Paying Less for Train Connections with MOTIS

Matthias Müller–Hannemann and Mathias Schnee

Darmstadt University of Technology, Computer Science,
64289 Darmstadt, Hochschulstraße 10, Germany
muellerh,schnee@algo.informatik.tu-darmstadt.de,
http://www.algo.informatik.tu-darmstadt.de

Abstract. Finding cheap train connections for long-distance traffic is algorithmically a hard task due to very complex tariff regulations. Several new tariff options have been developed in recent years, partly to react on the stronger competition with low-cost airline carriers. In such an environment, it becomes more and more important that search engines for travel connections are able to find special offers efficiently.
We have developed a multi-objective traffic information system (MOTIS) which finds all attractive train connections with respect to travel time, number of interchanges, and ticket costs. In contrast, most servers for timetable information as well as the theoretical literature on this subject focus only on travel time as the primary objective, and secondary objectives like the number of interchanges are treated only heuristically. The purpose of this paper is to show by means of a case study how several of the most common tariff rules (including special offers) can be embedded into a general multi-objective search tool.
Computational results show that a multi-objective search with a mixture of tariff rules can be done almost as fast as just with one regular tariff. For the train schedule of Germany, a query can be answered within 1.9s on average on a standard PC.

Keywords: timetable information system, multi-criteria optimization, shortest paths, fares, special offers, long-distance traffic

1 Introduction

In recent years, there has been strong interest in efficient algorithms for timetable information in public transportation systems (with emphasis on public railroad systems). For a given customer query, the problem is to find all attractive train connections with respect to several objectives. We concentrate on travel time, number of interchanges, and ticket costs.

Most work has considered optimization subject to a single criterion, namely to find the fastest connection. Such a problem can easily be modeled as a shortest path search in a graph where the edge lengths correspond to travel times. Likewise it is not difficult to extend these graph models so that also the minimum number of train interchanges can be solved as a shortest path problem with $\{0, 1\}$-lengths on the edges.

ATMOS 2005
5th Workshop on Algorithmic Methods and Models for Optimization of Railways
http://drops.dagstuhl.de/opus/volltexte/2006/657

However, difficulties arise with fares as objectives. Pricing systems of railway companies are very complex and actual fares depend on many parameters. In recent years, railway companies faced higher competition caused by the strong increase of low-cost airlines. As a reaction on this development, marketing departments of railway companies answer with the introduction of different types of special offer tariffs. For origin-destination pairs with a low-cost competitor the relation-based prices are occasionally decreased.

For this and several other reasons, the fare of a connection cannot be modeled in an exact way as an additive function on the edges of a graph which can simultaneously be used for a fastest connection search.

Previous work. Two main approaches have been proposed for modeling time table information as a shortest path problem: the *time-expanded* [1,2,3,4,5,6,7], and the *time-dependent* approach [6,8,9,10,11,12,13,14]. The common characteristic of both approaches is that a query is answered by applying some shortest path algorithm to a suitably constructed graph. These models and algorithms are described in detail in a recent survey [15]. The time-expanded model is much more flexible than the time-dependent model. It is therefore preferred if all side constraints of a real-world scenario have to be respected.

As mentioned above, most of the cited papers consider fastest connections only. Multi-criteria search for train connections in a fully-realistic environment has been studied in [7]. The latter paper already used a simplified model to search for regular fares. Apart from initial work in [7], we are not aware of any previous work which takes fares as an optimization criterion into account.

Contribution of this paper. Usually, marketing experts design a new tariff with respect to expected sales but without considering how such an offer can be searched for in an efficient way. It seems that Germany has one of the most complicated tariff systems of the world, providing us with the most challenging task to find cheap connections systematically.

In this paper, we analyze the different tariff options with respect to searchability. We show that a systematic, simultaneous search for different tariffs can be integrated into a suitable graph model and a generalized version of Dijkstra's algorithm.

In particular, we focus on tariff options which are based on the availability of contingents, yielding either a fixed price or a certain discount.

Currently we develop the information server MOTIS (multi objective traffic information system) in cooperation with datagon GmbH, Waldems, Germany. The main features of MOTIS are the following:

– It contains a Dijkstra-based multi-objective search algorithm (travel time, number of interchanges, ticket costs).
– It provably yields *exact* minimization of travel time and number of interchanges. In contrast, the electronic timetable information system HAFAS [16], which is used by many European railway companies provides only heuristic solutions.

– It delivers many attractive alternatives by using the concept of relaxed Pareto optimality [7].
– MOTIS is extensible to add further criteria like the possibility of seat reservation or to incorporate safety margins for train changes in case of delays.

An extensive computational study shows that the computational cost increases only very slightly when we combine the search with respect to regular fares and other tariff options. On average, a query can be answered within 1.9s on a standard PC.

Overview. The rest of the paper is organized as follows. In Section 2, we first give a brief description of MOTIS. Afterwards, in Section 3, we present a systematic overview on fare regulations. For each tariff class we analyze the algorithmic consequences for efficient searchability of connections which fall into this class. Thereafter we explain more details on the search algorithm of MOTIS in Section 4. Then, we provide computational results based on a large test set of real customer queries. Finally, we conclude with a summary and directions for future work.

2 The Information Server MOTIS

This section is intended to give a brief introduction to MOTIS and the main ideas behind it. In the following subsections we first explain what kind of queries can be handled, and define what we understand by "finding all attractive connections". Then we briefly touch upon the graph model used and the general search algorithm.

2.1 Queries

A *query* to a timetable information system usually includes:

The (start or) *source station* of the connection, the *target station* and an *interval* in time in which either the departure or the arrival of the connection has to be, depending on the *search direction*, the user's choice whether to provide the interval for departure ("forward search") or arrival ("backward search"). Additional query options include:

Vias and duration of stay. A query may contain one (or more) so called *vias*, stations the connection has to visit and where at least the specified amount of time can be spent, e.g. from Cologne to Munich via Frankfurt with a stay of at least two hours for shopping in Frankfurt.

Train class restrictions. Each train has a specific *train class* assigned to it. These classes are high-speed trains such as the German ICE and French TGV; ICs and ECs; Interregios and the like; local trains, "S-Bahn" and subway; busses and trams. The *query* may be restricted to a subset of all *train classes*. Certain train tariffs exclude some of the higher-valued train classes. Hence, by excluding high speed trains one can search for special tariffs.

Attribute requirements. Trains have *attributes* describing additional services they provide. Such attributes are for example: "bike transportation possible", "sleeping car", "board restaurant available". A user can specify attributes a connection has to satisfy or is not allowed to have. We allow Boolean operators for specifying *attribute requirements* like: (a restaurant OR a bistro) AND bike transportation.

Passenger related attributes. Additional attributes are relevant for the fare calculation. One has to choose the desired *comfort class* (i.e. first or second class). In order to determine possibilities for discounts, the query has to provide the number of passengers, and for each passenger the type of discount card which is available (if any). Families with children also have to specify the age of each child.

2.2 Attractive Train Connections

A simple measurement for the "attractiveness" of a connection does not exist. Different kinds of costumers have differing (and possibly contrary) preferences. Key criteria for the quality of a connection are travel time, ticket cost and convenience (number of interchanges, comfort of the used trains, time for train changes). In order to build a traffic information system that can provide attractive connections we avoid the drawbacks of weighted target functions or "preference profiles". Instead we want to serve each possible costumer by presenting him a selection of highly attractive alternatives with one single run of the algorithm.

When dealing with multiple criteria a standard approach is to look for the so-called Pareto set. For two given k-dimensional vectors $x = (x_1, \ldots, x_k)$ and $y = (y_1, \ldots, y_k)$, x *dominates* y if $x_i \leq y_i$ for $1 \leq i \leq k$ and $x_i < y_i$ for at least one $i \in \{1, \ldots, k\}$. Vector x is *Pareto optimal* in set X if there is no $y \in X$ that dominates x. Here, we assume for simplicity that all cost criteria shall be minimized. In our scenario we compare 3-dimensional vectors (travel time, ticket costs, number of interchanges) for our connections.

We argued in [7] that the set of Pareto optima still does not contain all attractive connections and proposed to apply the concept of *relaxed Pareto optimality.* It provides more alternatives than Pareto optimality can give. Under relaxed Pareto dominance

- connections that are nearly equivalent but differ slightly do not dominate each other;
- the bigger the difference in time between start or end of two connections the less influence they have on each other;
- traveling longer needs to yield a fair *hourly wage* (i.e. the amount of money saved divided by the extra time in hours) to make a cheaper alternative attractive. The latter also excludes irrelevant Pareto optima.

We used the following rules to compare connections A and B which have departure times d_A, d_B, arrival times a_A, a_B, travel times t_A, t_B (all data given in minutes), i_A, i_B interchanges and associated costs c_A, c_B in Euros, respectively. Connection A *dominates* connection B

– with respect to the criterion travel time if B does not overtake A and

$$t_A + \alpha(t_A) \cdot \min\{|d_A - d_B|, |a_A - a_B|\} + \beta(t_A) < t_B,$$

where, $\alpha(t_A) := t_A/360$ and $\beta(t_A) := 5 + \sqrt{t_A}/4$;
– with respect to the number of interchanges only if $i_A < i_B$;
– with respect to the cost criterion only if

$$c_A + \frac{t_A - t_B}{60} \cdot \Delta < c_B,$$

where the required hourly wage Δ is set to 5 Euros.

2.3 Time-Expanded Graph Model

The basic idea of a so-called *time-expanded graph model* is to introduce a directed search graph where every node corresponds to a specific event (departure, arrival, change of a train) at a station.

A connection served by a train from station A to station B is called *elementary*, if the train does not stop between A and B. Edges between nodes represent either elementary connections, waiting within a station, or changing between two trains. For each optimization criterion, a certain length is associated with each edge.

Traffic days, possible attribute requirements and train class restrictions with respect to a given query can be handled quite easily. We simply mark train edges as *invisible* for the search if they do not meet all requirements of the given query. With respect to this visibility of edges, there is a one-to-one correspondence between feasible connections and paths in the graph.

More details of the graph model can be found in [7].

2.4 The Search Algorithm in MOTIS

Our algorithm is a "Pareto-version" of Dijkstra's algorithm using multi-dimensional labels. See Möhring [2] or Theune [17] for a general description and correctness proofs of the multi-criteria Pareto-search.

Each label is associated with a node v in the search graph. A label contains key values of a connection from a start node up to v. These key values include the travel time, the number of interchanges, a ticket cost estimation and some additional information. For every node in the graph we maintain a list of labels that are not dominated by any other label at this node. Every time a node is extracted from the priority queue, its outgoing edges are scanned and (if they are not infeasible due to traffic days, attributes and train class restrictions etc.)

labels for their head nodes are created. Such a new label is compared to all labels in the list at the head node. It is only inserted into that list and the priority queue if it is not dominated by any other label in the list. On the other hand, labels dominated by the new label are removed.

As a further means of dominance we keep a short list of Pareto-optimal labels at the terminal station and compare each new label to these labels. To compare labels at an intermediate node v with a node at the terminal, we use lower bounds on the key values of a shortest, a most convenient, and a cheapest path from v to the terminal station. We increase the criteria of the label at v by lower bounds on the according values. If the label with its increased values is dominated by any label at the terminal, it is excluded from further search.

To make lower bounds available, we determine a guaranteed fastest connection from source to target using a goal-directed single criterion search in an initialization phase before the actual multi-criteria search. This search is by orders of magnitude faster than the multi-criteria search and can be performed in less then 50ms on average.

2.5 Black-Box-Pricing Component

As noted in the introduction, the fare regulations are extremely complex. Furthermore, the system undergoes rapid change. Therefore, it is reasonable to have a *black-box pricing component (BPC)* that can be used to calculate the exact ticket cost for some connection. Unfortunately, one call to this black-box routine is very costly. Hence, it is impossible to calculate the correct price for every label and achieve a bearable running time.

As a consequence, we use fast to compute price estimates in the labels that are updated during the search. To this end, we associate an estimated base fare with each travel edge in our search graph. (How we derive these estimations will be described in more detail in Section 3.1.)

This simplified model provides helpful estimates for the search. In order not to loose low cost connections due to this approximation we need a safety margin which is incorporated into the corresponding relaxation function for the relaxed Pareto dominance. After a search is completed, all connections are correctly priced by the BPC and relaxed Pareto dominance can be applied to true fares.

3 Modeling Regular Fares and Special Offers

The purpose of this section is to provide an overview on the many different classes of tariffs commonly used by train companies.

As the number of different tariffs being in use is very large, tariffs differ considerably from country to country, and they are subject to frequent changes, this overview is far from being comprehensive. However, we try to group the most commonly used tariffs into certain classes. For each tariff class, we analyze how a search for connections which fall under this class can be modeled and incorporated into our general framework of MOTIS.

In some rare cases it might be profitable to partition the desired connection into smaller connections. To each partial connection a different tariff option may apply, yielding an overall saving if several tickets are bought. However, this is very impractical and potentially confusing for the customer. In this paper, we therefore restrict our discussion to a single tariff for each connection. [1]

3.1 Regular Fares

Regular fares apply at any time to everyone without any restrictions. To calculate regular fares, two main principles are in use: distance-based and relation-based fares.

Distance-based fares. For this type, regular fares are modeled by piecewise affine-linear functions which depend on the number of kilometers of the connection and the used train classes. These functions are encoded in tables and the calculation of fares is done with a table look-up. For example, regular fares in France (SNCF) follow this scheme. [2]

Relation-based fares. For long-distance travel in a highly connected network like that of Germany the regular fare is more often based on relations, i.e. origin-destination pairs associated with a regional corridor. The corridor of a relation describes what is considered as a common route. A relation can only be applied to a connection if the connection passes stations from a relation-specific set which specifies the corridor.

If a connection leaves the corridor of a relation, the fare has to be determined by partitioning the entire connection into smaller connections. The details of this procedure are beyond the scope of this paper.

Marketing considerations influence the price for each relation. In general, the fare of a relation is derived from the travel distance, but it may be changed for marketing reasons in either direction.

Properties of regular fares. In most cases, we can assume that regular fares are monotonously increasing and subadditive. That is, for a connection c from station s to station t via station v, the price $p_c(s,t)$ satisfies

$$p_c(s,t) \leq p_c(s,v) + p_c(v,t).$$

Distance-based fares are degressive functions in the travel kilometers. Hence, they are always strictly subadditive.

In dominance tests, good lower bounds are of crucial importance for the efficiency of the search. Hence, we need a lower bound on the price of a connection. With distance-based fares, we get a lower bound on the distance of a connection

[1] Note that a combination of tariffs is necessary in multi-vendor systems.

[2] `http://www.voyages-sncf.com/info_resa/guide_du_voyageur/Calcul_PT.htm`

from the distance traveled from s to v plus a lower bound on the distance from v to t.

In sharp contrast, valid lower bounds are hard to obtain for relation-based fares as these may even violate our subadditivity assumption. But even if we assume subadditivity, it is not clear how to get a lower bound on the price of a connection from s to t given the prices from s to v and from v to t.

Frequent user cards. For holders of frequent user cards (like "BahnCard") a general $x\%$ discount applies to the regular fare. As this kind of discount yields the same reduction rate for all connections, our price estimation merely needs a flag indicating whether such a card is available or not. Such a flag is necessary for a comparison with other tariff options.

Approximation of regular fares. We use a very simple but efficiently computable model to approximate regular fares. Basically, we simulate a distance-based fare and associate a travel distance with each edge. The distance between the two stations of a train edge is taken as the straight line distance obtained from the coordinates of the stations. During the search, we add for each train edge the travel distance times a constant factor (in Euros/km) depending on the train class used. If true regular fares are based on relations, we have to incorporate relatively large safety margins in order not to loose too many attractive connections.

3.2 Surcharges

An additive surcharge applies to certain trains (night trains, ICE sprinter) or train classes (IC,EC). It has to be paid once, if such a train is used. If a connection uses several trains to which a surcharge applies, then usually only the highest surcharge has to be paid once.

During the search, the amount of the surcharge is added to the price estimation when a partial connection first enters a train with a surcharge. In order to guarantee that a surcharge is paid only once, the labels characterizing a partial connection store in flags which surcharges have already been applied.

3.3 Contingent Based Discount Fares

Contingent-based offers are intended to increase the average passenger load on high-speed trains. For each train in a connection for such an offer, a contingent of available seats must not be exceeded by previous bookings. For high-speed trains the contingent may be something like 10% of all seats. For local trains, there is typically no contingent restriction, i.e. the contingent is regarded as being unlimited. As a consequence, such offers are only valid for connections which contain at least one contingent-restricted train.

Many train companies offer discounted fares on long-distance travel under certain restrictions. These restrictions typically include that

– the ticket has to be bought a certain time in advance (for example, at least three days in advance);
– passengers restrict themselves to a particular day and a certain connection which has a contingent available;
– passengers make a return journey to and from the same station.

Discount rates may also be subject to weekend restrictions. For example, Deutsche Bahn AG offers "Savings Fare 50" ("Sparpreis 50") only if the following restrictions apply: For trips starting from Monday to Friday, the return trip cannot be any sooner than the following Sunday. If you travel on Saturday or Sunday you may return that same day.

To incorporate such types of offers into the search, we add and maintain a contingent flag in our labels. The *contingent flag* is a Boolean flag which is set to true if and only if all previous train edges of this connection have a contingent available.

3.4 Fixed Price Offers

Contingent-Based Restrictions. Certain special tariffs offer fixed price tickets within a limited time period (of several weeks or even months, like "Summer Special") subject to the availability of contingents.

A further restriction is that the itinerary of a connection from station A to B must use a "common route". This rule is to prevent from possible misuse by making round-trips or stop-overs during the travel for which one usually would have to buy several tickets or at least to pay for the deviation.

The easiest way to model common routes is to impose the restriction that the length of an itinerary of a connection has to be at most a certain percentage, say 20%, longer than the shortest route from A to B. Alternatively, the travel time should not be more than a certain percentage longer than the fastest route from A to B.

The modification of our model for this kind of tariff is similar to the previous case. We also maintain a contingent flag in each label indicating whether a contingent has been available on all previous edges. As contingents for discounts and for fixed prices may be different, we use different kind of contingent flags. At each intermediate station, we also check whether the partial connection up to this station can still be extended in such a way that it stays on a "common route". To this end, we use lower bounds for the remaining path from this intermediate station to the final destination.

Time Interval Restrictions. Tickets allowing unlimited travel may be available for a fixed price provided the time of the trip falls into a certain time interval.

For example, Deutsche Bahn AG offers a "Happy-Weekend-Ticket" which can be used on all trains except high-speed trains on Saturdays or Sundays between 12 a.m. until 3 a.m. of the following day for a fixed price. Another example would be a fixed price ticket valid from 7 p.m. until the end of the same business day ("Guten-Abend-Ticket").

Such offers can be handled in the following way. For a given query, we first check whether the given start interval falls into the interval of a special offer. If not, the corresponding tariff is definitely not applicable. If the offer has no train class restrictions, we can use the standard multi-objective search. For each alternative found by this search, we finally have to check whether the complete connection falls into the time interval. If this is the case, the price for this connections is the minimum of the regular fare and the fixed price.

If train class restrictions apply, we could use two independent searches, one with train class restriction and one without. However, it is more efficient to treat train class restrictions as a further criterion in the multi-criteria search and to run just a single simultaneous search for both cases.

Rail Passes. Many train companies also offer different kinds of so-called *rail passes* which allow unlimited travel. Prices depend on country and number of days. Rail passes may be restricted to special user groups (students, disabled, unemployed), restrictions may be based on the age (children, seniors), or restrictions on the place of permanent residence apply.

Further restrictions may be imposed on the set of allowed train classes. For example, a regional rail pass like "Hessenticket" offered by Deutsche Bahn AG is only valid for local trains.

Passengers with rail passes can use the standard multi-objective search on the basis of regular fares which delivers, in particular, all attractive connections with respect to travel time and convenience. The price information can simply be ignored. The search has only to make sure that the whole connection lies within the region where the rail pass is valid.

3.5 Discounts for Groups

Groups of 2 or more passengers either get an $x\%$ discount on the regular tariff which can be applied to all trains, or they get an even larger discounts of $y > x\%$ based on the availability of certain contingents. During the search, both options can be handled in the same way as for single passengers.

3.6 Further Possibilities for Discounts

Discounts for single passengers or groups may also be restricted to certain Boolean conditions which depend only on properties of the travelers but not on the particular trip they are going to make. For example, if the group is a family with children below a certain age, then special discounts apply. Another example would be discounts for employees of certain companies (corporate clients).

4 More Details on the Search Algorithm

4.1 Simultaneous Search

The aforementioned modeling of the various tariffs allows the search for combinations of tariffs simultaneously. This is preferable over having individual searches for each of the tariff rules that apply in a scenario and - as we will show in the subsequent section - can be done without sacrificing search speed.

However, as the number of tariff rules increases, more and more labels become mutual incomparable. For example, consider two labels representing partial connections that can gain a fixed price or discounted fare, respectively. Either connection might not be extendable to a connection from source to target with contingents available on all edges. So neither of them can dominate the other depending on an estimate of the special price. Furthermore, they cannot even be compared regarding the estimation for the regular price, as the final price may differ substantially if a special tariff is applicable.

The dominance test between a connection that has already reached the terminal station and a partial connection has to compare the lowest possible price reachable by extending the partial connection to the actual price of the complete connection. So it is even more important to have a fast and cheap connection at the terminal fairly early in the search process (compare Section 4.2).

4.2 Fast Search for the Fastest Fixed Price Connection

For several reasons we implemented a specialized version of our algorithm to search for fixed price connections. Our motivation was

1. to have a stand alone tool to find one fixed price connection, and
2. to strengthen our dominance with terminal labels, or
3. to have a certificate that no fixed price connection is available at all. In the latter case, we can turn off our fixed price search.

Our specialized algorithm for fixed price search ("fixed price Dijkstra") is a single-criterion goal-directed search algorithm. It determines a fastest connection among all connections using only available contingent edges and edges without contingent restrictions.

4.3 Determining Lower Bounds in the Preprocessing Phase

The initialization phase now consists of up to two searches: First we use the standard single-criterion goal-directed search algorithm to determine a fastest connection from source to target. It keeps track of the contingent information and

– either finds a connection with a fixed price (it includes a high-speed train and contingents are available on all contingent edges),

number of stations	8,861
number of trains	45,370
number of high-speed trains	1,006
number of nodes	1,427,726
number of edges	2,395,703

Table 1. Size parameters of the time-expanded graph.

– or finds a connection without high-speed train (therefore no fixed price is possible for it). As it is the fastest connection, we may use it for dominance testing later on. It is also quite often cheaper than the fixed price (see Section 5.4).
– Otherwise, it triggers the specialized algorithm for fixed price search.

If triggered, the "fixed price Dijkstra" algorithm

– either finds a connection with a fixed price (it includes a high-speed train, contingents are available on all contingent edges, and it is within the allowed margin (here 20% more travel time) compared to the fastest connection),
– or finds a connection without high-speed train (therefore no fixed price is possible for it). If a fixed price connection exists, it must be slower than this connection. Such a connection is also quite often cheaper than the fixed price (see Section 5.4) and therefore very useful for later dominance testing.
– Otherwise it finds a connection with contingents available on all contingent edges but that does not stay within the allowed margin. In this case no fixed price connection exists (as all other connections with contingents available are even slower).

In the latter case the following multi-criteria search is performed with the option to search for fixed price connections turned off. Note, that the algorithm sometimes fails to compute a connection with a fixed price although one may exist. However, it delivers an alternative connection for dominance testing that is faster than any fixed price connection, if there are any, and in most cases cheaper than the fixed price (see Section 5.4).

5 Computational Results

5.1 Test Cases

We took the train schedule of trains within Germany from 2003. For our experiments, we used a snapshot of about 5000 real customer queries of Deutsche Bahn AG falling within the week January 13-19, 2003. For all queries, we searched for valid connections within a two-hours time interval. This schedule and the derived time-expanded graph have sizes as shown in Table 1.

Ticket contingents exist for high-speed trains (like ICE, Thalys, TGV, IC, EC) or night trains. Each train t has a certain capacity $cap(t)$ (depending on the

scenario	average CPU time in msec.	average extract min operations	average # of Pareto optima	average # of relaxed Pareto optima
MOTIS	1,702	169,114	3.93	7.26
C10	1,889	176,861	4.22	8.05
C20	1,839	175,221	4.13	7.92
C40	1,776	170,976	3.90	7.73
C60	1,734	167,114	3.67	7.46
C80	1,676	161,446	3.43	7.32
C100	1,605	155,219	3.19	7.06

Table 2. Computational results for simultaneous search of several tariff types (minimum of regular fare, contingent-restricted special offer and contingent-restricted 50% discount.)

train type). We do not have access to real pre-booking data for trains. Therefore, we simulate the booking status for each train.

A random number of passengers uses each train with contingent restrictions. This number is based on the train class and some other criteria (number of stops, importance of the served stations, etc.). For each of the passengers a random station for entering and leaving the train is chosen evenly distributed from the stations the train visits. We then set thresholds $x_A(t)$ for the number of passengers required to exhaust the contingent on a train edge of train t according to the desired level of availability $A = x\%$. A travel edge which may have a contingent restriction is called *contingent edge*. For two availabilities A, A' with $A < A'$ we require $x_A(t) \geq x_{A'}(t)$ for all trains t. So the contingent edges that are not available for some availability A are not available for every availability $A^* < A$.

We consider the following scenarios for the availability of contingents: C10, C20, C40, C60, C80 and C100, where Cx has an availability of $A = x\%$ on the contingent edges. For comparison we also include the numbers for the search for regular fares (denoted by MOTIS).

For all queries, we assume the same type of passenger, namely a single adult booking early enough to get a 50% discount if a contingent is available. The fixed price for special offers is assumed to be 29 Euros.

5.2 Computational Environment

All computations are executed on a standard Intel P4 processor with 3.2 GHz and 4 GB main memory running under Suse Linux 9.2. Our C++ code has been compiled with g++ 3.x and compile option -O3.

5.3 Searching for Multiple Tariffs

In the following, we compare computational results for running our code with regular fares only (this version is called MOTIS in the following) and a simultaneous search of several tariff types for different scenarios of available contingents.

In the simultaneous search, we finally select the relaxed Pareto-optimal connections where the fare is taken as the minimum of the regular fare, a contingent-restricted special offer and a contingent-restricted 50% discount on the regular fare if the contingent is available. Table 2 summarizes the key figures obtained in our experiment. In the first column of numbers we present the average CPU time in milliseconds for a single query. The average CPU running time lies within the relatively small range of $1.6s$ and $1.9s$ for all scenarios.

As CPU times are very hardware-dependent, we prefer to add representative operations counts for the performance evaluation of algorithms. Previous studies [7] indicated that a suitable parameter for operation counts of a multi-criteria version of Dijkstra's algorithm is the number of extract minimum operations from the priority queue. This parameter is highly correlated with the CPU running time for the corresponding query. Therefore, we display in the second column of numbers in Table 2 also the average number of these extract operations.

The computational effort increases with decreasing availability of contingents mainly due to two reasons: On the one hand, very few available contingent edges force the algorithm to take longer detours to find cheap contingent prices. On the other hand, a high availability of contingent edges leads to many cheap connections. These help in dominance. There are actually less connections to explore to find cheap alternatives. If about half or more of the contingent edges are available, the contingent version has less operations than the version MOTIS not considering different tariffs.

We note that dominance rules are faster to evaluate if only regular fares are considered (case MOTIS) as less connections are mutually incomparable, see Section 4.1. Therefore, the workload per extract minimum operation is smaller in this version. For all versions using contingent information the correlation between running time and number of extract min operations is plain to see.

In Figure 1, we also show a histogram on the distribution of extract minimum operations. Case MOTIS mostly lies between the easiest (C100) and most difficult (C10) contingent scenario. The overall distribution looks very similar for all versions of our algorithm. It turns out that about half of all test cases require less than 50,000 extract operations. Such queries are very easy and take only a few milliseconds.

The two remaining columns of Table 2 display the average number of true Pareto optima and the number of relaxed Pareto optima, respectively. These numbers are visualized in Figure 2.

MOTIS offers about 7-8 attractive connections on average, i.e. about four additional connections in comparison to standard Pareto filtering. The more contingents are available, the smaller is the number of Pareto optima, since more fast connections have a cheaper price.

Figure 3 shows the distribution of the number of Pareto optima and relaxed Pareto optima over the test cases for MOTIS and the most difficult contingent version C10.

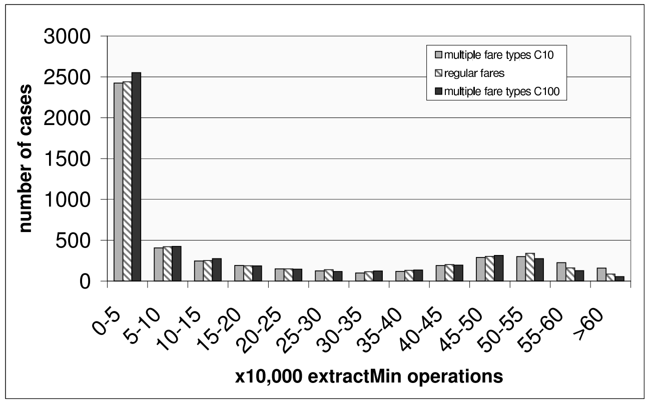

Fig. 1. Histogram showing the distribution of the number of extract min operations from the priority queue. We compare MOTIS (search only for regular fares) with a new version which simultaneously searches for a mixture of fare types.

5.4 Fast Search for Fixed Price Connections

We also evaluated the results of the preprocessing phase with our test set. In this experiment, we have run the subroutines "fastest travel time Dijkstra" (FTTD) and our "specialized fixed price Dijkstra" (SFPD). Recall that the purpose of these routines is to find either a fixed price connection, a suitable connection for dominance testing or a certificate, that no fixed price connection exists.

Table 3 shows the average running time, the number of calls to the SFPD, the number of different types of connections and the number of certificates that no fixed price connection exists. The connections are either fixed price connections

scenario	average CPU time in msec.	# calls to SFPC	# fixed price conn. from		# certificate no fixed price conn. exists	# non high-speed conn.	
			FTTD	SFPD		total	too expensive
C10	204	3641	82	317	2790	1811	373
C20	153	3502	221	841	2224	1714	321
C40	111	3101	622	1490	288	2450	256
C60	90	2579	1144	1742	194	1920	216
C80	70	1534	2189	1275	59	1477	171
C100	45	0	3723	-	0	1277	152

Table 3. Results for the fast search for fixed price connections. Either a fixed price connection was found, a certificate that no fixed price connection exists was computed, or a non high-speed connection was found which is cheaper than the fixed price in most cases.

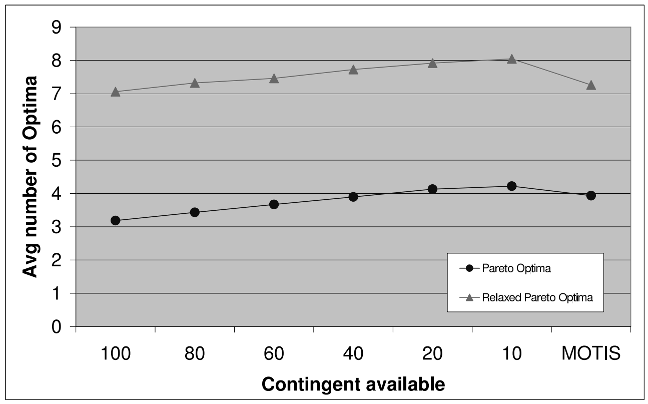

Fig. 2. Distribution of the average number of Pareto optima and relaxed Pareto optima for different scenarios of contingent availability.

found by either of the algorithms or non-high-speed connections. In the last column we give the number of cases where such a non-high-speed connection was more expensive than the fixed price. These cases are the only ones, where we have neither a connection to use in dominance testing (either a fixed price connection or a connection without high-speed train that is faster than any fixed price connection) nor the knowledge that no fixed price connection exists. This only happens in 152 to 373 cases, which is 3.04% to 7.5% of the cases, depending on the availability of contingent edges. This is acceptable for a heuristic that runs in at most a fifth of a second on average.

Not surprisingly the total number of fixed price connections increases with the availability of contingents. With decreasing availability the running time, the number of calls to the SFPD and the number of certificates that no fixed price connection exists increase. As the availability of contingent edges increases, the number of fixed price connections determined by the FTTD increases and the number of calls to the SFPD decreases, therefore the running time improves. The number of fixed price connections SFPD determines increases with the availability but decreases if many fixed price connections have already been found by FTTD.

Fixed price search in MOTIS becomes harder the less contingent edges are available (as more detours have to be investigated). Fortunately, with decreasing availability of contingents the number of queries increases significantly for which we can turn off the tariff option fixed price search in the multi-criteria search due to the preprocessing phase.

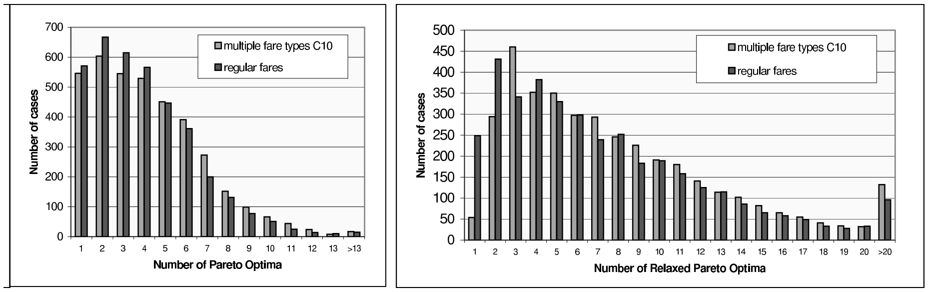

Fig. 3. Histogram showing the distribution of the number of Pareto optima and relaxed Pareto optima.

6 Conclusion and Outlook

The focus of this paper was to demonstrate how a large variety of different tariff classes can be incorporated into a multi-objective shortest path framework for travel information. We successively integrated a combined search for regular tariffs and contingent-based tariffs into MOTIS. In our computational experiments we observed that the computational cost of this advanced search increases only slightly over the regular fare search. Sometimes the contingent-restricted versions run even faster. The computational time for a query is less than 1.9s on average. This is significantly more than for a single-criteria search, and further speed-up is desirable.

We also observed that our simple model to represent regular fares within Germany is not as accurate as desired. Hence, future work should concentrate on improved approximations of regular fares. A tighter approximation would allow stricter dominance rules. We do expect significant savings of computational time from stricter dominance rules.

Within this paper we did not consider a specialized search for night trains. Night train search differs from ordinary search in one of the main objectives. A night train passenger typically does not wish to have the fastest connection. Instead, he wishes to have a long sleeping period without interruptions caused by train changes.

Acknowledgments

This work was supported by the DFG Focus Program 1126 "Algorithmic Aspects of Large and Complex Networks", grant Mu 1482/2-2. This work was done in cooperation with datagon GmbH, Waldems, Germany. In particular, we wish to thank Wolfgang Sprick for fruitful discussions and close collaboration in the development of MOTIS.

References

1. Pallottino, S., Scutellà, M.G.: Shortest path algorithms in transportation models: Classical and innovative aspects. In: Equilibrium and Advanced Transportation Modelling. Kluwer Academic Publishers (1998)
2. Möhring, R.H.: Verteilte Verbindungssuche im öffentlichen Personenverkehr: Graphentheoretische Modelle und Algorithmen. In: Angewandte Mathematik - insbesondere Informatik, Vieweg (1999) 192–220
3. Schulz, F., Wagner, D., Weihe, K.: Dijkstra's algorithm on-line: An empirical case study from public railroad transport. ACM Journal of Experimental Algorithmics **5** (2000) Article 12
4. Müller-Hannemann, M., Weihe, K.: Pareto shortest paths is often feasible in practice. In: Algorithm Engineering – WAE 2001. Volume 2141 of LNCS. Springer (2001) 185–198
5. Müller-Hannemann, M., Schnee, M., Weihe, K.: Getting train timetables into the main storage. In: Proceedings of the 2nd Workshop on Algorithmic Methods and Models for Optimization of Railways (ATMOS 2002). Volume 66 of Electronic Notes in Theoretical Computer Science. Elsevier (2002)
6. Pyrga, E., Schulz, F., Wagner, D., Zaroliagis, C.: Experimental comparison of shortest path approaches for timetable information. In: Proceedings of the Sixth Workshop on Algorithm Engineering and Experiments, SIAM (2004) 88–99
7. Müller-Hannemann, M., Schnee, M.: Finding all attractive train connections by multi-criteria Pareto search. In: Proceedings of the 4th Workshop in Algorithmic Methods and Models for Optimization of Railways (ATMOS 2004). LNCS, Springer (2005), to appear.
8. Cooke, K.L., Halsey, E.: The shortest route through a network with time-dependent internodal transit times. Journal of Mathematical Analysis and Applications **14** (1966) 493–498
9. Orda, A., Rom, R.: Shortest-path and minimum-delay algorithms in networks with time-dependent edge-length. Journal of the ACM **37** (1990)
10. Orda, A., Rom, R.: Minimum weight paths in time-dependent networks. Networks **21** (1991)
11. Kostreva, M.M., Wiecek, M.M.: Time dependency in multiple objective dynamic programming. Journal of Mathematical Analysis and Applications **173** (1993) 289–307
12. Nachtigal, K.: Time depending shortest-path problems with applications to railway networks. European Journal of Operations Research **83** (1995) 154–166
13. Brodal, G.S., Jacob, R.: Time-dependent networks as models to achieve fast exact time-table queries. In: Proceedings of the 3rd Workshop on Algorithmic Methods and Models for Optimization of Railways (ATMOS 2003). Volume 92 of Electronic Notes in Theoretical Computer Science. Elsevier (2004)
14. Pyrga, E., Schulz, F., Wagner, D., Zaroliagis, C.: Towards realistic modeling of time-table information through the time-dependent approach. In: Proceedings of the 3rd Workshop on Algorithmic Methods and Models for Optimization of Railways (ATMOS 2003). Volume 92 of Electronic Notes in Theoretical Computer Science. Elsevier (2004) 85–103
15. Müller-Hannemann, M., Schulz, F., Wagner, D., Zaroliagis, C.: Timetable information: Models and algorithms. In: Algorithmic Methods for Railway Optimization. LNCS, Springer (2005), to appear.

16. (HAFAS) A timetable information system by HaCon Ingenieurgesellschaft mbH, Hannover, Germany. http://www.hacon.de/hafas/.
17. Theune, D.: Robuste und effiziente Methoden zur Lösung von Wegproblemen. Teubner Verlag, Stuttgart (1995)