

A robust schedule for Montpellier's Tramway network

Oliver Ullrich, Daniel Lückerath, Ewald Speckenmeyer
ullrich@informatik.uni-koeln.de

July 8, 2013

Abstract

The city of Montpellier in the Languedoc-Roussillon region of France features a fast growing tram network as a central part of its public service infrastructure. Here, as in many other tram networks, resources like tracks and stations are shared between different lines. Because of the resulting dependencies, small inevitable delays can spread through the network and affect its global performance.

This article examines whether a robust tram schedule may help to raise punctuality in Montpellier's tram network. To accomplish this, we apply a tool set designed to generate schedules optimized for robustness, which also satisfy given sets of planning requirements. These tools allow to compare time tables with respect to their punctuality and other key indicators.

After an introduction to the goals of this paper, we continue with a description of the tool set focusing on optimization and simulation modules. These software utilities are then employed to generate and simulate robust and non-robust schedules for Montpellier's tram network, which are subsequently compared for the resulting delays.

1 Introduction

The city of Montpellier in southern France is growing fast, its population has tripled in the last fifty years (see [6]). As major part of the city's public service infrastructure, the Tramway tram network is provided by Transports de l'agglomération de Montpellier (TAM). The first Tramway line was launched in 2000, it connects the eastern and western suburbs to the city center. Since then three more lines commenced operation. By now, about 282,000 passengers are served on each weekday (see [17]), which amounts to about half of the population of Montpellier's metropolitan area. Three more tram lines are commissioned, the first of which is planned to commence operation in 2017.

In Montpellier's tram network, several lines share resources like platforms, switches and tracks. Because of the resulting dependencies, small local delays

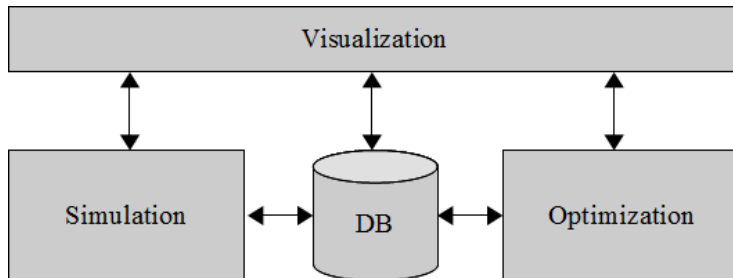


Figure 1: Architecture of project CATS

can propagate to succeeding trams, build up to larger delays, and thus affect the network’s global performance.

In this paper, we explore whether a robust schedule can help to reduce delays in Montpellier’s Tramway network. We define robustness as the degree to which inevitable small delays are kept local to the immediately affected tram and do not spread through the network. To examine this, we apply a software tool chain which enables us to generate robust schedules, compare their feasibility and evaluate their punctuality and other key indicators.

This paper continues with a description of our approaches on optimization and simulation of tram schedules (section 2). It then focuses on the modeling and simulation of Montpellier’s Tramway system. Robust and non-robust schedules are generated, simulated, and compared concerning the resulting delays (section 3). The paper closes with a short summary of lessons learned and some thoughts on further research (section 4).

2 Simulation and optimization of tram schedules¹

The project “Computer Aided Traffic Scheduling” (CATS) is built around a database complying with the ÖPNV5 data model proposed by the association of German transport companies (Verband Deutscher Verkehrsunternehmen, see [22]). Visualization, optimization, and simulation modules are connected via operations on the database and through XML configuration files (see figure 1). Due to its compliance with the ÖPNV5 data model our framework is capable of working on many European tram networks.

For an in-depth description of the optimization method, see [21]. A more detailed discussion of the simulation software can be found in [7].

2.1 Optimization of tram schedules

Various approaches to optimize tram and railway schedules are known (see e.g. [1, 3, 4, 5, 9, 10, 12]). Most of them aim at one general objective like minimizing

¹This section is an abbreviated version of [21], section 1.

vehicle delay (see [10, 12]) or maximizing robustness to restrict the global impact of small, local disturbances (see [4, 5]). Others use a combination of objectives, like operational profit and robustness in [3], or combining social opportunity cost and operational cost in [11].

Because of the complex nature of the problem, many authors use heuristic approaches like Lagrangian heuristics (see [3]) or simulated annealing (see [11]). Others, like Bampas et al. in [1] introduce exact algorithms for restricted subclasses, like chain and spider networks.

In our project, we combine heuristics and exact methods to generate optimal synchronized time tables for tram networks, targeting maximal robustness and adherence to transport planning requirements at the same time. Those planning requirements originate from political, economic and feasibility reasons. Thus it is not sufficient to exclusively consider a general goal like robustness when generating time tables.

We use the scheduled time offset between two consecutively departing vehicles at a platform as an indicator for robustness. In an assumed tact interval of ten minutes, two lines could be scheduled with equidistant offsets of five minutes, which means that one or both involved vehicles could be late for more than four minutes without consequences for the following tram. Under an extremely unequal split of the available time span into a nine minute offset followed by a one minute offset, the first tram could have a delay of more than eight minutes without consequences to the following vehicle. On the other hand, would the second vehicle be even slightly late, the delay would spread to the follow-up tram. Since we are assuming typically small delays, we see an equidistant distribution as very robust, the occurrence of very small offsets as not robust.

So, to calculate the robustness of a time table λ we examine at each platform h of the network the scheduled time offset $\delta_{f,pred(f)}(h, \lambda)$ between any trip f and its predecessor $pred(f)$, i.e. the time elapsed between the departures of $pred(f)$ and f at platform h .

To reduce complexity we aggregate subsequent similar platforms operated by the same lines to a maximal platform type h' , weighted by the number of included platforms $\varphi_{h'}$ (see figure 2). The reduced set of platforms is denoted by H' .

To calculate the robustness Φ_a of schedule λ , we add the inverse of $\delta_{f,pred(f)}(h', \lambda)$ for all platform types $h' \in H'$ and all its trips, thus applying a penalty for small safety distances. With $f \in F_{h'}$ representing all trips that serve platform type h' under schedule λ , and $\varphi_{h'}$ as the number of platforms represented by h' , the resulting function is as follows:

$$\Phi_a(\lambda) = \sum_{h' \in H'} \sum_{f \in F_{h'}} \frac{1}{\delta_{f,pred(f)}(h', \lambda)} * \varphi_{h'} \quad (1)$$

Given is a set R of planning requirements, with r denoting a single requirement $r \in R$. In order to calculate the compliance with transport planning requirements we introduce $\rho_r(\lambda) \in \{1, 2, 3, \infty\}$ the compliance factor of requirement r under a schedule λ . A compliance factor of 1 means that the requirement

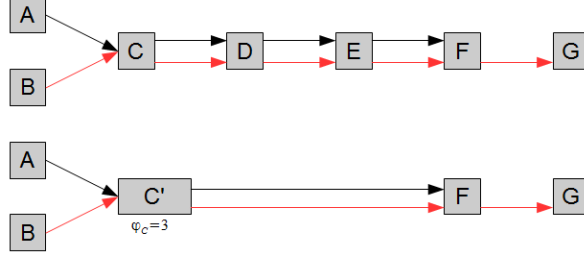


Abbildung 2: Example for platform reduction

is completely satisfied, 2 and 3 denote tolerable compliance, and ∞ means that the constraint is not met and the time table candidate λ must be rejected. We add the compliance values for all $r \in R$ and get the following:

$$\Phi_b(\lambda) = \sum_{r \in R} \rho_r(\lambda) \quad (2)$$

Depending on the network under consideration and the number of planning requirements, the two parts of the objective function may not be comparable directly. Thus we define a normalizing factor σ , which reflects the relationship between the lower bounds of both parts. The theoretically optimal distance $\delta_{f,pred(f)}^{opt}(h')$ of two trips $pred(f)$ and f on platform type h' is obtained by dividing the tact interval by the number of serving lines at that platform type. The best possible compliance factor ρ_r^{min} of a planning requirement $r \in R$ is the minimal value assigned by the planner, independent of the characteristics of the examined solution candidate. Typically $\rho_r^{min}=1$. We define σ as:

$$\sigma = \left(\sum_{h' \in H'} \sum_{f \in F_{h'}} \frac{1}{\delta_{f,pred(f)}^{opt}(h')} \right) / \sum_{r \in R} \rho_r^{min} \quad (3)$$

Combining $\Phi_a(\lambda)$ and $\Phi_b(\lambda)$ yields the overall objective function $\Phi(\lambda)$ (see formula 4), normalized by σ and weighted by a factor $0 \leq \alpha \leq 1$, the relative weight of the fulfillment of planning requirements.

$$\Phi(\lambda) = (1 - \alpha) * \sum_{h' \in H'} \sum_{f \in F_{h'}} \frac{1}{\delta_{f,pred(f)}(h', \lambda)} * \varphi_{h'} + \alpha * \sigma * \sum_{r \in R} \rho_r(\lambda) \quad (4)$$

In our experiments, this weight is set to $\alpha = 0.5$, so that robustness and the fulfillment of requirements are equally important.

A valid solution also has to adhere to some other constraints. The first restriction requires each start time μ_i of each line variant i to be inside the tact interval, with $t_{interval}$ being the duration of the interval (see formula 5).

$$\forall i \leq |\lambda| : 0 \leq \mu_i < t_{interval} \quad (5)$$

Another restriction requires an offset of at least one minute between two departures of the trips f and $pred(f)$ at each platform type $h' \in H'$ (see formula 6). This means that no platform can be blocked by more than one train at any point of time, the schedule has to be free of collisions.

$$\forall h' \in H' : \forall f \in F : \delta_{f,pred(f)}(h', \lambda) > 0 \quad (6)$$

We identify seven types of transport planning constraints: Interval constraints, start time constraints, core line constraints, bidirectional track constraints, turning point constraints, warranted connection constraints and follow-up connection constraints. Upon closer inspection (see [20], section 6.2.3) it becomes clear that interval and start time constraints are fundamental and all other constraint types can be expressed using these two. E.g. a bidirectional track constraint can be expressed by two interval constraints covering opposing platforms. Subsequently only interval and start time constraints are considered in the remainder of this paper.

The presented model is implemented as a branch-and-bound solver, which starts with an initial solution computed by a genetic algorithm for performance reasons. For implementation details see [21] or [20], sections 6.2.4 and 6.2.5.

2.2 Simulation of tram schedules

Most rail-bound traffic simulation models are designed for long distance train or railway networks, see e.g. [8, 9]. While those systems feature similarities to tram networks, e.g. passenger exchange or maneuvering capabilities, they differ significantly in important aspects. Tram networks are often mixed, i.e. trams travel on underground tracks as well as on street level, and are thus subject to individual traffic and corresponding traffic regulation strategies. Subsequently, tram behavior is a mixture between train and car behavior, e.g. line-of-sight operating/driving. Therefore a simple adaption of railway simulation methodologies is not feasible.

Our application is based upon a model-based parallelization framework (described in [20]), which exploits the embedded model’s intrinsic parallelism. The mixed tram network is modeled as a directed graph with platforms, tracks and track switches represented by nodes. Connections between nodes are represented as edges. The distributions for the duration of passenger exchange are specific to platform and tram type with the combined duration of opening and closing the vehicle doors as minimum value. Vehicles encapsulate most of the simulation dynamics, which are based upon the event based simulation approach (as described in [2]). Thus trams change their state at events of certain types, like stopping or accelerating, which happen at discrete points in time. These

state changes may trigger a change in the over-all system state and generate follow-up events, which are administrated in a priority queue.

Tram attributes are specified by the type of tram, which holds functions for the maneuvering capabilities, e.g. acceleration and braking. Main parameters of the simulation are the maximum driving velocity v_{max} , the dawdle probability $0 \leq p_d \leq 1$ (which maps the chance that a tram’s driver does not accelerate at a given moment due to external causes), and the dawdle factor $d > 1$ (which maps the amount of the delay caused by dawdling). For the experiments, these values are set as $p_d = 0.3$ and $d = 1.3$. A more detailed description of model and implementation can be found in [7].

3 Examining Montpellier’s Tramway network

We apply the described software suite to Montpellier’s Tramway network (for an overview see figure 3) based on the time table data of 2013 (gathered from [13], [14], [15], and [16]). The system consists of 84 stations with 176 platforms and 46 track switches, connected via 232 tracks (see [19]). These tracks cover a total length of ca. 56 kilometers, resulting in an average track length of about 241 meters. 1,215 trips per operational day are executed on four lines with 24 line routes (see figure 4), about 282,000 passengers are served on each weekday (see [17]).

3.1 Schedule generation

The schedule implemented by TAM has no global tact interval, trains serve the routes in varying patterns through the day. At peak times, lines 1 and 2 are traversing the city center every four to five minutes, but in changing patterns. Line 3 is served every six to eight minutes, the intervals between consecutive trains of line 4 are alternating between eight and nine minutes. To find an appropriate approximation of this, we assume a tact interval of eight minutes, and insert additional core lines 1A and 2A to double the frequency of lines 1 and 2 to four minutes. A set of planning requirements is defined, which can be decomposed to 16 interval constraints (see table 1). These include the additional core lines 1A and 2A (requirement 1 and 2), and minimum turn-around times at line ends (requirement 3).

The genetic algorithm is initialized with a population of 450 randomly generated individuals. The best fitness value of this first generation is 75.55 (average: 83.58, worst value: 95.00). In the course of 500 generations and a runtime of 313 seconds, the algorithm finds a best solution candidate with a fitness value of 75.25 (average: 75.51, worst value: 80.11). The branch-and-bound solver further enhances the minimal fitness value in the course of a 200 seconds run down to 75.22, and finds 128 optimal solutions.

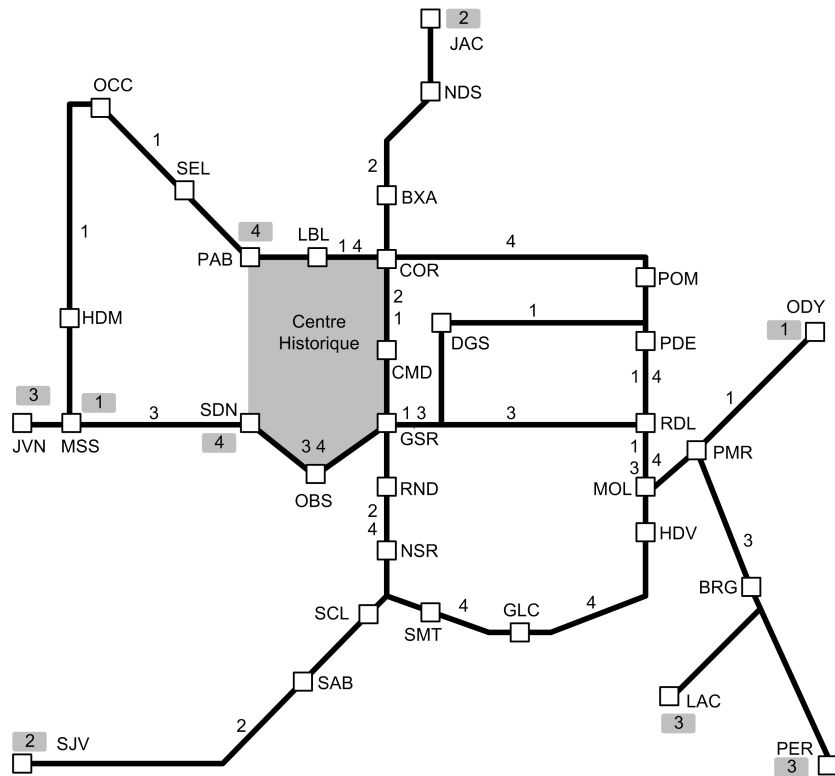


Figure 3: Montpellier's Tramway network



Figure 4: Montpellier's line routes

#	Req.	Variant 1	Variant 2	Hp. 1	Hp. 2	Interval priorities							
						0	1	2	3	4	5	6	7
1	1	1T01	1T05	9	-	0	0	2	1	1	1	2	0
2	1	1T02	1T06	115	-	0	0	2	1	1	1	2	0
3	2	2T01	2T05	35	-	0	0	2	1	1	1	2	0
4	2	2T02	2T06	136	-	0	0	2	1	1	1	2	0
5	3	1T01	1T02	30	115	0	0	1	1	1	1	1	1
6	3	1T02	1T01	173	174	0	0	1	1	1	1	1	1
7	3	1T05	1T06	30	115	0	0	1	1	1	1	1	1
8	3	1T06	1T05	94	9	0	0	1	1	1	1	1	1
9	3	2T01	2T02	55	140	0	0	1	1	1	1	1	1
10	3	2T02	2T01	116	31	0	0	1	1	1	1	1	1
11	3	2T05	2T06	51	136	0	0	1	1	1	1	1	1
12	3	2T06	2T05	120	35	0	0	1	1	1	1	1	1
13	3	3T01	3T02	78	163	0	0	1	1	1	1	1	1
14	3	3T02	3T01	151	56	0	0	1	1	1	1	1	1
15	3	4T01	4T02	152	67	0	0	1	1	1	1	1	1
16	3	4T02	4T01	100	15	0	0	1	1	1	1	1	1

Table 1: Distance constraints

3.2 Comparing generated schedules

We pick ten schedules each out of both the pool of initial solutions and the optimum solution pool and execute ten simulation runs for each of those 20 schedules. The maximum velocity is set to $v_{max} = 40$ km/h, a compromise between the observed inner city maximum speed of 30 km/h and the higher speed in some regions outside the city with exclusive track usage.

The runs under the initial schedules yield an average delay of all departures of 9.8 seconds. Under the best schedules the average delay is 8.2 seconds, which means a reduction of 16.3 percent or 1.6 seconds (see figure 5, left). The average delay of all delayed departures is reduced from 25.8 by 2.3 seconds or 8.9 percent to 23.5 seconds.

The frequency distribution of occurring delays was also collected (see figure 6). Under the optimal schedules, the numbers of delays in each bucket are reduced. This effect is especially significant for the larger delays of more than 60 seconds (see figure 7). The total number of departures with a larger delay is reduced from 521.3 under the random schedules by 210.7 departures or 40.4 percent down to 310.6 (see figure 5, right).

As seen, robust schedules reduce the average delay in the Tramway network, though only by a small amount, and significantly reduce the number of larger delays. Under optimal schedules with their better distributed time offsets, many small delays can be made up for fast and do not spread to consecutive departures. A higher robustness can thus help to reduce the number of larger delays by preventing inevitable small delays from accumulating over the simulation

Route	Schedule A					
	1	1A	2	2A	3	4
01	0	6	3	1	7	3
02	5	3	3	1	6	2

Route	Schedule B					
	1	1A	2	2A	3	4
01	6	1	4	0	1	5
02	3	7	7	3	0	5

Table 2: Randomly generated (left) and optimized (right) schedule

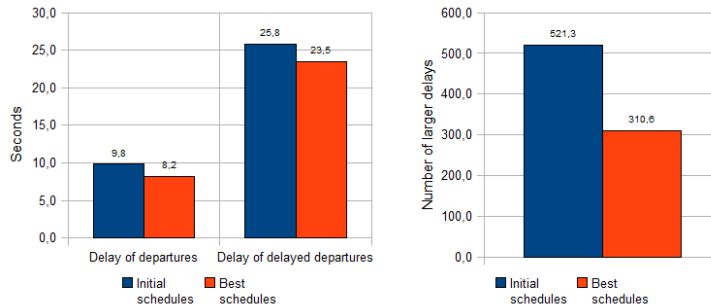


Figure 5: Average delay (left) and number of larger delays (right)

run.

To take a more detailed look at the model’s behavior, we pick a typical schedule A (see table 2, left) with an objective function value of 92.69 from the genetic algorithm’s initial pool of valid solution candidates, and a schedule B (see table 2, right) from the pool of best solutions. We examine both schedules by executing 100 simulation runs each and comparing the results.

On average, schedule A yields a line delay of 8.7 seconds, which is reduced under schedule B by 17.2 percent or 1.5 seconds to 7.3 seconds. The only line that yields a significantly lower delay under the optimal schedule is line 2, with a reduction of 24.3 percent from 21.6 to 16.3 seconds (see figure 8).

To examine this, we take a closer look at trips 3 and 4 of tram 2005 (see figure 9), which serves the shorter routes 205 and 206 of line 2A. While the measured delays at several platforms vary, the most obvious differences are found in the regions of the town center around Corum (COR, see figure 3) and Gare Saint-Roch (GSR).

Serving trip 3 in the direction of Sabines (see figure 10), trams of line 2A enter an array of switches they share with lines 1, 1A, 2 and 4 after the departure at Corum. Under schedule A, the vehicle has to wait to access these common resources, and cannot regain the resulting delay until after the stop at Nouveau Saint-Roch (NSR). Under schedule B with its better distributed time offset, these resources are instantly accessible to the tram.

On the return trip in the direction of Notre-Dame de Sablassou (see figure 11), the tram has to navigate four consecutive switches between the stations Rondelet (RND) and Gare Saint-Roch. It shares some of these switches with all other lines. Under the random schedule A, the vehicle gets behind a tram

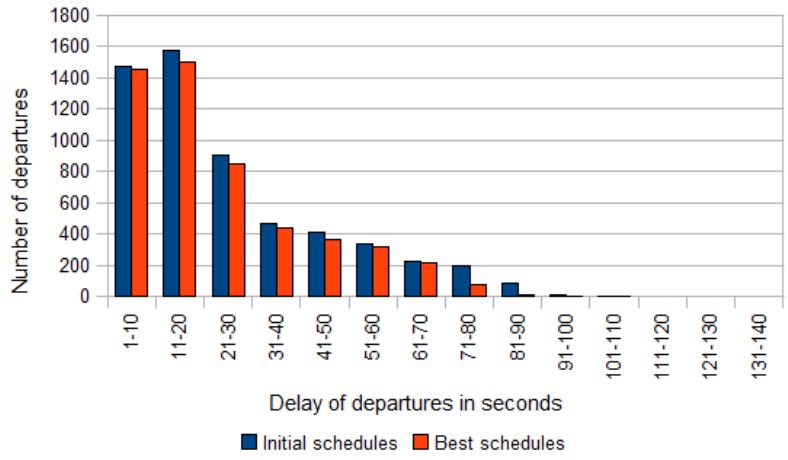


Figure 6: Frequency distribution of delays

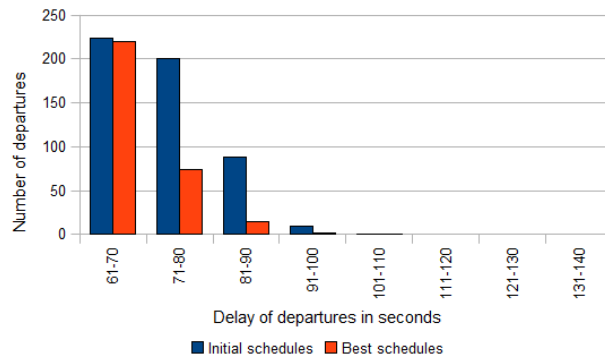


Figure 7: Frequency distributions of larger delays

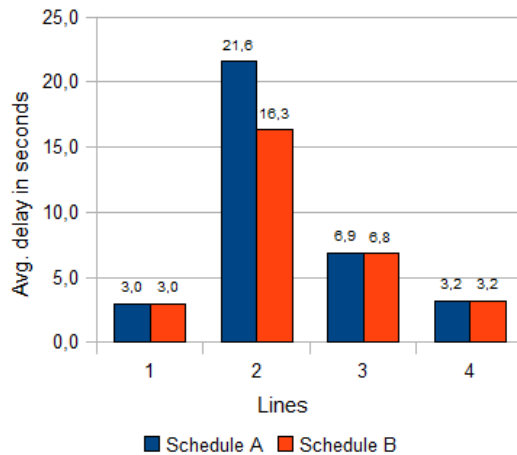


Figure 8: Average delay of lines

serving line 1, although it is scheduled to precede it by one minute. It therefore has to wait for that tram to clear the Gare Saint-Roch platform and thus gets a delay of about 80 seconds. It can start to regain the delay after lines 1 and 2 split course before Corum station.

As described, only line 2 (and its companion line 2A) shows a significantly lower delay under the robust schedule B, the other lines yield the same values under both schedules. Lines 1 and 4 run in parallel for a while, then part way and rejoin after sections of different track lengths and planned driving times (see figure 3). Because of this, and considering that the optimizer can only generate valid schedules with a time offset of at least one minute at each platform, these lines are locked in relation to each other under all valid schedules. There is no way the optimizer can generate a better (or worse) schedule concerning these two lines. The same applies to the combination of lines 3 and 4: These are also locked under any valid schedule.

This phenomenon does not occur with line 2. It runs parallel to lines 1 and 4, but only in one contiguous section each. It does then split from these lines but does not rejoin them later. These lines are therefore not locked, the optimizer can schedule line 2 more freely.

The experiments show that the application of a robust schedule can help to reduce delays in Montpellier’s tram network. They also show that robustness has its main impact in those regions of the networks where resources are shared by most line routes. In the Tramway network these regions are switch arrays near the stations Gare Saint-Roch and Corum.

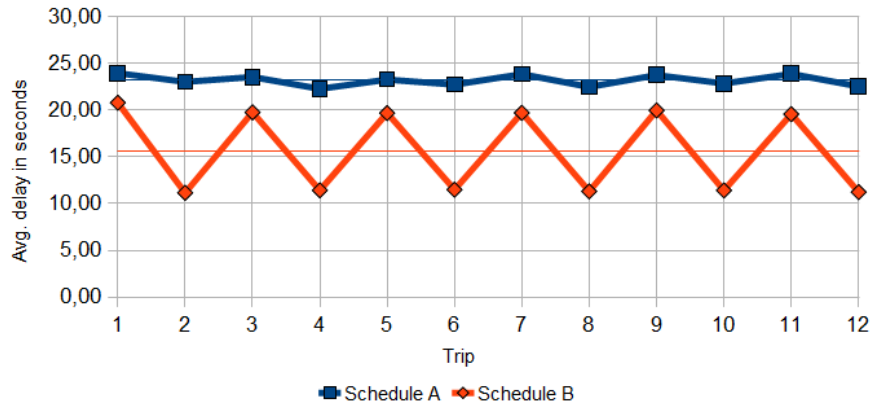


Figure 9: Average trip delays of tram 2005 serving line 2A

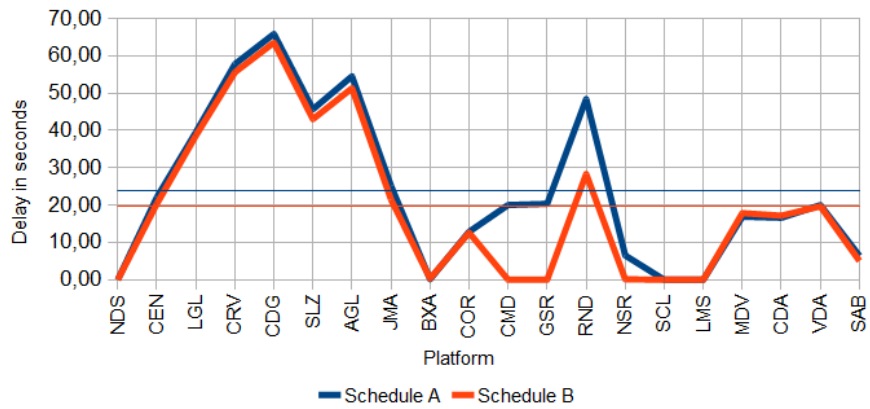


Figure 10: Average delays at platforms of trip 3 of tram 2005

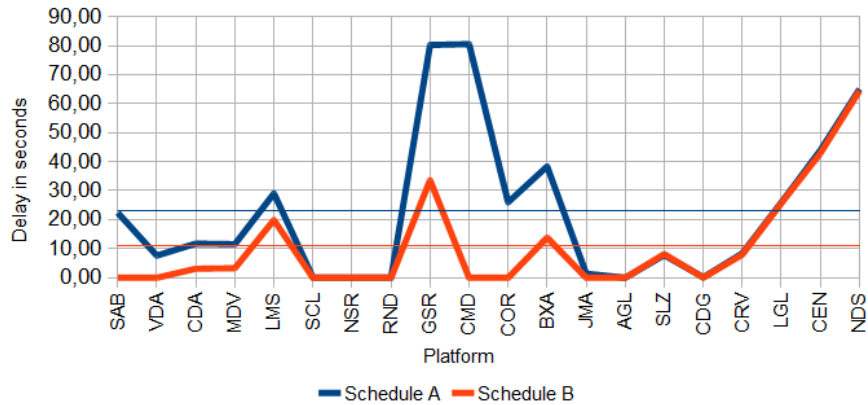


Figure 11: Average delays at platforms of trip 4 of tram 2005

3.3 TAM’s applied schedule

To complete the picture of Montpellier’s Tramway network, we also examine the schedule applied by TAM at the time of this writing. Because it adheres to no common tact interval and comprehends planning requirements unknown to the authors, the results cannot be compared directly to the generated schedules. Therefore, no insights about special traffic phenomenons should be assumed.

As described in section 3.1, TAM’s schedule has no common tact interval. Therefore, the numbers of the started trips per hour deviate in the sample period of 08.00 to 16.59 (see figure 12) from their counterparts of the generated schedules.

The data gathered by executing 100 simulation runs with the described parameters shows TAM’s schedule to be in general range with the generated schedules. The average delay of departures of 8.1 seconds is slightly smaller than the value yielded by schedule B, and 1.7 seconds smaller than that of schedule A (see figure 13). The average delay of delayed departures has a value of 24.4 seconds and is therefore splitting the distance between schedule A (25.8 seconds) and schedule B (23.5 seconds). The number of larger delays is 314.4, on about the level of schedule A’s value.

The frequency distribution shows that TAM’s schedule yields a lower number of small delays, which are compensated by a higher number of delays of more than 70 seconds (see figure 14).

TAM’s schedule yields line delays which are comparable to the values resulting from the generated schedules (see figure 15): line 1 has the same value under all three schedules, line 2 lies between the values of schedules A and B, lines’ 3 and 4 delay values are a bit higher than their counterparts.

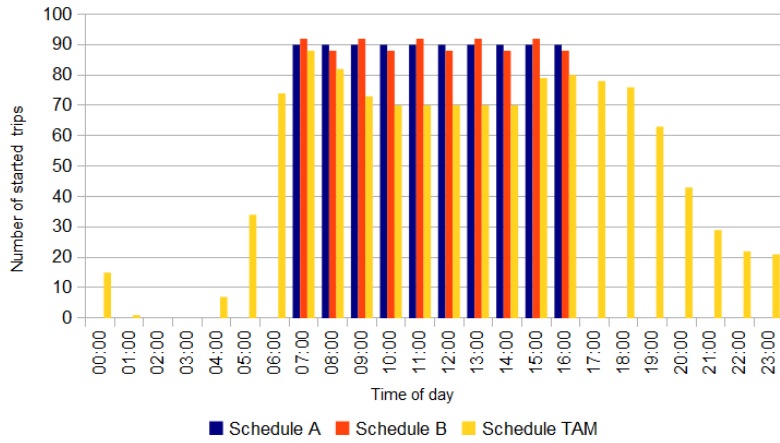


Figure 12: Number of started trips per hour

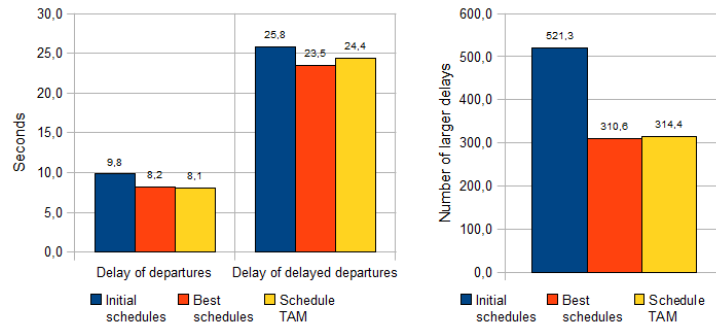


Figure 13: TAM's applied schedule - Average delay (left) and number of delays (right)

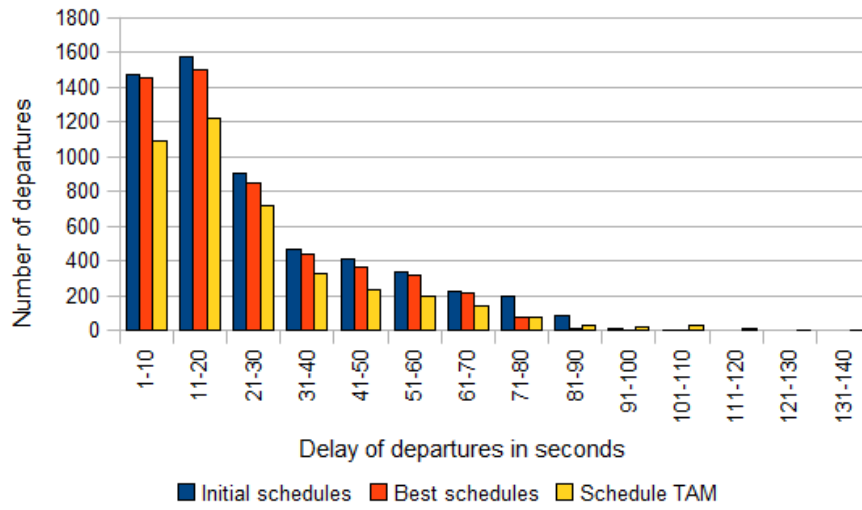


Figure 14: TAM's applied schedule - Frequency distribution

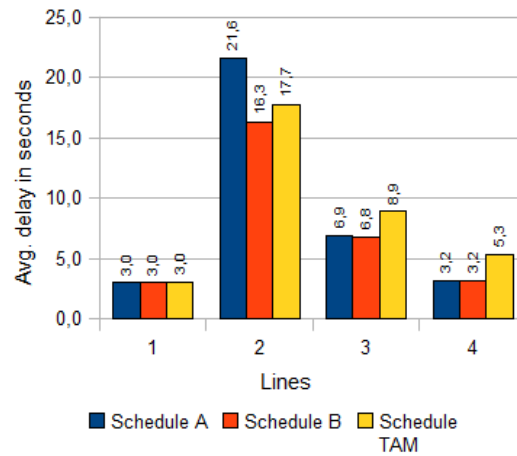


Figure 15: TAM's applied schedule - Average delay of lines

4 Conclusions and further research

This article showed an approach to examine the influence of robustness on a tram network. To accomplish this, we applied optimization and simulation tools designed to evaluate schedules optimized for robustness. These software utilities were employed to generate and compare robust and non-robust schedules for Montpellier's tram network, demonstrating that a robust schedule can indeed help to reduce delays in the Tramway network. The experiments showed that the main improvements center in those regions of the networks where resources are shared by most line routes. In the presented case these regions are the switch arrays near the stations Gare Saint-Roch and Corum.

Montpellier's Tramway network is expanding: a line 5 is currently being built and will connect the fast growing suburbs in the north and west to the inner city (see [18]). Supporting a rerouted line 4, this line will complete the ring track around the historical city center. Line 5 is planned to commence operation in 2017. The city of Montpellier already commissioned lines 6 and 7, their exact routes are still under consideration. Our model will be expanded with representations of these lines, the resulting model's behavior will be analyzed and compared to the existing model's.

We also plan to analyze under which general circumstances a robust schedule will increase punctuality in a tram network. The presented results of Montpellier's Tramway and of our hometown Cologne's KVB network (see [21]) will be utilized as a base for this.

References

- [1] Bampas, E., Kaouri, G., Lampis, M., Pagourtzis, A.: Periodic Metro Scheduling. In: Proceedings of ATMOS, 2006.
- [2] Banks, J., Carson, J.S., Nelson B.L., Nicol D.M.: Discrete-Event System Simulation. Pearson, 2010.
- [3] Cacchiana, V., Caprara, A., Fischetti, M.: A Lagrangian Heuristic for Robustness, with an Application to Train Timetabling. Transportation Science, to appear.
- [4] Caimi, G., Fuchsberger, M., Laumanns, M., Schüpbach, K.: Periodic Railway Timetabling with Event Flexibility. In: Networks, 2010, Volume 57, Number 1, pp. 3-18.
- [5] Genç, Z.: Ein neuer Ansatz zur Fahrplanoptimierung im ÖPNV: Maximierung von zeitlichen Sicherheitabständen. Dissertation, Mathematisch-Naturwissenschaftliche Fakultät, Universität zu Köln, 2003
- [6] Institut national de la statistique et des études économiques: La population de Montpellier Agglomération a triplé au cours des cinquante dernières années.

- http://www.insee.fr/fr/themes/document.asp?ref_id=16088, accessed on June, 26th, 2013.
- [7] Lückcrath, D., Ullrich, O., Speckenmeyer, E: Modeling time table based tram traffic. In: Simulation Notes Europe (SNE), ARGESIM/ASIM Pub., TU Vienna, Volume 22, Number 2, August 2012, pp. 61-68.
 - [8] Middelkoop, D., Bouwman, M.: SIMONE: Large Scale Train Network Simulations. In: B.A. Peters, J.S. Smith, D.J. Medeiros, M.W. Rohrer (Ed.): Proceedings of the 2001 Winter Simulation Conference, Arlington, 2001.
 - [9] Nash, A., Huerlimann, D.: Railroad Simulation Using OpenTrack. In: Allan, J., R.J. Hill, C.A. Brebbia, G. Sciutto, S. Sone (Ed.): Computers in Railways IX, WIT Press, Southampton, 2004, pp. 45-54.
 - [10] Schöbel, A.: A Model for the Delay Management Problem based on Mixed-Integer-Programming. In: Proceedings of ATMOS, 2001.
 - [11] Speckenmeyer, E., Li, N., Lückcrath, D., Ullrich, O.: Socio-Economic Objectives in Tram Scheduling. Technical Report, Universität zu Köln, to appear.
 - [12] Suhl, L., Mellouli, T.: Managing and preventing delays in railway traffic by simulation and optimization. In: Mathematical Methods on Optimization in Transportation Systems 2001, pp. 3–16.
 - [13] Transports de l'agglomération de Montpellier (Ed.: Subra, R.): Horaires tram 1. 2012.
 - [14] Transports de l'agglomération de Montpellier (Ed.: Subra, R.): Horaires tram 2. 2012.
 - [15] Transports de l'agglomération de Montpellier (Ed.: Subra, R.): Horaires tram 3. 2012.
 - [16] Transports de l'agglomération de Montpellier (Ed.: Subra, R.): Horaires tram 4. 2012.
 - [17] Transports de l'agglomération de Montpellier: Un réseau en étoile. http://www.montpellier-agglo.com/tam/page.php?id_rubrique=31, accessed on Feb, 21st, 2013.
 - [18] Transports de l'agglomération de Montpellier: Le tracé du ligne 5. http://www.ligne5-montpellier-agglo.com/?page_id=16, accessed on Feb, 21st, 2013.

- [19] Transports de l'agglomération de Montpellier (Ed.: Subra, R.): Plan du réseau du centre de l'agglomération. 2012.
- [20] Ullrich, O.: Modellbasierte Parallelisierung von Anwendungen zur Verkehrssimulation - Ein dynamischer und adaptiver Ansatz. Dissertation, Univ. Köln, to appear.
- [21] Ullrich, O., Lückcrath, D., Franz, S., Speckenmeyer, E.: Simulation and optimization of Cologne's tram schedule. In: Simulation Notes Europe (SNE), ARGESIM/ASIM Pub., TU Vienna, Volume 22, Number 2, August 2012, pp. 69-76.
- [22] Verband Deutscher Verkehrsunternehmen e.V.: VDV-Standardschnittstelle Liniennetz/Fahrplan, VDV-Schriften 452, 2008.