Technische Universität Berlin

FAKULTÄT II
MATHEMATIK UND
NATURWISSENSCHAFTEN
Institut für Mathematik

A Cut-based Heuristic to Produce Almost Feasible Periodic Railway Timetables

by

Christian Liebchen
TU Berlin, Institut für Mathematik, Sekr. MA 6-1
Strasse des 17. Juni 136, D-10623 Berlin, Germany
liebchen@math.tu-berlin.de

No. 2005/06

A Cut-based Heuristic to Produce Almost Feasible Periodic Railway Timetables*

Christian Liebchen

TU Berlin, Institut für Mathematik, Sekr. MA 6-1
Straße des 17. Juni 136, D-10623 Berlin, Germany
liebchen@math.tu-berlin.de

Abstract. We consider the problem of satisfying the maximum number of constraints of an instance of the Periodic Event Scheduling Problem (PESP). This is a key issue in periodic railway timetable construction, and has many other applications, e.g. for traffic light scheduling.
We generalize two (in-) approximability results, which are known for MAXIMUM-K-COLORABLE-SUBGRAPH. Moreover, we present a deterministic combinatorial polynomial time algorithm. Its output violates only very few constraints for five real-world instances.

1 Introduction

The Periodic Event Scheduling Problem (PESP) has been introduced by Serafini and Ukovich ([20]). This powerful model has many practical applications, e.g. traffic light scheduling ([8]) and periodic railway timetabling. In the context of the second application, it has been exemplified in several studies that exact optimization can be very difficult for real-world instances ([18, 11, 13]). These studies made even use of sophisticated MIP models and several problem specific classes of valid inequalities.

Nevertheless, the PESP has proven to be sufficiently powerful to model the vast majority of the requirements which practitioners impose. For a concise review of the modeling capabilities of the PESP— including the minimization of the amount of rolling stock required to operate a timetable, and even some decisions of line planning—we refer to [10]. In particular, the 2005 timetable of the Berlin underground is the first one which has been computed by mathematical optimization methods ([9]). At the level of strategical planning, both Nederlandse Spoorwegen and Deutsche Bahn AG use software which is based on the PESP-model ([19, 11]).

The reported difficulties motivate to have a look at local search procedures. This does also hold for models for timetabling that are based on the quadratic semi-assignment problem (QSAP), for which Daduna and Voß consider the minimization of passenger transfer times ([4]). For some PESP-models, the genetic algorithm that has been proposed by Nachtigall and Voget ([15]), constitutes a competitive alternative ([11, 5]). However, sometimes it is already difficult to come up with a feasible solution, and the performance may depend on the quality of the initial population.

* Supported by the DFG Research Center “Mathematics for key technologies” (MATHEON) in Berlin

Contribution. This is the first time, the question of satisfying as many PESP-constraints as possible is addressed formally. We are able to generalize two (in-) approximability results that were established for MAXIMUM-K-COLORABLE-SUBGRAPH.

Moreover, we present a deterministic combinatorial polynomial time algorithm, whose output violates only very few constraints. In a computational study on five practical instances—ranging from long-distance traffic over regional traffic down to undergrounds—we exhibit its superiority compared to three other heuristics, two of these being previously published in related contexts.

2 Problem Definition

We start by defining the PESP formally, and then derive the problem of satisfying the maximum number of constraints of a PESP instance. We include the constant integer period time T of the input network in the name of the problem—just as with the constant K for K-VERTEX-COLORABILITY.

An instance $I = (D, \ell, u)$ of T-PESP consists of a directed graph $D = (V, A)$ and vectors ℓ and u of lower and upper time bounds for the arcs. As usual, we set $n := |V|$ and $m := |A|$, and we denote by $G(D)$ the underlying undirected graph of D. We may assume $\ell_a \leq u_a$. In the case of ℓ and u being integral vectors, we call an instance of T-PESP *integral*. A (feasible) solution of a T-PESP instance is a vector $\pi : V \to [0, T)$—which may represent time values of, say, hourly recurring departure/arrival events within a public transportation network—fulfilling periodic constraints of the form

$$(\pi_j - \pi_i - \ell_a) \bmod T \leq u_a - \ell_a,\ a = (i, j) \in A, \qquad (1)$$

or $\pi_j - \pi_i \in [\ell_a, u_a]_T$ for short. Sometimes, we will refer to a constraint (1) only by its arc a. In practice, often a (linear) objective function over the *slack times* $(\pi_j - \pi_i - \ell_a) \bmod T$ has to be minimized. We mention two simple but useful properties of T-PESP.

Lemma 1 ([20]). *If the underlying undirected graph $G(D)$ of D of an instance I of* T-PESP *is a forest, then I has a feasible solution.*

Remark 1. For every feasible integral instance of T-PESP, there exists an integral solution $\pi \in \{0, \ldots, T-1\}^V$. To that end, consider the MIP which results from modeling the mod-operators in the constraints by integer variables p. Then, for every fixed vector p the resulting LP is totally unimodular. Observe that if we multiply ℓ and u with the least common multiple of their denominators, we have to scale T by the very same factor.

Remark 2. Notice that we allow parallel arcs explicitly. They provide the ability to model disjunctive constraints ([20]), which are extremely useful in practice ([10]).

The input for MAX-T-PESP is identical to that of T-PESP. A solution of MAX-T-PESP is a vector $\pi : V \to [0, T)$ which maximizes

$$|\{A' \subseteq A \,|\, \pi_j - \pi_i \in [\ell_a, u_a]_T,\ \forall a = (i, j) \in A'\}|.$$

Besides providing initial solutions for local search algorithms for PESP minimization problems, MAX-T-PESP has an intrinsic application in practice. To that end, let us associate a weight with every constraint, which reflects its importance. Then, the goals of practical applications, as they are presented in [19] for Dutch railways (NS) and in [9] for Berlin underground, can be modeled immediately.

For instance, during the evenings' service—where T equals 10 minutes—the Berlin underground aims at offering a changeover waiting time of at most five minutes for a maximum number of connections. Currently, this can be offered for the 48 most important connections. Among the next 86, of approximately 110 remaining connections, 55 do not exceed this bound.

3 Approximability of MAX-T-PESP

Odijk ([16]) proposed the most convenient proof of the NP-completeness of T-PESP by polynomially transforming K-VERTEX-COLORABILITY to T-PESP. Recall that there are two canonical optimization variants of K-VERTEX-COLORABILITY. The most popular is to compute the chromatic number of a graph. But this question is of no practical relevance for the construction of periodic railway timetables, because the period time of the transportation system is a fixed constant—passengers would never accept a period time of, say, 53 minutes. If of any use at all, the chromatic number sometimes may provide a very rough idea of the total tightness of the system, and thus its theoretical capability to absorb delays.

In contrast, the other optimization variant, namely MAXIMUM-K-COLORABLE-SUBGRAPH, in which we seek for a K-coloring of the vertices such that a minimum number of edges relates two vertices sharing the same color, is indeed relevant to MAX-T-PESP—even to T-PESP, as has been motivated in Section 2. We summarize some of the main properties of MAXIMUM-K-COLORABLE-SUBGRAPH and relate them to MAX-T-PESP.

Theorem 1 ([17]). MAXIMUM-K-COLORABLE-SUBGRAPH *is MAXSNP-hard.*

Theorem 2. MAX-T-PESP *is MAXSNP-hard.*

Proof. We prove that MAXIMUM-K-COLORABLE-SUBGRAPH is L-reducible to MAX-T-PESP. Consider an instance of MAXIMUM-K-COLORABLE-SUBGRAPH being defined on a graph $G(V, E)$. Let $D = (V, A)$ be an arbitrary orientation of G. Set $T = K$ and define $\ell_a = 1$, $u_a = T - 1$ for every $a \in A$.

Let A' be any subset of A and consider the instance $I' := (D' = (V, A'), \ell|_{A'}, u|_{A'})$ of T-PESP. Due to Remark 1, we know that from every solution of I' we can derive an integral solution $\pi' \in \{0, \ldots, T-1\}^V$ of I' in polynomial time. By the choice of ℓ and u, we may interprete π' as a vertex coloring.

Let $E' \subseteq E$ be the projection of A' into G. Then, there exists a bijection between K colorings of G which respect the edges in E' and vectors $\pi \in \{0, \ldots, T-1\}$ which respect the constraints (1) for a set A'. Trivially, the above construction constitutes an L-reduction—in particular $\alpha = \beta = 1$. □

Corollary 1. *There is no PTAS for* MAX-T-PESP, *unless P=NP.*

Corollary 2. MAX-T-PESP *can be approximated within some fixed constant ratio.*

Proposition 1 ([21]). MAXIMUM-K-COLORABLE-SUBGRAPH *can be approximated with ratio* $\frac{K-1}{K}$.

Remark 3. Recall that for dense graphs, a PTAS for MAXIMUM-K-COLORABLE-SUBGRAPH can be derived using the techniques of Arora, Karger, and Karpinski ([1, 3]). Further, notice that the ratio of $\frac{K-1}{K}$ can be improved by applying the semidefinite programming techniques due to Goemans and Williamson ([6, 3]). However, this improvement becomes arbitrarily small for large values for K.

Proposition 1 might sound promising for railway timetabling. There, often T is at least 60 minutes. Moreover, the German railway infrastructure company computes at a precision of $\frac{1}{10}$ minute, which yields even $T = 600$. Further, bear in mind that we are able to respect the K-COLORING requirement for every arc being incident to vertices with degree at most $K - 1$.

We have to analyze, to what extent we may profit from Proposition 1 for MAX-T-PESP. Unfortunately, this turns out to be limited, in particular if an instance of MAX-T-PESP involves many constraints with small *span ratio* $\rho_a = \frac{\Delta_a+1}{T}$, $\Delta_a := u_a - \ell_a$ being the *span* of a constraint, i.e. with $\rho_a \ll 1$. Notice that the span ratio is invariant under any scaling of the time precision.

We say that a constraint (1) is *symmetric*, if $u_a = T - \ell_a$. Further, we call an instance of T-PESP *span homogeneous* if there exists an $\alpha \in [0, T)$ such that $\Delta_a = \alpha$ for every $a \in A$.

On the one hand, neither symmetric nor span homogeneous instances seem to have any practical motivation in railway timetabling. On the other hand, as they constitute the bridge to MAXIMUM-K-COLORABLE-SUBGRAPH, they will enable us to generalize both, algorithms and approximation guarantees from MAXIMUM-K-COLORABLE-SUBGRAPH to MAX-T-PESP. More specifically, in the following section we provide a ρ-approximation algorithm for the span homogeneous integral MAX-T-PESP.

4 Heuristics for MAX-T-PESP

We present four heuristics for MAX-T-PESP. Originally, the first and the second one were proposed to compute initial vectors for a backtracking procedure for T-PESP, or to compute starting vectors for local search procedures for T-PESP minimization. The third one has been inspired by Vitanyi's approximation algorithm for MAXIMUM-K-COLORABLE-SUBGRAPH, and serves as a kind of theoretical benchmark, too. The fourth one constitutes a considerable improvement of the first heuristic, and is the main target of the computational study in Section 5.

4.1 MST Heuristic

Algorithm 1 can already be found in the pioneering work of Serafini and Ukovich ([20]). It is based on Lemma 1 and thus ensures $n-1$ constraints of I to be satisfied. By choos-

Algorithm 1 MST heuristic ([20]) for MAX-T-PESP

Input: Instance $I = (D, \ell, u)$ of MAX-T-PESPwith D being a connected digraph
Output: Vector $\pi \in [0, T)^V$ and spanning tree $F \subseteq A$ of D, such that the vector π is feasible for all arcs $a \in F$

1: Compute a minimum spanning tree $F \subseteq A$ of D with respect to the arcs' span $\Delta_a = u_a - \ell_a$.
2: Choose an arbitrary $v \in V$.
3: Set $\pi_v := 0$ and $V' := \{\pi_v\}$.
4: **while** $V' \neq V$ **do**
5:     Choose $v \in \Gamma(V')$, i.e. a neighbor of V'. and let $a \in F$ be the arc which connects v to V'.
6:     Set π_v to a value in $[0, T)$ such that constraint a is satisfied.
7:     $V' \leftarrow V' \cup \{v\}$
8: **end while**

ing the spans as weights for the MST computation, we ensure that the $n - 1$ tightest—i.e. hopefully the most difficult—constraints are always satisfied.

A more detailed analysis requires to specify how to choose the value for π_v in Step 6. By the following lemma, we illustrate that a global rule which does not take into account local configurations of a specific instance provides only poor results.

Lemma 2. *For every $T \geq 4$, the approximation ratio of Algorithm 1 for* MAX-T-PESP *is $\Theta(\frac{n}{m})$, if in Step 6 π_v is selected such that the lower bound of constraint a becomes tight.*

Proof. Consider the complete graph K_n. Orient its edges from the smaller-indexed vertex to the larger-indexed vertex. Set $\ell_a = 0$ and $u_a = 1$ for all $a = (1, v)$, $v \in \{2, \ldots, n\}$, $\ell_a = 1$ and $u_a = T - 1$ for the remaining arcs.

Algorithm 1 builds a solution in which all vertices v have the same value π_v. Hence, precisely the $n - 1$ constraints induced by the tree arcs are satisfied.

In contrast, by setting

$$\pi_v := \begin{cases} 0, \text{ for } v \text{ odd, and} \\ 1, \text{ otherwise,} \end{cases}$$

the constraints of the $n-1$ tree arcs are still satisfied. Moreover, any arc which connects an odd vertex with an even vertex is satisfied. Roughly speaking, this is every second arc of K_{n-1}. Hence, $\Theta(m)$ constraints can be satisfied for this instance of MAX-T-PESP, providing an approximation ratio of only $\Theta(\frac{n}{m})$, or $\Theta(\frac{1}{n})$, for the selection we assume. □

Notice that if we apply Algorithm 1 to a planar graph, we are able to guarantee a fixed constant approximation ratio of $\frac{1}{3}$, because then we have $m \leq 3n - 6$. Let us close by mentioning that with a global rule for Step 6, the result of Algorithm 1 does not depend on the choice of v in Step 5.

4.2 Local Improvements

A vector π computed by Algorithm 1 can be improved locally by the following algorithm, which is due to Nachtigall and Voget ([15]). Observe that the performance of

Algorithm 2 Local improvement algorithm for MAX-T-PESP

Input: Instance $I = (D, \ell, u)$ of MAX-T-PESP and a vector $\pi \in [0, T)^V$
Output: Vector $\pi' \in [0, T)^V$
1: $\pi' \leftarrow \pi$
2: **for all** $v \in V$ **do**
3: Compute the minimal value $t \in [0, T)$ such that with $\pi'_v + t$ a maximum number of arcs in $\delta(\{v\})$ are satisfied.
4: $\pi'_v \leftarrow \pi'_v + t$
5: **end for**

Algorithm 2 depends on the order in which the vertices are processed.

Given a set of vertices X and a vector π, let us introduce sets $P(X, \pi)$ such that for the special case $X = \{v\}$ the set $P(\{v\}, \pi')$ contains candidate values for t in Step 3 of Algorithm 2,

$$P(X, \pi) := \bigcup_{a=(i,j)\in\delta^+(X)} \{((\pi_j - \pi_i) - \ell_a) \bmod T, ((\pi_j - \pi_i) - u_a) \bmod T\} \cup \bigcup_{a=(i,j)\in\delta^-(X)} \{(\ell_a - (\pi_j - \pi_i)) \bmod T, (u_a - (\pi_j - \pi_i)) \bmod T\} \cup \{0\}. \tag{2}$$

Lemma 3. *Let t be the choice of Algorithm 2, when processing vertex v. Then, $t \in P(\{v\}, \pi')$.*

Proof. We know that $\pi'_v + t$ is in the intersection of periodic intervals whose bounds we collect in $\{\pi'_v + t' \mid t' \in P(\{v\}, \pi')\}$. □

Corollary 3. *Algorithm 2 can always be implemented with runtime $O(m^2 n)$. For integral instances and integral input vectors π, we obtain $O(n \cdot \min\{m, T\} \cdot m)$.*

Proof. We know $|P(\{v\}, \pi')| \leq 2m + 1$. Remark 1 guarantees $|P(\{v\}, \pi')| \leq T$ for integral instances. □

Notice that we may face $n \in o(\delta(\{v\}))$ because of Remark 2.

Remark 4. Further improvements can be achieved by executing Algorithm 2 repeatedly.

4.3 An Approximation Algorithm for Span Homogeneous MAX-T-PESP

The main idea of Algorithm 3 is similar to the one of Algorithm 2. Surely, they are of a different flavor, as Algorithm 3 does not require any input vector π. However, observe that the only difference between Algorithm 3 and applying Algorithm 2 to $\pi = \mathbf{0}$ is that here only such arcs are taken into account that connect the current vertex with vertices that were already processed. Hereby, the following lemma holds. It will enable us to compute a lower bound on the fixed constant approximation ratio of MAX-T-PESP restricted to span homogeneous integral instances, cf. Corollary 2.

Lemma 4. *For every arc $a \in A$ there exists precisely one vertex $v \in V$ such that $a \in A'$ in Step 3 of Algorithm 3.*

Algorithm 3 Approximation algorithm for span homogeneous integral MAX-T-PESP

Input: Instance $I = (D, \ell, u)$ of MAX-T-PESP
Output: Vector $\pi \in [0, T)^V$

$V' \leftarrow \emptyset$
for all $v \in V$ **do**
  $A' \leftarrow \{a \in A \mid \exists u \in V' : a = (u, v)$ or $a = (v, u)\}$
  Set π_v to the minimum value in $[0, T)$, such that a minimum number of constraints $a \in A'$ are violated.
  $V' \leftarrow V' \cup \{v\}$
end for

Proof. The arc $a = (u, v) \in A$ is in A', if and only if the second of its two vertices is processed by the for-loop. □

Theorem 3. *When applied to a span homogeneous integral instance $I = (D, \ell, u)$ of* MAX-T-PESP *with span Δ, the output vector π produced by Algorithm 3 satisfies at least $\frac{\Delta+1}{T}|A|$ constraints of I.*

Proof. Due to Lemma 4, we may decompose the analysis into the n iterations of the for-loop. Moreover, by Remark 1 and an analogon of Lemma 3, we may focus on integer valued vectors π.

Let us count the number of arcs that are feasible if we set π_v to t,

$$f_v(t) := |\{a \in A' \mid a \text{ is feasible for } \pi_v = t\}|.$$

By our assumption on the span homogeneity of I, for every $a \in A'$ there are precisely $\Delta + 1$ values for π_v, such that a is respected. This yields

$$\sum_{t=0}^{T-1} f_v(t) = |A'| \cdot (\Delta + 1).$$

Trivially, there exists some $t \in \{0, \ldots, T-1\}$ such that

$$f_v(t) \geq \left\lceil \frac{\Delta + 1}{T} |A'| \right\rceil.$$

□

Corollary 4. *Algorithm 3 is a ρ-approximation algorithm for the span homogeneous integral* MAX-T-PESP *with span ratio ρ.*

Proposition 2. *The runtime of Algorithm 3 can be bounded by the runtime of Algorithm 2.*

Proof. From applying Algorithm 2 to the input vector $\pi \equiv 0$ it becomes obvious that we are able to find π_v immediately in a subset of $P(\{v\}, \pi)$. Hence, the bounds of Algorithm 2 in Corollary 3 apply. □

Remark 5. By considering symmetric span homogeneous instances of MAX-T-PESP with $\ell = \mathbb{1}$ it gets obvious that Theorem 3 is a generalization of Vitanyi's result ([21]) for MAXIMUM-K-COLORABLE-SUBGRAPH. However, both the algorithm and its analysis became more direct than in the original paper. Finally, we may conclude that an approximation ratio of ρ cannot be tight—at least for symmetric instances, cf. Remark 3.

4.4 Cut Improvements of the MST Heuristic

The last heuristic which we propose is motivated by the poor quality of Algorithm 1, and by Lemma 1. For a spanning tree $F \subseteq A$ and one of its arcs $a \in F$ consider the set C_a of arcs of the fundamental cut induced by a and F. Observe that for every $a' \in C_a \setminus \{a\}$ there is a unique cycle in $F \cup \{a'\}$. Algorithm 4 will take care of these cycles.

Algorithm 4 Cut-based improvements for MAX-T-PESP

Input: Instance $I = (D, \ell, u)$ of MAX-T-PESP with D being a connected digraph, a vector $\pi \in [0, T)^V$, and a spanning tree F
Output: Vector $\pi' \in [0, T)^V$
1: $\pi' \leftarrow \pi$
2: **for all** $a \in F$ **do**
3: Let $C_a \subseteq A$ be the arc set of the fundamental cut induced by the arc $a = (i, j)$ and the tree F, and define $X \subset V$ such that $\{i\} \in X$ and $\delta(X) = C_a$.
4: Compute the minimal value $t \in [0, T)$ such that $\pi' + t \cdot \chi_X$ satisfies a maximum number of constraints in $\delta(X)$. $\{\chi_X \in \{0, 1\}^V$ being the characteristic vector of $X\}$
5: $\pi' \leftarrow \pi' + t \cdot \chi_X$
6: **end for**

We refer to applying the cut improvements of Algorithm 4 to an output vector π of Algorithm 1 as *cut heuristic*.

Our computational study in Section 5 will reveal the notable benefi t Algorithm 4 achieves when compared to Algorithm 3 and to the MST heuristic—even after the local improvement heuristic has been applied. Unfortunately, the following theoretical results are not able to reflect the practical quality of the cut heuristic. This is mainly caused by non-empty pairwise intersections of the fundamental cuts and by the two-stage character of the cut heuristic—even three stage, when including local improvements.

Lemma 5. *Consider the strategy in which the lower bounds of the constraints become tight for the tree arcs. Then, for every $T \geq 6$ there are feasible instances of* T-PESP, *for which the cut heuristic fails to produce a feasible solution.*

Proof. Consider the 6-PESP instance in Figure 1. After shifting any solution such that $\pi_1 = 0$, $\pi^\star = (0, 0, 1, 2)^t$ becomes the unique feasible solution.

Algorithm 1 produces $\pi \equiv 0$ as output vector. The only non-tree arc is violated. Hence, the cut heuristic will only change π, if it can obtain $\pi = \pi^\star$—occasionally after shifting. But this is impossible, because $\pi^\star$ carries three distinct values and in every iteration of Algorithm 4, only one single offset can be applied to some set of vertices. □

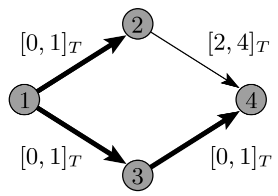

Fig. 1. Feasible instance of 6-PESP, but the cut heuristic fails to produce a feasible solution

Proposition 3. *The runtime of the cut heuristic is $O(m^2 n)$. For integral instances and integral input vectors π, we achieve $O(n \cdot \min\{m, T\} \cdot m)$.*

Proof. We know $|\delta(X)| \leq m$ and $|F| = n-1$. Further, we will find t in the set $P(X, \pi)$ and, hence, the analysis of Corollary 3 deploys. □

Proposition 4. *For every $T \geq 4$ there are instances of* MAX-T-PESP*, where Algorithm 4 examines $\Theta(nm)$ arcs in total.*

Proof. Consider the complete graph K_n. Again, orient its edges from the smaller-indexed vertex to the larger-indexed vertex. Define the feasible intervals of the constraints as follows:

$$[\ell_a, u_a]_T = \begin{cases} [0,1]_T, \text{ if } a = (i, i+1),\ i = 1, \ldots, n-1, \\ [1,3]_T, \text{ otherwise.} \end{cases}$$

Thus, the spanning tree will be a path.

Let us assume $n = 3k+1, k \in \mathbb{N}$, for notational convenience. Consider the $\frac{n-1}{3}$ fundamental cuts that are induced by the tree arcs $a = (i, i+1), i = k+1, \ldots, 2k$. Each of these contains all the

$$(k+1)^2 = \left(\frac{n+2}{3}\right)^2 = \left(\frac{\sqrt{2m + \frac{1}{4}}}{3} + \frac{5}{6}\right)^2 = \Theta(m)$$

arcs $a = (i, j)$ with $i = 1, \ldots, k+1$ and $j = 2k+1, \ldots, 3k+1$. □

Remark 6. Observe that the cut heuristic immediately extends to the so-called EXTENDED-PESP, in which the vertices may have different period times. There, local search algorithms are even more relevant, because MIP solvers face yet more problems, although sophisticated problem formulations are used (Hermite normal form ([14]), generalizations of valid inequalities and of the cycle periodicity formulation, which are known for T-PESP ([7])).

5 Computational Study

We will compare the cut heuristic for MAX-T-PESP to Algorithm 1. Further, we analyze how the local improvement heuristic performs when applied to the output vectors of these two algorithms. We will also apply the approximation algorithm for span homogeneous instances to the five practical data sets that we consider. We start by describing these. Notice that each of these five instances permits feasible timetables.

The first pair of data sets, ICE small and ICE big, share the same basic network. In particular, ICE small is a subset of ICE big, resulting from the deletion of certain traffic lines. In turn, the lines contained in ICE big are a subset of a strategic planning scenario of Deutsche Bahn AG. Beyond the 31 pairs of directed two-hourly traffic lines which are contained in ICE big, it consists of seven more pairs of two-hourly lines, as well as several four-hourly variants. Hence, ICE small and ICE big share large parts of their structure. However, since the underlying infrastructure has the same capacity for the two scenarios, it shall be easier to construct a feasible timetable for ICE small than for ICE big. Notice that many constraints are to ensure a minimal headway between two successive trains. And there are even some single tracks[1] in these high-speed networks, e.g. the famous “Hildesheimer Kurve”.

The data set ICE big has been the subject of an earlier extended computational study ([11]). To motivate the potential for local search techniques, we quote its major results. Most of them relate to the performance of CPLEX©, version 8.1 ([2]). Notice that the goal was to minimize a linear objective function over ICE big.

- For each of about ten promising parameter settings, CPLEX© is not able to solve ICE big to optimality within two days (Intel 2.8 GHz, memory limit 512 MB).
- A genetic algorithm outperforms CPLEX© under default parameter settings, i.e. finds better solutions earlier.
- With the parameter settings, which yield the best solution, it takes CPLEX© more than two hours to construct a first feasible solution—with other parameter settings, this takes only a couple of minutes.
- A constraint programming algorithm is able to construct a feasible solution instantaneously—however, it is not able to improve its poor quality.

The second pair of data sets model the regional service of one of the five largest federal states of Germany. In PS Regio, only the regional trains within that state are considered. In PS all, we also include the long-distance traffic which serves that state and fix the timetables on tracks outside the federal state to a planning scenario provided by Deutsche Bahn AG. Notice that many of the regional trains have to use single tracks. Moreover, in the geographic intersection of PS all and ICE big, there is much more traffic in PS all, because the full passenger traffic is included. Due to Remark 6, the algorithms to be investigated are able to deal with the three different periods that occur in these data sets.

The last data set models the Berlin underground. More specifically, one tries to ensure a maximal waiting time of five minutes for the 48 most important connections. But this must not happen at the price of stopping times which exceed 2.5 minutes—travel times between stations being fixed. Also, some technical constraints have to be obeyed, e.g. crossings in front of terminus stations.

Tables 1 and 2 provide additional information on the real-world instances, and on the resulting graph models, respectively. Notice that for the latter, we only mention

[1] A single track is a track that is used in both directions.

classification numbers for the graph, in which redundancies have been eliminated by so-called contraction steps ([12]). Furthermore, we ignore arcs with $\Delta_a \geq T - 1$.

Table 1. Classification numbers of the real-world problems

Quantity	ICE small	ICE big	PS Regio	PS all	U Berlin
Period times of the lines	120’	120’	60’, 120’	30’, 60’, 120’	10’
Time precision	60”	60”	6”	6”	30”
Pairs of traffic lines	11	31	66	98	8
# fixed pairs of lines	0	0	0	40	1
# partly fixed pairs of lines	0	0	0	9	0

Given the fact that the first four data sets make even use of parallel arcs, cf. Remark 2, none of the data sets induces a dense graph, cf. Remark 3.

Table 2. Classification numbers of the resulting graph models

Quantity	ICE small	ICE big	PS Regio		PS all	U Berlin
Period time T	120	120	600	1200	$\leq$1200	20
Number of vertices n	69	173	160	192	343	38
Number of arcs m	304	1102	341	387	1224	83
Cyclomatic number μ	236	930	377		882	46
Average span ratio $\overline{\rho}$	73.6%	82.5%	61.4%	55.9%	—	32.3%
Max. number of violated arcs for $\overline{\rho}$ span homog. instances	80	192	—		—	57

Computational Results. Let us start by filling the theoretical benchmarks in the last row of Table 2 with life. Just as it has been experienced with many other approximation algorithms, when applied to instances that arise in practice, Algorithm 3 leaves much less PESP-constraints violated than can be guaranteed in general.

Notice that we investigate three different strategies for Algorithm 3 to select the next vertex v: by index (increasingly), by the vertices’ degree (decreasingly), and by the intensity of the incident constraints, $\sum_{a \in \delta(\{v\})} T - (\Delta_a + 1)$, (decreasingly). Table 3 shows that none of these strategies dominates the two others.

Before presenting the results of our computations for the cut heuristic, we specify how we made use of the degrees of freedom left by Algorithms 1 and 4. For the MST heuristic, we examine several strategies for setting the values of the vertices. More precisely, we run Algorithm 1 for eleven *target spans* $p \in \{\frac{k}{10} \,|\, k \in \{0, \ldots, 10\}\}$, and compute π such that

$$(\pi_v - \pi_u - \ell_a) \bmod T = p \cdot \Delta_a,$$

for every arc $a = (u, v) \in F$.

Table 3. Number of infeasibilities left by Algorithm 3

Strategy	ICE small	ICE big	PS Regio	PS all	U Berlin
Algorithm 3					
index	**6**	**22**	**31**	81	12
degree	15	30	58	72	19
intensity	18	28	42	**52**	**7**
Algorithm 3 plus local improvement					
index	**4**	12	**25**	47	8
degree	8	**11**	36	41	14
intensity	9	14	34	**38**	**6**

For the cut improvements, we choose cuts that have (many) infeasible arcs and many arcs with small span ratio foremost.

Definition 1 (Score of a Fundamental Cut). *For a fundamental cut with arc set* C_a *that is induced by some tree arc* a*, and a vector* π *we define its* score $\sigma(a, \pi)$ *as follows*

$$\sigma(a,\pi) := \frac{\sum_{e \in C_a,\, e \text{ feasible for } \pi} w(e)}{\text{SHIFT} + \sum_{e \in C_a} w(e)},$$

with

$$w(e) := \begin{cases} \text{TIGHT\_FACTOR}, & \text{if } \rho_e < \text{TIGHT\_LIMIT and} \\ 1, & \text{otherwise.} \end{cases}$$

In Step 2 of Algorithm 4 we always select an arc which induces the fundamental cut with highest score but not being processed so far. Our choice for the remaining parameters is as follows: SHIFT $= 2$, TIGHT_FACTOR $= 3$, and TIGHT_LIMIT $= 0.75$.

Furthermore, we apply the local improvement (Algorithm 2) repeatedly, until no more change appears, cf. Remark 4.

Figure 2 reports the performance of the four combinations of heuristics (cf. Table 4) that we investigate. Every chart represents the results obtained for one of the five data sets. For every data set, the four heuristics have been executed eleven times each, with

Table 4. Heuristics for the computational study

Name	Algorithm 1 (MST)	Algorithm 4 (cuts)	Algorithm 2 (local)
tree	√	–	–
tree locImp	√	–	√
cut	√	√	–
cut locImp	√	√	√

different values for the target span to be applied by the MST heuristic. On the ordinate, the number of arcs that are violated by the output vector of a heuristic is given.

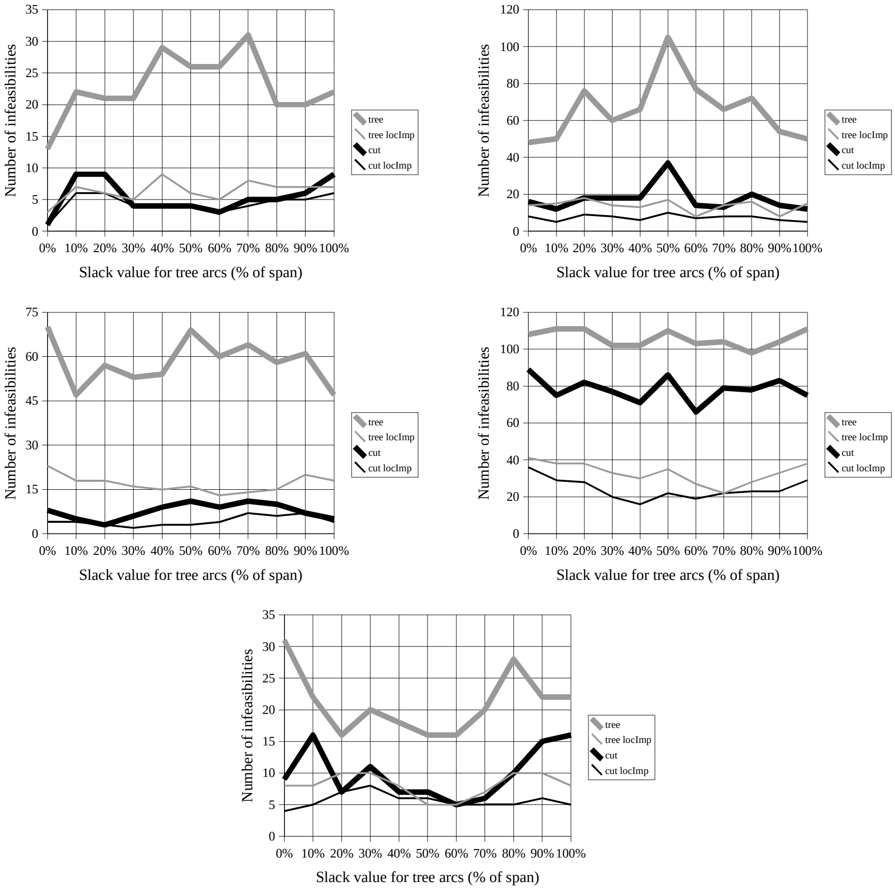

Fig. 2. Performance of MST heuristic and cut heuristic plus occasional local improvement for five data sets: ICE small, ICE big, PS Regio, PS all, and U Berlin (from left to right)

Let us summarize the main observations to be torn out of Figure 2 and relate them to the practical performance of the approximation algorithm for span homogeneous instances:

– Both, Algorithm 3 and the cut heuristic are much superior to the pure MST heuristic. With the exception of PS all, the cut heuristic also outperforms Algorithm 3 significantly.

– For each of the five data sets, there is a target span such that the cut heuristic plus local improvements behaves better than both, the MST heuristic plus local improvements and Algorithm 3 plus local improvements.
– For PS Regio, the worst solution obtained by the pure cut heuristic is still better than the best solution obtained by any locally improved output of the MST heuristic and of Algorithm 3.

Notice that the local improvement is most powerful for the MST heuristic and for the cut heuristic. But this is not very surprising, because only there it adds a completely different perspective.

Recall that one motivation for considering heuristics for MAX-T-PESP has been to compute good initial input vectors for local search algorithms. Unfortunately, we have to admit that in some spot tests on ICE big, we did not perceive any significant improvement in the performance of a genetic algorithm, when fed with locally improved output vectors of the cut heuristic.

6 Conclusions

We addressed the problem of satisfying as many constraints of a PESP-instance as possible. We proved it to be MAXSNP-hard and provided an approximation algorithm with fixed constant approximation ratio ρ, ρ being the span ratio of a span homogeneous integral instance of MAX-T-PESP. Moreover, we proposed a new heuristic which provides much better computational results for MAX-T-PESP than two heuristics that were previously published.

Unfortunately, the theoretical analysis of the cut heuristic stays limited. Hopefully, our promising computational results attract other researchers to join the theoretical analysis of the cut heuristic—or even to design further approximation algorithms for MAX-T-PESP. This would really be of importance, because PESP-techniques have just entered the practice of timetable design. And practice bears many instances, on which the existing algorithms still leave some space for improvements that, in turn, are really required by practice. . .

7 Acknowledgments

I would like to thank Stefan Felsner and Marco Lübbecke for initial hints and helpful discussions.

References

1. Arora, S., Karger, D., and Karpinski, M. (1995) Polynomial Time Approximation Schemes for Dense Instances of NP-hard Problems. Proceedings of the 27th Annual ACM Symposium on Theory of Computing, 284–293, ACM Press, New York
2. CPLEX 8.1 (2004) http://www.ilog.com/products/cplex, ILOG SA, France.
3. Crescenzi, P. and Kann, V. (2005) A compendium of NP optimization problems. http://www.nada.kth.se/~viggo/problemlist/compendium.html

4. Daduna, J.R. and Voß, S. (1995) Practical Experiences in Schedule Synchronization. In Daduna, J.R. et al.: Computer-Aided Transit Scheduling—Proceedings of the Sixth International Workshop on Computer-Aided Scheduling of Public Transport, LNEMS 430, 39–55, Springer
5. Engelhardt-Funke, O. and Kolonko, M. (2004) Analysing stability and investments in railway networks using advanced evolutionary algorithms. International Transactions in Operational Research **11**, 381–394
6. Goemans, M. and Williamson, D. (1994) .878-Approximation Algorithms for MAX CUT and MAX 2SAT. Proceedings of the 26th Annual ACM Symposium on Theory of Computing, 422–431, ACM Press, New York
7. Haenelt, S. (2004) Taktfahrplanoptimierung mit unterschiedlichen Taktzeiten—Verallgemeinerung von Lösungsverfahren für den Eintaktfall. Diploma Thesis, TU Berlin, Germany, In German
8. Hassin, R. (1996) A Flow Algorithm for Network Synchronization. Operations Research **44**, 570–579
9. Liebchen, C. (2005) Der Berliner U-Bahn Fahrplan 2005: Realisierung eines mathematisch optimierten Angebotskonzepts. Tagungsbericht Heureka '05 Optimierung in Transport und Verkehr, FGSV Verlag, to appear, in German
10. Liebchen, C., Möhring, R.H. (2004) The Modeling Power of the Periodic Event Scheduling Problem: Railway Timetables—and Beyond. Preprint 020/2004, Mathematical Institute, TU Berlin, Germany
11. Liebchen, C., Proksch, M., and Wagner, F.H. (2004) Performance of Algorithms for Periodic Timetable Optimization. Preprint 021/2004, Mathematical Institute, TU Berlin, Germany
12. Lindner, T. (2000) Train Schedule Optimization in Public Rail Transport. Ph.D. Thesis, TU Braunschweig, Germany
13. Martin, A., Achterberg, T., and Koch, T. (2003) MIPLIB 2003, `http://www.zib.de/miplib`
14. Nachtigall, K. (1996) Periodic network optimization with different arc frequencies. Discrete Applied Mathematics **69**, 1–17
15. Nachtigall, K. and Voget, S. (1996) A genetic algorithm approach to periodic railway synchronization. Computers and Operations Research **23**, 453–463
16. Odijk, M. (1997) Railway Timetable Generation. Ph.D. Thesis, TU Delft, The Netherlands
17. Papadimitriou, C.H. and Yannakakis, M. (1991) Optimization, Approximation, and Complexity Classes. Journal of Computer and System Sciences **43**, 425–440
18. Peeters, L. (2003) Cyclic Railway Timetable Optimization. Ph.D. Thesis, Erasmus Universiteit Rotterdam, The Netherlands
19. Schrijver, A. and Steenbeek, A. (1993) Dienstregelingontwikkeling voor Nederlandse Spoorwegen N.V.—Rapport Fase 1. Report, CWI, The Netherlands, In Dutch
20. Serafini, P., Ukovich, W. (1989) A mathematical model for periodic scheduling problems. SIAM Journal on Discrete Mathematics **2**, 550–581
21. Vitanyi, P.M.B. (1981) How Well Can a Graph be n-Colored? Discrete Mathematics **34**, 69–80

Reports from the group

"Combinatorial Optimization and Graph Algorithms"

of the Department of Mathematics, TU Berlin

2005/06 *Christian Liebchen:* A Cut-based Heuristic to Produce Almost Feasible Periodic Railway Timetables

2005/03 *Nicole Megow, Marc Uetz, and Tjark Vredeveld:* Models and Algorithms for Stochastic Online Scheduling

2004/037 *Laura Heinrich-Litan and Marco E. Lübbecke:* Rectangle Covers Revisited Computationally

2004/35 *Alex Hall and Heiko Schilling:* Flows over Time: Towards a more Realistic and Computationally Tractable Model

2004/31 *Christian Liebchen and Romeo Rizzi:* A Greedy Approach to Compute a Minimum Cycle Bases of a Directed Graph

2004/27 *Ekkehard Köhler and Rolf H. Möhring and Gregor Wünsch:* Minimizing Total Delay in Fixed-Time Controlled Traffic Networks

2004/26 *Rolf H. Möhring and Ekkehard Köhler and Ewgenij Gawrilow and Björn Stenzel:* Conflict-free Real-time AGV Routing

2004/21 *Christian Liebchen and Mark Proksch and Frank H. Wagner:* Performance of Algorithms for Periodic Timetable Optimization

2004/20 *Christian Liebchen and Rolf H. Möhring:* The Modeling Power of the Periodic Event Scheduling Problem: Railway Timetables — and Beyond

2004/19 *Ronald Koch and Ines Spenke:* Complexity and Approximability of k-splittable flow problems

2004/18 *Nicole Megow, Marc Uetz, and Tjark Vredeveld:* Stochastic Online Scheduling on Parallel Machines

2004/09 *Marco E. Lübbecke and Uwe T. Zimmermann:* Shunting Minimal Rail Car Allocation

2004/08 *Marco E. Lübbecke and Jacques Desrosiers:* Selected Topics in Column Generation

2003/050 *Berit Johannes:* On the Complexity of Scheduling Unit-Time Jobs with OR-Precedence Constraints

2003/49 *Christian Liebchen and Rolf H. Möhring:* Information on MIPLIB's timetab-instances

2003/48 *Jacques Desrosiers and Marco E. Lübbecke:* A Primer in Column Generation

2003/47 *Thomas Erlebach, Vanessa Kääb, and Rolf H. Möhring:* Scheduling AND/OR-Networks on Identical Parallel Machines

2003/43 *Michael R. Bussieck, Thomas Lindner, and Marco E. Lübbecke:* A Fast Algorithm for Near Cost Optimal Line Plans

2003/42 *Marco E. Lübbecke:* Dual Variable Based Fathoming in Dynamic Programs for Column Generation

2003/37 *Sándor P. Fekete, Marco E. Lübbecke, and Henk Meijer:* Minimizing the Stabbing Number of Matchings, Trees, and Triangulations

2003/25 *Daniel Villeneuve, Jacques Desrosiers, Marco E. Lübbecke, and François Soumis:* On Compact Formulations for Integer Programs Solved by Column Generation

2003/24 *Alex Hall, Katharina Langkau, and Martin Skutella:* An FPTAS for Quickest Multicommodity Flows with Inflow-Dependent Transit Times

2003/23 *Sven O. Krumke, Nicole Megow, and Tjark Vredeveld:* How to Whack Moles

2003/22 *Nicole Megow and Andreas S. Schulz:* Scheduling to Minimize Average Completion Time Revisited: Deterministic On-Line Algorithms

2003/16 *Christian Liebchen:* Symmetry for Periodic Railway Timetables

2003/12 *Christian Liebchen:* Finding Short Integral Cycle Bases for Cyclic Timetabling

762/2002 *Ekkehard Köhler and Katharina Langkau and Martin Skutella:* Time-Expanded Graphs for Flow-Dependent Transit Times

761/2002 *Christian Liebchen and Leon Peeters:* On Cyclic Timetabling and Cycles in Graphs

752/2002 *Ekkehard Köhler and Rolf H. Möhring and Martin Skutella:* Traffic Networks and Flows Over Time

739/2002 *Georg Baier and Ekkehard Köhler and Martin Skutella:* On the k-splittable Flow Problem

736/2002 *Christian Liebchen and Rolf H. Möhring:* A Case Study in Periodic Timetabling

Reports may be requested from:

Sekretariat MA 6–1
Fakultt II – Institut fr Mathematik
TU Berlin
Straße des 17. Juni 136
D-10623 Berlin – Germany

e-mail: klink@math.TU-Berlin.DE

Reports are also available in various formats from

`http://www.math.tu-berlin.de/coga/publications/techreports/`

and via anonymous ftp as

`ftp://ftp.math.tu-berlin.de/pub/Preprints/combi/Report-`*number*`-`*year*`.ps`