GEORG-AUGUST-UNIVERSITÄT
GÖTTINGEN

FAKULTÄT FÜR MATHEMATIK UND INFORMATIK
INSTITUT FÜR NUMERISCHE UND ANGEWANDTE MATHEMATIK

Bachelor Thesis

A Passenger-Orientated Approach to Analysing and Optimising Disorder Situations in Public Tranportation Networks

Submitted by / on:	Morten Tiedemann / 2 June 2010
Matriculation Number:	20723108
Adress:	Kreuzbergring 56c 37075 Göttingen
Email:	morten.tiedemann@stud.uni-goettingen.de

Primary Reviewer:	Prof. Anita Schöbel
Secondary Reviewer:	Prof. Jens Clausen

Contents

1 Introduction

The occurrence of disorder situations in public transportation networks is usually unavoidable, since unpredictable disturbances can make any sophisticated timetable unachievable. What is important is how to deal with a given disorder situation, because every delay of a vehicle in a public transportation network can have direct or indirect consequences for other vehicles. In this thesis a passenger-orientated approach to analysing and optimising disorder situations in public transportation networks is chosen.

For a better understanding of disorder situations an application visualising detailed information about disorder situations of past days of operation was developed in cooperation with the S-train department of Danish State Railways (DSB S-tog). Furthermore, the effects of skipping a station by a train were examined in order to provide an optimised decision support for the purpose of minimising the total delay of all passengers in the network. Since this thesis considers specifically Copenhagen's S-train network, first some facts about DSB S-tog and its network will be given in this section. Moreover, the implementation of algorithms and methods developed in this thesis will be addressed.

Danske Statsbaner (DSB) is the largest Danish train operating company and is responsible for the passenger train operation on most of the Danish railways. Moreover, DSB runs a suburban rail network, called S-train, in the area around Copenhagen that connects the city centre with the inner suburbs of Copenhagen. Each day more than 300.000 passengers are transported in the S-train network, from given datasets passenger counts add up to 329908 (11^{th} of January, 2008) and 332596 (5^{th} of January, 2009). The S-train network consists currently of 85 stations and 7 lines, and has a system length of 155km. In Figure 1 on page 3 a schematic line map of the S-train network as released for passengers is depicted. The data about DSB and Copenhagen's S-train network presented above are directly provided by DSB. Furthermore, the network can be subdivided in a central area framed by the yellow line F and five so-called fingers pointing to the inner suburbs of Copenhagen. This clear-cut design allows for a fragmentation of the network that will be helpful for later considerations. Additionally, in Figure 2 on page 4 a schematical infrastructure map of Copenhagen's S-train network is depicted.

The transformation of the origin-destination matrix (OD-matrix) described in Section 3, the analysis of the passenger delay described in Section 4 and especially

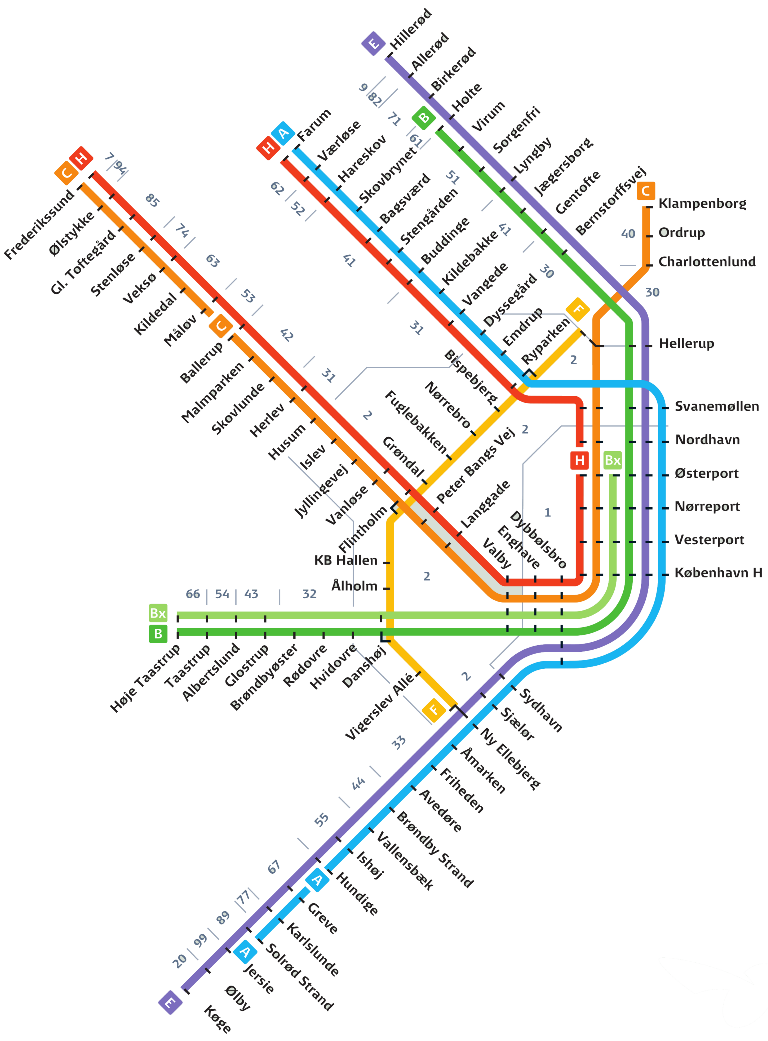


Figure 1: Schematic Line Map, [DSB10]

the replay, which visualises analysed (passenger) delay information, described in Section 5 have been completely implemented and are fully operational. Both the algorithmic and the graphical part have been implemented with Java Version 1.6.0. The optimised skipping of stations presented in Section 6 has not yet been implemented.

2 Model

DSB S-tog provided a lot of data for the analysis and optimisation of disorder situations. In order to develop immediately applicable results the model presented

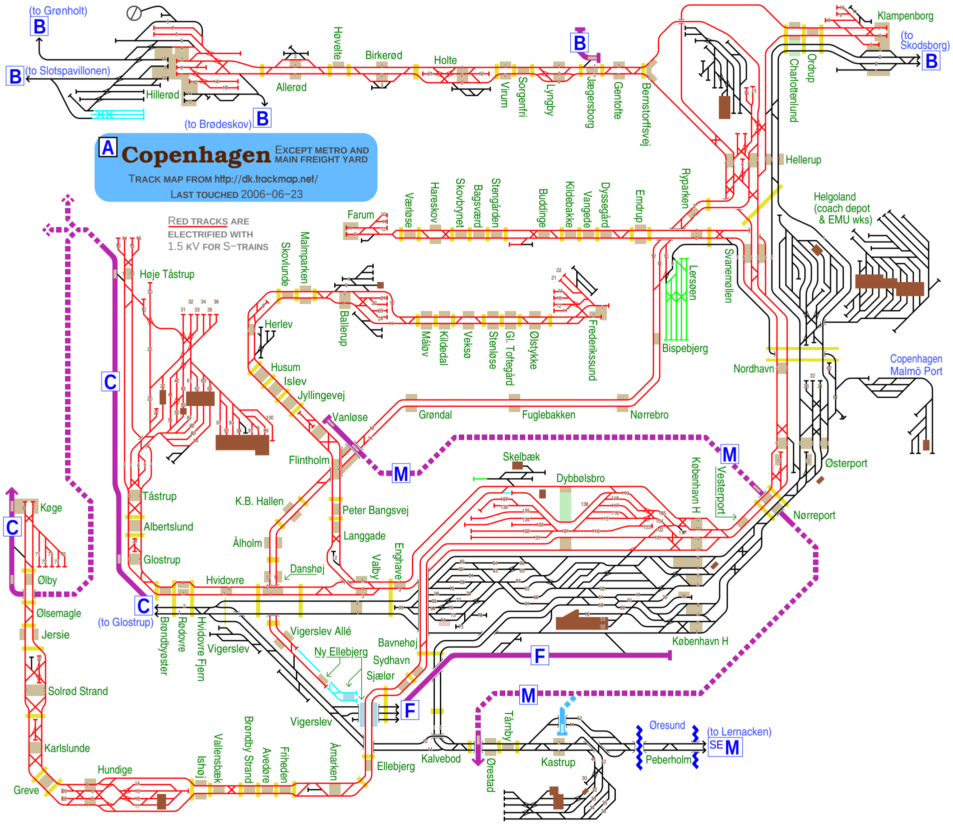


Figure 2: Schematic Infrastructure Map, [dk.06]

in this section is based upon the available data and no assumptions of additional data that is not on hand were made. This practical approach has the advantage that all results can be tested instantly with real data sets and permanent feedback can be received from the employees at DSB S-tog. Therefore the methods could be adjusted to the demand of the user step by step. For a better understanding of the given conditions first the different datasets provided by DSB S-tog will be described. Afterwards the model for the public transportation network will be introduced formally.

2.1 Datasets

2.1.1 Train Data

At DSB S-tog each ride of a train from one station to another station at a certain time without stop at any station in between is recorded. A formal definition of a ride is given in Section 2.2. All the rides of one day are summarised in a dataset, in the following referred to as $\mathcal{D}_{train}$. So every single line of $\mathcal{D}_{train}$ contains information about one ride. Below the fields of dataset $\mathcal{D}_{train}$ are described in detail:

Train number Every time a train is employed in the network a train number is assigned to this train. Consequently the same physical train can have different train numbers during a day.

End station This field states the end station of the ride. Since the data is primarily ordered by the train number and secondarily ordered by the scheduled arrival time, it is not necessary to state the start station explicitly, but it can be generated automatically.

Scheduled arrival time at end station This is the arrival time of the train at the end station of the ride as listed on the customer's timetable. All times in all datasets are given in seconds after midnight.

Actual arrival time at end station This is the arrival time of the train at the end station of the ride as it actually happened. With the help of the scheduled and the actual arrival time the delay of the train at the end station can easily be calculated.

Capacity of the train According to the number and type of cars trains have different capacities. This field states the capacity of the corresponding train.

Train type There is train type "P" for regular passenger trains and train type "M" for maintenance trains. Only the passenger trains are considered for further analysis.

Line This field indicates on which line this train runs.

Direction Every ride has a direction, either from south to north (N) or from north to south (S), which is stated in this field.

Number of passengers in the train The number of passengers in the train is calculated from the results of weighing machines in the track line that weigh the trains every time before they arrive at a station and after they have left a station. Since these calculations are approximations the number of passengers in the train as well as the number of entering and leaving passengers are always decimal numbers.

Number of entering passengers at end station The number of passengers entering the train at the end station either to start or to continue their journey.

Number of leaving passengers at end station The number of passengers leaving the train at the end station either for changing trains or because they have reached their final destination.

Delayed This field is set to zero, if a train is on time and set to one if a train is delayed. A train is marked as delayed if its delay is higher than 150 seconds.

Cancelled This field is set to one, if a train is cancelled. Otherwise, this field is set to zero.

Delay caused by Every time a train is delayed the train that is responsible for this delay is recorded. If there is no other train involved in the delay, the delayed train itself is recorded.

Delay cause In this field an error code of the delay can be found. With the help of the description of the error code more detailed information for the reason of the delay can be gained.

Thus, dataset $\mathcal{D}_{train}$ allocates not only very detailed information about all rides of that day, but also information about all (used) stations of the network. Furthermore, DSB S-tog provides sojourn times for all stations and, therefore, it is possible to generate both a scheduled and an actual timetable with arrival and departure times. A formal introduction of a scheduled and an actual timetable will be given in Section 2.2.

Dataset $\mathcal{D}_{train}$ is recorded by DSB S-tog for every single day. Consequently the analysis presented in this thesis can be applied to any user-defined day. Moreover, dataset $\mathcal{D}_{train}$ as well as the OD-matrix described in Section 2.1.2 are available as csv-files, which assures an uncomplicated further processing.

2.1.2 OD-Matrix

Especially for a passenger-orientated approach to analysing and optimising disorder situations an OD-matrix is essential. Fortunately, DSB S-tog could make an OD-matrix from passenger surveys on a specific workingday in March 2009 available. From now on this dataset will be referred to as $\mathcal{D}_{od}$. Dataset $\mathcal{D}_{od}$ compasses 71993 lines, containing the fields listed below:

Type of the day This field contains information about the type of the day on which the data for this dataset was collected, either workingday or weekend. The provided dataset is based on countings on a workingday.

Month This field contains information about the month in which the data for this dataset was collected. The provided dataset is based on countings in March.

Origin The number of the origin station.

Destination The number of the destination station.

Time period A day is subdivided into 42 time periods, each of them with a duration of 20 or 60 minutes, depending on the time of the day: From 06.00-10.00 and from 14.00-19.00 the periods have a duration of 20 minutes, the rest of the day the periods have a duration of 60 minutes. This field indicates the time period containing the starting time of the journey from origin to destination. Note that the actual starting time is not given.

Passenger count The number of passengers that travelled from the origin station to the destination station in the given time period.

In which way the information of the OD-matrix were handled is explicitly described in Section 3.

2.1.3 Map Data

In order to visualise the results of our analysis, information about the position of stations and track lines is essential. Due to the lack of adequate data, datasets for a schematic infrastructure map and a schematic line map were generated based on information provided by DSB S-tog. From now on these two datasets will be referred as $\mathcal{D}_{imap}$ (schematic infrastructure map) and $\mathcal{D}_{lmap}$ (schematic line map).

2.2 A Model for the Analysis of Disorder Situations

In this section the notation necessary for the analysis and optimisation of disorder situations is specified. First of all, a formal definition of a public transportation network analogue to [Sch06, p. 5] is given.

Definition 1. A *public transportation network* is a finite, undirected graph
PTN = (S, E) with

- a node set S representing stations, and
- an edge set E, where each edge (s_1, s_2) indicates that there exists a direct ride from station $s_1 \in S$ to station $s_2 \in S$ (i.e., a ride that does not pass any other station in between).

The set of stations S and the set of direct rides E can be extracted from $\mathcal{D}_{train}$. Furthermore, let L be the set of lines in Copenhagen's S-train network. The set of directed lines $\mathcal{L}$ is then given by

$$\mathcal{L} = \{l \times \{\mathrm{N}, \mathrm{S}\} \mid l \in L\}$$

where $\{\mathrm{N}, \mathrm{S}\}$ is the set of directions; N $\triangleq$ from south to north and S $\triangleq$ from north to south. For each directed line $l \in \mathcal{L}$ we introduce

- l_d as the direction of the line, and
- $l_f^s(b)$ as the frequency at which station s is served by line l in time period $b \in \mathcal{B}$.

$\mathcal{B}$ is the set of time periods mentioned in 2.1.2. Moreover, for each station s a set of directed lines $\mathcal{L}^s \subseteq \mathcal{L}$ that serve station s is introduced.

The public transportation network PTN $= (S, E)$ embodies the physical stations and edges. In order to deal with passenger flows in the network, the PTN is in the first place extended to a so-called change & go graph, analogue to [Sch09, p. 5f].

Definition 2. Given a public transportation network PTN $= (S, E)$ and a set of directed lines $\mathcal{L}$, its corresponding *change & go graph* consists of

- a set of nodes $V = \{s \times \mathcal{L}^s \mid s \in S\}$, i.e. every node $v = (s, l) \in V$ represents a station in combination with a directed line, and
- a set of edges $\mathcal{E} = \{\{(s_1, l_1), (s_2, l_2)\} \mid s_1 = s_2 \vee (l_1 = l_2 \wedge (s_1, s_2) \in E)\}$, i.e. two nodes of the change & go graph are linked if they belong to the same station, hence a transfer between the corresponding lines is possible, or if they are consecutive stations within the same line.

Additionally, the corresponding station and the corresponding directed line of a node $v = (s, l)$ are denoted by $v_{station}$ and v_{line} respectively. Furthermore, the set of trains T and for each train $t \in T$ a set $V^t \subseteq V$ as the set of nodes, where train t

stops, is introduced. Every line represents a set of repetitive trains, more precisely $V^t = V^{t'}$ for all $t, t' \in T$ on the same line.

As mentioned above the actual and the scheduled timetable can be extracted from $\mathcal{D}_{train}$. Now this is stated formally by means of two definitions similar to [Sch06, p. 100]:

Definition 3. A *scheduled timetable* Π is given by natural numbers $\Pi^v_{arr_t}$, $\Pi^v_{dep_t}$ for all $t \in T$, $v \in V^t$ with the following meaning:

- $\Pi^v_{arr_t}$ is the scheduled arrival time of train t at node v, and
- $\Pi^v_{dep_t}$ is the scheduled departure time of train t at node v.

Definition 4. An *actual timetable* Γ is given by natural numbers $\Gamma^v_{arr_t}$, $\Gamma^v_{dep_t}$ for all $t \in T$, $v \in V^t$ with the following meaning:

- $\Gamma^v_{arr_t}$ is the actual arrival time of train t at node v, and
- $\Gamma^v_{dep_t}$ is the actual departure time of train t at node v.

The scheduled and actual arrival times are directly given by $\mathcal{D}_{train}$ and the departure times can be calculated with the help of the given sojourn times. Let $S^v \in \mathbb{N}$ be the given sojourn time for every train at node $v \in V$. Here, the assumption is made that these sojourn times have to be maintained even in case of a delay. Thus, the departure times are given by

$$\Pi^v_{dep_t} = \Pi^v_{arr_t} + S^v$$
$$\Gamma^v_{dep_t} = \Gamma^v_{arr_t} + S^v$$

Now we have the necessary notations to define the term *ride* formally.

Definition 5. A *ride* $r = (v_1, v_2, t)$ encapsulates the possibility to travel from node $v_1 \in V$ to node $v_2 \in V$ with train $t \in T$, where (v_1, v_2) is an edge of the change & go graph. Consequently, the scheduled departure time at node v_1 and the scheduled arrival time at node v_2 are given by $\Pi^{v_1}_{dep_t}$ and $\Pi^{v_2}_{arr_t}$ respectively. Further information about ride r are given by

- ω_r as the number of passengers in train t while train t drives from node v_1 to node v_2,
- r_{start} as the start node v_1 of ride r and r_{end} as the end node v_2 of ride r,

- r_{enter} as the number of passengers entering at the end node v_2 and r_{leave} as the number of passengers leaving the train at the end node v_2,
- Π^r_{dep} and Γ^r_{dep} as the scheduled and actual departure time at the start node of r, and
- Π^r_{arr} and Γ^r_{arr} as the scheduled and actual arrival time at the end node of r. This notation of scheduled and actual arrival and departure times is equivalent to the timetable notation given above with the corresponding trains and nodes:

$$\begin{aligned} \Pi^r_{dep} &= \Pi^{v_2}_{dep_t} \\ \Pi^r_{arr} &= \Pi^{v_2}_{arr_t} \\ \Gamma^r_{dep} &= \Gamma^{v_2}_{dep_t} \\ \Gamma^r_{arr} &= \Gamma^{v_2}_{arr_t} \end{aligned}$$

The set of all rides can be extracted directly from $\mathcal{D}_{train}$ and will be denoted by R. Every edge of the change & go graph will now be extended by the corresponding rides:

- For every edge $e = \{v_1, v_2\} \in \mathcal{E}$ a set of rides

  $$R^e = \{r = (u, v, t) \in R \mid u = v_1 \wedge v = v_2\} \subseteq R$$

  is introduced as the possibilities of a passenger to travel from node v_1 to node v_2.

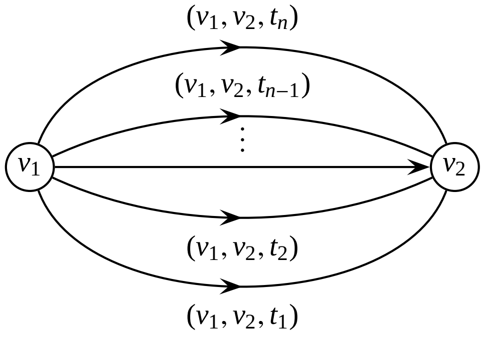


Figure 3: Extension of the Change & Go Graph

- For every train $t \in T$ we introduce $R^t = \{r_1, r_2, \ldots, r_{|R^t|}\} \subseteq R$ with $r_i = (u_i, v_i, t)$ as the set of rides used by train t ordered by departure time $\Pi^t_{dep_{u_i}}$ of train t at node u_i for $i = 1, \ldots, |R^t|$.

- For every node $v \in V$ we introduce $R^v = \{r = (v_1, v_2, t) \in R \mid v = v_1\}$ as the set of rides that depart from node v.

In Figure 4 on page 11 a small part of the Copenhagen S-train network transformed into a PTN is depicted. The change & go graph of the same section of the network is depicted in Figure 5 on page 12. For clarity reasons not all lines connecting the stations in the real network are shown in the graph.

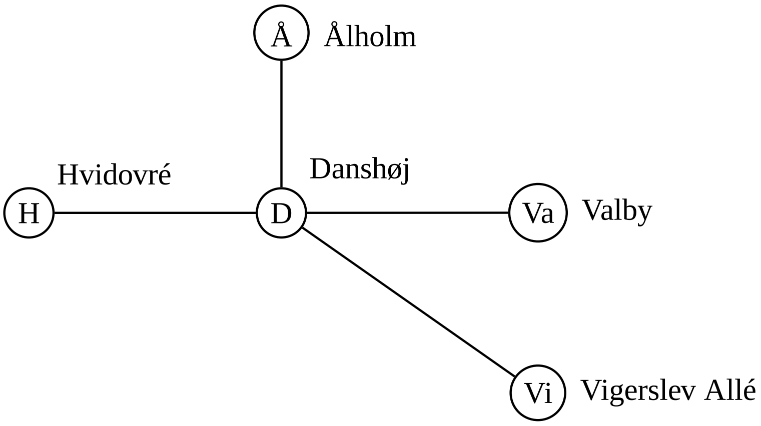


Figure 4: PTN

Furthermore, the necessary notation for dealing with the OD-matrix will be introduced. Each entry of $\mathcal{D}_{od}$ gives information about the number of passengers that want to travel from a given origin station to a given destination station in a certain time period. Analogue to [DHSS10] such an origin-destination pair (OD-pair) is denoted by

$$p = \{o, d, b\},$$

where $o \in S$ is the origin station, $d \in S$ is the destination station and $b \in \mathcal{B}$ is the time period that contains the starting time of the journey. P denotes the set of all OD-pairs from $\mathcal{D}_{od}$ and the number of passengers associated with an OD-Pair $p \in P$ is identified with ω_p. The OD-matrix is therefore based on the PTN introduced in this section. In order to analyse passenger delays a more detailed version of the OD-matrix, based on the extended change & go graph, is necessary.

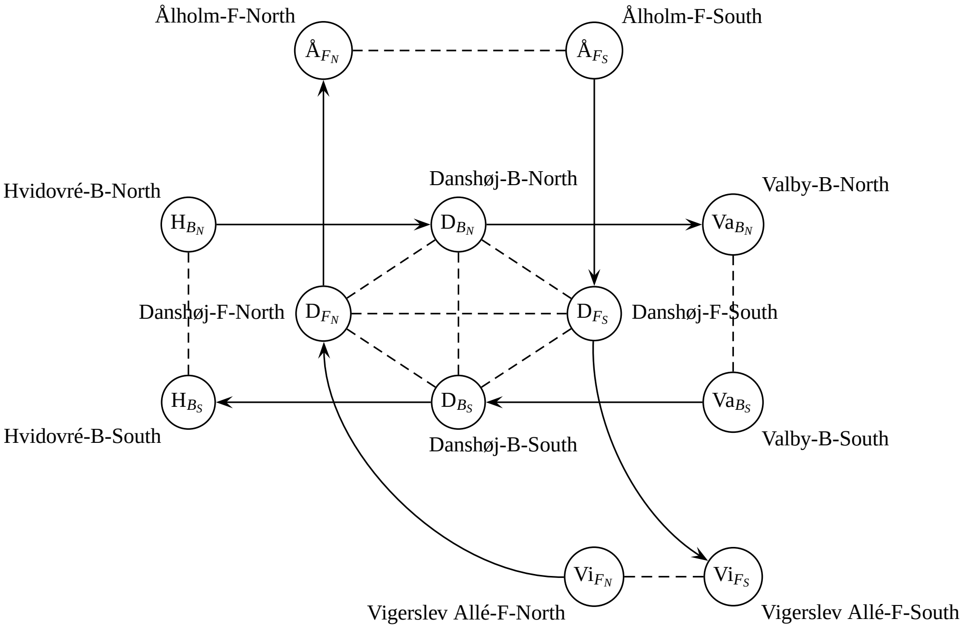


Figure 5: Change & Go Graph

3 Transformation of the OD-Matrix

Dataset $\mathcal{D}_{od}$ gives only information about how many passengers travel from a given origin to a given destination in a certain time period. We do not know which trains the passengers use to travel from the origin to the destination and thus we do not know either the paths of the passengers through the network. For example, we have the information that 80 passengers want to travel from Vesterport to Ishøj. They start their journey in the time period from 08.00 till 08.20. Vesterport and Ishøj are connected by line E and line A, in rush hours there are lines E+ and A+ as well. It is almost certain that more than one train of these two or even four lines will transport passengers from Vesterport to Ishøj in the time period from 08.00 till 08.20. In case of one-hour time periods or OD-pairs with transfers, the situation gets even more complicated. However, for a passenger-orientated approach to analysing and optimisation disorder situations it is very important to know the paths of the passengers through the network. With the help of the given dataset $\mathcal{D}_{od}$ and the knowledge of passenger counts of specific days — the number of passengers in the train and the number of passengers entering and leaving the trains — a way to transform $\mathcal{D}_{od}$ into a more detailed dataset $\mathcal{D}_{odt}$ will be developed in this section. Each entry of $\mathcal{D}_{odt}$ will provide information about the number of passengers that want to travel from a given origin node to a given destination station at a certain time. Additionally, the path from the origin node to the destination station will be given in $\mathcal{D}_{odt}$. Such an OD-pair is denoted by

$$\phi = \{o, d, s_{od}, \Lambda_{od}\},$$

where $o \in V$ is the origin node, $d \in S$ is the destination station, s_{od} is the starting time for the trip and Λ_{od} is a reasonable path in the change & go graph from origin node o to destination node (d, l) for some $l \in \mathcal{L}^d$. The calculated path is not necessarily the shortest path from o to d, hence, it is labelled as reasonable path, which will be explained more detailed in the following sections. The set of all OD-pairs from $\mathcal{D}_{od}$ is denoted by Φ and the number of passengers associated with an OD-Pair $\phi \in \Phi$ is identified as ω_ϕ. The algorithm for the transformation from $\mathcal{D}_{od}$ to $\mathcal{D}_{odt}$ is described below.

First of all the transformation of all OD-pairs can be accelerated if OD-pairs with direct connection and OD-pairs without direct connection are treated separately.

Definition 6. Two stations s_1,$s_2 \in S$ have a *direct connection* in time period $b \in \mathcal{B}$ if a passenger can travel from station s_1 to station s_2 in time period b without transfers. More formally, two stations s_1, s_2 have a *direct connection* in time period b if

$$\left\{l \in \mathcal{L}^{s_1} \cap \mathcal{L}^{s_2} \mid l_f^{s_1}(b) \neq 0\right\} \neq \emptyset.$$

Consequently, an OD-pair $p = \{o, d, b\}$ is identified as an OD-pair with direct connection if station o and station d have a direct connection in time period b. The assumption is that a passenger would never take a more complicated path from one station to another station than the direct connection between these two stations, even if another path would be shorter. The risk of missing connecting trains and the additional effort to find a shorter path at least level the saved amount of time. This argumentation may not be adequate for every public transportation network, but the design of Copenhagen's S-train network described in Section 1 justifies the underlying assumption.

3.1 OD-pairs with Direct Connection

Let $p = \{o, d, b\} \in P$ with $o \in S$, $d \in S$ and $b \in \mathcal{B}$ an OD-pair with direct connection, more precisely

$$\left\{l \in \mathcal{L}^o \cap \mathcal{L}^d \mid l^o_f(b) \neq 0\right\} \neq \emptyset.$$

For every directed line $l \in \left\{l \in \mathcal{L}^o \cap \mathcal{L}^d \mid l^o_f(b)\right\}$ the corresponding node $n = (o, l) \in V$ is considered, i.e. all departure nodes that could be used by a passenger who starts his journey at station o in time period b are considered. For each corresponding node $n = (o, l)$ each ride $r = (n, v, t)$ in its set of departing rides R^n is considered. If the scheduled departure time $\Pi^t_{dep_n}$ of ride r lies in time period b, a possible starting ride for the journey from station o to station d is found. The set of rides R^t of the corresponding train t provides the necessary rides to complete the journey to the destination station d. This procedure is formally summarised in Algorithm 1 on page 14.

Algorithm 1 Transformation of OD-pairs with Direct Connection

```
Require: OD-pair $p = \{o, d, b\} \in P$ with direct connection in time period $b$
for all directed lines $l \in \left\{l \in \mathcal{L}^o \cap \mathcal{L}^d \mid l^o_f(b)\right\}$ do
    node $n = (o, l) \in V$
    for all rides $r = (n, v, t) \in R^n$ with $\Pi^t_{dep_n} \in b$ do
        $i =$ index of $r$ in $R^t = \{r_1, r_2, \ldots, r_{|R^t|})\}$
        while $r_{i_{station}} \neq d$ do
            add $r_i$ to $\Lambda_{od}$
            $i = i + 1$
        end while
        add $\phi = \{n, d, \Pi^t_{dep_n}, \Lambda_{nd}\}$ to $\Phi$
    end for
end for
```

3.2 OD-pairs without Direct Connection

Now let $p = \{o, d, b\} \in P$ with $o \in S$, $d \in S$ and $b \in \mathcal{B}$ an OD-pair without direct connection, more precisely

$$\left\{l \in \mathcal{L}^o \cap \mathcal{L}^d \mid l^o_f(b) \neq 0\right\} = \emptyset.$$

In this case the passenger has to change at least once during his journey from station o to station d and the method to transform the OD-pair used in algorithm 1 is not

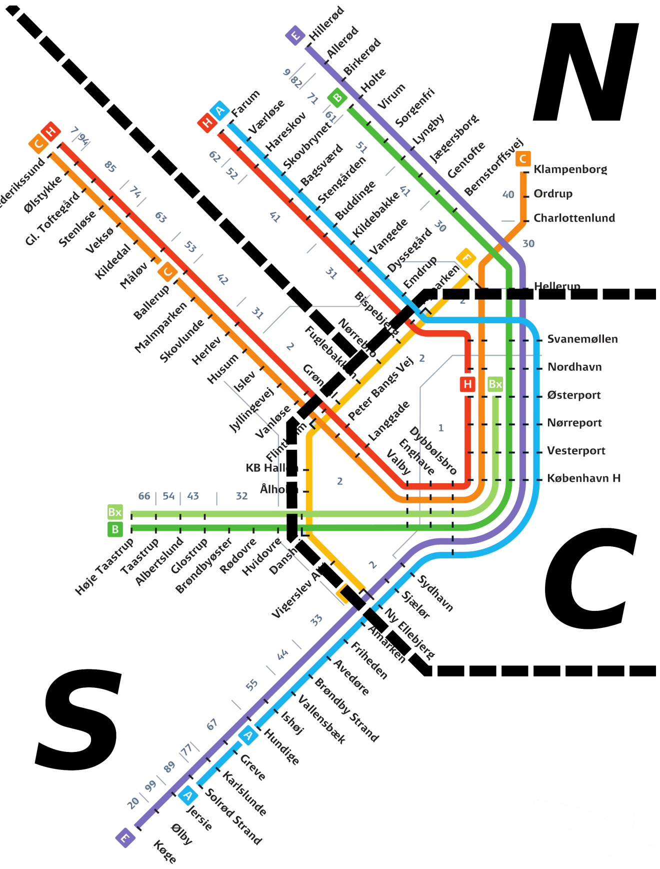


Figure 6: Fragmentation of Copenhagen's S-train Network

applicable any more. Consequently, all directed lines $l \in \mathcal{L}^o$ have to be considered for the determination of possible starting nodes. In order to rule out the generation of new OD-pairs with unreasonable starting nodes, some conditions that have to be met by l are introduced.

The stations in Copenhagen's S-train network can be subdivided in three parts: Stations that are located in the north, stations that are located in the south and stations that are located in the centre. Thus for each station $s \in S$ a location $s_{loc} \in \{$N (north), S (south), C (central)$\}$ is introduced. Furthermore, the location of a node $v = (s, l)$ is defined as the location of its corresponding station, i.e. $v_{loc} = s_{loc}$ with $v = (s, l)$. In Figure 6 on page 15 the fragmentation of Copenhagen's S-train network is depicted. With the help of this classification it is now possible to state the conditions that have to be met by l. So, $l \in \mathcal{L}^o$ is considered if it meets at least one of the conditions in Table 1 on page 16. These conditions ensure that the direction of the chosen starting nodes are corresponding to the location of

Origin Node Loc. o_{loc}	Destination Station Loc. d_{loc}	Line Direction l_d
N	C or S	S
S	C or N	N
C	N	N
C	S	S
C	C	N or S

Table 1: Conditions for Directed Lines in Algorithm 2

the origin station and the destination station. Here the assumption is made that a passenger would never take a train driving in the directly opposed direction of his destination station. Again the underlying assumption is justified by the design of Copenhagen's S-train network.

For the chosen directed lines l the corresponding nodes $n = (o, l)$ are considered. For each corresponding node $n = (o, l)$ each ride $r = (n, v, t)$ in its set of departing rides R^n is considered. If the scheduled departure time $\Pi^t_{dep_n}$ of ride r lies in time period b, a possible starting ride for the journey from station o to station d is found. This procedure for the transformation of OD-pairs without a direct connection is formally summarised in Algorithm 2 on page 16.

The critical point is the determination of the path Λ_{nd} for the new OD-pair $\phi = \{n, d, r_{dep}, \Lambda_{nd}\}$. For the computation of the paths Λ_{nd} a modified version of Dijkstra's shortest path algorithm [Dij59] is used. Primarily all OD-pairs without direct connection are transformed and an empty path Λ_{nd} is assigned to each OD-pair. Subsequently, the path for every OD-pair $\phi = \{n, d, \Pi^t_{dep_n}, \Lambda_{nd}\} \in \Phi$ is calculated. As described in Section 3.3 it is possible that paths for several OD-pairs are calculated in a single run of the algorithm. Consequently computing time is saved by this order of implementation.

Algorithm 2 Transformation of OD-pairs without direct connection

```
Require: OD-pair $p = \{o, d, b\} \in P$ without direct connection in time period $b$
for all directed lines $l \in \mathcal{L}^o$ do
    if $l$ meets one of the conditions in Table 1 on page 16 then
        node $n = (o, l) \in V$
        for all rides $r = (n, v, t) \in R^n$ with $\Pi^t_{dep_n} \in b$ do
            add $\phi = \{n, d, \Pi^t_{dep_n}, \Lambda_{nd}\}$ to $\Phi$
        end for
    end if
end for
```

3.3 Path Algorithm for OD-pairs

After the transformation of all OD-pairs without direct connection the path Λ_{nd} for each OD-pair $\phi = \{n, d, \Pi^t_{dep_n}, \Lambda_{nd}\} \in \Phi$ has to be determined. For this purpose a modified version of Dijkstra's shortest path algorithm [Dij59] is described in this section, before it is stated formally in Algorithm 3.

3.3.1 Basic Structure of the Path Algorithm

Let $\phi = \{o, d, \Pi^t_{dep_n}, \Lambda_{nd}\} \in \Phi$ with $o \in V$ and $d \in S$. For every node $v \in V$ a previous node $v_{prev} \in V$, a previous ride v_{ride}, and a distance v_{dist} is introduced. The distance v_{dist} is the time a passengers needs to travel from origin node o to node v with respect to the scheduled time table Π.

1. v_{prev} and v_{ride} are set empty, and v_{dist} is set to infinity for all $v \in V$. For the initial node o the distance o_{dist} is set to zero. Furthermore, the initial node o is set as current node c.

2. Choose node $v \in V \setminus \{(s, l) \in V \mid s = c_{station}\}$ with the smallest distance v_{dist} and set this node v as current node c. Nodes of the same station are not considered, because in case of a transfer the consecutive ride after the transfer is added immediately to the path. Thus, the arrival node instead of the departure node of the consecutive ride is taken into consideration.

3. (a) If the smallest distance c_{dist} equals infinity, all remaining nodes are inaccessible from the initial node and we are done.

   (b) If the destination is reached, i.e. $c_{station} = d$, we are done.

   (c) If the current node c and the destination station c have a direct connection, i.e. $c_{line} \in \mathcal{L}^d$ the path to the destination station can be completed as described in Section 3.1 and we are done.

   (d) Otherwise, the current node c is removed from V and each direct neighbour of c that is still in V is updated. This process of updating, including the update of the distance v_{dist}, the previous node $v_{prev} \in V$, and the previous ride v_{ride}, is described in Section 3.3.2. If V is not empty we return to step 2.

After the steps above are completed, it remains to add the corresponding rides to the path Λ_{od} of the OD-pair. This is simply done by iterating through the previous rides, beginning with the destination station. Consequently, a path from the origin node o to the destination station d is determined. Furthermore, all “subpaths” of $\Lambda_{nd} = \{r_1, r_2, \dots, r_{|\Lambda_{nd}|}\}$ can be assigned to the corresponding OD-pair in case it exists. More precisely, if an OD-pair ϕ with $r_{i_{start}}$ as start node, $r_{j_{end}}$ as end node and $\Pi^{r_i}_{dep}$ as departure time exists, the path $\left\{r_i, \dots, r_j\right\}$ is assigned to this OD-pair. In doing so for all $i < j$, $i,j \in \{1, \dots, |\Lambda_{nd}|\}$ many paths do not have to be determined any more. The determination of the paths for the OD-pairs without direct connection is formally summarised in Algorithm 3 on page 18.

Algorithm 3 Path Algorithm for OD-Pairs without Direct Connection

```
Require: OD-pair $\phi = \{o, d, \Pi^t_{dep_o}, \Lambda_{od}\} \in \Phi$ with $o \in V$ and $d \in S$
— Initialisation —
for all nodes $v \in V$ do
    $v_{prev} = \emptyset$, $v_{ride} = \emptyset$, $v_{dist} = \infty$
end for
$o_{dist} = 0$
— Main Loop —
while $V \neq \emptyset$ do
    $c = \min_{v \in V^*} v_{dist}$ with $V^* = V \setminus \{(s, l) \in V \mid s = c_{station}\}$
    if $c_{dist} = \infty$ then
        break
    else if $c_{station} = d$ then
        break
    else if $c_{line} \in \mathcal{L}^d$ then
        complete the path by setting the corresponding previous nodes as described in Algorithm 1 and break
    else
        $V = V \setminus c$
        for all $v \in V$ do
            update $v$ as described in Section 3.3.2
        end for
    end if
end while
— Path Assignation —
while $c_{prev} \neq \emptyset$ do
    insert $c_{ride}$ at the beginning of $\Lambda_{od}$
end while
```

3.3.2 Updating Nodes in the Path Algorithm

In this section the process of updating nodes employed in Algorithm 3 is described. Let c be the current node. First of all, only neighbour nodes of c that implicate a reasonable continuation of the journey are updated. Again the design of Copenhagen's S-train network justifies conditions which delimit the number of nodes that have to be considered. Two cases are differentiated:

Changing Edge If the neighbour node n of the current node c is connected by a changing edge, i.e. $n_{station} = c_{station}$ and $n_{line} \neq c_{line}$, it is checked whether line n_{line} is a reasonable choice to reach the destination. More precisely, it is reasonable to change from the current node $c = (s, l_1)$ to node $v = (s, l_2)$ if the directed line l_2 meets one of the conditions in Table 2 on page 19. These conditions accelerate the path algorithm significantly and make it possible to calculate all paths in an acceptable time.

o_{loc}	d_{loc}	Additional Condition
N	S	$l_{2_d} = \text{S}$
N	N	$(c_{loc} = \text{N} \wedge l_{2_d} = \text{S}) \vee c_{\text{loc}} = \text{C}$
N	C	$l_{2_d} = \text{S}$
S	N	$l_{2_d} = \text{N}$
S	S	$(c_{loc} = \text{S} \wedge l_{2_d} = \text{N}) \vee c_{\text{loc}} = \text{C}$
S	C	$(c_{loc} = \text{S} \wedge l_{2_d} = \text{N}) \vee c_{\text{loc}} = \text{C}$
C	N	$l_{2_d} = \text{N}$
C	S	$l_{2_d} = \text{S}$
C	C	$l_2 \neq \text{F}$
C	C	$l_2 = \text{F} \wedge c_{\text{loc}} = \text{C}$
C	C	$l_1 = \text{F} \wedge c_{\text{loc}} = \text{C}$

Table 2: Conditions for Nodes in case of a Transfer in Algorithm 3

The next possible ride $r^* \in R^n$ departing from this node is given by

$$r^* = \arg\min_{r \in R^n} \left\{ \Pi^n_{dep_{t_1}} - \Pi^c_{arr_{t_2}} \mid \Pi^n_{dep_{t_1}} - \Pi^c_{arr_{t_2}} \geq 0 \right\},$$

where t_1 is the train the passenger used to reach the current node c and t_2 is the train the passenger would use to continue his journey from node n. Now the distance at the start node of $r^* = (n, v, t)$ is given by $\Pi^t_{dep_n} - \Pi^{t_{start}}_{dep_o}$ and the distance at the end node of r^* is given by $\Pi^t_{arr_v} - \Pi^{t_{start}}_{dep_o}$, where t_{start} is the train the passenger used to start his journey at the origin node o. If the distances at the start and end node are smaller than the current distances n_{dist} and v_{dist} respectively, nodes n and v are updated, i.e.

- $n_{prev} = c$, $n_{dist} = \Pi^t_{dep_n} - \Pi^{t_{start}}_{dep_o}$, and

- $v_{prev} = n$, $v_{dist} = \Pi^{t}_{arr_v} - \Pi^{t_{start}}_{dep_o}$, $v_{ride} = r^*$.

Continuative Edge Otherwise if the neighbour node $n = (s, l)$ is connected by a continuativ edge, i.e. $n_{station} \neq c_{station}$ and $n_{line} = c_{line}$, it is checked whether the continuation is a reasonable choice to reach the destination. More precisely, it is reasonable to choose a neighbour node $v = (s_2, l)$ of the current node $c = (s_1, l)$ if the directed line l meets one of the conditions in Table 3 on page 20.

o_{loc}	d_{loc}	Additional Condition
C	N	$c_{loc} \neq$ S
C	S	$c_{loc} \neq$ N
C	C	$c_{loc} =$ C
N	—	—
S	—	—

Table 3: Conditions for Nodes in case of a Continuation in Algorithm 3

The next possible ride r^* departing from the current node c is given by

$$r^* = \arg\min_{r \in R^c} \left\{ \Pi^{c}_{dep_{t_1}} - \Pi^{c}_{arr_{t_2}} \mid \Pi^{c}_{dep_{t_1}} - \Pi^{c}_{arr_{t_2}} \geq 0 \right\},$$

where t_1 is the train the passenger used to reach the current node c and t_2 is the train the passenger would use to continue his journey from the current node c. Now the distance at the end node of $r^* = (c, v, t)$ is given by $\Pi^{t}_{arr_v} - \Pi^{t_{start}}_{dep_o}$, where t_{start} is the train the passenger used to start his journey at the origin node o. If this distance is smaller than the current distance v_{dist}, node v is updated, i.e.

- $v_{prev} = c$,
- $v_{dist} = \Pi^{t}_{arr_n} - \Pi^{t_{start}}_{dep_o}$, and
- $v_{ride} = r^*$.

3.4 Reallocation of Passenger Counts

The set of OD-pairs P has been transformed into a new set of OD-pairs Φ. Now the passenger counts ω_p for every $p \in P$ have to be reallocated to new passenger counts ω_ϕ for every OD-pair $\phi \in \Phi$. Let $p = \{o, d, b\} \in P$ and let $\Phi_p = \left\{\phi_1, \phi_2, \ldots, \phi_{|\Phi_p|}\right\} \subseteq \Phi$ be the set of OD-pairs generated from p ordered by departure time at the origin node. It is assumed that the passengers arrive uniformly at a station during time period $b \in \mathcal{B}$. Consequently, ω_{ϕ_i} with $\phi_i = \{o_i, d_i, \Pi^t_{dep_{o_i}}, \Lambda_{o_i d_i}\}$ is given by

$$\omega_{\phi_i} = \left(\Pi^t_{dep_{o_i}} - \Pi^t_{dep_{o_{i-1}}}\right)\frac{\omega_p}{l(b)}, \qquad \text{for all } i = 1, 2, \ldots, |\Phi_p|$$

where $l(b)$ is the length of the time period b. $\Pi^t_{dep_{o_0}}$ is set as the beginning of time period b and $\Pi^t_{dep_{o_{|\Phi_p|}}}$ is set as the end of time period b. This procedure is formally summarised in Algorithm 4 on page 21.

Algorithm 4 Reallocation of Passenger Counts

```
Require: OD-pair $p \in P$
for all $\phi_i \in \Phi_p = \left\{\phi_1, \phi_2, \ldots, \phi_{|\Phi_p|}\right\}$ do
    $\omega_{\phi_i} = \left(\Pi^t_{dep_{o_i}} - \Pi^t_{dep_{o_{i-1}}}\right)\frac{\omega_p}{l(b)}, \qquad \text{for all } i = 1, 2, \ldots, |\Phi_p|$
end for
```

3.5 Transformation of the OD-Matrix

In this section the complete transformation of all OD-pairs $p \in P$ to a new set of OD-pairs Φ is summarised in Algorithm 5. The resulting set of OD-pairs Φ will be needed for the calculation of the passenger delay at the final destination, described in Section 4.

Algorithm 5 Transformation of the OD-Matrix

```
Require: Set of OD-pairs $P$
for all OD-pairs $p \in P$ do
    if $\left\{l \in \mathcal{L}^o \cap \mathcal{L}^d \mid l^o_f(b) \neq 0\right\} \neq \emptyset$ then
        apply Algorithm 1 to transform $p$
    else
        apply Algorithm 2 to transform $p$
    end if
end for
for all OD-pairs $\phi = \{o, d, \Pi^t_{dep_o}, \Lambda_{od}\} \in \Phi$ with $\Lambda_{od} = \emptyset$ do
    apply Algorithm 3 to generate a path from $o$ to $d$
end for
for all OD-pairs $p \in P$ do
    apply Algorithm 4 to reallocate the passenger counts
end for
```

4 Analysis of Passenger Delay

After the transformation of the OD-matrix we are now able to start with the analysis of passenger delay. To this end different measures for passenger delay will be introduced, each of them stating information about the current situation of the passengers in the network. The visualisation of these measures, described in Section 5, will provide a better understanding of disorder situations. Furthermore, these measures will be used to determine whether a certain train should have skipped a station in order to decrease the total delay of all passengers. In Section 6 this is described in detail.

4.1 Passengers Waiting at Stations

If a train is delayed, not only passengers in the train are concerned with the delay but also passengers at the stations waiting for this train. It is assumed, that each passenger arrives at the station exactly at the time of the scheduled departure of the train he wants to catch. At least, if the passenger arrives earlier than the departure time of his train, this is not considered as waiting time. Consequently, the delay of the train equals the waiting time of the passengers waiting for this train at the station.

For each station s in S we introduce $s_{wp}(\tau)$ as the number of waiting passengers at station s at time τ and $s_{wt}(\tau)$ as the waiting time of all passengers waiting at station s at time τ. In order to calculate $s_{wp}(\tau)$ we sum up the entering passengers r_{enter} of all delayed rides $r = (v_1, v_2, t)$ with $v2_{station} = s$, $\Pi^r_{arr} < \Gamma^r_{arr}$ and $\Pi^r_{arr} < \tau < \Gamma^r_{arr}$.

$$s_{wp}(\tau) = \sum_{r \in R^*} r_{enter}, \qquad \forall s \in S,$$

where $R^* = \{r = (v_1, v_2, t) \in R \mid v2_{station} = s \wedge \Pi^r_{arr} < \Gamma^r_{arr} \wedge \Pi^r_{arr} < \tau < \Gamma^r_{arr}\}$. Consequently $s_{wt}(\tau)$ is given by

$$s_{wt}(\tau) = \sum_{r \in R^*} r_{enter}(\tau - \Pi^r_{arr}), \qquad \forall s \in S$$

where $R^* = \{r = (v_1, v_2, t) \in R \mid v2_{station} = s \wedge \Pi^r_{arr} < \Gamma^r_{arr} \wedge \Pi^r_{arr} < \tau < \Gamma^r_{arr}\}$.

4.2 Use to Capacity of Trains

For a rating of the importance of a train in the face of a passenger-orientated approach to analysing disorder situations the number of passengers in a train is decisive. Thus, the use to capacity u_t of a train t at time τ, given by

$$u_t(\tau) = \frac{\omega_{r^\tau}}{t_c}, \qquad \forall t \in T$$

where t_c is the capacity of train t and $r^\tau \in R^t$ such that $\Gamma^r_{dep} < \tau < \Gamma^r_{arr}$, is a reasonable measure for the relevance of a train in a disorder situation.

4.3 Passenger Delay at Final Destination

The extent of train delays for passengers can be measured by evaluating the delay of each passenger in the delayed train at his final destination, i.e. at the end of his journey, including transfers. By this means it is possible to distinguish between considerable and inconsiderable train delays. For instance, if a train has a delay of 10 minutes and all passengers in this train want to change trains at the next station and have to wait there 15 minutes for the connecting train anyway, this train delay is rather inconsiderable with regards to the passengers.

For each ride $r \in R$ we introduce Φ_r as the set of all OD-pairs whose path contains ride r, more precisely

$$\Phi_r = \{\phi = o, d, s_{od}, \Lambda_{od} \in \Phi \mid r \in \Lambda_{od}\}.$$

In order to calculate the delay of all passengers at their final destination for ride r, denoted by D^r_{pass}, all OD-pairs $\phi \in \Phi_r$ are considered. For each OD-pair

$$\phi = \{o, d, s_{od}, \Lambda_{od}\} \in \Phi_r$$

the following values are calculated:

Actual Arrival Time at Destination Station Γ^d_{arr} By means of the actual timetable Γ the actual arrival time at the destination station d can be calculated. Therefore, a similar path algorithm as described in Section 3.3 is employed. Instead of the scheduled times the actual times are applied and as a start node r_{end} is used. It is essential to determine the whole path again, since the most reasonable path of the passenger calculated for the OD-pair could have changed due to the delay.

Scheduled Arrival Time at Destination Station Π^d_{arr} In contrast to the actual arrival time at the destination station the scheduled arrival time at the destination station is directly given by

$$\Pi^{d}_{arr} = \Pi^{r_{|\Lambda_{od}|}}_{arr},$$

with $\Lambda_{od} = \{r_1, r_2, \dots, r_{|\Lambda_{od}|}\}$.

Number of Passengers Travelling to Destination Station κ_ϕ For every OD-pair $\phi \in \Phi$ the number of passengers travelling from the origin to the destination is known. Additionally, the number of passengers of ride r is known. An approximation for the number of passengers of ride r travelling to the destination Station d is therefore given by

$$\kappa_\phi = \omega_r \frac{\omega_\phi}{\sum\limits_{\rho \in \Phi_r} \omega_\rho}.$$

With the help of these values the delay of all passengers at their final destination for ride r is now given by

$$D^{r}_{pass} = \sum_{\phi \in \Phi_r} (\Gamma^{d}_{arr} - \Pi^{d}_{arr}) \kappa_\phi.$$

5 Replay

Dataset $\mathcal{D}_{train}$ and the calculations in Section 4 provide a lot of information about the passenger delay. In order to review and analyse the network with respect to the passenger delay a visualisation of the information is helpful. Consequently, a graphical refurbishment of all network activity including the passenger delay information was implemented. Since this is not a simulation in the first place — most of the information are given by the datasets — it is referred to as *replay*. The implementation of the replay results from a cooperation with Jonas Harbering, who also dealt with the analysis and optimisation of disorder situations in public transportation networks. For further information, especially on the replay, see [Har09].

In this section the different elements of the replay and the underlying information will be described.

Map The datasets $\mathcal{D}_{imap}$ and $\mathcal{D}_{lmap}$ mentioned in Section 2.1.3 provide the necessary data to draw a schematic infrastructure network and a schematic line network respectively. Two different maps have been implemented in order to give the user, who employs the replay to analyse disorder situations, a choice of which map type is more applicable to a specific situation. The schematic infrastructure map and the schematic line map are depicted in Figure 2 on page 4 and Figure 1 on page 3 respectively.

Stations Stations are symbolised by a thick black dash and a corresponding label with an abbreviation of the name of the station. The bounding-box of the label can be filled with a colour corresponding to the waiting time $s_{wt}(\tau)$ of all passengers waiting at station s, as described in Section 4.1. The colour ranges from green, indicating no waiting time, to red, indicating a very high waiting time. In Figure 7 on page 25 symbolised stations are depicted.

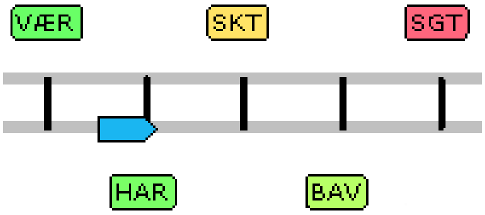


Figure 7: Stations

Trains Trains are drawn as small rectangles with a triangle pointing to the driving direction of the train. According to requirements the filling of the trains can either be corresponding to the line colour of the train or to the passenger delay at the final destination D^r_{pass}, as described in Section 4.3, combined with the use to capacity of the train $u_t(\tau)$, as described in Section 4.2. In the latter case the colour indicates the passenger delay at the final destination and the charging level of the train indicates the use to capacity. Again, the colour ranges from green, indicating no passenger delay, to red, indicating a very high passenger delay. In Figure 8 on page 25 both options are depicted. Furthermore, every delayed train has a grey shadow train, which appears at

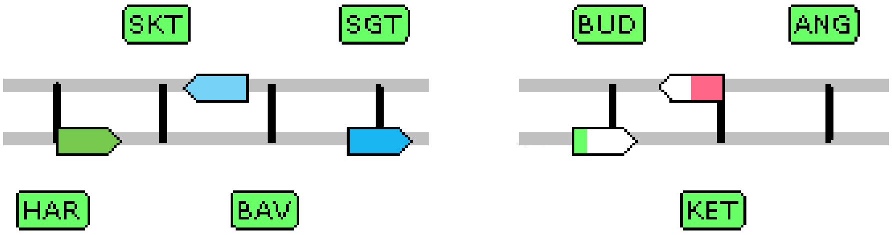


Figure 8: Different Train Types

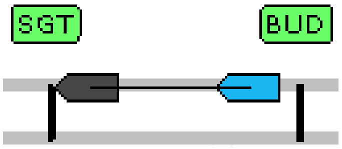
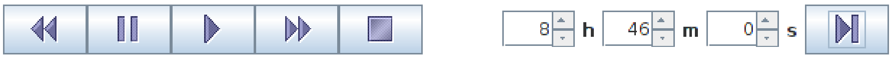


Figure 9: Shadow Train

the position, where the train should have been according to the scheduled timetable. A thin line connects the actual train and its shadow train in order to clarify the connection for the user. In Figure 9 on page 26 a train and its shadow train are depicted.

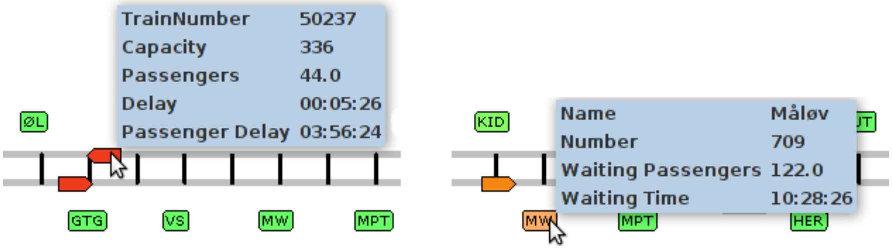
Control Bar With the help of the control bar the user is able to fast-forward, rewind, play, pause and abort the replay. Additionally it is possible to jump to a certain time of the day. In Figure 10 on page 26 the control bar is depicted.



8 h 46 m 0 s

Figure 10: Control Bar

Tool Tips The preservation of a clear arrangement is an important factor for the replay. Thus we decided to employ tool tips in order to show information about a train or a station only if required. For each train the train number, the train capacity, the number of passengers in the train, the train delay and the passenger delay at the final destination are shown in a tool tip, if the mouse is placed over the train.



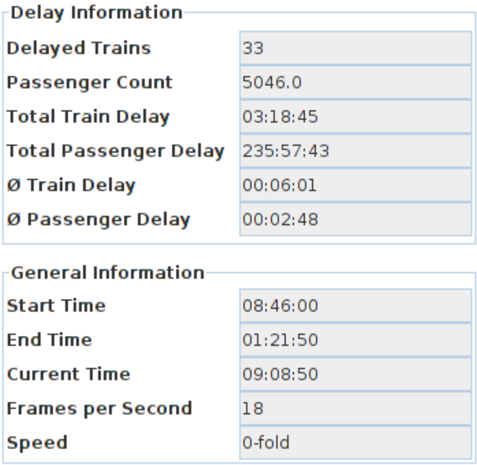
TrainNumber	50237
Capacity	336
Passengers	44.0
Delay	00:05:26
Passenger Delay	03:56:24

Name	Måløv
Number	709
Waiting Passengers	122.0
Waiting Time	10:28:26

Figure 11: Tool Tips for Trains and Stations

For each station the full name of the station, the station number, the number of waiting passengers at the station and the waiting time at the station is shown. In Figure 11 on page 26 a tool tip both for a train and a station is depicted. If the user clicks on a certain train, a new window opens providing detailed information about the train, including a full list of rides of the train.

Information Panel Further information about the current situation are given in the information panel.



Delay Information

Delayed Trains	33
Passenger Count	5046.0
Total Train Delay	03:18:45
Total Passenger Delay	235:57:43
Ø Train Delay	00:06:01
Ø Passenger Delay	00:02:48

General Information

Start Time	08:46:00
End Time	01:21:50
Current Time	09:08:50
Frames per Second	18
Speed	0-fold

Figure 12: Information Panel

Statistics such as the average passenger delay or the average train delay corresponding to the current time are listed in this panel. In Figure 12 on page 27 the information panel is depicted.

Delay Chart In an additional window the total train delay and the total passenger delay at the final destination are plotted. In order to keep the graphs in a common range the total passenger delay at the final destination is divided by 1000. In Figure 13 on page 28 the delay chart for the 5^{th} of January, 2009 is depicted. The horizontal axis represents the time in K-seconds and the vertical axis represents the delay in K-seconds. Moreover, the red line corresponds to the total train delay and the blue line corresponds to the total passenger delay at final destinations divided by 1000. The characteristics of both curves are quite similar.

Delay Causality Dataset $\mathcal{D}_{train}$ provides not only data about the cause of the delay but also about the train that is responsible for the delay. Thus, a delay causality tree can be drawn: Every train with a source delay, i.e. a train with itself in the “Delay Caused by” column, is represented by a root node and the

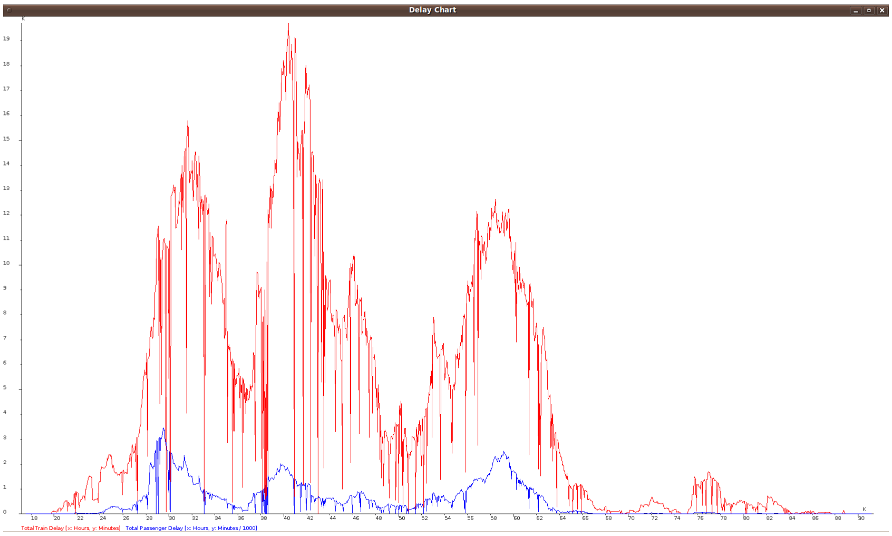


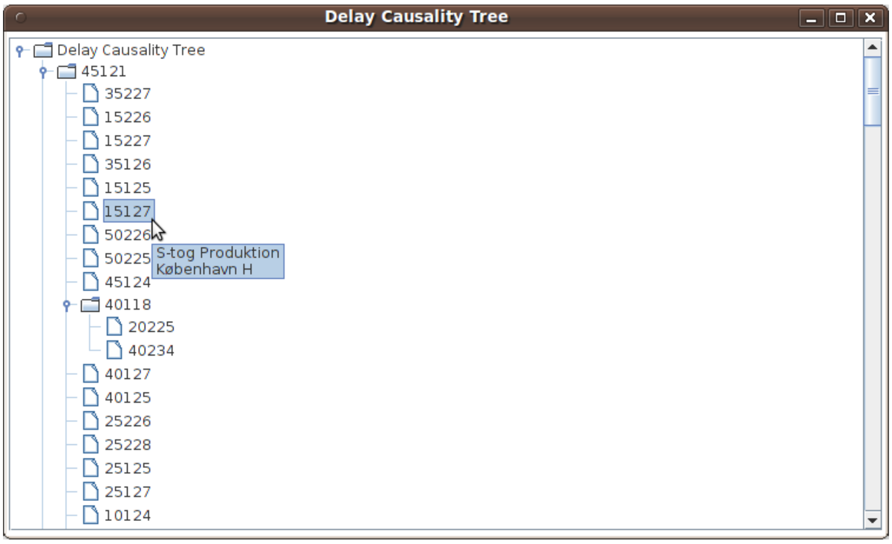
Figure 13: Delay Chart

trains, whose delay is caused by this train are listed as child nodes. If one of these child nodes causes a delay for another train in turn, this delayed train is shown as a child node of the child node, and so on. If the user clicks on one of the trains in the delay causality tree, the train is marked by a red circle on the map. By this way a certain train can easily be localised on the map. In Figure 14 on page 29 a small part of a delay causality tree is depicted.

6 Optimised Skipping of Stations

In order to delimit the extent of disorder situations there are several possibilities to intervene in the ongoing operation. Besides the possibility to cancel highly delayed trains, treated in [Har09], it is an option to instruct delayed trains to skip single stations in order to make up for lost time and to keep delays caused by this train within a limit. This option will be dealt with in this section. Again a passenger-orientated approach is chosen, i.e. the advantages and disadvantages for the passengers caused by the skipping of a certain station are measured.

First of all a particular train is considered. Furthermore, it is analysed whether this particular train should skip a station on his trip. The decision whether this particular train should skip a station or not is based on the following considerations. Let $t \in T$ be the fixed train, $R^t = \{r_1, r_2, \ldots, r_{|R^t|}\} \subseteq R$ the set of rides of t with $r_i = (v_i, v_{i+1}, t)$, $i = 1, \ldots, |R^t|$, and $V^t = \{v_1, v_2, \ldots, v_{|V^t|}\} \subseteq V$ the set of nodes of



Delay Causality Tree
45121
35227
15226
15227
35126
15125
15127
50226
50225
45124
40118
20225
40234
40127
40125
25226
25228
25125
25127
10124

S-tog Produktion
København H

Figure 14: Delay Causality Tree

the change & go graph where train t stops. Moreover, let $v_i \in V^t$ with $1 < i < |V^t|$ be the node skipped by train t.

Waiting Passengers If node v_i is skipped, the passengers that wanted to enter train t at this node have to wait for the next train. The additional waiting τ_1 time is given by

$$\tau_1 = r_{i-1_{enter}}(\Gamma^{r^n}_{arr} - \Gamma^{r_{i-1}}_{arr}),$$

where r^n is the next possible ride for the passengers waiting at node v_i to continue their journey.

Leaving Passengers It is assumed that the passengers who want to leave train t at node v_i will leave the train one station earlier and wait there for the next train in order to reach node v_i. Thus, we have an additional travelling time τ_2 given by

$$\tau_2 = r_{i-1_{leave}}(\Gamma^{r^n_{i-2}}_{arr} - \Gamma^{r_{i-2}}_{arr}),$$

where r^n_{i-2} is the next possible ride for the passengers leaving train t early to reach their destination node v_i.

Saved Time If node v_i is skipped, at least the sojourn time S^{v_i} is saved. Thus, we have a saved time τ_{save_1} given by

$$\tau_{save_1} = S^{v_i} \left((\omega_{r_{i-1}} - r_{i-1_{leave}}) + \sum_{j=i}^{|R^t|} r_{j_{enter}} \right)$$

Impact on other Trains When a delay is caused by another train this is recorded in dataset $\mathcal{D}_{train}$. Thus, if the train that skipped node v_i, is responsible for the delay of another train, the saved time can be handed down. At this point we assume, that if train t_1 is responsible for the delay of train t_2 and train t_1 skips a station before the delay of train t_2 occurs, the delay of t_2 is reduced by S^v. This process can be continued if further delay causalities exist. Formally, we introduce T^t_{delay} as the set of trains whose delay is caused by train t and t_{occ} as the time when the delay of train t primarily occurred. Thus, we have a saved time τ_{save_2} given by

$$\tau_{save_2} = \sum_{t' \in T^*} S^{v_i} \left(\omega_r + \sum_{j=i+1}^{|R^t|} r_{j_{enter}} \right),$$

where $T^* = \left\{ t' \in T^t_{delay} \mid t'_{occ} > \Gamma^{r_{i-1}}_{arr} \right\}$, r is the ride of train t' where the delay primarily occurred and i the index of r in $R^{t'}$. This calculation is repeated for every train $t' \in T^*$.

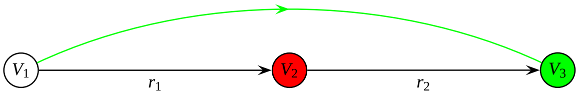


Figure 15: Skipping of Stations

The considerations above give rise to introduce v_{skip} for every $v \in V^t$ given by

$$v_{skip} = \tau_{save_1} + \tau_{save_2} - \tau_1 - \tau_2.$$

Consequently

$$v^* = \arg\max_{v \in V^t} v_{skip}$$

is the node most likely to be skipped by train t. It is reasonable for train t to skip this node, if

$$\max_{v \in V^t} v_{skip} > 0.$$

In order to limit the scope the thoughts for the passenger delay made in this section consider the delay at the current ride, not at the final destination as in Section 4.3. Of course the concept of calculating the passenger delay at the final destination could be employed in this section as well.

Furthermore, this approach is a local optimisation considering each train individually and not a global optimisation considering all trains. This approach is chosen because in this way a simple implementation is possible and the results for each train are comprehensible. Up to now a particular train is considered independently of skipping operations of other trains. For an extension of this basic approach all (delayed) trains of one line come into consideration. If a particular train skips a station, this affects the following trains on this line. Consequently, the decision whether following trains should skip a station has to be based on the results of the previous trains. This proceeding can be continued iteratively in chronological order until all (delayed) trains of a line are considered. By this way interdependencies of all trains of a particular line with respect to the skipping of stations are taken into account. A further extension of this approach to the whole network, i.e. consideration of interdependencies of different lines, would be even more complex and is not covered in this thesis.

The implementation of the basic approach developed in this section, i.e. highlighting of trains and corresponding stations that should be skipped by these trains in the replay, would provide a starting point for further analysis in order to optimise the timetable and prevent the escalation of disorder situations.

7 Conclusion

First of all, the conclusion is made that a passenger-orientated approach is a reasonable way to analysing and optimising disorder situations in public transportation networks. An exercisable method to transform the given OD-matrix corresponding to a PTN into a new OD-matrix corresponding to an extended change & go graph is presented and forms the basis for further analysis of passenger delay. Moreover, different measures to classify passenger delay were found and especially the calculation of the passenger delay at final destinations provides enlightening information about the impact of trains in disorder situations.

The graphical visualisation of all (passenger) delay information in the replay offers several possibilities for the employees at DSB S-tog to review and analyse past days of operation in order to develop rules of action in disorder situations. Furthermore, a passenger-orientated approach to optimised skipping of stations is provided, which could be implemented in order to gain even more information from the replay.

8 Acknowledgement

First of all, I would like to thank Jonas Harbering for the constructive and pleasant collaboration. Furthermore, I appreciate the engagement and all the valuable suggestions of Per Thorlacius and Torben Christensen at DSB S-tog, who significantly contributed to the success of this thesis. Last but not least I would like to thank Prof. Dr. Anita Schöbel for her support and for the possibility to write a bachelor's thesis with a high level of practical relevance.

Frequently Used Notation

Datasets:

$\mathcal{D}_{train}$	Dataset with information about all rides
$\mathcal{D}_{od}$	OD-matrix provided by DSB S-tog
$\mathcal{D}_{odt}$	Transformed OD-matrix used for passenger delay calculations
$\mathcal{D}_{imap}$	Information for the schematic infrastructure map
$\mathcal{D}_{lmap}$	Information for the schematic line map

Network:

$\text{PTN} = (S, E)$	Public Transportation Network
S	Set of stations in PTN
E	Set of edges in PTN
L	Set of lines
$\mathcal{L}$	Set of directed lines
l_d	Direction of line l
$l_f^s(b)$	Frequency at which station s is served by line l in time period b
$\mathcal{B}$	Set of time periods
$\mathcal{L}^s$	Set of directed lines that serve station s
V	Set of nodes in the change & go graph
$\mathcal{E}$	Set of edges in the change & go graph
$v_{station}$	Corresponding station of node v in the change & go graph
v_{line}	Corresponding line of node v in the change & go graph
T	Set of trains
V^t	Set of nodes in the change & go graph where train t stops

Timetable:

Π	Scheduled timetable
$\Pi^{v}_{arr_t}$	Scheduled arrival time of train *t* at node *v* in the change & go graph
$\Pi^{v}_{dep_t}$	Scheduled departure time of train *t* at node *v* in the change & go graph
Γ	Actual timetable
$\Gamma^{v}_{arr_t}$	Actual arrival time of train *t* at node *v*
$\Gamma^{v}_{dep_t}$	Actual departure time of train *t* at node *v*
S^{v}	Sojourn time at station *v*

Rides:

R	Set of rides
ω_r	Number of passengers of ride *r*
r_{start}	Start node of ride *r*
r_{end}	End node of ride *r*
r_{enter}	Number of entering passengers at the end node of ride *r*
r_{leave}	Number of leaving passengers at the end node of ride *r*
$\Pi^{r}_{dep}, \Gamma^{r}_{dep}$	Scheduled and actual departure time at the start node of ride *r*
$\Pi^{r}_{arr}, \Gamma^{r}_{arr}$	Scheduled and actual arrival time at the end node of ride *r*
R^{e}	Set of rides corresponding to edge *e* in the change & go graph
R^{t}	Set of rides of train *t*
R^{v}	Set of rides departing from node *v* in the change & go graph

OD-Matrix:

P	Set of OD-pairs from the original OD-matrix
ω_p	Number of passengers associated with OD-pair $p \in P$
Φ	Set of OD-pairs from the transformed OD-matrix
ω_ϕ	Number of passengers associated with OD-pair $\phi \in \Phi$
Λ_{od}	Path from node o to node v

Analysis of Passenger Delay:

τ	Time
$s_{wp}(\tau)$	Number of waiting passengers at station s at time τ
$u_t(\tau)$	Use to capacity of train t at time τ
D^r_{pass}	Delay of all passengers at their final destination for ride r

List of Figures

List of Tables

References

[DHSS10] T. Dollevoet, D. Huisman, M. Schmidt, and A. Schobel. Delay Management with Re-Routing of Passengers. Econometric Institute Report, Erasmus University Rotterdam, Econometric Institute, May 2010.

[Dij59] E.W. Dijkstra. A Note on Two Problems in Connection with Graphs. *Numerical Mathematics*, 1:269–271, 1959.

[dk.06] dk.trackmap.net. Schematic Infrastructure Map. Website, 2006. `http://dk.trackmap.net/a`; visited on 28th of March 2010.

[DSB10] DSB. Schematic line map. Website, 2010. `http://www.dsb.dk/Global/PDF/Koereplaner/S-tog/Alle%20linjer.pdf`; visited on 28th of March 2010.

[Har09] J. Harbering. *Train Delay in Public Transportation Networks*. Bachelor's thesis, Georg-August-Universität Göttingen, 2009.

[Sch06] A. Schöbel. *Optimization in Public Transportation*. Springer, Georg-August-Universität, Göttingen, Germany, 2006.

[Sch09] A. Schöbel. Line Planning in Public Transportation: Mathematical Programming Approaches. Technical report, Institut für Numerische und Angewandte Mathematik Göttingen, 2009.