

Strong Formulations for the Multi-module PESP and a Quadratic Algorithm for Graphical Diophantine Equation Systems

Laura Galli¹ and Sebastian Stiller²

¹ DEIS, Alma Mater University of Bologna, Italy

² Institut für Mathematik, Technische Universität Berlin, Germany

Abstract. The Periodic Event Scheduling Problem (PESP) is the method of choice for real-world periodic timetabling in public transport. Its MIP formulation has been studied intensely for the case of uniform modules, i.e., when all events have the same period. In practice, multiple periods are equally important. Yet, the powerful methods developed for uniform modules generally fail for the multi-module case. We analyze a certain type of Diophantine equation systems closely related to the multi-module PESP. Thereby, we identify a structure, so-called *sharp trees*, that allows to solve the system in $\mathcal{O}(n^2)$ time if the modules form a linear lattice. Based on this we develop the machinery to solve multi-module PESP on real-world scale. In our computational results the new MIP-formulations considerably improve the solvability of multi-module PESP.

1 Introduction

The Periodic Event Scheduling Problem (PESP) is a combinatorial optimization problem of great practical importance introduced by [16]. It is the model of choice for periodic timetabling in public transport [13],[14], has been used for periodic job shop [16] and traffic signal scheduling [7],[5], and has successfully and repeatedly been applied in practice.

The task is to schedule periodic recurring events, e.g., the arrivals and departures of trains, such that between pairs of periodic events periodic constraints are fulfilled. A periodic constraint between periodic events i and j means that for every realization of i there is at least one realization of j with time difference greater or equal some lower respectively less or equal some upper bound modulo the period of the application. Such constraints can model for periodic timetables the headway constraints, the stopping times, or that passengers can quickly transfer between trains of different lines. In fact, the high modeling power of the PESP (cf. [8]) even allows to include rolling stock minimization and crew scheduling into the timetabling. This facilitated the construction of the first mathematically optimized railway timetable, namely for the Berlin underground in 2005.

The PESP is most powerfully solved as a (mixed) integer linear program ((M)IP). Apart from their practical importance, the IPs arising from the PESP

are of independent theoretical interest. Any such IP is naturally associated to a digraph, whose nodes and arcs represent respectively periodic events and periodic constraints among them. There are node variables π_i for each node i and offset variables k_a for each arc a . Two constraints correspond to each arc $a = (i, j)$: $\ell_a \leq \pi_j - \pi_i + k_a P \leq u_a$. Thus, for each arc the difference of the node variables must be in the interval $[\ell_a, u_a]$ modulo a constant P . This constant is the period of the application, e.g., the time that elapses until the next train of the same line arrives. This special class of IPs has attracted a lot of research attention from combinatorics and integer programming perspectives. It has been shown to be NP-complete in [11,12], even MAXSNP-hard in [8], and to have an unbounded Chvatal rank [9]. Nevertheless, research efforts exploiting the graph structure have led to a rich understanding of these IPs. These insights facilitate methods that are eventually capable to solve instances of considerable and practically relevant size—e.g., the timetable optimization of a complete national railway system. Some of these instances are challenging enough to be found in the MIPLIB. (These are the instances on which we base the computational validation of our method.)

The strong methods for uniform-module PESP have three ingredients (cf. [10]): first, if the instance is feasible, then for any tree there exist optimal solutions that have offsets equal to zero on that tree; second, a strong MIP formulation can be constructed using an integral cycle basis; and finally, a class of rounding cuts (the Odijk inequalities), also based on a well chosen set of cycles, can further improve the strength of the model. All of this fails if the modules are not uniform. In general, solutions need non-zero offsets on all arcs. In general, an integral cycle basis does not give an equivalent formulation, and the Odijk inequalities become arbitrarily loose for multiple modules. Yet multi-module PESP are justified and even desired from the applications perspective: A public transport system is often comprised of bus, subway, and commuter lines with different periods. Also traffic lights in the same urban area may have different periods.

For solving PESP a special class of linear Diophantine equation systems (DES) naturally related to the PESP is important. Again, given a digraph one associates variables π to the nodes and k to the arcs, and each arc represents an equation of the form: $\pi_j - \pi_i + k_a P_a = x_a$. Note, the coefficients of the node variables are equal to 1. Setting the arc coefficients $P_a = 1$ would allow for a trivial solution $k \equiv x, \pi \equiv 0$. In this sense the proposed class is the simplest non-trivial class of DES one can associate to a digraph. Therefore, we call them Graphical Diophantine Equation Systems (GDES). GDESs are not only similar to PESP, they also play a role in the state-of-the-art solving methods for PESP. In the special case, when all modules P_a are equal, the GDES can be solved in linear time by a straight forward algorithm. For the general case so far one has to resort to constructing the Hermite Normal Form (HNF) of the GDES matrix, which can be done in (high) polynomial time [15].

In this work, starting from an analysis of and a new algorithm for GDES, we develop a method capable of solving real-world multi-module PESP. We

test this on multi-period instances which we get by changing the periods in the uniform PESP instances that can be found in the MIPLIB, namely, `timtab1` and `timtab2`. These MIPLIB instances are real-world railway timetabling instances.

Related work: A concise exposition of the PESP and the state-of-the-art theory and solving methods can be found in [8]. In [12] a set of particularly useful rounding cuts have been proposed for the first time. In [3] optimization over the first Chvatal rank of the PESP has been studied.

Diophantine equations systems can be solved as any other linear equation system, once they are presented by their Hermite Normal Form (HNF). There is a standard polytime algorithm for constructing the HNF [15]. While further research on the HNF focuses on algorithms that use less space, we are not aware of a specific algorithm for GDES. The GDES are a special, and particularly simple class of DES. Similar flavor the Mixing Set Problem defines a particularly simple class of Diophantine inequalities that has been studied intensively [2],[1] and also has practical applications. Another very important class of DES that are closely related to GDES are the so-called *unique games* [6].

For practical instances which feature two different modules $P \mid P'$ (e.g., a system of trains of which some run every hour and others every two hours) the state-of-the-art [8] approach is to use a PESP with uniform module P' and double the events that have higher frequency. The duplications of such an event have to be mutually fixed by additional constraints. This increases the size of the PESP.

Our Contribution and Outline: In Section 2 we present the so-called *sharp tree* structure and show that if that exists then the offsets can be chosen zero on it. These trees exist and can be found in $\mathcal{O}(n^2)$ time, if the modules are nested, i.e., form a linear lattice with respect to division. In Section 3 we show that the existence of a sharp tree allows for a cycle basis formulation, which is equivalent to the original arc formulation and gives a stronger IP. Moreover, we show that we can prune a multi-module PESP, such that the fundamental cycles of a sharp tree give particularly good inequalities. The same algorithm that finds a sharp tree in case of nested modules, also solves a GDES in time quadratic in the number of nodes (cf. Section 2). We also show that without the existence of a sharp tree similar approaches to strengthen the IP formulation or to quickly solve a GDES do not extend in general. In the final section we report on twelve instances derived from the afore mentioned MIPLIB timetabling problems.

Definitions and Basics: Firstly, we define the two main mathematical objects under consideration.

Definition 1. A digraph $G(V, A)$ together with a natural valued function on the arc set, $P : A \rightarrow \mathbb{N}$, and an integer valued function on the arc set $x : A \rightarrow \mathbb{Z}$ is called a graphical representation of the following system of Diophantine equations on variable vectors $(\pi, k) \in \mathbb{Z}^{|V|+|A|}$:

$$\pi_j - \pi_i + k_a P_a = x_a \quad \forall a(= (i, j)) \in A. \tag{1}$$

A system for which a graphical representation exists is called a graphical Diophantine equation system (GDES).

The results we derive also hold if x maps to the rationals and P can be negative, but we can restrict w.l.o.g. (cf. [8]) to natural numbers for simplicity and also for its significance in a practical context. From a GDES one can straight forward construct its representation and this representation is unique up to isomorphism. So we speak of *the representation* of a GDES.

Definition 2. *Given a digraph $G(V, A)$ together with two rational functions on the arc set, $\ell : A \rightarrow \mathbb{Q}$ and $u : A \rightarrow \mathbb{Q}$, a third natural valued function on the arcs, $P : A \rightarrow \mathbb{N}$, and a cost vector $c \in \mathbb{Q}^{|A|}$, the following mixed integer program is called a classical formulation for the periodic event scheduling problem (PESP):*

$$\min \sum_{(i,j)=a \in A} c_a(\pi_j - \pi_i + k_a P_a) \tag{2}$$

$$\ell_a \leq \pi_j - \pi_i + k_a P_a \leq u_a \quad \forall a(= (i, j)) \in A \tag{3}$$

$$\pi \in \mathbb{Q}^{|V|}, k \in \mathbb{Z}^{|A|}. \tag{4}$$

In most applications the events, i.e., the nodes—not primarily the arcs—are periodic. So the constraints should rather read: $\ell_a \leq (\pi_j + k_j P_j) - (\pi_i + k_i P_i) \leq u_a$. It is easily checked that this is equivalent to the formulation above, when we choose $P_a = \gcd\{P_i, P_j\}$.

We will use subscripts for the arguments of the functions x, u, ℓ and P in the remainder. We abbreviate $n := |V|$ and $m := |A|$. We will use $V(G)$ and $A(G)$ to denote node and arc sets of a graph. Generally the values of P are called periods or modules, those of x tensions, those of π potentials, and those of k offsets. In the remainder we will assume w.l.o.g. the graphs to be connected. It is easy to see, that if the image of u and ℓ are in the integers there is always an optimal solution with all π integral. So basically, the PESP is an IP, although it is constantly referred to in the literature as a MIP.

We Summarize Some Basics on the PESP: Notice, the objective function only refers to pairwise differences of potentials. This is partly due to the pertinent applications, partly to the mathematical structure. The k variables model the modulo operator. It would be strange to count them in a practical objective. Moreover, for any feasible solution (π, k) and any $q \in \mathbb{Q}$ also $(q + \pi, k)$ is feasible and has the same objective value.

Assume all P_a are equal, and let x be the vector of arc differences of a solution (π, k) , i.e., $x_{(i,j)} = \pi_j - \pi_i + k_{(i,j)} P$. A vector x arising from a solution in this way is called its *tension*. It is easily checked—and we will re-prove it as a by-product of a more general theorem—that for any tree T there is a vector $(\pi, k)'$ with the same tension x but $k'_a = 0$ for all $a \in T$. Note, that $(\pi, k)'$ is also a feasible solution and has the same objective value as (π, k) , because it has the same tension. This gives rise to two important features of the PESP with uniform modules.

First, if we know the tension x , we can construct a feasible solution in a simple way: Set $\pi_i = 0$ for an arbitrary node i . Choose an arbitrary spanning tree T , and *propagate π starting from i along T with respect to x* . Propagation means, that we solve the equality system $\pi_j - \pi_i = x_a$ for all $a \in A(T)$ iteratively fixing the node values as we traverse the tree.

It is helpful to notice, that any arc a with $u_a - \ell_a \geq P$ states a redundant condition. Also, we can replace a directed arc by its antiparallel arc, simply by multiplying both constraint by (-1) .

The second important concept are *cycle bases*. A cycle basis is a basis for the linear subspace spanned by the incidence vectors of cycles in the vector space \mathbb{Q}^m spanned by the arc incidence vectors. Note that a cycle may have forward and backward arcs. For the latter the incidence vector of the cycle has a (-1) entry. A cycle basis is called *integral*, if all cycles are integer linear combination of the elements of the basis. Given a spanning tree T in the graph, the fundamental cycles $C(a, T)$ of all non-tree arcs $a \notin A(T)$ form a cycle basis. A *fundamental cycle* $C(a, T)$ of a non-tree arc a with respect to a spanning tree T is composed of the arc itself and the unique path in T connecting its endnodes. Such a basis is called a *fundamental cycle basis* (sometimes also: strictly fundamental). Every fundamental cycle basis is integral.

Finally, in the case of uniform modules, we can sum the constraints along a cycle C , yielding a new, valid constraint. Replacing $\pi_j - \pi_i$ by $x_{(i,j)}$ this constraint reads: $\sum_{a \in C} x_a = k_C P$, where $k_C = \sum_{a \in C} k_a$, and we assume w.l.o.g. all arcs to be directed in the orientation of the cycle. If a tension vector x fulfills this cycle constraint for all cycles of an integral cycle basis, then it is the tension of a solution (π, k) . If in addition $x_a \in [\ell_a, u_a]$, then it is the tension of a feasible solution (π, k) . In other words, for uniform modules an integral cycle basis gives rise to an equivalent MIP formulation. This *cycle basis formulation* has proven [10] significantly stronger than the original *arc formulation*.

For a cycle C we abbreviate $\text{gcd}(C) := \text{gcd}\{P_a : a \in C\}$.

2 Graphical Diophantine Equations Systems

Both, the Diophantine and the MIP results in this paper are based on Lemma 1 that guarantees the existence of well structured solutions under certain conditions. To state these conditions we define the following.

Definition 3. Let \mathcal{G} be a GDES and G the graph of its representation. A spanning tree T in G is called a sharp tree, if each of its fundamental cycles $C(a, T)$ has greatest common divisor equal to P_a , the module of the cycle's non-tree arc $a \in A(G) \setminus A(T)$.

Lemma 1. Let \mathcal{G} be a GDES and T a sharp tree in the graph of its representation. If \mathcal{G} has a solution, then there is a solution of \mathcal{G} with $k_a = 0$ for all $a \in T$.

Proof. Reorder the matrix of the GDES such that the following holds: The new matrix M starts with the $n - 1$ rows corresponding to the arcs in T . Restricted to the columns affecting the π variables, these rows form a lower triangular matrix (the first column omitted). The columns affecting k form a diagonal matrix.

Index the nodes according to their column and arcs according to their rows in M . For two nodes v and w denote by $\mathcal{P}(v, w, T)$ the unique path from v to w in T .

Let $(\pi, k)^0$ be some solution. We construct a solution $(\pi, k)^{n-1}$ over $n - 1$ stages denoted $(\pi, k)^i, i \in \{1 \dots n - 1\}$. For each $i \in \{1 \dots n - 1\}$ successively with increasing row index we take four steps:

1. Set $k_i^i = 0$.
2. Re-establish the correctness of the i -th equation by changing the node value π_i corresponding to $M_{i,i}$, the right most non-zero entry in the first $n - 1$ columns, i.e., $\pi_i^i := \pi_i^{i-1} + M_{i,i}k_i^0P_i$. Let $\ell(i)$ be minimal with $M_{i,\ell(i)} \neq 0$, i.e., the other node of arc i .
3. Propagate the new node value downwards along the tree. Formally: For all remaining tree rows $t \in \{i + 1 \dots n - 1\}$ successively with increasing row index set $\pi_t^i := \pi_t^{i-1} + M_{i,t}k_i^0P_i$ in case $\ell(i) \notin \mathcal{P}(i, t, T)$.
4. Re-establish correctness (in arbitrary order) for the equations of non-tree arcs by adjusting their arc variables. Formally: For all $j \in \{n, \dots, m\}$ let $1 \leq r < s \leq n - 1$ be the nodes of arc j , i.e., $M_{j,r}$ and $M_{j,s} \neq 0$. Set $k_j^i := k_j^{i-1} - \frac{M_{j,r}(\pi_r^i - \pi_r^{i-1}) + M_{j,s}(\pi_s^i - \pi_s^{i-1})}{P_j}$.

(We were a bit sloppy dropping exceptional handling of first row and column, and omitting when $\pi_j^i := \pi_j^{i-1}$ and likewise for k .)

Obviously, each $(\pi, k)^i$ fulfills all equalities. Observe, we touch the arc variable k_t of any tree row only in stage t . Therefore, $(\pi, k)^i$ the solution of any stage $i \in \{1 \dots n - 1\}$ has $k_t^i = 0$ for all $t \in \{1 \dots i\}$. It remains to show for the non-tree arcs $j \in \{n \dots m\}$ that every k_j^i is an integer, in particular, that

$$P_j \mid [M_{j,r}(\pi_r^i - \pi_r^{i-1}) + M_{j,s}(\pi_s^i - \pi_s^{i-1})].$$

For all nodes s we have $\pi_s^i - \pi_s^{i-1} = |k_i^0P_i|$. Now, distinguish whether $\ell(i) \in \mathcal{P}(r, s, T)$ or not. If $\ell(i)$ is in, so is i and the i -th arc is on the fundamental cycle of j in T . Thus, $P_j \mid P_i$ by condition of the lemma and we are done. In case, $\ell(i) \notin \mathcal{P}(r, s, T)$ both nodes are changed by the same value. Therefore, $M_{j,r}(\pi_r^i - \pi_r^{i-1}) + M_{j,s}(\pi_s^i - \pi_s^{i-1}) = 0$, which completes the proof. \square

To guarantee the existence of sharp trees we need the following property:

Definition 4. We say a GDES (or a PESP) has nested modules, if for each pair of its modules $P_a \leq P_b$ we have $P_a \mid P_b$.

We will show that GDES with nested modules have sharp trees, whereas sharp trees do not exist in general.

Nested Modules: The DENDI–algorithm (Diophantine Equations with Nested Divisors) (cf. [4] for pseudo-code) solves GDES with nested modules. In addition it constructs a sharp tree. The algorithm considers the arcs in subsequent levels ℓ according to their module. On the first level, it constructs spanning trees in the connected components of arcs with maximal modules. Then it shrinks these components to super-nodes and carries on with the next smaller level of modules. This way DENDI–algorithm constructs a subgraph, actually a tree, along which one can propagate the potentials π according to the tensions. Finally, DENDI verifies whether the equations of the non-tree arcs can

also be fulfilled for the chosen π . For the correctness of the algorithm we show two lemmata.

Lemma 2. *The subgraph T returned by the DENDI–algorithm is a sharp tree.*

Proof. Obviously, T is a spanning tree. Consider a non-tree arc a and its module P_a . Every other arc b in the fundamental cycle $C(a, T)$ either belongs to the same component as a on level ℓ or to a tree of a component shrunk into a super-node on an earlier level. In both cases $P^\ell \leq P_b$. The modules being nested, this implies $P_a \mid P_b$, and thus T is sharp. \square

Thus, we have:

Theorem 1. *A GDES with nested modules has a sharp tree.*

Lemma 3. *The DENDI–algorithm returns failure, iff the GDES is infeasible. Moreover, the offset $k_a = 0$ for all tree arcs $a \in A(T)$.*

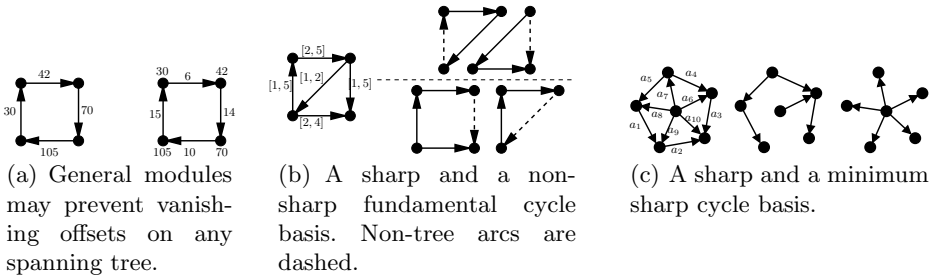
Proof. The offsets vanish on the tree arcs by construction, and because of the final test in DENDI–algorithm, (π, k) is a solution to the GDES, if they are returned. By Lemma 2 T is sharp. Hence, by Lemma 1 the GDES has a solution iff it has a solution with $k_a = 0$ for all $a \in A(T)$. If such a solution exists, it is fully determined by the potential of one node i , propagated along the spanning tree T . If π^* is such a solution, then $\pi^* + z$ is also one for all $z \in \mathbb{Z}$. In particular, the π constructed in DENDI–algorithm is one. Thus, if the π constructed by DENDI is infeasible, then the GDES is infeasible. \square

Observe that one can force any minimum spanning tree algorithm to return the same tree as the (weighted) DENDI–algorithm by introducing the following weights: The weight of an arc is (the sum of its original weight w and) a multiple of a large constant $M (> \sum w)$, where arcs with larger period get smaller multiples of M . Therefore, we can substitute DENDI–algorithm by any MST algorithm and conclude:

Theorem 2. *The DENDI–algorithm is correct, and has a running time in $\mathcal{O}(n^2)$, where n is the dimension of the solution vector.*

General Modules: Requiring nested modules may be suitable for the application but constitutes a strong mathematical restriction. Still, the example on the left of Figure 1(a) shows that we cannot hope for similar results in the general case.

Example 1. In Figure 1(a) the numbers next to the arcs and nodes give the modules of the arcs respectively the nodes. On the right, the arc modules result as gcd of the node modules. Assume the tensions along the cycle sum up to 1. This is feasible, because the gcd of all arcs is 1. Yet, a feasible solution must have $k \neq 0$ on all arcs for the left graph, and either on the two vertical or the two horizontal arcs for the right graph.



In general, a solution for any cycle C must have non-zero offsets k on a subset S of C 's arcs, such that $\gcd(S) = \gcd(C)$. As in the example this may require all offsets to be non-zero.

Looking at GDESs from the perspective of the PESP and its applications, the following objection is valid: In the application we are given periods for the events, i.e., the nodes. The period (module) of an arc $a = (i, j)$ arises in an equivalent formulation as $P_a = \gcd\{P_i, P_j\}$. Thus, the situation on the left of Figure 1(a) cannot occur. Still, (cf. the example on the right in Figure 1(a)) node modules can be such, that a solution must have non-zero offsets on a subset of the arcs, that forms a maximal matching on any cycle.

3 Solving PESP with Multiple Modules

The methods for the PESP with uniform modules rest on a strong formulation based on an integral cycle basis and on certain rounding cuts, that also stem from cycles. We will show that in general an integral cycle basis for a multi-module PESP does *not* give an equivalent formulation. Whereas, if a sharp tree exists, we will show that, its fundamental cycle basis provides for the desired strong formulation. In general, a multi-module PESP need not have a sharp tree. But we have seen in the previous section that a sharp tree can be found in case of nested modules.

The proper generalization of integral cycle bases for multi-module PESP is the following type of basis:

Definition 5. A fundamental cycle basis stemming from a sharp tree is called a sharp cycle basis.

In particular, this means for each cycle C in a sharp basis, that the $\gcd(C)$ is attained by the non-tree arc of C . Recall, that any fundamental cycle basis, and thereby any sharp cycle basis is integral.

Lemma 4. Let \mathcal{B} be a sharp cycle basis in a PESP model. For an arc vector x the following three statements are equivalent:

1. The vector x is the arc tension of a node potential π .
2. The vector x fulfills the cycle equality for every cycle C , i.e., there is $k_C \in \mathbb{Z}$ such that $\sum_{a \in C} x_a = k_C \cdot \gcd(C)$.
3. The vector x fulfills the cycle equality for every cycle $C \in \mathcal{B}$.

Proof. The inclusion $2 \Rightarrow 3$ being trivial, we show $1 \Rightarrow 2$ and $3 \Rightarrow 1$.

$1 \Rightarrow 2$: According to (1) we have $x_{(i,j)} = \pi_j - \pi_i + k_{(i,j)}P_{(i,j)}$ for all arcs $(i, j) \in A$. Summing along a cycle C (multiplying the equation of a with (-1) for arcs a that lie in C contrary to its orientation) we get $\sum_{(i,j) \in C} x_{(i,j)} = k_{(i,j)}P_{(i,j)}$.

$3 \Rightarrow 1$: Define a node potential π by setting $\pi_s = 0$ for some node s and propagate the x value along the sharp tree T . It remains to show that for every non-tree arc $(i, j) \in A \setminus T$ there is $k_{(i,j)} \in \mathbb{Z}$ such that

$$\pi_j - \pi_i + x_{(i,j)} = k_{(i,j)}P_{(i,j)}. \tag{5}$$

The fundamental cycle $C := (\mathcal{P}(i, j, T), (i, j))$ is in \mathcal{B} and $P_{(i,j)} = \text{gcd}(C)$, and Equation (5) follows. □

Together with Theorem 2 we get:

Theorem 3. *Let \mathcal{G} be a PESP in the arc formulation. If a sharp cycle basis formulation for \mathcal{G} exists, it is equivalent. If \mathcal{G} has nested modules, then a sharp cycle basis formulation exists and can be found in time in $\mathcal{O}(n^2)$.*

For multiple modules the cycle basis formulation is not equivalent to the arc formulation even if the modules are nested. In the following example we show that non-sharp trees do not guarantee an equivalent cycle basis formulation.

Example 2. Consider the graph on the left of Figure 1(b). Set the module $P_a = 6$ for all arcs except the diagonal one. For this set the module to 3. The interval next to an arc shows its upper and lower bounds. As cost vector choose the unit vector.

On the right of Figure 1(b) we show two fundamental cycle bases corresponding to two different trees. The non-tree arcs of a fundamental cycle are drawn dashed. On top, the cycle basis consists of two triangles. The corresponding tree is not a sharp tree, because for both cycles the non-tree arc has module 6 and the cycle’s gcd is 3. Again, the example cannot occur if the periods stem from the nodes. Still, if one replaces the diagonal arc by two arcs of period 3 one can choose the node periods accordingly.

Index the arcs clockwise starting left and put the diagonal arc last. Then the optimal solution for this cycle basis formulation is $x = (2, 2, 1, 2, 2)$, for which DENDI–algorithm returns failure, i.e., it is not a tension of a feasible solution and thus the cycle basis formulation is *not* equivalent to the original arc formulation.

Yet, if we consider the sharp tree consisting of the left, upper, and lower arc, the corresponding cycle basis, shown at the bottom right of Figure 1(b), gives a formulation which is equivalent to the arc formulation, as stated in Theorem 3. In particular, the optimal solution we get is $x = (3, 2, 3, 2, 1)$, for which DENDI–algorithm is able to find a feasible set of potentials.

Algebraic Pruning: Assume again nested modules. Consider an arc a that is in no cycle or only in cycles C with $\text{gcd}(C)$ strictly smaller than P_a . Assume we

have a solution $(x, k)^*$ to a sharp cycle basis formulation of the PESP. The arc a must lie in the sharp tree T of any such formulation. Therefore, recovering node potentials from x^* will result in a solution $(\pi, k)'$ with $k'_a P_a = k_{C(a,T)} \gcd(C)$. Thus, the inequality of arc a is also fulfilled modulo $\gcd(C)$ —which is strictly smaller than P_a . This observation allows to simplify a PESP with nested divisors:

Theorem 4. *Given a PESP \mathcal{G} with nested modules containing an arc a with $\gcd(C) < P_a$ for all cycles $C \ni a$. The PESP \mathcal{G}' , resulting from \mathcal{G} by replacing P_a by its largest divisor $P'_a \neq P_a$, is equivalent to \mathcal{G} .*

One may repeatedly apply Theorem 4 to simplify the PESP in a pre-processing. In case the considered arc is in no cycle, its module is ultimately set to 1, which is equivalent to removing the arc. Any solution for the (after the removal) disconnected graph can be amalgamated to a solution of the original PESP in a linear time post-processing. As one can remove arcs, for which the difference between upper and lower bound is greater or equal to the module, even reducing to a non-trivial module can result in the arc being obsolete. This reduces the dimension of the MIP and allow to assume the following property for the sharp basis found by the DENDI:

Observation 1. *W.l.o.g. for every arc a a sharp cycle basis contains a cycle C , such that $\gcd(C) = P_a$.*

The basis of the DENDI–algorithm is sharp and thus has a tight cycle for each arc. This will be exploited in the last section, where we seek to give a small set of strong cuts derived from cycle inequalities.

Cuts and Sharp Trees: Solving a uniform-module PESP cycles are also used to produce a special class of rounding cuts, the so-called *Odiijk* inequalities:

$$\left| \frac{\sum_{a_+ \in C} \ell_a - \sum_{a_- \in C} u_a}{P} \right| \leq k_C \quad \text{and} \quad \left| \frac{\sum_{a_+ \in C} u_a - \sum_{a_- \in C} \ell_a}{P} \right| \geq k_C \quad (6)$$

The key question is, for which cycles one should add the corresponding Odiijk inequalities to the MIP formulation. For the case of uniform modules there is a well established heuristic reasoning: The right-hand side is rounded down (or up) by a value between 0 and $P - 1$. If the total value of the right-hand side is large in comparison to P the effect of rounding cannot be large.

Therefore, one is interested in shortest integral cycle bases. There is no polynomial time algorithm known for this problem. Yet, there are many heuristics to find short integral cycle bases. A standard approach is to construct a fundamental cycle basis from a minimum spanning tree (MST). Here the heuristic idea is, that the non-tree arcs feature in exactly one cycle, whereas the minimized tree arcs can occur in several cycles of the basis. Thus, the sum of all cycles will be rather small, and the Odiijk inequalities likely tight. For multiple modules the rounding on a cycle C is between 0 and $\gcd(C)$. Assume the cycle C contains an arc a with a significantly smaller module P_a than that of all other arcs in C .

One can assume the difference between upper and lower bound on an arc to be less than its module. But, the contribution to the right-hand side by each other arc $b \in C$ can be much larger than P_a , because $P_b \gg P_a$. Thus, if an arc b is in no cycle C with $\gcd(C)$ close to its own module P_b , the set of cuts will not have a relevant effect on the number of choices for x_b .

Theorem 4 allows to assume that every arc $b \in A$ is in at least one cycle C with $\gcd(C) = P_b$. A sharp cycle basis for every arc b contains such a cycle C . So, for multiple-modules PESP we propose to choose cycle basis \mathcal{B} such that (1) \mathcal{B} is a sharp cycle basis and (2) \mathcal{B} arises from a sharp tree with minimal sum of arc weights (with respect to $(u - \ell)$) among all sharp trees. This can be found by the weighted version of DENDI-algorithm.

4 Computational Results

We study twelve instances derived from `timtab1` and `timtab2`, the MIPLIB PESP instances¹. These two MIPLIB instances are anonymized real-world timetabling problems of a major European railway provider. We changed the original periods of 60 on the *nodes* randomly to the nested periods 120, 60, 30, 15 and 5, giving lower probability to the small periods as they dominate in the transition from node to arc periods. After this transition, the bounds on the arcs were adjusted relative to the change in period.

On these instances we compare the standard formulation to a sharp cycle base formulation with the basis' Odijk inequalities. For each we use CPLEX 10.0 with a timelimit (TL) of 2 hours on a 2.4Ghz processor. The sharp cycle basis formulation is strikingly faster in detecting infeasibility and solves all except one of the feasible instances with a better gap (cf. Table 1).

Table 1. multi-period miplib PESP statistics

instance	status	Classical		Sharp Tree + Odijks		
		GAP%	time (sec.)	status	GAP%	time (sec.)
mpesp1	feasible	5.99	TL	feasible	4.19	TL
mpesp2	feasible	6.13	TL	feasible	5.73	TL
mpesp3	feasible	5.58	TL	feasible	3.83	TL
mpesp4	feasible	2.94	TL	feasible	2.50	TL
mpesp5	feasible	5.33	TL	feasible	5.29	TL
mpesp6*	feasible	9.81	TL	feasible	10.26	TL
mpesp7	feasible	12.09	TL	feasible	9.72	TL
mpesp8	feasible	12.87	TL	feasible	9.71	TL
mpesp9	-	-	TL	infeasible	-	0
mpesp10	-	-	TL	infeasible	-	3431
mpesp11	infeasible	-	6934	infeasible	-	0
mpesp12	infeasible	-	657	infeasible	-	0

¹ Elmar Swarat kindly provided us with the raw data of `timtab`.

Finally, note that nested periods in timetable optimization are also recommendable in the light of quality of service: They yield that more passenger actually experience the optimized transfer time, because, e.g., for co-prime periods *every* transfer time will be experienced by some passengers.

References

1. Conforti, M., Di Summa, M., Wolsey, L.: The mixing set with divisible capacities. In: Lodi, A., Panconesi, A., Rinaldi, G. (eds.) IPCO 2008. LNCS, vol. 5035, pp. 435–449. Springer, Heidelberg (2008)
2. Conforti, M., Zambelli, G.: The mixing set with divisible capacities: A simple approach. *Operations Research Letters* 37, 379–383 (2009)
3. Fischetti, M., Lodi, A.: Optimizing over the first Chvatal closure. *Mathematical Programming* 110(1), 3–20 (2006)
4. Galli, L., Stiller, S.: Strong Formulations for the Multi-module PESP and a Quadratic Algorithm for Graphical Diophantine Equation Systems. COGA Technical Report 009–2010 (2010)
5. Hassin, R.: A flow algorithm for network synchronization. *Operations Research* 44, 570–579 (1996)
6. Khot, S.: On the power of unique 2-prover 1-round games. In: Proceedings of the 34th Annual ACM Symposium on Theory of Computing, pp. 767–775. ACM Press, New York (2002)
7. Köhler, E., Möhring, R., Nökel, K., Wünsch, G.: Optimization of Signalized Traffic Networks. In: *Mathematics – Key Technology for the Future*, pp. 179–180. Springer, Heidelberg (2008)
8. Liebchen, C.: Periodic Timetable Optimization in Public Transport. Ph.D. thesis, Technische Universität Berlin (2006)
9. Liebchen, C., Swarat, E.: The Second Chvatal Closure Can Yield Better Railway Timetables. In: Proceedings of 8th Workshop on Algorithmic Approaches for Transportation Modeling, Optimization, and Systems. Online Schloss Dagstuhl (2008)
10. Liebchen, C., Prosch, M., Wagner, F.H.: Performance of Algorithms for Periodic Timetable Optimization. In: *Computer-aided Systems in Public Transport*, pp. 151–180. Springer, Heidelberg (2008)
11. Nachtigall, K.: Cutting planes for a polyhedron associated with a periodic network. DLR Technical Report 112-96/17
12. Odijk, M.: Construction of periodic timetables, Part1: a cutting plane algorithm. TU Delft Technical Report 94-61 (1994)
13. Odijk, M.: A constraint generation algorithm for the construction of periodic railway timetables. *Transportation Research B* 30(6), 455–464 (1996)
14. Peeters, L.: Cyclic Railway Timetable Optimization. Ph.D. thesis, Erasmus University of Rotterdam (2003)
15. Schrijver, A.: *Theory of Linear and Integer Programming*. Wiley & Sons, Chichester (1986)
16. Serafini, P., Ukovich, W.: A mathematical model for periodic scheduling problems. *SIAM Journal on Discrete Mathematics* 2(4), 550–581 (1989)