Multi-depot multi-vehicle-type vehicle scheduling for Cologne's tram network

Daniel Lückerath[1], Oliver Ullrich[2], Aleksander Kupicha[3], Ewald Speckenmeyer[1]

[1]Institut für Informatik, Universität zu Köln

[2]National Science Foundation's Industry-University Cooperative Research Center for Advanced Knowledge Enablement, Florida International University

[3]Prof. Roll & Pastuch – Management Consultants

lueckerath@informatik.uni-koeln.de

To be a feasible base for simulation studies of Cologne's tram network, a valid vehicle schedule has to consider several requirements, like multiple vehicle depots and multiple types of vehicles. The local transport provider utilizes both low-floor and high-floor vehicles, with high-floor vehicles being qualified to serve both high-floor and low-floor platforms. Therefore mixed vehicle rotations are acceptable, but generally not desired. This paper presents a set of models which adhere to these requirements, while also considering several possible optimization goals, like minimum number of deployed vehicles, and minimum combined length of maintenance trips.

1 Introduction

In recent work, some of the authors conducted simulation studies on the influence of robust time tables on punctuality in tram networks, especially in the tram networks of the cities of Montpellier (see [15]) and Cologne (see [13]). A combination of heuristic and exact optimization methods was applied to generate robust time tables, which then were simulated with a microscopic simulation model (see [11]). It has been shown that a tram network has to fulfill a set of structure constraints for robust time tables to have an effect on overall punctuality (see [14]). Up until now, the vehicle schedules, consisting of the assignment of the scheduled trips to a fleet of vehicles, were generated by simple heuristic methods. The resulting vehicle schedules were usually feasible, but were not considering optimization goals like cost minimization or maximizing robustness. They were thus not very realistic and restricting the accuracy of the simulation results.

To address this issue, this paper presents a network flow model and its accompanying integer linear model based on the model introduced in [8], which adhere to the requirements for a feasible vehicle schedule for Cologne's tram network while considering several optimization goals like minimizing the number of deployed vehicles, minimizing the combined lengths of maintenance trips, or minimizing overall cost. A CPLEX implementation of this model is then utilized to generate such schedules for Cologne's tram network.

This paper continues with sharing some background on vehicle scheduling and recent research on the subject (section 2). Following that, an optimization model for multi-depot, multi-vehicle-type vehicle scheduling for Cologne's tram network is presented (section 3). Several experiments are conducted, demonstrating the adaptivity of the model for different optimization goals (section 4). The paper closes with a short summary of lessons learned and some thoughts on further research (section 5).

2 Background

2.1 Vehicle scheduling

A vehicle schedule R consists of an assignment of scheduled trips $f \in F$ to one of a fleet of vehicles, with $F = F_s \cup F_m$, and F_s the set of planned service trips, and F_m the set of maintenance trips. A rotation $r = (f_{i_1}, f_{i_2}, \dots, f_{i_n})$ for a given vehicle usually starts with a maintenance trip from the depot, where the vehicle is stored, to the start platform of the first service trip. After this service trip the vehicle may continue with a maintenance trip to the start platform of the next service trip, etc. The rotation ends with a return trip to the depot. A vehicle schedule $R = (r_1, \dots, r_k)$ consists of a set of rotations covering all planned trips of an operational day. The vehicle scheduling optimization problem consists of finding the optimal vehicle schedule R^*, usually regarding minimal cost.

Vehicle scheduling problems are frequently solved using network flow models. **Figure 1** depicts an example of such a model for a simple single-depot vehicle scheduling problem with one vehicle type. The graph can be transformed into an integer linear problem which can then be solved by a software solver like CPLEX.

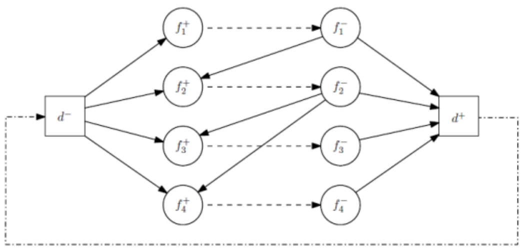

Figure 1. Simple network flow model for vehicle scheduling. Dashed lines are service trips, solid lines represent maintenance trips

As a first step, a given instance is represented by a graph $G(V,F)$ with vertices $v,w \in V$ representing start and end platforms of trips and edges $e = (v,w) \in F$ representing the trips. A cost function $c(e), c: F \to \mathbb{R}$ maps the cost of each trip, which is usually proportional to the length of the corresponding trip. A depot $d \in D$ is a marked vertex; its capacity λ_d represents the number of vehicles which can be stored in d. There is only a single depot in this example.

Each trip $f \in F$ starts at a platform f^+ and ends at a platform f^-. F is defined as $F = F_{d,f} \cup F_s \cup F_m \cup F_{f,d}$ and thus consists of the maintenance trips $F_{d,f} \subset F_m$ from the depot d to the start of each service trip $f \in F_s$, the regular service trips F_s, the maintenance trips $(f_i, f_j) \in F_m, i \neq j$ from the end of a trip f_i to the start platform of each trip f_j with$f_i < f_j$, and the return trips from the last platform of a rotation to the depot $F_{f,d} \subset F_m$. We define an order of time compatibility on the set of trips: $f_i < f_j$ means f_j can be served after f_i. This order considers the transfer time of a vehicle to get in time for the scheduled departure from the last platform of trip f_i to the first platform of trip f_j. If $f_i \nless f_j$, trips f_i and f_j cannot be served by the same vehicle.

The network flow model is then transformed into an integer linear model (see [16]), as shown in **Table 1**.

Minimize	$\sum_{f \in F} c_f x_f$		(OF)
Subject to	$x_f = 1$	$\forall f \in F_s$	(C1)
	$x_f \in \{0,1\}$	$\forall f \in F_m$	(C2)
	$\sum_{f \in \delta^+(v)} x_f = \sum_{f \in \delta^-(v)} x_f$	$\forall v \in V$	(C3)
	$x_{(d^-,d^+)} \leq \lambda_d$		(C4)

Table 1. Integer linear program for a simple vehicle schedule

The elements x_f of the solution vector x are interpreted as decisions whether a potential trip is covered by a vehicle. Obviously all service trips have to be covered, therefore $x_f = 1$ for all $f \in F_s$ (see (C1)). For each maintenance trip $f \in F_m$ the value of x_f can either be 1, if this trip is covered by a vehicle, or 0, if it is not (see (C2)). Constraint (C3) denotes that the number of outgoing edges $\delta^+(v)$ of a node v which are covered by a vehicle has to be equal to the number of covered incoming edges $\delta^-(v)$. This means that each vehicle which enters a platform has to leave it subsequently. The edge between d^- and d^+ denotes a virtual trip and can be interpreted as a counter of deployed vehicles. Because of (C3) all vehicles leaving the depot at d^+ have to return to it eventually via d^-. Constraint (C4) therefore sets an upper bound to the number of deployed vehicles.

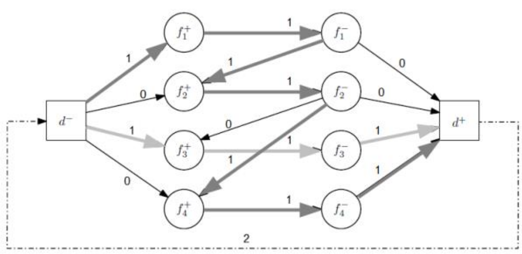

Figure 2. Vehicle 1 executes (f_1, f_2, f_4), vehicle 2 executes (f_3)

A valid solution to the example is shown in **Figure 2**. The value $x_{(d^+,d^-)} = 2$ denotes that two vehicles are employed: the first vehicle leaves the depot, executes the trips f_1, f_2, and f_4, and then returns to the depot;

the second vehicle leaves the depot, executes trip f_3 and then also returns to the depot.

The optimizer finds the vector x which minimizes the objective function (OF) value, and thus yields the combination of rotations with the minimum cost. Because all service trips have to be covered under any such vehicle schedule, this yields a schedule R^* with minimum cost for maintenance trips.

For each instance of this simple vehicle scheduling problem, a network flow model and a corresponding integer linear model can be generated which allows for an optimal solution to be found by the well known methods of integer linear programming (see [16]).

2.2 Related Work

The vehicle scheduling problem has been extensively covered in the past 50 years and several different formulations and approaches exist (see e.g. [1, 2, 3, 5, 6, 7, 8, 9, 10] or, for an overview, see [4]). While the general vehicle scheduling optimization problem is known to be NP-hard (see [1]) some special cases are known to be in P and can be solved efficiently. Gavish and Shlifer in [7] for example use a (quasi-) assignment model to minimize cost resulting from fleet size and maintenance trips for solving the single-depot vehicle scheduling problem with only one vehicle type. Similarly, Bodin et al. in [2] use the network flow approach to convert the single-depot vehicle scheduling problem into a minimum cost flow problem.

More realistic (albeit NP-hard) instances arise when multiple depots and vehicle types are considered. In those cases the problem is often solved using multi-commodity models (as in [1, 8, 10]) or set partitioning formulations (as in [9]). Kliewer, Mellouli and Suhl in [10] for example apply a two-stage aggregation process to reduce the number of decision variables before solving the multi-depot multi-vehicle-type vehicle scheduling problem using a multi-commodity approach. Grötschel, Schöbel and Völker in [8] on the other hand first solve the corresponding single-depot problem before applying heuristic methods to repair invalid rotations, i.e. rotations including service trips not compatible with the respective depot. Hadjar, Marcotte and Soumis in [9] in turn develop a branch-and-bound algorithm combining column generation, variable fixing and cutting planes to solve the problem with the set partitioning formulation.

3 Vehicle scheduling for Cologne’s tram network

Cologne's local transport provider utilizes both low-floor and high-floor vehicles based in several depots, with high-floor vehicles being qualified to serve both high-floor and low-floor platforms. Therefore mixed vehicle rotations (i.e. rotations containing both low-floor and high-floor service trips) are acceptable to some extend, but generally not desired. As a result a feasible vehicle schedule for Cologne's tram network has to consider several requirements, like multiple vehicle depots and multiple types of vehicles. A feasible model should also enable several optimization goals: a minimum number of employed vehicles (as acquisition and maintenance of vehicles is expensive), a minimum combined length of connecting trips (as too many non-service trips congest the network), minimum overall cost, or a balance of those.

The simple model shown in section 2.1 does obviously not accommodate those requirements, but it can be utilized as a starting point to build a more complex model.

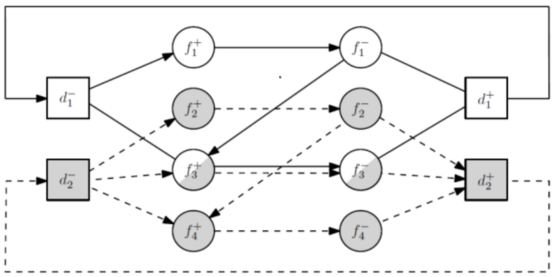

Figure 3. Multi-depot vehicle scheduling

To accommodate for multiple depots, we use a multi-commodity model based on the one presented in [8] (see **Figure 3**), which allows for several depots $d \in D$, with each depot storing only one type of vehicles. The set F^d denotes the service trips a depot d can serve, thus considering multiple vehicle types (indicated by node color in **Figure 3**). Set $D^f \subseteq D$ denotes the subset of all depots from which a service trip f can be served. There exist trips which can be served by several depots and vehicle types (trip f_3 in **Figure 3**), therefore $F^{d_1} \cap F^{d_2}$ will typically not be empty.

Service trips have to be covered under each valid vehicle schedule. Thus, the combined cost of service trips can be considered constant, it can therefore be parametrized. The resulting compressed model (see

Figure 4) manages on less decision variables and can thus be computed faster.

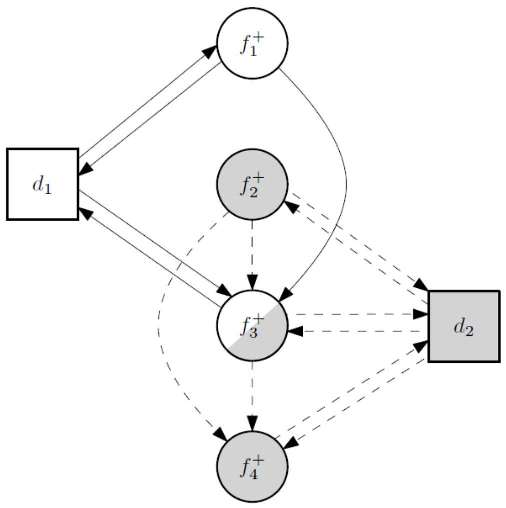

Figure 4. Multi-depot vehicle scheduling with compression

Table 2 shows an integer linear program for the multi-depot multi-vehicle-type vehicle schedule problem. Here, the elements x_{f_1,f_2} of the solution vector x are interpreted as decisions whether a potential maintenance trip between the end platform of trip f_1 and the start platform of trip f_2 should be covered by a vehicle.

Min.	$\sum_{d\in D}[\sum_{f\in F_s^d}((v_{(d,f)}+c)*x_{(d,f)})$		(O1, O2)
	$+\sum_{f_1\in F_s}\sum_{f_1<f_2}(v_{(f_1,f_2)}*x_{(f_1,f_2)}^d)$		(O3)
	$+\sum_{f\in F_s^d}(v_{(f,d)}*x_{(f,d)})]$		(O4)
S.t.	$\sum_{d\in D}(x_{(f_1,d)}+\sum_{f_1<f_2}x_{(f_1,f_2)}^d)=1$	$\forall f_1\in F_s^d$	(C1)
	$[x_{(d,f_1)}+\sum_{f_2<f_1}x_{(f_2,f_1)}^d$ $-\sum_{f_1<f_2}x_{(f_1,f_2)}^d-x_{(f_1,d)}]=0$	$\forall d\in D,$ $\forall f_1\in F_s^d$	(C2)
	$0\le\sum_{f\in F_s}x_{(d,f)}\le\lambda_d$	$\forall d\in D$	(C3)
	$x_{(i,j)}\in\{0,1\}$	$\forall i,j$ $\in D\cup F_s$	(C4)

Table 2. Integer program for a multi-depot multi-vehicle-type vehicle schedule

The objective function considers (O1) the fixed cost c of a vehicle's deployment, (O2) the cost of the first maintenance trip from the depot to the first platform of its first trip, (O3) the combined cost of the maintenance trips connecting service trips, and (O4) the return trip to the depot from the last platform of the last service trip.

Constraint (C1) guarantees that for every service trip f only one of the possible succeeding trips is selected. Together with the network flow conservation constraint (C2) this guarantees that each trip is covered by at most one vehicle and has only one preceding trip. Constraint (C3) guarantees for each depot a number of deployed vehicles that is within this depot's capacity, while (C4) guarantees that each potential maintenance trip is either covered by a vehicle or not.

Several optimization goals can be reached by varying the fixed cost c: If c is set to a value greater than the maximum length of maintenance trips v_{max}, then executing maintenance trips is generally preferred to deploying another vehicle, thus minimizing the total number of deployed vehicles. If $0 < c < v_{min}$, then the model prefers deploying another vehicle to executing any maintenance trips with length greater than zero, thus minimizing the combined lengths of maintenance trips. By sweeping fixed cost c between v_{min} and v_{max} a trade-off between number of deployed vehicles and lengths of maintenance trips may be observed.

4 Experiments

4.1 Modeling Cologne's tram network

We apply the developed model to our hometown Cologne's tram network based on the time table data of 2001 (see **Figure 5**). It consists of 528 platforms and 58 track switches connected via 584 tracks. These tracks cover a total length of 407.4 kilometers, resulting in an average track length of 697.6 meters. 15 lines with 182 line routes are served by 178 vehicles which execute 2,814 trips per operational day. The vehicles are stored in three maintenance depots, two of them store high-floor vehicles (near stations Aachener Straße/Gürtel (ASG) and Niehler Straße/Gürtel (NSG)), and one stores low-floor vehicles (near station Kalk Kapelle (KKP)).

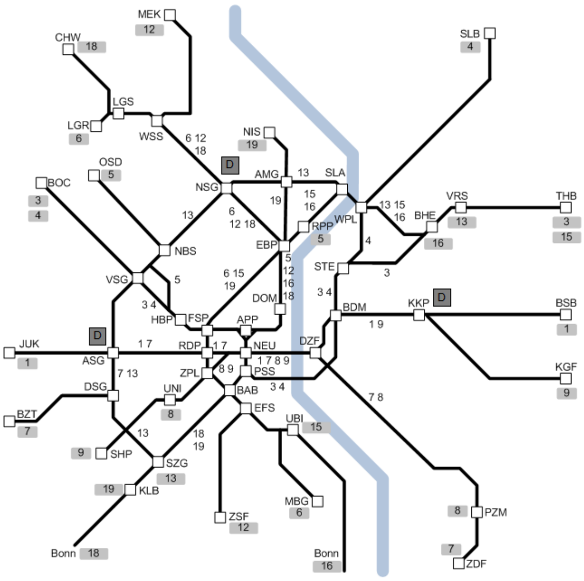

Figure 5. Cologne's tram network

As test scenario we chose the tram schedule of 2001 from 3 am to 12 pm, which covers most of the planned service trips of a typical operational day. The described instance is solved via the CPLEX software package. We conduct five experiments:

(E1) The fixed cost is set to a value $c > v_{max}$ to minimize the number of deployed vehicles. For Cologne's tram network v_{max} is established as $v_{max} = 41.907$ kilometers, describing the distance from station Chorweiler (CHW) to station Bad Godesberg Stadthalle in the neighboring town of Bonn. The fixed cost are accordingly set to $c = 41.908$.

(E2) The fixed cost is set to a value $c < v_{min}$ to minimize the combined length of maintenance trips. The value of v_{min} is established as $v_{min} = 1.342$ kilometers, occurring between stations Zollstock Südfriedhof (ZSF) and Klettenbergpark (KLB). For this experiment the fixed cost is set to $c = 1.341$.

(E3) A sweep over $v_{min} \leq c \leq v_{max}$ is conducted to explore the trade-off between the number of deployed vehicles and the lengths of maintenance trips.

(E4) Up until now, low-floor platforms could be served by both high-floor and low-floor vehicles. For this experiment we explicitly forbid mixed vehicle rotations, which results in two separated problem instances.

(E5) This experiment allows mixed vehicle rotations, but sets a penalty by doubling the cost of low-floor trips served by high-floor vehicles. For (E4) and (E5) c is again set to 41.908.

4.2 Results and discussion

Table 3 shows the results of experiments (E1) and (E2). Setting the fixed cost to a value greater than v_{max} results in a vehicle schedule with 109 vehicles serving 18.72 service trips on average. In comparison, a fixed cost value less than v_{min} raises the number of utilized vehicles by 4.4 percent (or 5 vehicles) to 114 vehicles in total, which serve 17.91 service trips on average. By utilizing more vehicles variable cost can be lowered by 1.05 percent.

	(E1): Minimizing number of vehicles	**(E2): Minimizing length of maintenance trips**
Run time	2,470 s	1,037 s
Overall cost	6,007	1,577
Fix cost	4,567.86	152,87
Variable cost	1,439.14	1,424.13
Vehicles	109	114
μ_L	18.72	17.91
σ_L	8.79	8.15
$\min_L$	6	6
$\max_L$	50	46

Table 3. Results of experiments (E1) and (E2). μ_L and σ_L denote average rotation length and standard deviation. $\min_L$ and $\max_L$ denote minimum and maximum rotation length

A general trade-off between the number of vehicles and the length of the maintenance trips is highlighted by the results of experiment (E3) (see **Figure 6**). For $0 \leq c \leq 5.4$ a reduction of vehicles is compensated by longer maintenance trips. For fixed cost values of $c \geq 5.4$ both the number of utilized vehicles and the length of the maintenance trips stagnate, indicating that it is not possible to serve all planned service trips with fewer vehicles.

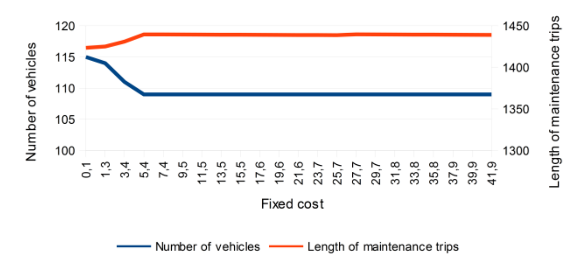

Figure 6. Results of experiment (E3)

The results of the last experiments (E4) and (E5) are shown in **Table 4**. As expected, banning mixed rotations reduces the set of valid solutions and subsequently results in a less efficient vehicle schedule compared to the solutions from experiments (E1) and (E2). On the other hand, penalizing mixed rotations in (E5) does not result in significant changes compared to (E1).

	(E4): No mixed rotations	**(E5): Penalty for mixed rotations**
Run time	436.92 s	7,650.91 s
Overall cost	6,592	6,251
Fix cost	4,651.68	4,567.86
Variable cost	1,940.32	1,683.14
Vehicles	111	109
μ_L	18.10	18.66
σ_L	9.27	8.08
$\min_L$	4	6
$\max_L$	51	52

Table 4. Results of experiments (E4) and (E5)

5 Summary and further research

In this paper, we shared an optimization model to generate multi-depot, multi-vehicle-type vehicle schedules for Cologne's tram network. This model can be tuned to consider optimization goals like minimizing the number of deployed vehicles, minimizing the combined lengths of maintenance trips, or minimizing overall cost. Several series of experiments showed the applicability of the model while exploring its tuning capabilities.

In a further step, the described model will be applied to generate vehicle schedules for given time tables, which in turn are generated by the optimization tools described in [12] and [13]. These combined schedules will then be simulated with the simulation engine described in [11] to further validate their applicability.

6 Acknowledgements

This material is based in part upon work supported by the National Science Foundation under Grant Nos. I/UCRC IIP-1338922, AIR IIP-1237818, SBIR IIP-1330943, III-Large IIS-1213026, MRI CNS-0821345, MRI CNS-1126619, CREST HRD-0833093, I/UCRC IIP-0829576, MRI CNS-0959985, FRP IIP-1230661 and U.S. Department Transportation under a TIGER grant.

7 References

[1] Bertossi, A. A., Carraresi, P., Gallo, G.: On Some Matching Problems Arising in Vehicle Scheduling Models. In: Networks, Volume 17, Number 3, 1987, pp. 271-281.

[2] Bodin, L., Golden, B., Assad, A., Ball, M.: Routing and scheduling of vehicles and crews: the state of the art. In: Computers & Operations Research, Volume 2, Number 10, 1983, pp. 63-211.

[3] Borndörfer, R., Löbel, A., Weider, S.: A bundle method for integrated multi-depot vehicle and duty scheduling in public transport. In: Computer-aided systems in public transport. Springer, 2008, pp. 3-24.

[4] Bunte, S., Kliewer, N.: An overview on vehicle scheduling models. In: Public Transport, Volume 1, Number 4, 2010, pp. 299-317.

[5] Ceder, A.: Public-transport vehicle scheduling with multi vehicle type. In: Transportation Research Part C, Volume 19, 2011, pp. 485-497.

[6] Daduna, J. R., Branco, I., Paixao, J. M. P.: Vehicle Scheduling Problem with Multiple Types of Vehicles and a Single Depot. In: Computer-Aided Transit Scheduling. Springer, 1995, pp. 115-129.

[7] Gavish, B., Shlifer, E.: An approach for solving a class of transportation scheduling problems. In: European Journal of Operational Research, Volume 3, 1978, pp. 122-134.

[8] Grötschel, M., Löbel, A., Völker, M.: Optimierung des Fahrzeugumlaufs im öffentlichen

Nahverkehr. In: Mathematik Schlüsseltechnologie für die Zukunft. Springer, 1997, pp. 609-624.

[9] Hadjar, A., Marcotte, O., Soumis, F.: A Branch-and-Cut Algorithm for the Multiple Depot Vehicle Scheduling Problem. In: Operations Research, Volume 54, Number 1, 2006, pp. 130-149.

[10] Kliewer, N., Mellouli, T., Suhl, L.: A new solution model for multi-depot multi-vehicle-type vehicle scheduling in (sub)urban public transport. In: Proceedings of the 13th Mini-EURO Conference and the 9th meeting of the EURO working group on transportation, 2002.

[11] Lückerath, 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.

[12] Ullrich, O.: Modellbasierte Parallelisierung von Anwendungen zur Verkehrssimulation - Ein dynamischer und adaptiver Ansatz. Dissertationsschrift, Univ. Köln, 2014.

[13] Ullrich, O., Lückerath, 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.

[14] Ullrich, O.; Lückerath, D.; Speckenmeyer, E.: Reduzieren robuste Fahrpläne Verspätungen in Stadtbahnnetzen? - Es kommt drauf an! In: Proceedings of HEUREKA 2014, to appear.

[15] Ullrich, O.; Lückerath, D.; Speckenmeyer, E.: A robust schedule for Montpellier's Tramway network. Technical Report, Univ. Köln, 2013, 17 pg.

[16] Vanderbei, R. J.: Linear Programming - Foundations and Extensions. Third Edition. Springer, 2008.