Cost-Benefit-Analysis of Investments into Railway Networks with Periodically Timed Schedules

Michael Kolonko[1] and Ophelia Engelhardt-Funke[2]

Institute for Mathematics, Technical University of Clausthal, Germany

Abstract. An efficient planning of future investments into a railway network requires a thorough analysis of possible effects. Therefore a tool is needed for a cost-benefit-analysis at an early stage of the planning process. We present a method to obtain a cost-benefit-curve that shows the effect of investments (cost) on the quality of the network measured by the waiting time of passengers (benefit).

This curve is obtained from the solutions of a multi-criteria time-table optimization problem. Time-tables are evaluated with respect to the investment they require and the benefit they bring to passengers in terms of shorter waiting times. Moreover, we show how the notion of stability of a time-table under random delays can be included into our approach. The analysis is done on a strategic level without consideration of all operational details. We use genetic algorithms to find approximate solutions to the optimization problem.

A prototype system is presently tested on a network of regional lines in Germany. We report on the first very promising results.

1 Introduction

To increase the attractiveness of public transport it is important to improve the quality of the service for the passengers and in particular to reduce the travel and waiting times in the network. An improvement usually requires a major investment, therefore the traffic providers have to decide how the available money should be invested into the network to obtain a maximal benefit for the passengers. This is an urgent issue e.g. on some of the regional lines in the new states of Germany. Here, the condition of rails, crossings and switches allows only a limited speed of the trains on some of the sections. Usually there are different levels of investment possible e.g. one could simply provide a level crossing with automatic barriers, renew a switch or rebuild the whole section. Therefore, apart from deciding which sections are to be modernized the level of modernization has to be fixed for each section.

This requires a detailed cost-benefit-analysis of possible investments taking into account the different investment scenarios, see Fig. 1 for a simple example of a cost-benefit curve.

From such an analysis the decision makers could expect answers to questions like:

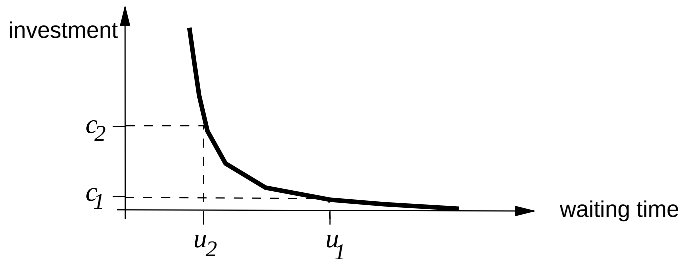

Fig. 1. A simple cost-benefit-curve, where benefit is measured by the waiting time in the network.

- What is the (maximal) benefit passengers can derive from any given amount of investment? How can it be obtained, i.e. what are the detailed investment decisions? Here, the 'benefit' is the reduction of waiting time, other choices are discussed below.
- How much money would it cost to increase the quality (i.e. to reduce the waiting time) by a certain percentage ?
- What is the return of investment (measured in terms of benefit), i.e. how much additional benefit could be obtained from increasing the investment over a certain level (see e.g. levels c_1 and c_2 in Fig. 1) ?

Generally, the benefit for the passengers is the quality improvement of the time-table : for example, higher speed of the trains shortens the travel times and also gives room to design time-tables with better connections and shorter waiting times. The link between investment and passengers is provided by the time-table which therefore will be the main ingredient of our analysis.

The necessary investment as well as the benefit for the passengers can be formulated as properties (more formally: cost functions) of a time-table. Finding optimal time-tables with respect to these multiple targets then yields a cost-benefit-analysis as is shown below. In addition, this result will also give the planner all the information on how to allocate the investment in the network and how to schedule the trains to obtain the maximal benefit.

It should be pointed out that we are only concerned with the strategic, long-term planning problem in which operational details like safety headways and capacity restrictions are not considered.

This research is performed in cooperation with the Nahverkehrsservice Sachsen-Anhalt (NASA) GmbH. NASA is a provider of regional rail traffic in the state of Sachsen-Anhalt in Germany. A prototype system for the optimization and the evaluation of the results has been developed and is presently tested on the network of NASA as explained below.

2 A Mathematical Model of Time-Table Optimization

2.1 The Time-Table

We consider a network with fixed **lines** $\mathfrak{L} = \{L_1,\dots,L_N\}$ that are served periodically each with a fixed period τ_{L_i}. A line is represented by the list of consecutive stations the train passes through. Reverse directions are modeled as separate lines. We assume that lines are strictly periodical. That means e.g. that if some stations are skipped on a line during weak traffic hours, this has to be modeled as a separate line with a possibly large period.

Let $\mathfrak{S} = \{S_1,\cdots,S_K\}$ denote the set of all **stations** of the network and $\mathfrak{Z}$ the set of all **sections** of tracks, i.e. stretches of tracks between two neighbouring stations S, S'. Note that there may be more than one such section between two stations, in particular, if two lines have different speed on the same physical track, we model this by different sections. Let (S, S', L) denote the section from $\mathfrak{Z}$ that line L uses to travel from station S to its next station S'. To each section, there is a list of potential improvements with their costs and their effect on the running time of all lines on that section. When investing into that section one has to choose among these improvements.

Let

$$\mathfrak{D} = \{(L,S) \mid L \in \mathfrak{L}, S \in \mathfrak{S}; L \text{ departs from } S\}$$

be the set of possible **departures**. Then a **time-table** T consists of two lists:

$$T = \Big(\ (\pi(L,S) \mid (L,S) \in \mathfrak{D}), \quad (\delta(z) \mid z \in \mathfrak{Z}) \ \Big)$$

where $\pi(L,S) \in \{0,1,\cdots,\tau_L - 1\}$ is the **departure time** (modulo the period τ_L) of line L in station S and $\delta(z)$ is the **running time** scheduled for all trains on section $z \in \mathfrak{Z}$. From this the scheduled **arrival time**

$$\gamma(L,S') := \pi(L,S) + \delta((S,S',L))$$

of L in S' can be calculated. Incorporating the running times instead of the arrival times into the time-table is more convenient for our purpose.

As was mentioned above, we only consider a strategic planning situation, in which the details of the network like capacities of sections or stations and safety constraints (headways) are not taken into account. This greatly simplifies the problem of finding a (mathematically) feasible time-table. In fact, any list $(p(L,S) \mid (L,S) \in \mathfrak{D})$ of integers can be interpreted as a list of departure times simply by reducing them modulo the appropriate line period: $\pi(L,S) := p(L,S) \mod \tau_L$. We assume that for the running times $\delta(z)$ there is a lower bound $\underline{\delta}(z)$ that could be achieved if all improvements on that section were realized, and an upper value $\overline{\delta}(z)$, e.g. the running time of the present time-table. Then the set of feasible time-tables is given as

$$\mathfrak{T} := \mathop{\times}_{(L,S)\in\mathfrak{D}} \{0,\dots,\tau_{L-1}\} \times \mathop{\times}_{z\in\mathfrak{Z}} \{\underline{\delta}(z),\dots,\overline{\delta}(z)\}.$$

Any $T \in \mathfrak{T}$ represents a valid time-table within our framework. Note that all times are treated as integers, interpreted e.g. as 0.1 minute.

2.2 Cost Function I : Investment

Each feasible time-table $T \in \mathfrak{T}$ requires a certain running time on each section $z \in \mathfrak{Z}$. To calculate the amount of investment necessary to enable that running time one has to be given the **local cost functions**

$$c_z : \{\underline{\delta}(z), \cdots, \overline{\delta}(z)\} \to \mathbb{N}$$

for each $z \in \mathfrak{Z}$. $c_z(\delta)$ gives the minimal amount of money needed to enable the running time δ on section z, see Fig. 2 for an example of such a local cost function. As mentioned above there is a list of possible improvements on section z e.g. building a new level crossing, modernizing a switch or rebuilding the whole track. The local cost function is built from these data by calculating the cost and the reduction in running time for all possible combinations of improvements. In particular the system checks all possible combinations for the cheapest way to achieve a certain reduction in running time. In Fig. 2, a running time δ would require to install a new crossing and a new switch at the (cumulative) cost of $c_z(\delta) = c' + c$.

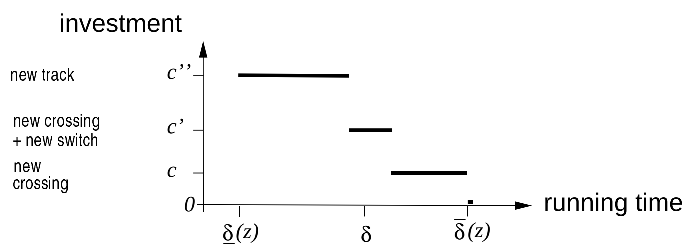

Fig. 2. A simple local cost function $c_z(\cdot)$

From these local cost functions we may calculate the **total amount of investment** required by time-table T as

$$C(T) := \sum_{z \in \mathfrak{Z}} c_z(\delta(z)).$$

Note that now the investment is just a cost function value, i.e. a property of the time-table.

2.3 Cost Function II : Waiting Time

When changing from line L to line L' at station S the waiting time in a periodic network can be given by

$$w(L, L', S) := \pi(L', S) - (\gamma(L, S) + \alpha(L, L', S)) \mod \gcd(\tau_L, \tau_{L'}).$$

Here, $\gamma(L, S)$ is the arrival time of line L at station S as defined above in Sec. 2.1. For $L \neq L'$, $\alpha(L, L', S)$ denotes the minimal transfer time at station S from line L to L'. Then $(\gamma(L, S) + \alpha(L, S))$ denotes the earliest possible time at which a passenger changing from line L to line L' can reach the train of line L' and $\pi(L', S)$ is his or her actual departure time. For $L = L'$, $\alpha(L, L, S)$ denotes the minimal stopping time of line L in station S. Therefore, $w(L, L, S)$ gives the amount by which the actual stopping time of line L exceeds its minimal stopping time in station S. This is the waiting time of passengers continuing their journey on line L at station S.

It is known from the literature that for any connection in a periodic network the smallest, the largest and the average waiting time occurring during a day differ only by constants from $w(L, L', S)$, see Nachtigall (1996) and the references given there. As these constants do not depend on the timetable, minimizing $w(L, L', S)$ is equivalent to minimizing any of the above target functions. The **total weighted waiting time** is now given by

$$W(T) := \sum_{S \in \mathfrak{S}} \sum_{L, L' \in \mathfrak{L}} w(L, L', S) \cdot g(L, L', S).$$

Here $g(L, L', S)$ denotes a weight, e.g. the average number of passengers changing from line L to line L' at station S. $g(L, L', S)$ will be 0 if there is no reasonable connection from L to L' at S, e.g. if L' is the reverse line of L. For $L = L'$, $g(L, L', S)$ denotes the number of passengers who continue their journey on line L as described above. There is an additional penalty factor for $L \neq L'$ to give waiting on the platform a heavier weight than waiting in the train during stops.

The weights $g(L, L', S)$ can be entered into our system if such numbers are available e.g. from traffic counts. If such information is not available it can be estimated by our system: if there is an $OD-$matrix $M = (m(o, d))$ available giving the number of passengers $m(o, d)$ that travel from origin o to destination d then our system can calculate weights by sending the $m(o, d)$ passengers along shortest routes through the network. If even less data are available, then we can estimate the $OD-$matrix using Lill's law, see Section 6.2 below.

In some situations it is important to consider additional waiting times like waiting for connections to other networks or waiting when entering the system at particular stations at given time points (schools, large factories). To include this kind of service into our system (e.g. arriving at 8 am at the school station) we model the corresponding events as artificial lines with fixed arrival and/or departure times. Minimizing the total waiting time $W(T)$ will then lead to small waiting times for the connections to these artificial lines (e.g. the train will arrive shortly before 8 am at the school station).

Note that we do not take into account waiting times that occur when entering the railway system from outside at some random point of time. These waiting times only depend on the periods of the lines which are considered to be fixed in our context.

2.4 Multi-Criteria Optimization

The two cost functions, investment $C(T)$ and waiting time $W(T)$, reflect the quality of a time-table from different viewpoints: the traffic provider will be interested in low investments whereas the passengers will insist on short travel times and on short waiting times as the most unpleasant part of the journey.

Further aspects of time-table quality may be considered by using additional cost functions like the total weighted travel time:

$$R(T) := \sum_{o,d \in \mathfrak{S}} r(o,d) \cdot g(o,d),$$

where $r(o,d)$ is the minimal scheduled travel time for a route from origin o to destination d and $g(o,d)$ is the weight for the importance of that route. We are also considering

$$U(T) := \text{ no. of vehicles necessary to run } T$$

which can be calculated from a particular periodic scheduling problem by linear optimization (see e.g. Orlin (1982), Serafini and Ukovich (1989)). Another cost function taking into account the delays will be discussed in the next section.

Thus we are faced with an optimization problem where each time-table $T \in \mathfrak{T}$ has a multi-dimensional vector of cost function values e.g.

$$\Big(C(T), W(T), R(T), U(T)\Big).$$

The aim is to find good time-tables under this multidimensional criterion. Obviously, these cost functions are not independent of each other.

Fig. 3 shows a simple example. There are four lines starting at the oval stations. The rectangles indicate stations where lines can be changed, we assume that the stopping and transfer times are 0. The numbers on the sections denote their running times. The trains are assumed to run every 60 minutes. The schedule is determined by the departure times in the starting stations which are chosen such that there are no waiting times at the first changing stations. This necessarily leads to a conflict at the last, fourth station. Whatever departure times we prescribe at the starting stations, the waiting

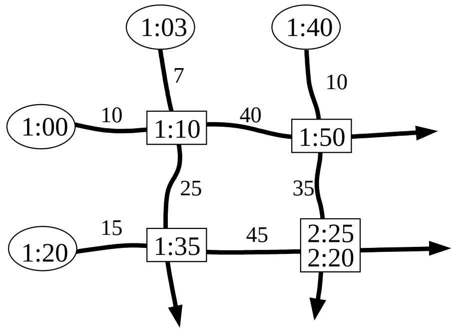

Fig. 3. A simple network with four lines

time in the system will be 5 min resp. 55 min as the running times in the directed cycle of the changing stations sum to $5 = 40 + 35 - 45 - 25$. Only an investment that will shorten these running times e.g. from 40 to 35 minutes will reduce the waiting time (to 0 in this case).

In general one cannot expect that there is a single time-table minimizing all criteria simultaneously. Instead, one is looking for the so-called **Pareto-optimal** or undominated solutions. A time-table T is **dominated** by T' if (for the cost values considered above)

$$\begin{array}{lll} C(T') \leq C(T) & , & R(T') \leq R(T), \\ W(T') \leq W(T) & \text{and} & U(T') \leq U(T), \end{array}$$

and at least one of the "$\leq$" is a "$<$". In this situation T' is better than T and T should not be used. In Fig. 4 the shaded area indicates the cost values of all time-tables for two cost functions. The time-table belonging to the black dot dominates all time-tables in the hatched quadrant. The bold line marks the Pareto-optimal solutions.

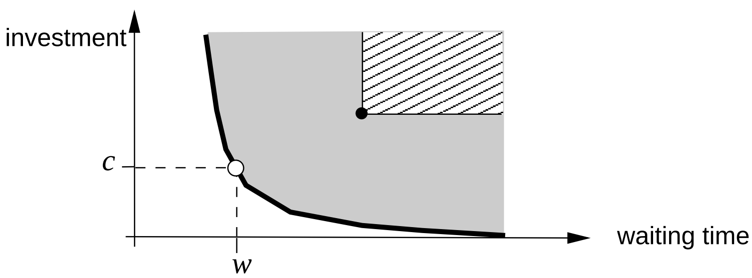

Fig. 4. Pareto-optimal solutions for two cost functions

The cost function values of the Pareto-optimal time-tables constitute a cost-benefit curve (or surface in case of more than two criteria). Each point (c, w) on the curve in Fig. 4 gives the maximal benefit w achievable by an investment c resp. the least investment one has to make to obtain benefit w. From the time-table T that is represented by the point $(c, w) = (C(T), W(T))$ we can also determine how the total investment

$$c = C(T) = \sum_{z \in \mathfrak{Z}} c_z(\delta(z))$$

should be allocated to the different sections $z \in \mathfrak{Z}$ and how the 'benefit', i.e. the waiting time $W(T)$, spreads over the stations of the network.

An algorithm for the approximate solution of this multi-criteria optimization problem is presented below in Section 5.

3 Stability of Time-Tables Under Delays

3.1 Scheduled vs. actual Times

A particular focus of our research is on the stability of time-tables. If we optimize time-tables only with respect to their *scheduled* travel or waiting time, we shall end up with time-tables that are highly synchronized but have only small time buffers at stations. These time-tables may turn out to be very instable in real operation as small delays seem to be inevitable in complex networks. But then, the resulting real travel and waiting times may be much larger than the scheduled ones due to missed connections. It is therefore important to take small delays into account when designing time-tables (whereas untypically large delays caused e.g. by accidents shall not be considered here). We can incorporate this aspect into our approach by defining a suitable cost function, e.g.

$$M(T) = \text{mean travel time under delays.}$$

Minimizing $M(T)$ or minimizing the scheduled travel time $R(T)$ and the difference $M(T) - R(T)$ would result in time-tables that have the additional quality of 'stability'. It could also be of interest to examine the *variation* of travel times that occur during a day. Typically, one would expect that large waiting time (= large time buffers) correlates with high stability. Then a cost-benefit-analysis including $M(T)$ would show how much stability can be gained e.g. by investing or by increasing the waiting time, see Goverde (1999) for a result on time buffers for a single isolated connection. Another target would be to examine the effect different 'waiting rules' (stating how long a train has to wait for its delayed feeder train) have on the delays and the waiting times.

To be able to calculate $M(T)$ one has to model the typical small operational delays on lines and at stations, their propagation through the network by the waiting rules and their absorption by time buffers. More precisely, one has to know the joint probability distribution of the delays in the whole network at every point in time. This is an extremely complex stochastic process which at present cannot be handled analytically.

Therefore, simulation of the mean delays seems to be the only way, see e.g. Suhl and Mellouli (1998) for a slightly different context. In our system however, the delays are a cost function which has to be evaluated over and over again. Exact simulation of the complete network is too time-consuming to be included into our system at present.

Instead, we are extending the analytical model of local delays on a section to simple tree-like (sub)networks. We intend to derive a fast approximate macroscopic simulation of the whole network using analytical representations of its subnets. In this program we have achieved a major step by exploring analogies between the accumulation of delays on a single section and the operation of a queuing system which is explained in the next Subsection.

3.2 Modeling Delays in Simple Nets

We start with a simple model of external disturbances and possible reactions to it. We assume that along lines, perturbations occur randomly at places $O_i, i = 1, 2, \dots$ and cause a sudden stop of length $Z_i, i = 1, 2, \dots$. As long as the train is in time, it travels at the scheduled speed of say a km/h. As soon as it is delayed, the driver turns to the maximal speed b at which he drives until the train is in time again or arrives at the next station. We neglect all braking or acceleration processes. See Fig. 5 for a possible place-time-diagram between stations S and S'. Here, perturbations occur at places O_1, O_2, O_3 causing delays of random amounts Z_1, Z_2, Z_3. If delayed, the train runs at increased speed b, indicated by bold lines. δ indicates the scheduled arrival time at S', ε is the arrival delay.

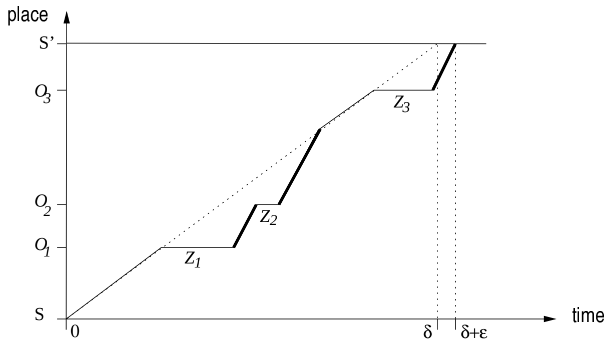

Fig. 5. A typical place-time diagram with random disturbances

$c := 1/a - 1/b$ denotes the possible rate of delay reduction, i.e. $c \cdot s$ are the minutes of delay that can be made up for on a section of length s km. Hence the accumulation of delay along a section may look like Fig. 6.

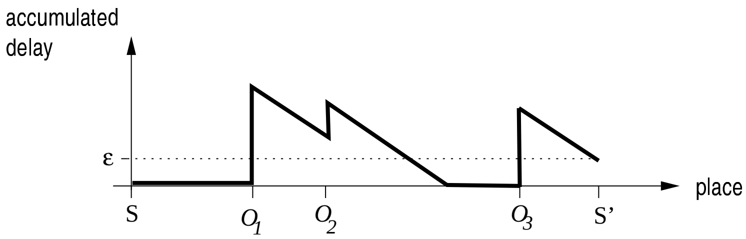

Fig. 6. The delays accumulated on a trip as in Fig. 5

There is a strong analogy to queueing theory in which customers arrive at random points of time and require a random amount of service time from the server. Here the total load of service time D_t lying ahead of the server at time t (the so-called virtual waiting time) has the same profile as the accumulated delay. In fact the train can be viewed as a server that serves (reduces) all requests (perturbations) with a service rate corresponding to c. Therefore results on viritual waiting time can be used to derive the distribution of the delay accumulated on a section.

We assume that the places $(O_i)_{i\geq 1}$ at which the perturbations occur form a Poisson process with rate λ, that the amounts $(Z_i)_{i\geq 1}$ of delay they cause are i.i.d., exponentially distributed with parameter μ and that both processes are independent. Then the accumulated delay $D_{S'}$ acquired along a section from S to S' approximately has the distribution

$$\mathbf{P}(D_{S'} \leq t) \;=\; \begin{cases} 1 - \frac{\lambda}{c\mu} & \text{if } t = 0; \\ \frac{\lambda}{c\mu}(1 - \mathrm{e}^{t(\mu - \lambda/c)}) & \text{if } t > 0 \end{cases} .$$

Note that $1 - \lambda/c\mu$ is the probability of arriving in time and $1 - e^{-t(\mu-\lambda/c)}$ is the conditional distribution of the delay given the train is delayed. A similar result holds under more general assumptions on the (O_i)-process, see Engelhardt-Funke and Kolonko (2000) for details.

Moreover, if the train leaves station S with a departure delay of random amount B_S then this delay can be reduced at rate c during the 'idle time' of the server which has approximate duration of $s(1 - \frac{\lambda}{c\mu})$ if s is the length of the section. Hence the arrival delay at the next station will be approximately

$$D_{S'} + [B_S - sc(1 - \frac{\lambda}{c\mu})]^+.$$

Assuming that all perturbations are independent, this scheme can be iterated to give the delay distributions along a line. The analytically derived distributions have the same structure as those extracted from empirical delay data (see e.g. Mühlhans (1990) and Herrmann (1996)).

We can use this approach also to derive the propagated delays on connecting trains with delayed feeders at least in simple tree-like net structures, see also Weigand (1981).

Note that in more complex structures containing (undirected) circles, delays of consecutive trains may no longer be independent due to feedback effects. Another problem arises from the dependencies caused by the circulation of (delayed) vehicles. These problems are at present beyond our analytical model and remain to be simulated in a future version of our system.

4 The Reduced Internal Network

For an efficient solution of the multidimensional optimization problem sketched in Section 2.4 above, in particular when calculating the waiting times, we have to reduce the network to the data relevant for that calculation (see also Nachtigall and Voget (1997)).

We restrict ourselves to **transfer stations** $\hat{\mathfrak{S}}$ in which passengers can change between different lines and determine all **change-or-stop-relations** (L, L', S) : 'change' for $L \neq L'$and 'stop' for $L = L'$. We also aggregate the sections connecting two transfer stations into a **segment** $\hat{z}$. Note that there may be more than one segment between two transfer stations if there are different routes or if the trains have different speeds and different minimal stopping times on their way. The corresponding local cost functions $c_z(\cdot)$ are then added into one cost function $\hat{c}_{\hat{z}}(\cdot)$ for the segment. This operation requires some care as only favourable combinations of improvements for the aggregated sections should be used. $\hat{c}_{\hat{z}}(\delta)$ then gives the minimal costs to achieve a running time of δ on $\hat{z}$.

The corresponding reduced time-tables $\hat{T}$ only contain departure times $\hat{\pi}(L, S)$ for $S \in \hat{\mathfrak{S}}$ and running times $\hat{\delta}(\hat{z})$ for segments $\hat{z}$. Again, the resulting set of feasible time-tables $\hat{\mathfrak{T}}$ has a very simple structure.

Note that the effort for reduction has to be spent only once at the beginning of the optimization. Internally, the reduced network is a stored as an activity-on-arc-network, with the change-or-stop-relations as activity arcs and the segments $\hat{z}$ with their cost functions as vertices.

After the optimization the reduced time-tables and cost functions have to be 'inflated' again. In particular, the aggregated investments have to be decoded carefully to yield the costs and actions on the original sections.

5 A Solution with an Evolutionary Algorithm

In Nachtigall (1998) it is shown that the solution of the one-dimensional waiting time problem with fixed running times is a very complex periodic optimization problem, see also Nachtigall (1996) and Zimmermann and Lindner

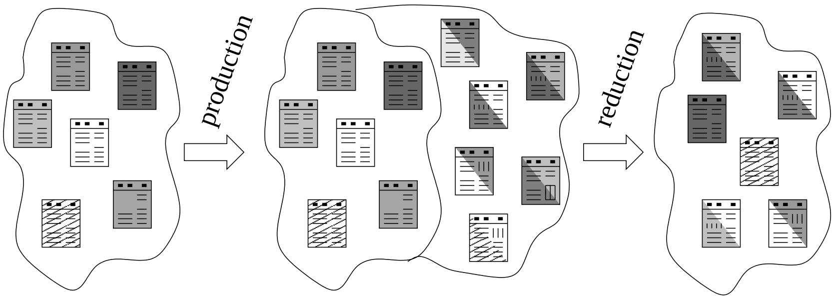

Fig. 7. A production-reduction cycle of the evolutionary algorithm

(2000). Here, we added more cost functions and increased the dimension of the solution space (by introducing the running times) so that the problem becomes far too complex for the methods of exact mathematical optimization. In particular, if we include cost functions like $M(T)$ that can only be simulated, exact methods are excluded.

In our prototype implementation we have successfully applied an evolutionary algorithm for the approximate solution of the multi-dimensional optimization problem. The algorithm is based on a population of time-tables that are chosen randomly at the beginning, see Fig. 7. The population is then enlarged by producing 'offspring' using genetic operators like crossover and mutation. This is indicated by the left arrow in Fig. 7. For the crossover two time-tables are chosen at random from the present population. The two lists of the time-tables are then crossed either with standard operations like one-point or uniform crossover or by more sophisticated methods taking into account the regional structure of the time-table. The resulting time-table is mutated by randomly changing departure and/or runnning times. To make sure that the mutated time-table is still feasible the running times on each section z must be restricted to their respective ranges $[\underline{\delta}(z), \overline{\delta}(z)]$ (see 2.1). The departure times are easy to handle as they are given as offsets to their line periods. Any result of a random mutation may therefore be interpreted as a departure time, possibly after modulo reduction. So the results of crossover and mutation are time-tables from $\hat{\mathfrak{T}}$ again.

These crude stochastic operations are complemented by local search heuristics that may be used to improve the result of the crossover and mutation. Here, the running times on all segments of tracks are increased (within their limits) until the waiting time of the time-table becomes (locally) minimal. An alternative heuristic varies the departure times at all stations so that the waiting time at this station becomes locally minimal. As for these heuristics all segments resp. all departures have to examined, these improvements are very time-consuming compared to the genetic operators.

Invoking crossover, mutation and possibly local improvement repeatedly, a number of offspring time-tables is produced enlarging the present population. This enlarged population will be reduced to its original size by selecting the 'fittest' time-tables, see the step 'reduction ' in Fig. 7. There are different ways to take care of the multidimensional cost function during the reduction, see Ishibuchi and Murata (1996). Particulary successful is a selection procedure that adapts the type of reduction to be used to the present state of the population, see Kolonko and Voget (1998) for details on this. The reproduction-reduction cycle is repeated for a number of generations. Typically the population of time tables tends to improve quite fast (see Fig. 8).

The evolution of the population can be visualized on the screen, see Fig. 8 for a screenshot. The cost function values of the individuals (time-tables) of each population form a cloud in the space spanned by the cost functions. Its lower envelope are the present Pareto-best solutions. They form an approximation to the Pareto-optimal set and the cost-benefit-curve. The visualization can also be used to examine the impact of the different parameter settings of the algorithm.

6 Practical Results with the NASA Network

Our prototype implementation is presently applied to the network of the NASA GmbH in the state of Sachsen-Anhalt, Germany. This network consists of 467 stations and 295 lines (in the sense of our model, see Section 2.1 above). The lines are formed of 551 different sections z and NASA decided that 190 of them could be modernized. The rest of 361 sections already have been renewed or are considered to be of less importance. After reducing the network as described in Sec. 4 there are 1010 different departure pairs (L, S) describing that line L departs from station S. We have 3910 change-or-stop-relations (L, L', S), so there are 3910 reasonable possibilities for passengers to change from line L to line L' in station S or – in case $L = L'$ – to stay in the train of line L during its stop at station S.

The two cost functions of interest to our partners fom NASA are the necessary investment $C(T)$ and the passenger waiting time $W(T)$. Since not all necessary data were available we agreed with our partners to estimate the missing data to get a first analysis on a strategic planning level.

6.1 Local Cost Functions

On each of the 190 sections open for renovation there is only one type of general reconstruction possible that enables the train to run at an increased speed. Hence the local cost function (see Section 2.2) attains only two values because no combinations of different improvements are available. As data we were given the present maximal speed and the maximal speed after a renovation of the section. We assumed that the actual average speed would increase with the same proportion. We then calculated the amount of running time that could be saved on each section from its length, its present running time and the possible increase in average speed.

Our system can cope with quite general cost functions for the improvements on the sections. However, the exact price for such an improvement is not known yet. So NASA suggested to use a linear cost estimator, meaning that reconstructing one km of the track costs one unit of money. Of course this is only a rough estimator but it was considered adequate for now by our partners. From the estimated data the investment costs $C(T)$ are calculated as described above.

Note that in this situation $C(T)$ is proportional to the minimal length of tracks that have to be renewed to enable time-table T, with the restriction, that only complete sections can be reconstructed.

6.2 Waiting Time

There were no data on the number of passengers changing at stations and no $OD-$matrix available. Therefore we used an estimation for the weights $g(L, L', S)$, used in the expression $W(T)$ for the waiting time (see 2.3). This estimator is based on the so-called Lill's law. There are five categories of cities. To each station S its category $\mathrm{Cat}(S)$ is assigned that characterizes its size and economical importance. The average number $m(S, S')$ of passengers travelling from S to S' is then estimated by

$$m(S, S') = \frac{\mathrm{Cat(S)} \cdot \mathrm{Cat(S')}}{\mathrm{distance}(S, S')^2} \cdot \mathrm{const}$$

where distance (S, S') is the length of a shortest path from S to S' in the network. Here, 'shortest' refers to the running times on the tracks from the present time-table (without any investment). The $m(S, S')$ are then collected into an OD-matrix M.

To obtain the weights $g(L, L', S)$ for each change-or-stop-relation (L, L', S), it is counted how many passengers will pass through that relation while following their shortest connection from S to S' as described above. For simplification, we do not take into account a possible split of passenger streams if there are several shortest paths. Also, we neglect a possible feedback effect of time-tables on the routes passengers choose. The weights calculated in this way present a kind of public pressure on the respective change-or-stop-relations. Time-tables should respect them as they also reflect the goal to drive passengers through the network on shortest routes for economical and ecological reasons.

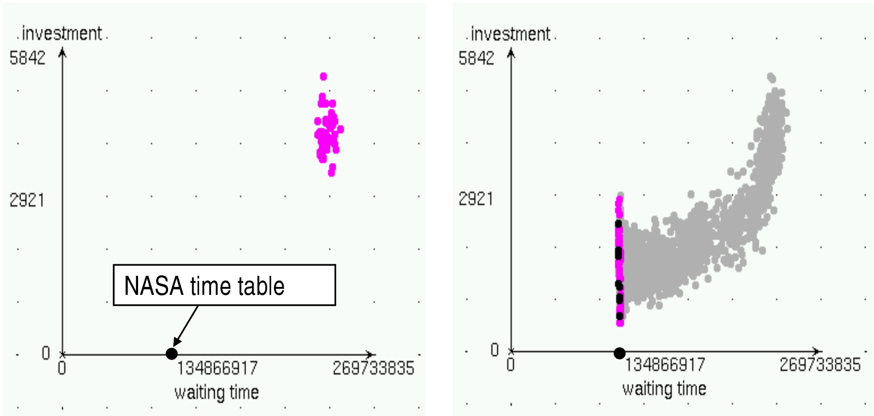

Fig. 8. The evolution of the time-tables

6.3 Optimization and Results

We included the actual NASA time-table (or rather a strictly periodic version of it) into our starting population. The cost function values of a typical starting population are shown in the screen-shots in Fig. 8. In the left picture, the random solutions of the starting population form a cloud in the upper right corner, the present NASA time-table (with investment 0) is included. Naturally, it dominates all random starting solutions. During the optimization the cloud moves towards the lower left corner as the quality of the solutions increases (less costs and less waiting time). The NASA time-table is the only Pareto-optimal solution for a number of generations but it is typically reached by other solutions as shown in the right picture of Fig. 8 after just a few seconds. Here, the dark grey dots represent the present population. Former solutions that have not 'survived' are drawn as light grey dots, whereas the black dots indicate the new Pareto-best solutions, which require some investment but have less waiting time than the NASA time-table.

Note that the waiting time on the $x-$axis has the unit 'passengers × minutes' and includes the penalty factor for waiting on the platform as mentioned in Sec. 2.3. It attains very high values which have no direct practical meaning. They are only used as a relative measure of the quality of time-tables when compared with the present one.

The diagram in Fig. 9 shows the Pareto-best solutions found in several runs with different parameter settings including local improvement operators. The scale of the axis has been adjusted to give a better overview; for comparison with Fig. 8, a screenshot has been added showing the same cost-benefit curve with the scaling as used in Fig. 8. The NASA time-table has been included for illustration though it is dominated by other time-tables. It is shown at the lower right corner of the diagram and has waiting time 94334724 units

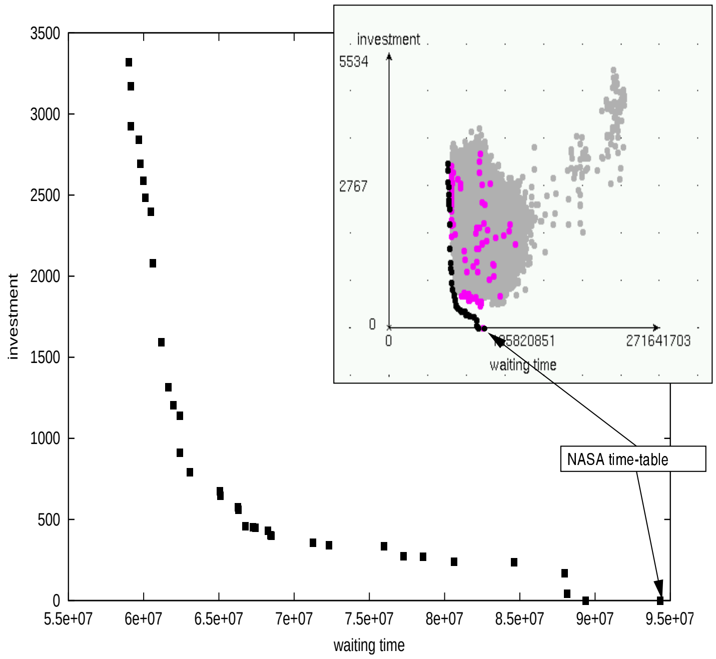

Fig. 9. The lower envelope of the cloud approximates the cost-benefit curve

(and 0 investment). The best solution at 0 investment that our system has found (by adjusting the departure times) has waiting time 89362695, which is an improvement of 5.2 %. Fig. 9 also shows that investments between 100 and 500 units are particularly efficient as the waiting time decreases drastically in this region. On the contrary, investments of more than 900 units seem not very efficient.

One should note however, that a direct comparison between the actual time-table of NASA and our Pareto-best results is difficult, as in contrast to the NASA time-table we do not take into account operational constraints as was mentioned above. Moreover, we had to 'rectify' the NASA time-table at some points to make it strictly periodic.

As the algorithm is highly stochastic, each run gives a different picture. The results of Fig. 8 were obtained within 30 sec on a medium sized Pentium PC. With runs of about 5 minutes we are able to find time-tables that dominate the present NASA time-table, i.e. they have less waiting time without any investment. The results in Figure 9 were obtained from several runs taking a total of about 5 h.

The result as shown in Fig. 9 represents an approximation of the Pareto-optimal solutions. It can be examined in detail using our interactive cost-benefit-analyser. Here, the cost-benefit curve is displayed in a diagram similar

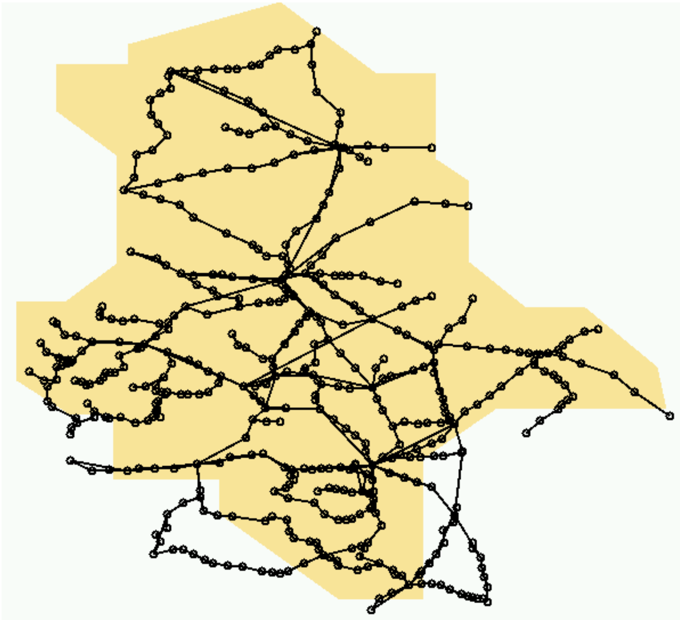

Fig. 10. The interactive cost-benefit-analyzer explains the results of the optimization

to Fig. 9. The map of the network is displayed in a separate window as in Fig. 10. If a point (i.e. a time-table) in the diagram is activated by a mouse-click, the allocation of the investment it requires is displayed on the map by marking the corresponding sections red. In addition, there is a pop-up list of the improvements selected for each section. In a similar fashion it can be shown how the benefit of the time-table, i.e. the reduction of waiting time spreads over the stations of the network. Of course, the time-table and the allocation of investments can be output as text.

7 Future work

Our future work will focus on three points: First, we want to integrate the macroscopic simulation of delays into the optimization to be able to measure the robustness of a time-table under random delays. Secondly, the optimization will be improved incorporating additional local search heuristics like simulated annealing to complement the global search aspect of genetic algorithms. Using simulated annealing in a multidimensional cost environment is a particularly challenging task. Finally, we are investigating how capacity and safety contraints could be integrated into our model. As the present implementation works very fast with a network as that of NASA, we are quite optimistic about the chance to include these aspects into our system.

Bibliography

Engelhardt-Funke, Ophelia and Michael Kolonko (2000). Optimal Time-Tables: Modelling Stochastic Pertubations. Proc. 7. International Workshop on Project Management and Scheduling (PMS 2000), Osnabrück, pp. 125-127.

Goverde, R. (1996). Optimal Transfertimes in Railway Time Tables. in : Abstracts of the 6-th Meeting of the EURO Working Group on Transportation, 1999.

Herrmann, Udo (1996). Untersuchung zur Verspätungsentwicklung von Fernreisezügen auf der Datengrundlage der RZÜ Frankfurt am Main. Dissertation, TH Darmstadt.

Ishibuchi, H. and T. Murata (1996). Multi-objective genetic local search algorithm. *Proc. IEEE Internat. Conf. on Evolutionary Computation,* pp. 119-124.

Kolonko, Michael and Stefan Voget (1998). Multidimensional Optimization Using Fuzzy Genetic Algorithms . *J. of Heuristics* (4), pp. 221 -244.

Mühlhans, Edmund(1990). Berechnung der Verspätungsentwicklung bei Zugfahrten. *Eisenbahntechnische Rundschau* (39), pp.465-468.

Nachtigall, Karl (1996). Periodic Network Optimization with Different Arc Frequencies. *Discrete Applied Mathematics* (69), pp. 1-17.

Nachtigall, Karl (1998). Periodic Network Optimization and Fixed Interval Timetables. Habilitationsschrift, Universität Hildesheim.

Nachtigall, Karl and Stefan Voget (1997). Minimizing waiting times in integrated fixed intervall timetables by upgrading railway tracks. *European Journal of Operations Research* (103), pp. 610-627.

Orlin, James B (1982). Minimizing the Number of Vehicles to Meet a Fixed Periodic Schedule: An Application of Periodic Posets. *Operations Research* (30), pp. 760-776.

Serafini, P. and W. Ukovich (1989). A Mathematical Model for Periodic Scheduling Problems. *SIAM J. Disc. Math.* (2), pp. 550-581.

Suhl, L. and T. Mellouli (1998). Requirement for, and Design of an Operations Control System for Railways. 7^{th} *Proc. int. Workshop on Computer Aided Scheduling of Public Transportation, Cambridge-Boston* Springer 1998

Weigand, W. (1981). Verspätungsübertragung in Fernverkehrsnetzen *Eisenbahntechnische Rundschau* (30), pp. 915-919.

Zimmermann, U. and T. Lindner (2000). Train Schedule Optimization in Public Transportation. preprint available from http://www.math.tu-bs.de/mo/research/preprints.html