The Stochastic Line Planing Problem - A problem case for the BVL BASF-Hackathon

Markus Frey, Bastian Kennel, Qingwei Li, Claus-Juergen Neumann, Minal Singhania, Tobias Heilmann and Jan Buchmann

Abstract—The BVL Hackathon took place during the "35. Deutscher Logistik Kongress" (English "35th German Logistic Convention") in Berlin 2018. Hosted by BASF, the event featured a competition held during one day around transportation in a futuristic digital city. As problem case we selected a new variant of the well-known line planning problem in which the demand of passengers, who have to be routed between locations, is stochastic and the lines for the vehicles have to be created. In this paper we describe the problem case in detail, the provided data and the structure of the game.

Index Terms—Hackathon, stochastic line planing problem, transportation demand simulation

1 Introduction

The federal association for logistics (German "Bundesvereinigung für Logistik", acronym BVL) Hackathon hosted by BASF was held during the "35. Deutscher Logistik Kongress" in Berlin in 2018 (see [1]). When creating the case, we strive for a design such that no specific industry knowledge, like from airline, manufacturing or logistic industry, gave any advantage, i.e. all participating companies had the same chance to succeed in the contest. Moreover, to meet the conference's motto "Digitalization meets reality", the focus of the problem should be in a futuristic field of logistics which is handled by means of predictive and prescriptive analytical methods of significant depth. As the time-slot for the competition was seven hours on one day, the problem case had to be solvable within this time.

Due to emissions and traffic in cities, new forms of mobility will arise in the near future. Public transportation should offer a high service quality with flexible lines adjusting to passengers' demand while running costs for required energy and infrastructure are minimized. One idea to reach both goals is using autonomous electric vehicles (AEVs). The idea of shared autonomous mobility is the motivation for our problem case. The participating teams, consisting of up to four people, took the role of the planning mangers for a fleet of AEVs. Their task is to create lines for AEVs to serve passengers' demand over the course of a day. The demand of people to be transported between two locations varies due to different factors like weather or distance. To react on these demand fluctuations, the lines for the AEVs can be changed between days, but not during the day. The objective is to minimize the running costs of the AEVs consisting of vehicle provision, energy and a penalty cost for unmet demand.

The problem is similar to the well-known line planing problem (LPP) in which lines have to be selected such that the running costs are minimal. Normally these lines cannot be changed from day to day. So the problem is typically on a tactical or strategical level, while our focus is on the operational level where lines can be changed for the next day. The lines of the AEVs can be created from scratch, i.e. no predefined lines are given as it is also very common for the LPP. Moreover, while in the LPP the demand for passengers is assumed to be deterministic, we consider the demand to be stochastic. To the best of our knowledge this leads to a new variant of the LPP which we denote as the stochastic LPP (SLPP). For the SLPP we present a model formulation and the generation of the demand.

• *All authors work with the Advanced Business Analytics unit in BASF Global Supply Chain Strategy and Performance.*
• *For contact use* firstname dot lastname at *basf.com*

Thanks to BASF, especially to Ralf Busche (ralf.busche@basf.com) for sponsoring the internal development and the event.

2 The Stochastic Line Planning Problem

This section defines the detailed universe of the problem statement. Section 2.1 describes the problem in detail with all met assumptions while section 2.2 introduces the required sets and parameters. The model formulation is presented in Section 2.3.

2.1 Problem Statement

The AEVs are deployed in a city to satisfy the transportation demand of the population within a day of 24 hours. The fleet of the AEVs is homogeneous, i.e. the number of passengers who can be transported is equal for all AEVs. Given a line through the city, the board computer of an AEV has to be programmed some time in advance such that it can follow the line automatically and stops at the locations to pick-up or drop-off passengers. Once the route for the AEV is programmed, it cannot be changed during the day. To each route, one or multiple AEVs can be assigned; there is no limit to the number of total AEVs in use. All locations on a line are visited by the AEV and one location can be visited by several lines. Lines may not contain circles.

We do not consider the exact timetabling or scheduling of the AEVs at the stations. The frequency of the line is given by the number of stops in a line per period indicating the basic timetable period and controlling the lines' transportation capacity (see [2], [3]). The frequency yields the number of times a station is visited by any of the assigned AEVs per period. To increase the frequency of a line, more AEVs can be assigned. The frequency multiplied with the number of AEVs and their capacities yields the number of passengers who can be transported on that line per period. As the passengers' perceived service quality should be as high as possible with that futuristic concept, we do not allow transfers between lines (see [4]).

The stochastic demand for the passenger transports between two locations is given by the stochastic origin-destination matrix (SODM, Γ) which does not need to be symmetric.

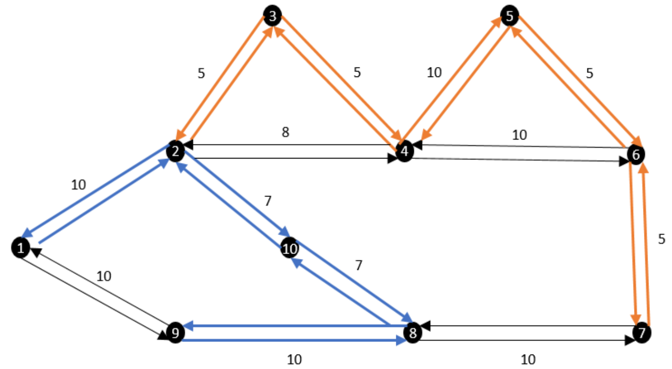

Fig. 1. Two lines represented by two bi-directed paths p_1 (blue line) and p_2 (orange line) in a city network.

2.2 Sets, Parameters and Assumptions

The planning day: is divided into time periods of equal length $\Delta = 60$ minutes. All periods of the day are contained in $\mathcal{T} := \{t_1, t_2 \ldots, t_{24}\}$ in which we have the order $t_k < t_{k+1}$ for all $k = 1, \ldots, 24$, i.e. period t_k is before period t_{k+1}.

The City: is represented by a network $N = (G, D)$ where a directed graph $G = (\mathcal{V}, \mathcal{A})$ represents the street network with the pick-up and drop-off locations for passengers given in node set $\mathcal{V}$. An arc $a = (i, j) \in \mathcal{A} \subset \mathcal{V} \times \mathcal{V}$ gives the connection between two locations $i, j \in \mathcal{V}$. Arc functions $D_a = D_{i,j}$ yields the travel time for arc $a = (i, j)$.

Passenger demand: As we do not have complete information about the passengers' demand let Ω be the set of possible demands realizations. The probability for demand realization ω in period t is given by weight $p^t_\omega \geq 0$, with $\sum_\omega p^t_\omega = 1$ for all $t \in \mathcal{T}$. For scenario $\omega \in \Omega$, the demand of the passengers between two locations in period $t \in \mathcal{T}$ is given in the SODM $\Gamma(t, \omega)$. An entry $\gamma^{t,\omega}_{i,j}$ of matrix $\Gamma(t, \omega)$ yields the demand for passengers traveling from location i to location j in scenario ω during time t. The tuples for which we have $\sum_{t \in \mathcal{T}} \gamma^{t,\omega}_{i,j} \neq 0$ (i.e. there is some demand between i and j in scenario ω) are contained in $\Gamma(\omega) := \left\{(i, j) \in \mathcal{V} \times \mathcal{V} | \exists t \in \mathcal{T} : \gamma^t_{i,j} > 0\right\}$. We assume that there is no time needed to embark or disembark passengers at a location. Each vehicle has S seats available.

A Line: is a bi-directed path, in both SLPP and LPP. A bi-directed simple path $p \in \mathcal{P}$ is the combination of two directed paths. A path starts at a start node i_1 and ends at a end node i_n by passing through a sequence of unique arcs $a_1, \ldots, a_{n-1}$ where head node i_k of arc $a_k = (i_k, i_{k+1})$ is the tail node for previous arc $a_{k-1} = (i_{k-1}, i_k)$ for $k = 2, \ldots, n-1$. To complete the bi-directed path we combine the first path with a second path going in opposite direction from node i_n to i_1 by using the backward arcs from the first path. This requires the assumption that for each arc $a = (i, j)$ there is the corresponding backward arc $\bar{a} = (j, i)$. The nodes in the path are unique, i.e. there is no repetition of nodes. The path represents a Hamilton-path. In Figure 1 we use two lines (both bi-directed paths) in blue and orange to serve the demand. At least one AEV has to be assigned to a line serving all nodes in the line. We write $a \in p$ if arc a is contained in path p.

The Frequency: of the line depends on the length of the path and the number of AEVs assigned. Let $p \in \mathcal{P}$ be a path in the graph, d_p be the travel time along the path from the start node to the end node and back to the start (full round-trip) and $v \in \mathbb{N}$ the number of AEVs assigned. The frequency defines how often an AEV will visit[1] a node along p in one period. Frequency f_p of path p is then given by

$$f_p = v \cdot \frac{\Delta}{d_p}$$

For example, let us consider the network in Figure 1 with two lines represented by the bi-directed paths p_1 (blue line) and p_2 (orange line). If we assume a period length of $\Delta = 60$ minutes, the frequency of the blue line $p_1, v = 1$, is equal to $f_{p_1} = 1 \cdot \frac{60}{2 \cdot 34} = 0.88$, i.e. 0.88 AEVs visit a node in 60 minutes. For the orange line $p_2, v = 1$ we have the frequency $f_{p_2} = 1 \cdot \frac{60}{2 \cdot 30} = 1$. If the available seat capacity of one AEV is given by $S = 50$, then $50 \cdot 0.88 = 44.12$ and $50 \cdot 1 = 50$ passengers can be transported in 60 minutes between two locations in the line p_1 and p_2, respectively. We bound each arc $a \in \mathcal{A}$ by frequency capacity f_a.

Objective Function: For the whole planing horizon an AEV has constant provisioning costs C^{AEV}. The energy costs C^{en} for the AEV is linear in the time traveled on the path, meaning that per distance unit cost units have to be paid for energy. If demand is not met in a period, penalty costs C^{pen} for persons not transported incur. Continuing the example from Figure 1, $\Delta = 60 \rightsquigarrow \mathcal{T} = \{t\}$. Assuming energy cost $C^{\text{en}} = 1$ Euro per minute driving time, provisioning costs of $C^{\text{AEV}} = 50$ Euro and penalty costs of $C^{\text{pen}} = 10$ Euros per person not transported. If only one scenario $\omega \in \Omega$ is given, let us consider the SODM[2]

$$\Gamma(\omega, t) = \begin{array}{c} \\ 1 \\ 2 \\ 7 \\ 10 \end{array} \begin{array}{c} \begin{array}{cccccc} 2 & 3 & 4 & 5 & 8 & 9 \end{array} \\ \begin{pmatrix} 30 & & & & 40 & 60 \\ & 50 & & 55 & 60 & \\ & & 55 & & & \\ & & & & & 34 \end{pmatrix} \end{array}$$

With $v = 1, S = 50$ for each line, the amount of unmet demand for lines p_1 and p_2 is equal

1. Note that visits are directional, so an AEV will arrive at any node twice the frequency, once in each direction.

2. the numbers above the columns and right of the rows represent the nodes

to $(30-44.12)^{+} + (40-44.12)^{+} + (60-44.12)^{+} + (60-44.12)^{+} + (34-44.12)^{+} = 31.76$ and $(50-50)^{+} + (55-50)^{+} + (55-50)^{+} = 10$, where $(x)^{+}$ is the maximum function $\max\{0,x\}, x \in \mathbb{R}$. The total cost is given by $50+50+34+30+31.76 \cdot 10 + 10 \cdot 10 = 581.6$ Euros. The different costs values are shown in Table 1.

Line	C^{AEV}	C^{dist}	C^{pen}
p_1	50	34	317.6
p_2	50	30	100

TABLE 1
Cost values for the two lines shown in Figure 1 where one vehicle with a seat capacity of $S = 50$ is assigned to each line. As cost parameters we have $C^{\text{en}} = 1$, $C^{\text{AEV}} = 50$ and $C^{\text{pen}} = 10$.

2.3 Mathematical Model

The problem statement leads to a non-linear two-stage stochastic program with the following **decision variables:**

- $x_{i,j}^{p}$ $\rightarrow$ 1, if path p goes from location i to j, 0 otherwise;
- $y_{i,j}^{p}$ $\rightarrow$ Number of passengers transported on path p from location i to j;
- f_p $\rightarrow$ Frequency of path p;
- $u_{i,j,t}^{\omega}$ $\rightarrow$ Measuring of unmet demand from location i to j in scenario ω and period t;
- v_p $\rightarrow$ Number of vehicles assigned to route p;
- d_p^{path} $\rightarrow$ Total travel time of path p.

The model **target function** is to minimize the sum of all accumulated costs:

$$\sum_{p\in\mathcal{P}} \left(v_p \cdot \left(\cdot C^{\text{AEV}} + C^{\text{dist}} \cdot d_p^{\text{path}}\right)\right) + \sum_{\omega\in\Omega}\sum_{t\in\mathcal{T}}\sum_{(i,j)\in\Gamma(\omega)} C^{\text{pen}} \cdot p_{\omega}^{t} \cdot u_{i,j,t}^{\omega} \tag{1}$$

subject to the constraints

$$\sum_{p\in\mathcal{P}} y_{i,j}^{p} + u_{i,j,t}^{\omega} \geq \gamma_{i,j}^{t,\omega} \ \forall\, (i,j) \in \Gamma, t \in \mathcal{T}, \omega \in \Omega \tag{2}$$

$$\sum_{(j,i)\in\mathcal{A}:} x_{j,i}^{p} - \sum_{(i,j)\in\mathcal{A}} x_{i,j}^{p} = 0 \ \forall\, s,e,i \in \mathcal{V} : s \neq e \neq i, p \in \mathcal{P} \tag{3}$$

$$\sum_{(i,k)\in\mathcal{E}} x_{i,k}^{p} \geq 0 \tag{4}$$

$$\sum_{(i,j)\in\mathcal{A}} D_{i,j} \cdot x_{i,j}^{p} \geq d_p^{\text{path}} \ \forall\, p \in \mathcal{P} \tag{5}$$

$$\sum_{(i,j)\in\mathcal{A}} D_{i,j} \cdot x_{i,j}^{p} \leq d_p^{\text{path}} \ \forall\, p \in \mathcal{P} \tag{6}$$

$$\sum_{(i,k)\in\mathcal{A}} S \cdot F_{i,k} \cdot x_{i,k}^{p} \geq y_{i,j}^{p} \ \forall\, (i,j) \in \Gamma \tag{7}$$

$$\sum_{(k,j)\in\mathcal{A}} S \cdot F_{k,j} \cdot x_{i,j}^{p} \geq y_{i,j}^{p} \ \forall\, (i,j) \in \Gamma \tag{8}$$

$$y_{i,j}^{p} \leq f_p \cdot S \ \forall\, (i,j) \in \mathcal{A}, p \in \mathcal{P} \tag{9}$$

$$\sum_{p\in\mathcal{P}:a=(i,j)\in p} f_p \leq F_{i,j} \ \forall\, (i,j) \in \mathcal{A} \tag{10}$$

$$2 \cdot f_p \cdot d_p^{\text{path}} \leq v_p \cdot \Delta \ \forall\, p \in \mathcal{P} \tag{11}$$

$$d_p^{\text{path}}, f_p, v_p \geq 0 \ \forall p \in \mathcal{P} \tag{12}$$

$$u_{i,j}^{t} \geq 0 \ \forall\, (i,j) \in \mathcal{A}, t \in \mathcal{T} \tag{13}$$

$$y_{i,j}^{p} \geq 0 \ \forall\, i,j \in \mathcal{V}, p \in \mathcal{P} \tag{14}$$

$$x_{i,j}^{p} \in \{0,1\} \ \forall\, (i,j) \in \mathcal{A}, p \in \mathcal{P} \tag{15}$$

Objective function (1) consists of two parts. The first part minimizes the costs for the used AEVs and the running costs of the lines, while the second part minimizes the expected penalty costs for unmet demand.

Constraint (2) specifies the partition of the total demand into met and unmet demand in scenario ω. The frequency of path p multiplied with the seat capacities of the AEVs give the upper bound for the maximal number of passengers who can be transported from i to j on path p at a time, see constrains (9). The frequency of each arc contained in p is bounded from above in constraint (10). Given frequency f_p of path p and the total length d_p^{path}, constraint (11) determines the corresponding number of required AEVs v_p (see Section 2.2).

The paths to satisfy the demand are generated in constraint (3). The resulting travel time of the paths is calculated in constraint (6). Constraints (7) and (8) ensure that passengers can only be transported from i to j on path p if both locations are visited. The domain of the used variables are given in (12) to (15).

3 GAME PLAY

The teams of size between two or four people take the role of a manager for a fleet of the AEVs. At the beginning of the competition they are given the problem description, the data and documentation.

The Problem Description: is derived from a real world city and the verbal introduction as a real-world problem, alongside a visualization of the city graph and the solution to be generated. The description is not too verbose (5-10 minutes talk) and not too detailed (as in section 2) to not spent too much time before the teams get going on their own.

The data: consists a number of JSON [5] files containing the

- representation of the city nodes (called blocks)
- connections and distances between the blocks
- structural information on the city, influencing the demand.
- the past demand data to derive the future demand from

Documentation: has the goal of providing starting points for the teams to familiarize with the problem. As teams are more than one person, it makes sense to provide more than one angle on the problem: First a class diagram to illustrate how the provided data types are interconnected and which properties the classes possess. Second a python [6] interface to those data-types, alongside with methods to read the JSON files and produce a representation of a produced solution to be handed in for evaluation and finally an online browsable interface documentation of the provided code.

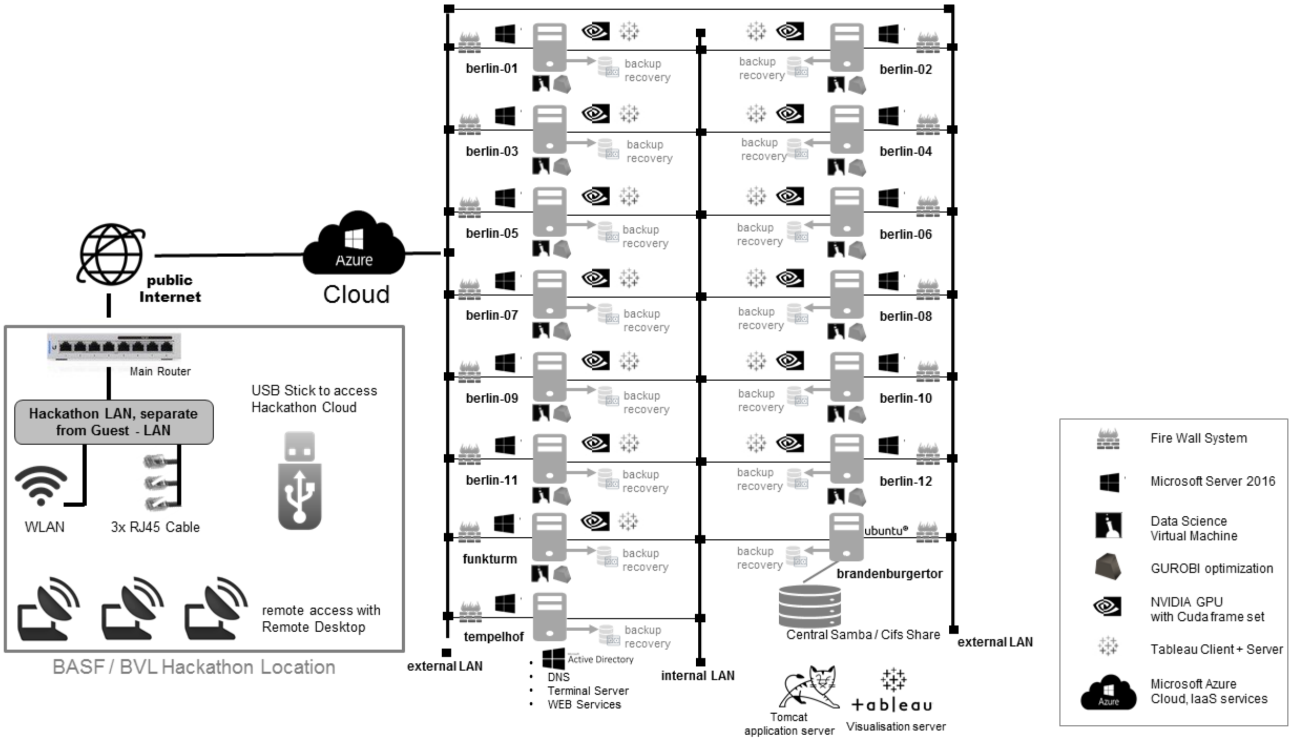

Fig. 2. The competition infrastructure of the Hackathon event.

3.1 Infrastructure

Each team member had to use his/her hardware to connect with a remote desktop server. The remote connection grants access to a virtual machine provided in the cloud with pre-installed software packages and sufficient computation power. The venue provided a dedicated WLAN to ensure sufficient bandwidth. This WLAN was connected to the public internet, so the participants could install software as required. The complete infrastructure is shown in figure 2.

4 The Demand Calculation

To anticipate future demand the major input for the participants has been the past demand. All demand (whether for the past or the future) was generated using the same function. The function depends on the attributes of the city and the weather. All of the input parameters were given to the teams but not the function itself. In detail:

- structural attributes of the city blocks: population, GDP, land value, age, latitude, longitude
- the past weather data for every town for every hour of every day
- the future weather data for every hour of the day the competition is for (like a 100% accurate forecast)
- the past demand for every hour of every day

So the only data the teams did not posses during the competition is the resulting future demand.

4.1 The Demand Function

Demand is calculated in block pairs (origin, destination). As input there are the time static parameters of the blocks, the hours of the day and the weather. Participants are given all that information, but not the logic how to combine it. They are provided, each for origin and destination: population, position on the map, weather in hourly ticks, land value, GDP, and age. We assume part of the population at any block leaves in the first 12 hours and returns in the second 12 hours.

The percentage of demand from origin to destination is proportional to the total population of origin since more people are available to leave in the morning. Richer blocks (higher GDP and land value) have more jobs and attract more workers. Age acts as a separator allocating inhabitants into buckets. The youngest 20% all leave for education. 60% of the working-age people leave their homes and 10% of the retired people. We assume that people will cover shorter distances without transportation. So the demand is proportional to the distance between the blocks.

Besides the static properties of the city weather is a function over time and influences the demand. Weather is represented by two binary functions: sun and rain that interact and change over time. Other aspects of weather are not considered. Each tick of weather (i.e. the conditions in one hour during the day) creates a demand impact based on the previous weather. sun s and rain $r \in 0, 1$ define 4 weather modes:

- $s = 1, r = 0$ sunny
- $s = 1, r = 1$ rainbow
- $s = 0, r = 0$ fair
- $s = 0, r = 1$ rain

Constant weather does not invoke a change of the demand as the situation does not change. If weather changes, however, it has an impact on demand.

- sunny to rain has a big positive impact as a lot of people that planned on walking/biking wanna get transportation
- fair to rain has a moderate impact with the same reason
- rainbow to rain has a slightly negative impact as the few people who wanted to skip transportation to see the rainbow now don't
- sunny to fair has no impact as nobody changes their play without rain
- rain to fair has a slightly negative impact as a few people are now motivated to walk/bike
- rainbow to fair does not have impact
- rain to sunny has a massive negative impact as many people will now switch to walk/bike
- fair to sunny has a slight negative effect
- rainbow to sunny has a negative effect as it is no longer raining but the sun still shines
- sunny to rainbow is constant as the same amount of persons who are drawn be the rainbow are repelled by the rain
- fair to rainbow is also constant for the same reason
- rain to rainbow has a slight negative impact as a few people will want to see the rainbow

The process steps for the demand calculation are as follows.

1) select demand pairs (origin, destination)
2) calculate the base demand value depending on the static parameters of the blocks
3) distribute the outgoing demand of origin to the 12 time buckets in the morning according to the predefined hourly weights
4) generate the returning 12 time buckets at the destination. Again there are predefined weights per hour and weather impacts. The return bucket is normalized to contain the same amount of demand as the outgoing bucket
5) minimal noise is introduced in all demand points
6) the sum of the buckets is normalized to the population of the origin block. No more than the population of the block can leave in the morning. For all of the demand pairs the maximum factor of over-demand is calculated and applied to all origin demands as to preserve the relation between the block demands and this maintain the relation to the input data

The above process is continued over several days. This demand is given to the participants where the most recent day is kept hidden as it is used for the evaluation of the produced solutions.

5 Scoring & Competition

The score of a solution is equivalent to the accumulated cost. As the cost are equivalent to the target function value as introduced in 2.3 the scoring of the competition is equal to the performance of the solutions in respect to the cost minimization.

The competition is hosted during one day. The problem case is not known to the participants beforehand and will be opened/presented at the start of the competition. After the closing of the competition, it is a good practice to reserve about one hour time to talk about the experience of the teams and the different solution approaches in detail. This leaves about 7 hours for the competition. Given the technical challenges of understanding the data and representing the model in a suitable modeling/programming language, solving the problem is a major achievement. Therefore, there is no need for multiple rounds with different problem settings. The space between the first solution and the theoretical optimum is sufficient to be a challenge for the teams.

6 Real World related Problems

The problem presented in this competition is a classical network design problem. The green field approach of designing a new network from scratch is almost never found in realistic settings. The chosen simplifications (non interference with other traffic, no switching of lines, neglect of boarding times, and route travel times) are suitable for a competition but significantly limit the real-world applicability[3]).

The design of the German rail network is a close example. Neither the rails nor the stations are easy to move or have a lot of possible options. The ability of the connections and stations to satisfy demand is the focus of the ongoing monitoring of the network. As network adjustments are constant, a variation of this problem is also relevant.

Airplane routes are another close example. Airports are even harder to move/build than train stations, but the routes between nodes are easier to adjust [7].

References

[1] BVL-Hackathon. (2018, Oct.) Hackathon. BVL. [Online]. Available: http://www.bvl.de/hackathon

[2] R. Borndörfer, M. Grötschel, and M. Pfetsch, "A column generation approach to line planning in public transport," *INFORMS Transportation Science*, vol. 41, no. 1, pp. 123 – 133, 2007.

[3] A. Schöbel, "Line planning in public transportation: models and methods," *OR Spectrum*, vol. 34, pp. 491–510, 2012.

[4] A. Schöbel and S. Scholl, "Line planning with minimal transfers," in *5th workshop on algorithmic methods and models for optimization of railways*, ser. Dagstuhl seminar proceedings, vol. 09601, 2006.

[5] "The JSON Data Interchange Format," ECMA, Tech. Rep. Standard ECMA-404 1st Edition / October 2013, Oct. 2013. [Online]. Available: http://www.ecma-international.org/publications/files/ECMA-ST/ECMA-404.pdf

[6] G. Rossum, "Python reference manual," Amsterdam, The Netherlands, The Netherlands, Tech. Rep., 1995.

[7] G. Desaulniers, J. Desrosiers, Y. Dumas, S. Marc, B. Rioux, M. Solomon, and F. Soumis, "Crew pairing at air france," *European Journal of Operational Research*, vol. 97, no. 2, pp. 245 – 259, 1997. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0377221796001956

3. The most realistic abstraction would be an underground network of really fast subway trains. The setting of AEVs was chosen as the reordering of routes is much more realistic than the reordering of underground tubes.