Universität Konstanz

Recognizing Bundles in Time Table Graphs — A Structural Approach

Annegret Liebers
Dorothea Wagner
Karsten Weihe

Konstanzer Schriften in Mathematik und Informatik
Nr. 96, September 1999, rev. März 2000

ISSN 1430–3558

Recognizing Bundles in Time Table Graphs — A Structural Approach*

Annegret Liebers Dorothea Wagner Karsten Weihe

University of Konstanz, Department of Computer and Information Science, Box D 188,
78457 Konstanz, Germany. Email: FirstName.LastName@uni-konstanz.de

(Last revision: March 24, 2000)

Abstract

We are dealing with an application problem arising in a cooperation with the national German railway company which occurs in the analysis of time table data. The purpose is to infer the underlying railroad network and the actual travel route of the trains when only their time tables are known. The structural basis of our considerations in this paper is a directed graph constructed from train time tables, where train stations correspond to vertices, and where pairs of consecutive stops of trains correspond to edges. Determining the travel route of trains corresponds to an edge classification problem in this graph. Exploiting the structure of this graph, we approach the edge classification problem by locating vertices that intuitively correspond to train stations where the underlying railroad network branches into several directions, and that induce a partition of the edge set into *bundles*.

We describe the modeling process of the classification problem resulting in the bundle recognition problem. Given the NP-hardness of the corresponding optimization problem, we present a heuristic that exploits among other things the geometry of the embedded directed graph. We perform a computational study using time table data from 13 European countries.

1 Introduction

We are given train time tables of long distance, regional, and local trains consisting of more than 140 000 trains. Together, they stop at about 28 000 train stations all over Europe. Each time table contains a list of consecutive stops for one particular train, indicating the times of arrival and departure for each stop. Such a set of train time tables induces a directed *time table graph* as follows: Each train station appearing in some time table becomes a vertex of the graph, and if there is a train leaving station r and having station s as its next stop, then there is a directed edge (r, s) in the graph. Typically, many trains contribute to one edge. So by definition, a time table graph has no multiple edges.

The original problem as posed by the TLC^1, the subsidiary of the national German railway company *Deutsche Bahn AG* that is responsible for collecting and publishing time table information, is the following: Decide, solely on the basis of the train time tables, for each edge of the time table graph, whether trains traveling along it pass through other train stations on the way

*Partially supported by DFG grant Wa 654/10-2.

[1] *Transport-, Logistik- und Consulting GmbH/EVA-Fahrplanzentrum*

without stopping there. If so, the sequence of these other train stations is also sought. In other words, the edges have to be classified into *real* edges representing segments of the physical railroad network and *transitive* edges corresponding to paths in this network, and for each transitive edge (s,t), the s-t-path consisting of only real edges that corresponds to the travel route of the trains contributing to (s,t) is sought. Figure 1 shows an example of a time table graph, with real edges in black, and transitive edges in grey.

In connection with some of their projects, the *TLC* is interested in an algorithmic solution for this problem based solely on train time tables. Since a European time table graph consists of more than 80 000 edges, a "manual" solution by consulting maps or other resources outside the train time tables themselves is infeasible simply because of the large size of the time table graph.

Using various elements of time table data (types of trains, travel times between stops, days of operation, etc.), many ways of approaching this edge classification problem are conceivable. The goal of this paper is to explore how the time table graph structure can be exploited towards the solution of the problem.

Exploration of railway data suggests that a railroad network decomposes into lines of railroad tracks between branching points of the underlying railroad network, where the lines of railroad tracks correspond to bundles consisting of (usually many) transitive edges and one path of real edges. Guided by this intuition, our structural approach consists of identifying bundles and according branching points, and in this paper we concentrate on classifying edges through identifying bundles and evaluate the usefulness of this approach for the edge classification problem. Section 2 illustrates the notion of bundles of edges in time table graphs. It then suggests to formulate and solve a *bundle recognition problem* in order to solve the edge classification problem. Section 3 formally defines the bundle recognition problem as a graph theoretic optimization problem by formalizing the notion of a bundle, and summarizes known NP-completeness results [2]. Section 4 presents a heuristic for bundle recognition, and Section 5 describes a computational study using train time tables from European countries, as well as an evaluation of our approach.

So far, this type of problem has not been studied under an algorithmic aspect. On the other hand, some previous work addresses related yet different problems and questions: [5] discusses general problems of experimental algorithm design and refers to the problem in this paper as one of four examples. [4, 3] deal with other algorithmic optimization problems on time table data, while [1] addresses visualization of edge classifications.

2 Modeling

The original problem consists in determining the actual travel route of trains, solely on the basis of their time tables. In discussions with our cooperation partner and in view of this and other time table analysis issues, we have developed the time table graph as an abstraction of time table data. In this graph, the original problem translates into classifying edges as real and transitive, and of finding the paths of real edges that correspond to the travel routes along transitive edges. After first experimenting with immediate approaches using local properties such as travel times of trains between stops, we decided to attack the problem on a structural level, and the focus of this paper is this structural approach.

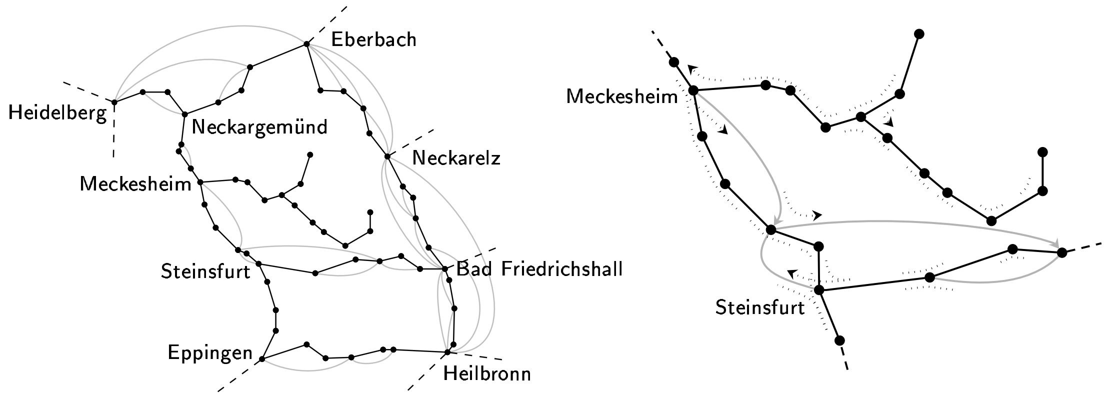


Figure 1: *Left:* An example of the underlying undirected graph of a time table graph. Real edges are drawn in black, transitive edges are drawn in grey. *Right:* Line graph edges induced by the trains operating on this part of the time table graph are drawn in dotted lines; edges shown undirected mean that there is a directed edge in either direction.

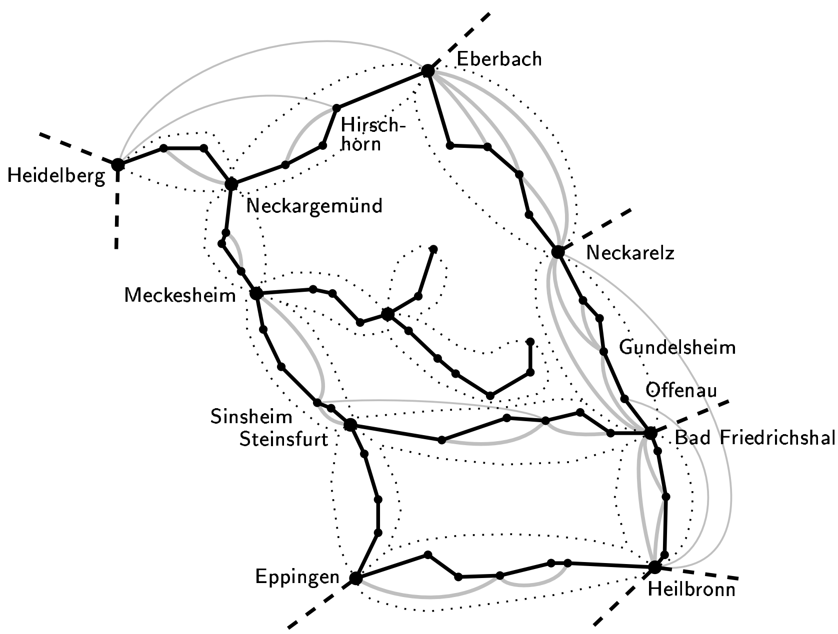


Figure 2: Branching points of the physical railroad network (big black vertices) and bundles of edges between them (encircled in dotted lines). The thin edges do not belong to any bundle.

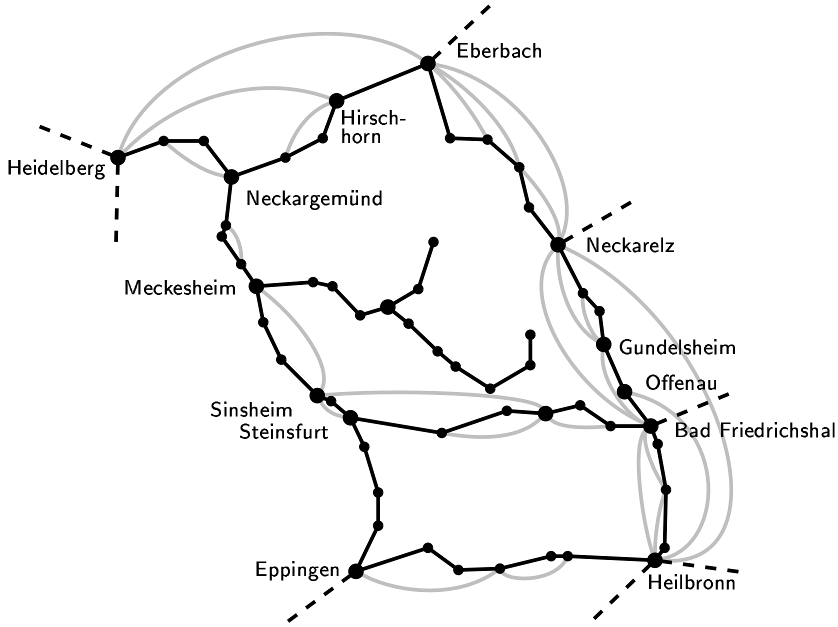


Figure 3: A vertex set (big black vertices) of potential bundle end points

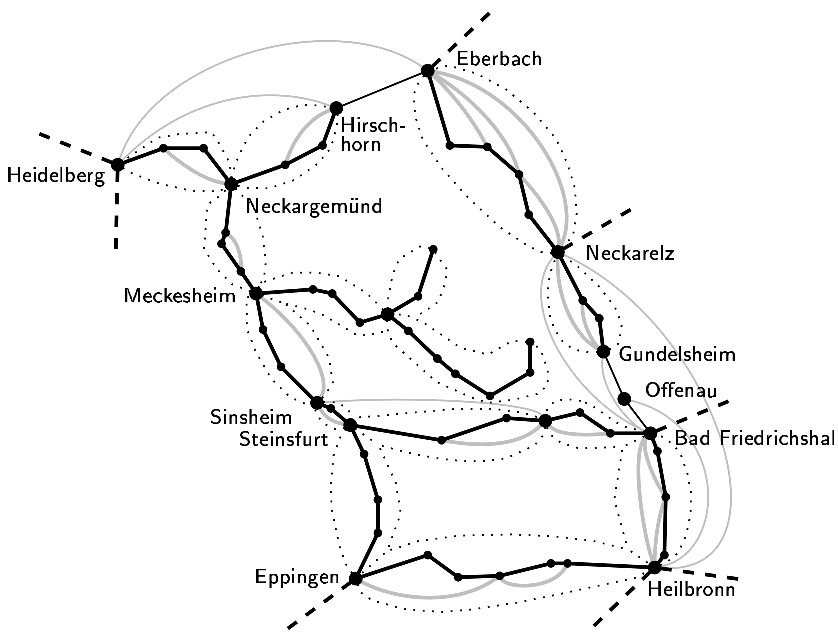


Figure 4: The edge partition resulting from applying steps 1. through 3. to Fig. 3

It is based on the following observation: Often there are bundle-like edge sets along a line of railroad tracks that are delimited by the branching points of the underlying railroad network (in Fig. 1 for example between Heilbronn and Eppingen, or between Neckarelz and Bad Friedrichshall). If we can find sets of edges that form such bundles, then we can, within each bundle, classify the edges: The physical railroad network appears as a (unique) Hamilton path of the bundle, and the Hamilton path also provides the desired paths for transitive edges. Notice that since a time table graph is a directed graph, along one line of railroad tracks there are usually two bundles, one in each direction.

In Fig. 2, the branching points of the physical railroad network are marked with big black vertices as bundle end points, and bundles of edges between them are encircled with dotted lines. There are some edges (drawn in thin lines) that do not belong to any bundle. Ignoring the thin edges for a moment, the idea for finding edge sets that form bundles is the following: If a train travels along the edge (r, s), and then along the edge (s, t), and if s is not a branching point, then (r, s) and (s, t) belong to the same bundle. So the following procedure could partition the edge set of the time table graph in Fig. 2 into the bundles that are encircled in dotted lines:

1. Initially, every edge of the time table graph is its own set.

2. For every train and for every triple (r, s, t) of consecutive stops of the train, if s is not a big black vertex, (and if neither (r, s) nor (s, t) are thin edges,) then unite the edge sets that (r, s) and (s, t) belong to.

   Each resulting bundle in Fig. 2 forms a directed acyclic graph that contains a Hamilton path. Within each bundle, the edges can now be classifyed as real or transitive depending on whether or not they belong to the Hamilton path. But edges connecting the end points of a bundle like (Eberbach, Neckarelz) are still singletons in the partition. We can easily assign each such edge to the bundle to which it belongs:

3. For every singleton (a, b) in the partition, if there is a unique edge set A' in the partition that contains a Hamilton path starting with a and ending with b and consisting of at least two edges, then (a, b) is a transitive edge, and the trains traveling along it actually pass through the train stations of the Hamilton path of A' without stopping there.

But in contrast to the ideal szenario described so far, in reality, we of course do not know the bundle end points of a given time table graph, nor do we know which edges to ignore because they do not belong to any bundle. Also, some edges might not be assigned to "their" bundle simply because there is no train causing the necessary unite operation in step 3. For example, no train traveling along the grey edge (Steinsfurt, Sinsheim) actually continues beyond Sinsheim towards Meckesheim, and so the edge (Steinsfurt, Sinsheim) would in reality remain a singleton instead of becoming part of the edge set that forms a bundle from Steinsfurt to Meckesheim.

Now consider Fig. 3, where more vertices than just the branching points of the railroad network are marked as bundle end points. If we start with this vertex set and with the assumption that every edge potentially belongs to some bundle (and is therefore drawn in thick lines in Fig. 3), then the edge partition resulting from steps 1. through 3. forms the bundles encircled in dotted lines and the singletons drawn in thin lines in Fig. 4. Within the encircled bundles, edges are correctly classified, while edges in singletons remain unclassified by this approach. So by finding a vertex set of potential bundle end points (that possibly contains more vertices than just the branching points of the railroad network) such that the edge partition resulting from steps 1.

through 3. consists of at least some bundles (or parts of bundles such as the one from Neckarelz to Gundelsheim in Fig. 4) and then some singletons, we are able to classify many of the edges in a time table graph. Notice that it is edges starting or ending in the middle of a bundle like (Offenau, Heilbronn) that cause us to include more vertices than just the branching points of the railroad network in the set of potential bundle end points. As long as time table graphs do not contain too many such edges, and as long as only few anomalies like the situation of the edge (Steinsfurt, Sinsheim) occur, this idea for bundle identification seems promising.

We now proceed as follows. In Section 3, we will formulate conditions that edge sets have to fulfill to constitute bundles. We seek this formalization of the intuitive notion of bundles with the following goal in mind: If a vertex set of potential bundle end points is guessed, and if steps 1. and 2. of the above procedure are applied (treating every edge as if it belongs to some bundle), then the resulting edge partition should have the following property: If the edge sets fulfill the conditions for being bundles, and if within each edge set that contains more than one element the edges are classifyed as real or transitive depending on whether or not they belong to the Hamilton path of the edge set, then there are no wrong classifications. Obviously, there cannot be a proof that our formalization of the notion of a bundle achieves this goal. But the heuristic results in Section 5 indicate that it is a workable definition. Note that we drop step 3. from our considerations for now and concentrate on the core of bundle recognition given by steps 1. and 2. Once bundles have been found with steps 1. and 2., singletons like (Eberbach, Neckarelz) can be assigned to their bundle with step 3. as postprocessing.

Once the notion of a bundle has been formally defined, the bundle recognition problem then consists of finding a vertex subset of a given time table graph such that the resulting edge partition forms bundles (according to the formal definition), and such that there are few singletons in the partition.

3 Formalization and known NP-Completeness results

Recall the following definition:

Definition 1 *Given a directed graph $G = (V, A)$, its* line graph $L = (A, A_{\mathrm{L}})$ *is the directed graph in which every edge of G is a vertex, and in which there is an edge (a, b) if and only if there are three vertices r, s, and t of G so that $a = (r, s)$ and $b = (s, t)$ are edges in G.*

A set of train time tables induces the time table graph $G = (V, A)$ together with a subset A'_{L} of its line graph edge set: If there is a train with train stations r, s, and t as consecutive stops, in that order, then the line graph edge $((r, s), (s, t))$ is in A'_{L}. Figure 1 shows an example for a subset of line graph edges in a time table graph.

Definition 2 *Given a directed graph $G = (V, A)$ and a subset A'_{L} of its line graph edges, a vertex subset $V' \subseteq V$ induces a partition of A by the following procedure:*

1. *Initially, every edge of A is its own set.*
2. *For every line graph edge $((r, s), (s, t)) \in A'_{\mathrm{L}}$, if s is not in V', then unite the edge sets that (r, s) and (s, t) belong to.*

We still need to specify how to recognize whether the edge sets of a directed graph $G = (V, A)$ induced by a line graph edge subset A'_{L} and a vertex subset $V' \subseteq V$ form bundles that are suitable

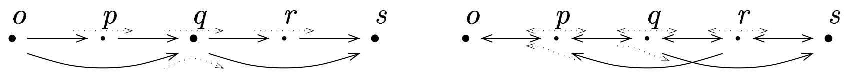


Figure 5: *Left:* $V' = \{o, s\}$ would induce two edge sets, one consisting of the four short edges, and one consisting of the two long edges — for note that there are no line graph edges $((p,q),(q,s))$ or $((o,q),(q,r))$. Classifying edges along Hamilton paths as real would lead to classifying all six edges as real (which is obviously wrong). $V' = \{o, q, s\}$, however, induces four edge sets, among them two singletons, resulting in four edges classified as real and two edges remaining unclassified. *Right:* $V' = \{o, s\}$ induces two edge sets that are opposite of each other.

for our edge classification approach. To begin with, such an edge set should form a directed acyclic graph that contains a Hamilton path. And since we classify edges along Hamilton paths as real, a situation as depicted on the left side of Fig. 5 would lead to wrongly classifying two edges as real. We can avoid such wrong classifications by demanding that two different bundles do not share vertices outside V', unless, of course, the two bundles are *opposite* of each other as defined below and as illustrated on the right side of Fig. 5.

Definition 3 *Consider the partial order induced by the transitive closure of a directed acyclic graph: For a directed acyclic graph G, define a partial order "$\leq$" on its vertex set by $r \leq s$ for two (not necessarily distinct) vertices of G if and only if there is a directed path (possibly of length zero) from r to s in G. If neither $r \leq s$ nor $s \leq r$ for two vertices r and s, we write $r||s$.*

We are now ready to cast the intuitive notion of opposite edge sets into a formal definition, and to state the bundle recognition problem as a graph theoretic problem:

Definition 4 *For a directed graph $G = (V, A)$ and a subset $A' \subseteq A$ of its edge set, let $G[A']$ denote the directed graph induced by A'. Given two disjoint edge subsets $A_1 \subseteq A$ and $A_2 \subseteq A$ of G such that $G[A_1]$ and $G[A_2]$ are acyclic, A_1 and A_2 are called* opposite *if the following two conditions hold:*

1. *There is an edge $(r, s) \in A_1$ such that $(s, r) \in A_2$.*

2. *If r and s are two distinct vertices belonging both to $G[A_1]$ and to $G[A_2]$, and if $r \leq s$ in $G[A_1]$, then either $s \leq r$ or $s||r$ in $G[A_2]$.*

Definition 5 (Bundle Recognition Problem) *Given a directed graph $G = (V, A)$ and a subset A'_{L} of its line graph edge set, the* bundle recognition problem *consists of finding a vertex set $V' \subseteq V$ such that the sets of the induced edge partition fulfill the following three conditions:*

1. *Each edge set induces a directed acyclic graph.*

2. *This graph contains a Hamilton path.*

3. *Two distinct edge sets do not have vertices outside V' in common, except possibly if the edge sets are opposite.*

If a vertex set $V' \subseteq V$ solves the bundle recognition problem, then we call the edge sets of the induced partition *bundles.* Bundles that are not singletons are called *nontrivial.* Note that a solution to the bundle recognition problem will consist in a subset V' of the vertices of the given time table graph, and that the whole vertex set $V' := V$ of a time table graph will always constitute a (trivial) solution. But this solution is useless because it induces an edge partition consisting only of singletons, and our goal is to minimize the number of induced singletons. Since adding a vertex to a solution of the bundle recognition problem cannot reduce the number of induced singletons, minimizing the cardinality of V' seems to be a reasonable optimization criterion as well. Note, however, that a minimum cardinality set V' solving the bundle recognition problem does not necessarily yield the smallest possible number of induced singletons. In fact, there are instances where all vertex sets solving the bundle recognition problem and inducing the minimum number of singletons possible contain more vertices than a vertex set of minimum cardinality.

[2] shows that finding a solution for the bundle recognition problem that contains a small number of vertices or that induces a small number of singletons is NP-hard in a very strong sense. It is already NP-complete to decide whether there is any other solution besides the trivial one:

Theorem 1 *Given a directed graph $G = (V, A)$ and a subset A'_{L} of its line graph edge set, it is NP-complete to decide whether there is any proper subset $V' \subset V$ solving the bundle recognition problem. It is also NP-complete to decide whether there is any vertex set $V' \subseteq V$ solving the bundle recognition problem and inducing less than $|A|$ singletons.*

4 A Heuristic for the Bundle Recognition Problem

We use the fact that branching points of the physical railroad network are useful for solving the bundle recognition problem on time table graphs in practice: We make an educated guess at the set of branching points and take it as an initial vertex set V' for solving the bundle recognition problem. We then assure that wherever the edge partition induced by the initial V' does not fulfill the three bundle conditions of Def. 5, the violations of the conditions are repaired by adding more vertices to V'. Recall that $V' = V$ is a feasible, albeit practically useless, solution to the bundle recognition problem, so augmenting V' until the bundle conditions are fulfilled always yields a feasible solution. Additionally to augmenting V', we may systematically take vertices out of V' if the edge sets incident to such vertices can be united to form (larger) bundles. These ideas lead to the following heuristic:

initial Guess an initial set $V' \subseteq V$ and determine the edge partition it induces.

first augment For each induced edge set that is not a DAG or that is a DAG but does not contain a Hamilton path, add all end vertices of its edges to V'. Also, for each vertex not in V' that is common to two non-opposite edge sets, add that vertex to V'. Determine the edge partition induced by the augmented V'.

unite-singleton If there is a singleton $\{(s,t)\}$ and exactly one other bundle whose Hamilton path begins with s and ends with t, then unite the singleton with this bundle.

reduce If the bundles incident to a vertex $v \in V'$ can be united to form new, larger bundles, then delete v from V' and update the edge partition accordingly. Perform *unite-singleton* and *reduce* alternatingly to grow bundles as illustrated in Figure 6. Note that so far we do not guarantee that the resulting vertex set V' is a solution to the bundle recognition problem.

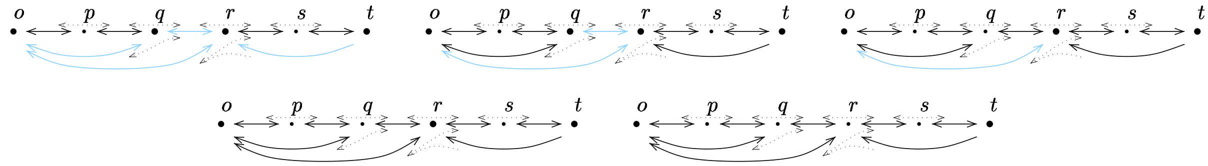


Figure 6: Growing bundles by alternately applying *unite-singleton* and *reduce*: Suppose we are starting with the instance in the top left diagram with 6 vertices, 15 edges and 13 line graph edges. The initial vertex set $V' = \{o, q, r, t\}$ (thick black dots) induces a partition with 4 nontrivial bundles (two in each direction; in black) and 7 singletons (in grey). *unite-singleton* yields the diagram in the top middle, where 3 singletons were united with nontrivial bundles. *reduce* then eliminates q from V', leaving only two singletons (top right). Another *unite-singleton* (bottom left) and *reduce* (bottom right) yield $V' = \{o, t\}$, two nontrivial bundles, and no singletons for this instance.

final augment Perform the augment step again so that the resulting vertex set V' is a solution to the bundle recognition problem.

Resulting from the original edge classification problem, as well as from observations with time table graphs, two additional elements have been added to the heuristic:

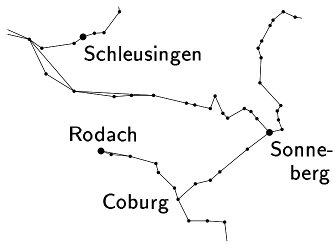
artificial line graph edges It occurs in time table graphs that a vertex v should according to intuition be an inner vertex of a bundle but remains in V' because all trains visiting v begin or end there (see Figure 7 for an example). When such an absence of line graph edges is detected in a *reduce* step, we add artificial line graph edges to the instance of the bundle recognition problem so that in the induced edge partition, bundles incident to v are united to form larger bundles.

final unite-singleton After a solution V' to the bundle recognition problem has been found, there are often singletons such as (Eberbach, Neckarelz) in Figure 1 where there is a unique bundle with the end points of its Hamilton path equal to the end points of the singleton. In this case, we assume that the singleton is a transitive edge, and that the trains traveling along it actually travel along the Hamilton path of the bundle. Therefore, we perform a final *unite-singleton* after the *final augment* to take care of all such singletons with unique corresponding bundles.

Note that *final unite-singleton* is the implementation of step 3. in Section 2. Observe that these additional elements result in an edge partition that is strictly speaking not induced by the set V' that resulted from *final augment*.

5 Computational Study

The bundle recognition heuristic was implemented in C++ and tested on a set of European time tables that was made available to us by the *TLC*. Given the fact that no quality criterion for the bundle recognition heuristic is available (except for studying the visualization of the results with a human eye), we simply calculate the percentage of classified edges. However, we do not propose a solution for the bundle recognition problem as a final solution for edge classification.



At big vertices, trains only begin or end, whereas for every small vertex, there is at least one train that induces a line graph edge at that vertex. Rodach is the end of a railroad line, and Rodach and Coburg should be recognized as bundle end points. Since no train induces a line graph edge at Rodach, it does not matter for the induced partition whether the heuristic places Rodach in V' or not. Sonneberg is a branching point of the railroad network, and the heuristic will insert it in V'. But Schleusingen should intuitively be an inner vertex of a bundle. In such cases the heuristic will add artificial line graph edges.

Figure 7: A small excerpt of the European time table graph in the Thuringian Forest in Germany.

It turns out that the percentage of edges classified through bundle recognition even indicates to what extent the bundle structure is actually present in different time table graphs.

As a result of the bundle recognition heuristic, 81 percent of the European time table graph edges can be classified as *real* or *transitive*. Counted separately for 13 countries, the highest percentage of classified edges (93) is reached for Slovakia, and the lowest (62) is obtained for Great Britain. The percentages for the other 11 countries are spread between 75 and 78 for 5 of them, and between 85 and 92 for the remaining 6.

In the first part of the experimental study, we tried (combinations of) the following different initial vertex sets: vertices of high degree in relation to the vertex's neighbors, vertices where trains begin or end, vertices that have degree at least three in a minimum spanning forest (MSF) of the (underlying undirected graph of the) time table graph (where the edge weights are either Euclidean distances or travel times along the edges), and vertices where the angles between consecutive edges in the embedding of the time table graph indicate branching vertices. To check this *angle condition*, we consider the underlying undirected graph and its straight line embedding given by the geographical positions of the vertices. For each vertex v with (undirected) degree at least three, we determine the three largest angles α_1, α_2, and α_3 between consecutive (undirected) edges incident to v. If $\alpha_1 + \alpha_2$ is not too large, or if α_3 is not too small, then we say that v fulfills the angle condition. Figure 8 shows the results of the heuristic for four combinations of these choices for initial vertex sets, broken up for 13 European countries. For comparison, we also included the results when the initial vertex set consists simply of all vertices.

The second part of the experimental study varies the set of time tables inducing the time table graphs considered. It is interesting to compare for different countries the results based on the European time table graph (Figure 8) with the results based on the time table graph induced by the time tables of trains traveling entirely within the country (Figure 9). Note that for Great Britain, France, and Germany, the best choice of initial V' for each of the three countries did not change. And comparing the results, it turns out that for each of the three countries, the extra edges in the European time table graph stemming from international trains mostly end up among the transitive and among the unclassified edges.

We visualize the results for time table graphs that are particularly interesting for the evaluation of our structural approach for the edge classification problem: Figure 10 show the time table graphs of Great Britain, France, and four selected areas of Germany corresponding to Figure 9 with the final vertex set V' and the recognized nontrivial bundles in black, and remaining singletons in grey. In Great Britain and France, there are many very long edges that stem from

	PL	**CZ**	**SL**	**GB**	**S**	**D**	**A**	**I**	**NL**	**CH**	**DK**	**F**	**B**	**Europe**
$\lvert V\rvert$	3 382	2 719	919	2 537	587	6 505	1 667	2 383	375	1 756	476	3 304	534	28 280
$\lvert A\rvert$	8 858	7 369	2 399	8 511	1 606	17 793	4 463	8 196	1 108	4 586	1 217	11 132	1 668	82 594
Choose the initial vertex set $V_i' = V$:														
s	987	872	212	3 581	391	3 115	649	2 743	300	591	235	3 381	462	18 928
t	1 216	1 199	391	997	173	2 159	719	1 564	140	635	132	2 138	254	12 107
H	6 655	5 298	1 796	3 933	1 042	12 519	3 095	3 889	668	3 360	850	5 613	952	51 559
V_i' contains the vertices with degree larger than that of 60 % of their neighbors:														
s	939	729	150	3 237	381	2 642	525	1 981	306	561	169	2 681	433	16 066
t	1 250	1 268	**419**	1 162	177	**2 388**	791	1 956	138	650	160	2 503	267	13 555
H	6 669	5 372	**1 830**	4 112	1 048	**12 763**	3 147	4 259	664	3 375	888	5 948	968	52 973
V_i' contains vertices where trains begin or end, or that have degree at least three in the Euclidean MSF:														
s	923	759	150	3 231	369	2 675	513	1 808	302	513	187	2 557	422	15 790
t	1 251	1 252	**419**	**1 171**	**183**	2 367	**790**	2 048	140	**677**	154	2 570	274	13 695
H	6 684	5 358	**1 830**	**4 109**	**1 054**	12 751	**3 160**	4 340	666	**3 396**	876	6 005	972	53 109
V_i' contains vertices where trains begin or end, or that have degree at least three in the MSF based on travel times:														
s	906	727	150	3 255	369	2 677	529	1 736	288	513	197	2 455	412	15 595
t	**1 256**	**1 266**	**419**	1 156	**183**	2 365	784	**2 085**	146	**677**	146	**2 635**	**276**	**13 793**
H	**6 696**	**5 376**	**1 830**	4 100	**1 054**	12 751	3 150	**4 375**	674	**3 396**	874	**6 042**	**980**	**53 206**
V_i' contains vertices fulfilling the angle condition $\alpha_1 + \alpha_2 \leq 280°$ or $\alpha_3 \geq 55°$, or having degree at least three in the Euclidean MSF:														
s	938	774	173	3 300	376	2 696	547	1 922	270	540	169	2 808	433	16 283
t	1 242	1 245	408	1 146	180	2 360	780	1 999	**154**	664	**160**	2 446	269	13 477
H	6 678	5 350	1 818	4 065	1 050	12 737	3 136	4 275	**684**	3 382	**888**	5 878	966	52 834
%	**89**	**92**	**93**	**62**	**77**	**85**	**88**	**78**	**75**	**88**	**86**	**77**	**75**	**81**

Figure 8: Results of the heuristic on European time table data: Columns for Poland, the Czech Republic, Slovakia, Great Britain, Sweden, Germany, Austria, Italy, The Netherlands, Switzerland, Denmark, France and Belgium indicate the counts for vertices and edges that lie in each country, respectively. European countries not listed have less than 150 vertices in the set of time table data available for the study. Their vertices and edges only appear in the column *Europe*. Also, edges crossing a country border are only accounted for in this column.

The rows $|V|$ and $|A|$ list the numbers of vertices and edges within each country. Then, the results of the heuristic are given for five different choices of the initial vertex set V_i'. For each choice, the numbers of singletons (rows "s"), transitive (rows "t") and Hamilton (rows "H") edges in the resulting edge partition are given. When classifying edges as *real* or *transitive* using the results of the heuristic, the Hamilton edges are classified as *real*, and the singletons remain unclassified. The choice of V_i' that classified most edges is indicated in bold face for each column, and the percentage of edges classified by this best choice is given in the last row of the table.

	GB	**F**	**D**	***sparsest***	***sparse***	***dense***	***urban***
$\lvert V\rvert$	2 525	3 291	6 473	165	119	425	321
$\lvert A\rvert$	8 485	10 897	16 892	404	317	1 005	906
$\lvert V_i'\rvert$	719	690	1 020	43	10	59	61
$\lvert V_f'\rvert$	871	753	1 071	22	12	75	68
s	3 209	2 305	1 988	18	10	78	153
t	**1 171**	**2 573**	**2 146**	**61**	**75**	**93**	**141**
H	**4 105**	**6 019**	**12 758**	**325**	**232**	**834**	**612**
b	1 005	1 130	2 473	57	30	187	135
%	**62**	**78**	**88**	**95**	**96**	**92**	**83**

Figure 9: Results of the heuristic for the time table graphs induced by trains traveling within Great Britain, France, and Germany, respectively. Within Germany, the results are broken down for four selected areas *sparsest*, *sparse*, *dense* and *urban* with increasing density of vertices and edges per area. Rows $|V_i'|$ and $|V_f'|$ show the cardinality of the initial vertex set and of the vertex set after *final augment*, respectively. The time table graphs for Great Britain and France and the time table graph excerpts for the four German areas with the edge classifications resulting from the heuristic are shown in Figure 10. The four excerpts in Figure 10 are drawn to scale, showing the different densities of the train network in the four German areas.

For Great Britain, the initial vertex set V_i' was chosen to contain vertices where trains begin or end as well as vertices that have degree at least three in the (undirected) Euclidean MSF. For France, the same choice was made, except that the MSF with respect to travel times along edges was used. For Germany, the *sparsest* and the *urban* areas, V_i' was chosen to contain all vertices whose vertex degree is larger than that of at least 60 percent of the vertex's neighbors. For the *sparse* and the *dense* areas, V_i' was chosen to contain vertices fulfilling the angle condition $\alpha_1 + \alpha_2 \leq 280°$ or $\alpha_3 \geq 55°$ or vertices having degree at least three in the (undirected) Euclidean MSF. It turned out that for the *sparsest*, *sparse* and *dense* areas, the result yielded by the choice of V_i' listed here was actually identical to the result for two other choices of V_i' among the five choices considered. Note that in the *sparse* area, only 4 pairs of directed unclassified edges are visible. The fifth one is very short and obscured by a vertex in V'.

high speed and overnight trains that sometimes travel for hours without stopping anywhere in-between. Particularly in Great Britain, the percentage of such edges appears to be high, while the time table graph shows comparatively few areas with the bundle structure that the heuristic is trying to exploit. This explains why the heuristic achieves such a low percentage of classified edges in Great Britain. This lack of bundle structure contrasts sharply with the four time table graph excerpts of Germany, where the bundle structure is clearly visible, and where the heuristic works well. It recognizes (most) existing (nontrivial) bundles and thus allows the classification of edges within such bundles as *real* or *transitive*.

6 Conclusion

As an approach for an edge classification problem on graphs induced by train time tables, we evaluated a structural approach based on the intuition that in such graphs many edges can be partitioned into *bundles*. We conducted a computational study on time tables of 140 000 European trains. It turns out that our heuristic works quite well for time table graphs exhibiting a structure according to this intuition. The classification results based on the bundle recognition heuristic are a very good basis for improvements using further elements of time tables besides the underlying graph structure and the geometry of the graph's embedding.

Acknowledgements

We would like to thank Stephan Schmidt, Frank Schulz, and Thomas Willhalm for software implementations that contributed to this computational study.

References

[1] Ulrik Brandes and Dorothea Wagner. Using Graph Layout to Visualize Train Connection Data. In Sue H. Whitesides, editor, *Proceedings of the 6th International Symposium on Graph Drawing, GD'98, Montréal, Canada, August 13-15, 1998*, pages 44–56. Springer-Verlag, Lecture Notes in Computer Science, vol. 1547, 1998.

[2] Annegret Liebers, Dorothea Wagner, and Karsten Weihe. On the Hardness of Recognizing Bundles in Time Table Graphs. In Peter Widmayer, Gabriele Neyer, and Stephan Eidenbenz, editors, *Proceedings 25th International Workshop on Graph-Theoretic Concepts in Computer Science, WG'99*, pages 325–337. Springer-Verlag, Lecture Notes in Computer Science, vol. 1665, 1999.

[3] Frank Schulz, Dorothea Wagner, and Karsten Weihe. Dijkstra's Algorithm On–Line: An Empirical Case Study from Public Railroad Transport. In Jeffrey Scott Vitter and Christos D. Zaroliagis, editors, *Proceedings of the 3rd Workshop on Algorithm Engineering, WAE' 99*, pages 110–123. Springer-Verlag, Lecture Notes in Computer Science, vol. 1668, 1999.

[4] Karsten Weihe. Covering trains by stations or the power of data reduction. In *On-Line Proceedings of the 1st Workshop on Algorithms and Experiments (ALEX'98)*, 1998. http://rtm.science.unitn.it/alex98/proceedings.html.

[5] Karsten Weihe, Ulrik Brandes, Annegret Liebers, Matthias Müller-Hannemann, Dorothea Wagner, and Thomas Willhalm. Empirical Design of Geometric Algorithms. In *Proceedings of the 15th ACM Symposium on Computational Geometry, SCG'99, Miami, Florida, USA, June 13 - 16, 1999*, pages 86–94, 1999.

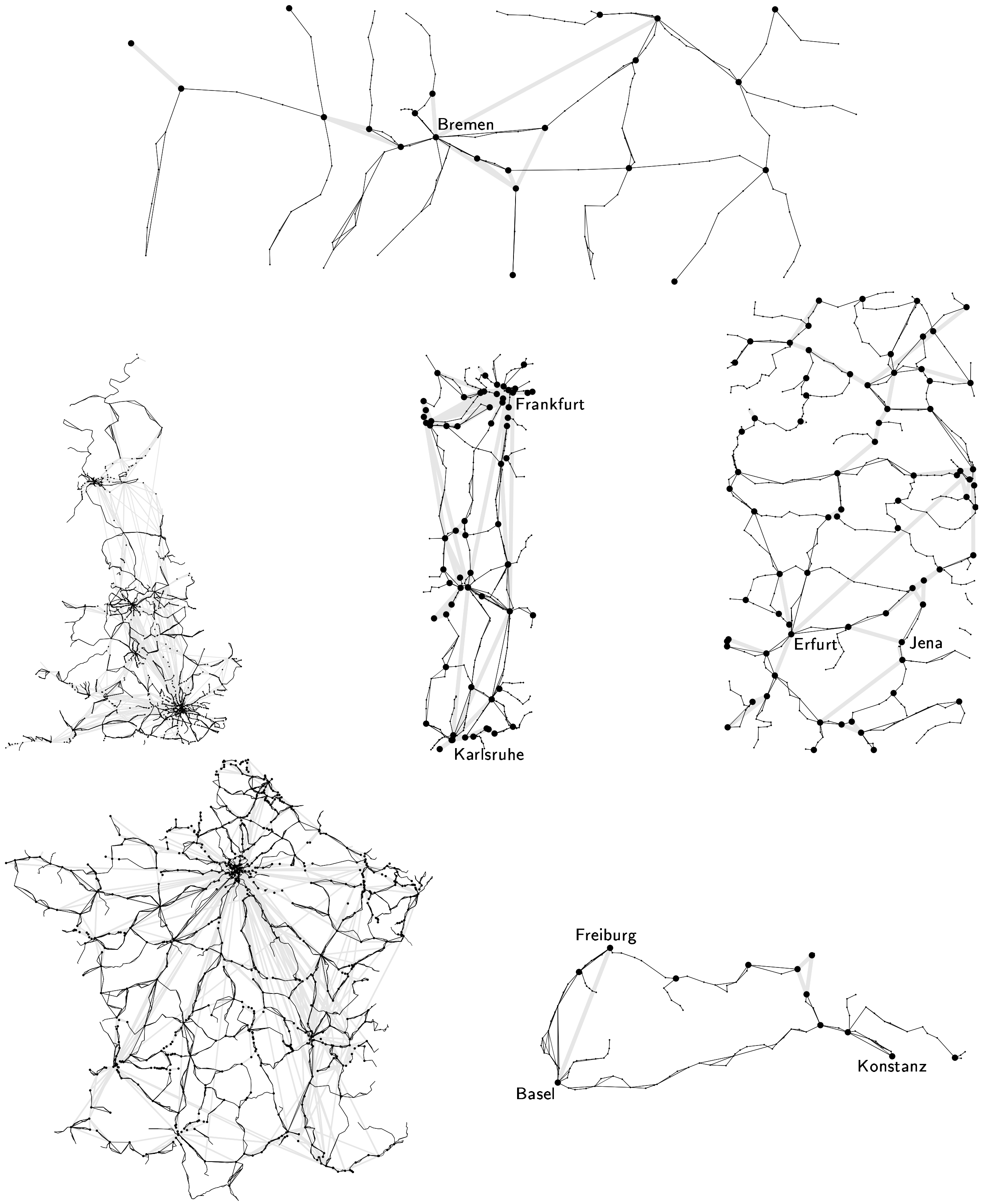


Figure 10: Time table graphs of Great Britain and France. Excerpts of the German time table graph: *sparsest* (top), *urban* (middle), *dense* (middle right) and *sparse* (bottom right)