research, Vol. 183, 2007, 643-657.

[4] D'Ariano, A.; Pranzo, M.: *An Advanced Real-Time Train Dispatching System for Minimizing the Propagation of Delays in a Dispatching*

Area Under Severe Disturbances. Networks and Spatial Economics, Vol. 9, 2009, pp. 63-84.

[5] Darmanin, T.; Lim, C.; Gan, H.: *Public Railway Disruption Recovery Planning: A New Recovery Strategy for Metro Train Melbourne.* Proceedings of the 11th Asia Pacific Industrial Engineering and Management Systems Conference, 2010.

[6] Ginkel, A.; Schöbel, A.: *To wait or not to wait? The bicriteria delay management problem in public transportation.* Transportation Science, Vol. 41, 2007, pp. 527-538.

[7] Lückerath, D.; Ullrich, O.; Speckenmeyer, E.: *Modeling time table based tram traffic.* Simulation Notes Europe (SNE), ARGESIM/ASIM Pub., TU Vienna, Vol. 22, No. 2, 2012, pp. 61-68.

[8] Törnquist, J.: *Railway traffic disturbance management: an experimental analysis of disturbance complexity, managements objectives and limitations in planning horizon.* Transportation Research Part A: Policy and Practice, Vol. 41, 2007, pp. 249-266.

[9] Törnquist Krasemann, J.: *Design of an effective algorithm for fast response to the rescheduling of railway traffic during disturbances.* Transportation Research Part C: Emerging Technologies, Vol. 20, 2012, pp.62-78.

[10] Törnquist, J.; Persson, J.: *N-tracked railway traffic re-scheduling during disturbances.* Transportation Research Part B: Methodological, Vol. 41, 2007, pp. 342-362

[11] Ullrich, O.: *Modellbasierte Parallelisierung von Anwendungen zur Verkehrssimulation: Ein dynamischer und adaptiver Ansatz*. Dissertation, University of Cologne, Germany, to appear.

[12] Ullrich, O.; Lückerath, D.; Franz, S.; Speckenmeyer, E.: *Simulation and optimization of Cologne's tram schedule.* Simulation Notes Europe (SNE), ARGESIM/ASIM Pub., TU Vienna, Vol. 22, No. 2, 2012, pp. 69-76.

[13] Zeng, A. Z.; Durach, C. F.; Fang, Y.: *Collaboration decisions on disruption recovery service in urban public tram systems.* Transportation Research Part E: Logistics and Transportation Review, Vol. 48, 2012, pp. 578-590.