



FINDING SHORT INTEGRAL CYCLE  
BASES  
FOR CYCLIC TIMETABLING

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. 2003/12

# Finding Short Integral Cycle Bases for Cyclic Timetabling\*

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

June 2, 2003

## Abstract

Cyclic timetabling for public transportation companies is usually modeled by the periodic event scheduling problem. To deduce a mixed-integer programming formulation, artificial integer variables have to be introduced. There are many ways to define these integer variables.

We show that the minimal number of integer variables required to encode an instance is achieved by introducing an integer variable for each element of some integral cycle basis. An integral cycle basis consists of  $|A| - |V| + 1$  oriented cycles of a directed graph  $D = (V, A)$  that enable any oriented cycle of the directed graph to be expressed as an *integer* linear combination.

The solution times for the originating application vary extremely with different integral cycle bases. However, our computational studies show that the width of integral cycle bases is a good empirical measure for the solution time of the MIP. Clearly, integral cycle bases permit a much wider choice than the former standard approach, in which integer variables are associated with the co-tree arcs of some spanning tree. Hence, to formulate better solvable integer programs, we present algorithms that construct integral cycle bases of small width.

To that end, we investigate classes of directed cycle bases that are closely related to integral cycle bases, namely (generalized) fundamental and undirected cycle bases. This gives rise to both, a compact classification of directed cycle bases and notable reductions of running times for cyclic timetabling.

## 1 Introduction and Scope

Cycle bases play an important role in various applications. Recent investigations cover ring perception in chemical structures ([7]) and the design and analysis of electric networks ([3]). What cyclic timetabling shares with these applications is that the construction of a short cycle basis is an important preprocessing step to improve solution methods for real world problems.

Since the pioneering work of Serafini and Ukovich[23], the construction of periodic timetables for public transportation companies, or cyclic timetabling for short, is usually modeled by the periodic event scheduling problem (PESP). For an exhaustive presentation of practical requirements that the PESP is able to meet, we refer to Krista[11]. The feasibility problem has been shown to be  $\mathcal{NP}$ -complete, by reductions from Hamiltonian Cycle ([23] and [17]) or Coloring ([19]). With a linear objective given,  $\mathcal{NP}$ -hardness has been deduced by a reduction from Linear Ordering ([14]). We want to solve such instances by using the mixed integer solver of the CPLEX<sup>®</sup>[4] optimization suite.

*Related Work.* Of course, the performance of implicit enumeration algorithms can be improved by reducing the number of integer variables. It has already been Serafini and Ukovich to detect that

---

\*Supported by the DFG research center "Mathematics for key technologies" (FZT 86) in Berlin

there is no need to introduce an integer variable for every arc of the directed constraint graph. Rather, one can restrict the integer variables to those that correspond to the co-tree arcs of some spanning tree. These arcs can be interpreted to be the representatives of a strictly fundamental cycle basis. Of course, there is a huge number of those cycle bases.

Nachtigall[16] did immediately profit from the spanning tree approach when switching to a tension-based problem formulation. Notice that our results on integral cycle bases apply to that tension-perspective as well.

Odiijk[19] provided box constraints for the remaining integer variables. Hereby, it becomes possible to quantify the differences of cycle bases. But the implied objective function is rather bulky.

De Pina[21] observed that a cycle basis that minimizes a much simpler function also minimizes our original objective. What remains to solve is a variant of the minimal cycle basis problem.

*Contribution and Scope.* In our computational studies we show that the width of a cycle basis is highly correlated with the solution time of the mixed-integer solver. Thus, it serves as a good empirical measure for the run time and provides a way to speed up the solver by choosing a short basis.

Hence, in order to supply MIP solvers with promising problem formulations, we are going to compute short cycle bases. But there is a certain dilemma when analyzing the two most popular types of directed cycle bases: On the one hand, there are directed cycle bases that induce undirected cycle bases. For these, we can minimize a linear objective function efficiently, due to a polynomial time algorithm by Horton[10]. But, contrary to a claim of de Pina[21], undirected cycle bases unfortunately are *not* applicable to cyclic timetabling in general, which we will demonstrate by giving a counter-example. On the other hand, cycle bases that stem from spanning trees, or strictly fundamental cycle bases for short, form a feasible choice. But for them, minimization is  $\mathcal{NP}$ -hard, as has been proved by Deo et al.[5]

To cope with this dilemma, we investigate if there is a class of cycle bases lying in *between* general undirected cycle bases and strictly fundamental cycle bases, hopefully combining both good algorithmic behavior and the usability to express PESP instances. To that end, a compact classification of directed cycle bases will be given. Efficient characterizations will be based on properties of the corresponding cycle matrices, e.g. its determinant, which we find out to be well-defined. This allows a natural definition of the *determinant of a directed cycle basis*.

The first important special class are integral cycle bases. We will show them to be the most general structure when limiting an instance of the PESP to only  $|A| - |V| + 1$  integer variables. However, the complexity status of minimizing a linear objective function over the integral cycle bases remains unknown to the author.

Since (generalized) fundamental cycle bases<sup>1</sup> are a subset of integral cycle bases, they will play a central role in the powerful algorithms presented for constructing short integral cycle bases. For example, being unsatisfied with a running time of  $\mathcal{O}(m^3n)$  for his initial algorithm to find a shortest cycle basis of a graph, Horton proposed an approximation-algorithm with running time  $\mathcal{O}(n^2m)$ . We will prove that this algorithm in fact always constructs a fundamental – and in particular integral – cycle basis.

The computational results provided in section 6 show the enormous benefit of generalizing the spanning tree approach to integral cycle bases for the originating application of cyclic timetabling. These results point out the need of deeper insights into integral cycle bases and related structures. Some open problems are stated at the end of this paper.

## 2 Periodic Scheduling and Short Cycle Bases

An instance of the Periodic Event Scheduling Problem (PESP) consists of a directed constraint graph  $D = (V, A, \ell, u)$ , where  $\ell$  and  $u$  are lower resp. upper time bounds for the arcs, together with a period time  $T$  of the transportation network. A solution of a PESP instance is a node potential  $\pi : V \rightarrow [0, T)$  – or time vector for the periodically recurring departure/arrival events within the

---

<sup>1</sup>We follow the notation of Whitney[24] where he introduced the concept of matroids.

public transportation network – that fulfills periodic constraints of the form  $(\pi_j - \pi_i - \ell_{ij}) \bmod T \leq u_{ij} - \ell_{ij}$ , which we resolve by introducing artificial integer variables  $p_{ij}$ ,

$$\ell_{ij} \leq \pi_j - \pi_i + p_{ij}T \leq u_{ij}, (i, j) \in A. \quad (1)$$

Our computational results will show that the running times of a mixed-integer solver on instances of cyclic timetabling correlate with the volume of the polytope spanned by the box constraints provided for the integer variables. Formulation (1) can only yield  $0 \leq p_a \leq 2$  for  $a \in A$  in general,<sup>2</sup> even with scaling to  $0 \leq \ell_{ij} < T$ .

Serafini and Ukovich observed that the above problem formulation may be simplified by eliminating  $|V| - 1$  integer variables that correspond to the arcs  $a$  of some spanning tree  $H$ , when relaxing  $\pi$  to be some real vector. Formally, we just fix  $p_a := 0$  for  $a \in H$ . Then, in general, the remaining integer variables may take more than three values. For example, think of the directed cycle on  $n$  arcs, with  $\ell \equiv 0$  and  $u \equiv T - \frac{1}{n}$ , as constraint graph. With  $\pi = \mathbf{0}$ , the integer variables of every arc will be zero. But  $\pi_i = (i - 1) \cdot (T - \frac{1}{n})$ ,  $i = 1, \dots, n$  would be a feasible solution as well, implying  $p_{n1} = n - 1$  for the only integer variable that we did not fix to zero. Fortunately, theorem 1 provides box constraints for the remaining integer variables.

**Theorem 1 (Odiijk[19]).** *A PESP instance defined by the constraint graph  $D = (V, A, \ell, u)$  and a period time  $T$  is feasible if and only if there exists an integer vector  $p \in \mathbb{Z}^{|A|}$  satisfying the cycle inequalities*

$$a_C \leq \sum_{a \in C^+} p_a - \sum_{a \in C^-} p_a \leq b_C, \quad (2)$$

for all (simple) cycles  $C \in G$ , where  $a_C$  and  $b_C$  are defined by

$$a_C = \left\lceil \frac{1}{T} \left( \sum_{a \in C^+} \ell_a - \sum_{a \in C^-} u_a \right) \right\rceil, \quad b_C = \left\lfloor \frac{1}{T} \left( \sum_{a \in C^+} u_a - \sum_{a \in C^-} \ell_a \right) \right\rfloor, \quad (3)$$

and  $C^+$  and  $C^-$  denote the sets of arcs that, for a fixed orientation of the cycle, are traversed forwardly resp. backwardly.

For any co-tree arc  $a$ , the box constraints for  $p_a$  can be derived by applying the cycle inequalities (2) to the unique oriented cycle in  $H \cup \{a\}$ .

## Directed Cycle Bases and Undirected Cycle Bases

Let  $D = (V, A)$  denote a connected directed graph. An *oriented cycle*  $C$  of  $D$  consists of *forward arcs*  $C^+$  and *backward arcs*  $C^-$ , such that  $C = C^+ \dot{\cup} C^-$  and reorienting all arcs in  $C^-$  results in a directed cycle. A *directed cycle basis* of  $D$  is a set of oriented cycles  $C_1, \dots, C_k$  with incidence vectors  $\gamma_i \in \{-1, 0, 1\}^{|A|}$  that permit a unique linear combination of the incidence vector of any (oriented) cycle of  $D$ . By  $k$  we denote the cyclomatic number  $k = |A| - |V| + 1$  of  $D$ , and of course, arithmetic is performed over the field  $\mathbb{Q}$ .

For a directed graph  $D$ , we obtain the *underlying undirected graph*  $G$  by removing the directions from the arcs. A *cycle basis of an undirected graph*  $G = (V, E)$  is a set of undirected cycles  $C_1, \dots, C_k$  with incidence vectors  $\phi_i \in \{0, 1\}^{|E|}$ , that again permit to combine any cycle of  $G$ . But here, arithmetic is over the field  $\text{GF}(2)$ . A set of directed cycles  $C_1, \dots, C_k$  *projects onto an undirected cycle basis*, if by removing the orientations of the cycles, we obtain a cycle basis for the underlying undirected graph  $G$ .

**Lemma 2.** *Let  $\mathcal{C} = \{C_1, \dots, C_k\}$  be a set of oriented cycles in a directed graph  $D$ . If  $\mathcal{C}$  projects onto an undirected cycle basis, then  $\mathcal{C}$  is a directed cycle basis.*

<sup>2</sup>For  $T = 10$ ,  $\ell_{ij} = 9$ , and  $u_{ij} = 11$ ,  $\pi_j = 9$  and  $\pi_i = 0$  yield  $p_{ij} = 0$ ;  $p_{ij} = 2$  is achieved by  $\pi_j = 0$  and  $\pi_i = 9$ .

This can easily be verified by considering the mod 2 projection of  $\mathcal{C}$ , cf. Liebchen and Peeters[13]. But the converse is not true, as is illustrated in the following example.

**Example 1.** *The following set of ten cycles defined on  $K_6$ , with edges oriented arbitrarily, constitutes a directed cycle basis. But since every arc is hit exactly twice, it does not project onto an undirected cycle basis.*

$$\begin{aligned} &(1, 2, 3) \ (1, 2, 4) \ (1, 3, 5) \ (1, 4, 6) \ (1, 5, 6) \\ &(2, 3, 6) \ (2, 4, 5) \ (2, 5, 6) \ (3, 4, 5) \ (3, 4, 6) \end{aligned}$$

Moreover, it is a minimal directed cycle basis, and altogether this is a node-minimal example for this effect.

## Objective Function for Short Cycle Basis

Considering the co-tree arcs in the spanning tree approach as representatives of the elements of a directed cycle basis enables us to formalize the desired property of cycle bases we need to construct a promising MIP formulation for cyclic timetabling instances.

**Definition 1 (Width of a Cycle Basis).** *Let  $\mathcal{C} = \{C_1, \dots, C_k\}$  be a directed cycle basis of a constraint graph  $D = (V, A, \ell, u)$ . Let  $T$  be a fixed period time. Then, for  $a_{C_i}$  and  $b_{C_i}$  as defined in (3), we define the width of  $\mathcal{C}$  by*

$$W(\mathcal{C}) := \prod_{i=1}^k (b_{C_i} - a_{C_i} + 1). \quad (4)$$

The width is just our empirical measure for the estimated running time of the MIP solver on instances of the originating application. Hence, for the spanning tree approach, we should construct a spanning tree whose cycle basis minimizes the width function. Especially, if many constraints have small *span*  $d_a := u_a - \ell_a$ , the width will be much smaller than the corresponding value of  $3^{|A|}$  for the initial formulation of the PESP.

To manage the product and the rounding operation for computing  $a_{C_i}$  and  $b_{C_i}$ , we consider a slight relaxation of the width:

$$W(\mathcal{C}) \leq \prod_{i=1}^k \left\lceil \frac{1}{T} \sum_{a \in C_i} d_a \right\rceil. \quad (5)$$

De Pina[21] proved that an undirected cycle basis that minimizes the linearized objective

$$\sum_{i=1}^k \sum_{a \in C_i} d_a \quad (6)$$

does also minimize the right-hand-side in (5). However, there are pathological examples in that a minimal cycle basis for the linearized objective (6) does not minimize the initial width function (4): Consider  $K_7$  with  $\ell \equiv 3$ ,  $u \equiv 4$ , and  $T = 10$ . A minimal cycle bases with respect to (6) consists of 15 triangles that lead to width one. But every cycle basis that contains one of the infeasible cycles through seven arcs results in zero width.

Applying the above linearization to spanning trees yields the problem of finding a minimal strictly fundamental cycle basis. But two decades ago, Deo et al.[5] showed this problem to be  $\mathcal{NP}$ -hard. Currently, Amaldi[1] is even working out MAX-SNP-hardness.

## General Cycle Bases are Misleading

De Pina[21] proposed to keep an integer variable in the PESP only for each of the cycles of arbitrary undirected cycle bases. Consequently, he could exploit Horton's[10]  $\mathcal{O}(m^3 n)$ -algorithm<sup>3</sup> for

<sup>3</sup>Golynski and Horton[8] adapted it to  $\mathcal{O}(m^s n)$ , with  $s$  being the exponent of fast matrix multiplication. And by a substantially different approach, de Pina[21] achieved a nice  $\mathcal{O}(m^3 + mn^2 \log n)$ -algorithm for the same problem.

constructing a minimal cycle basis according to (6), in order to find a cycle basis which is likely to minimize the width function.

In more detail, for some directed cycle basis  $\mathcal{C}$ , define the *cycle matrix*  $\Gamma$  to be the arc-cycle-incidence matrix of  $\mathcal{C}$ . He claimed that the solution spaces stay the same, in particular

$$\{p \in \mathbb{Z}^m \mid p \text{ allows a feasible solution}\} \stackrel{?}{\subseteq} \{\Gamma q \mid q \in \mathbb{Z}^{\mathcal{C}}, q \text{ satisfies (2) on } \mathcal{C}\}. \quad (7)$$

We show that, in general, inclusion (7) does *not* hold. Hartvigsen and Zemel[9] provided a nice cycle basis  $\mathcal{C}$ , cf. figure 1. Assume the PESP constraints of  $D$  allow only the first unit vector  $e_1$

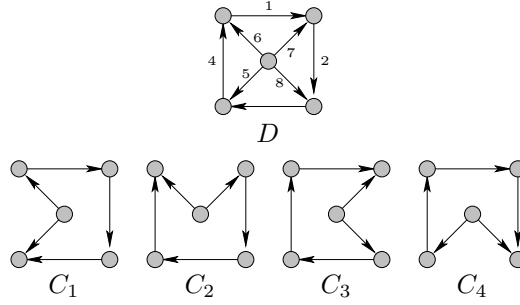


Figure 1: Cycle basis  $\mathcal{C} = \{C_1, \dots, C_4\}$  for which de Pina's approach fails

for  $p$  in any feasible solution, choosing the spanning tree  $H$  with  $p|_H = 0$  to be the star tree rooted at the center node.

For  $\mathcal{C}$ , the transpose of the cycle matrix  $\Gamma$  is

$$\Gamma^t = \begin{bmatrix} 1 & 1 & 1 & 0 & -1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & -1 & 1 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 & -1 & 1 \\ 1 & 1 & 0 & 1 & 1 & 0 & 0 & -1 \end{bmatrix}.$$

Restricting  $\Gamma$  to the rows that correspond to  $A \setminus H$ , the submatrix  $\Gamma'$  becomes regular and has the inverse matrix

$$(\Gamma')^{-1} = \frac{1}{3} \begin{bmatrix} 1 & 1 & 1 & -2 \\ -2 & 1 & 1 & 1 \\ 1 & -2 & 1 & 1 \\ 1 & 1 & -2 & 1 \end{bmatrix}.$$

Hence, the unique inverse image of  $p = e_1$  is  $q = (\Gamma')^{-1}p|_{A \setminus H} \notin \mathbb{Z}^k$ . Thus, the only feasible solution will never be seen, when working only on  $\mathbb{Z}^{\mathcal{C}}$ . In the following section we will establish that the crux in this example, is the fact that there is a regular  $k \times k$  submatrix of the cycle matrix  $\Gamma$  having determinant of absolute value different from one.

Thus, key information is lost, when only integer linear combinations of the cycles of some arbitrary cycle basis are considered. To summarize, our dilemma is the following: Cycle bases, over which minimization is easy, do not fit our purpose. But minimization over cycle bases that would be suitable to formulate instances of cyclic timetabling, becomes  $\mathcal{NP}$ -hard.

### 3 Matrix-Classification of Directed Cycle Bases

In order to develop algorithms that construct short cycle bases which we may use for expressing instances of cyclic timetabling, we want to identify an appropriate class of cycle bases. Fortunately, there is indeed some space left between directed cycle bases that project onto undirected ones, and cycle bases which stem from spanning trees. As our classification of what is in between will be based on properties of cycle matrices, we start by giving two algebraic lemmata.

**Lemma 3.** *Let  $\mathcal{C}$  be a directed cycle basis of a connected digraph  $D$ . A subset of  $k$  rows  $\Gamma'$  of its  $m \times k$  cycle matrix  $\Gamma$  is maximal linearly independent, if and only if the arcs chosen constitute the co-tree arcs of some spanning tree.*

*Proof.* To prove sufficiency, consider a spanning tree  $H$  of  $D$ , and  $\{a_1, \dots, a_k\}$  to become co-tree arcs. Consider the cycle matrix  $\Phi$  with the incidence vector of the unique cycle in  $H \cup \{a_i\}$  in column  $i$ . As  $\mathcal{C}$  is a directed cycle basis, there is a unique matrix  $B \in \mathbb{Q}^{k \times k}$  for combining the cycles of  $\Phi$ , i.e.

$$\Gamma B = \Phi.$$

By construction, the restriction of  $\Phi$  to the co-tree arcs of  $H$  is just the identity matrix. Hence,  $B$  is the inverse matrix of  $\Gamma'$ .

Conversely, if the  $n - 1$  rows we remove contain a cycle  $C$ , consider its incidence vector  $\gamma_C$ . As  $\mathcal{C}$  is a directed cycle basis, we have a unique solution  $x_C \neq \mathbf{0}$  to the system  $\Gamma x = \gamma_C$ . But removing  $n - 1$  rows that contain  $C$  cause  $x_C$  to become a non-trivial linear combination of the zero vector, proving  $\Gamma'$  to be singular in this case.  $\square$

**Lemma 4.** *Let  $\Gamma$  be the  $m \times k$  cycle matrix of some directed cycle basis  $\mathcal{C}$ . Denote by  $A_1$  and  $A_2$  two regular  $k \times k$  submatrices of  $\Gamma$ . Then we have  $\det A_1 = \pm \det A_2$ .*

*Proof.* By lemma 3, the  $k$  rows of  $A_1$  are the co-tree arcs  $a_1, \dots, a_k$  of some spanning tree  $H$ . Again, consider the cycle matrix  $\Phi$  with the incidence vector of the unique cycle in  $H \cup \{a_i\}$  in column  $i$ . Then,

$$\Phi A_1 = \Gamma, \tag{8}$$

cf. Berge[2] for instance. From Schrijver[22], we know that  $\Phi$  is totally unimodular. Restricting system (8) to the rows of  $A_2$ , we obtain  $\Phi' A_1 = A_2$ . Because  $\det \Phi' = \pm 1$ , and since the det-function is distributive, we get  $\det A_1 = \pm \det A_2$ .  $\square$

The above lemma allows to define the determinant of a directed cycle basis.

**Definition 2 (Determinant of a Directed Cycle Basis).** *For a directed cycle basis  $\mathcal{C}$  with  $m \times k$  cycle matrix  $\Gamma$  and regular  $k \times k$  submatrix  $\Gamma'$ , we define the determinant of  $\mathcal{C}$  by*

$$\det \mathcal{C} := |\det \Gamma'|.$$

We first collect how this determinant behaves for general directed cycle bases, as well as for those who project onto undirected cycle bases.

**Corollary 5.** *The determinants of directed cycle bases are positive integers.*

**Theorem 6.** *A directed cycle basis  $\mathcal{C}$  projects onto a cycle basis for the underlying undirected graph, if and only if  $\det \mathcal{C}$  is odd.*

Due to space limitations, we omit a formal proof and just outline that taking the mod 2 projection after every step of the Laplace expansion for the determinant of an integer matrix maintains oddness simultaneously over both,  $\mathbb{Q}$  and  $\text{GF}(2)$ .

The following definition introduces the largest class of cycle bases from that we may select elements to give compact formulations of instances of the PESP.

**Definition 3 (Integral Cycle Basis).** *Let  $\mathcal{C} = \{C_1, \dots, C_k\}$  be cycles of a digraph  $D$ , where  $k$  is the cyclomatic number  $k = |A| - |V| + 1$ . If for every cycle  $C$  in  $D$ , we can find  $\lambda_1, \dots, \lambda_k \in \mathbb{Z}$ , such that  $C = \sum_{i=1}^k \lambda_i C_i$ , then  $\mathcal{C}$  is called an integral cycle basis.*

**Theorem 7 (Liebchen and Peeters[13]).** *A directed cycle basis  $\mathcal{C}$  is integral, if and only if  $\det \mathcal{C} = 1$ .*

*Proof.* Sufficiency can be seen immediately by applying Cramer's rule. To prove necessity, we use the fact that the columns of cycle matrices  $\Gamma$  of integral cycle bases generate the lattice of integers  $\mathbb{Z}^k$ , when restricted to  $k$  linear independent rows  $\Gamma'$ . There, it is an elementary equivalence that  $\Gamma'x = \gamma'$  has an integral solution  $x$  for every integral vector  $\gamma'$ , if and only if  $\det \Gamma' = \pm 1$ , cf. Schrijver[22].  $\square$

By definition, for every pair of a strictly fundamental cycle basis and an integral cycle basis with cycle matrices  $\Gamma$  and  $\Phi$ , respectively, there are unimodular matrices  $B_1$  and  $B_2$  with  $\Gamma B_1 = \Phi$  and  $\Phi B_2 = \Gamma$ . Thus, integral cycle bases immediately inherit the capabilities of strictly fundamental cycle bases for expressing instances of cyclic timetabling. Moreover, the example in figure 1 illustrates that, among the classes we consider in this paper, integral cycle bases are the most general structure for keeping such integer transformations. Hence, they are the most general class of cycle bases allowing to express instances of the periodic event scheduling problem.

**Corollary 8.** *Every integral cycle basis projects onto an undirected cycle basis.*

The cycle basis in figure 1 did already provide an example of a directed cycle basis that is not integral, but projects onto an undirected cycle basis.

Theorem 7 provides an efficient criterion for recognizing integral cycle basis. But this does not immediately induce an (efficient) algorithm for constructing a directed cycle basis being minimal among the integral cycle bases. Interpreting integral cycle bases in terms of lattices (Liebchen and Peeters[13]) might allow to apply methods for lattice basis reduction, such as the prominent  $L^3$ [12] and Lovász-Scarf[15] algorithms. But notice that our objective function has to be adapted carefully in that case.

## 4 Special Classes of Integral Cycle Bases

There are two important special subclasses of integral cycle bases. Both give rise to good heuristics for minimizing the linearized width function. We follow the notation of Whitney[24], where he introduced the concept of matroids.

**Definition 4 ((Strictly) Fundamental Cycle Basis).** *Let  $\mathcal{C} = \{C_1, \dots, C_k\}$  be a directed cycle basis. If for some, resp. any, permutation  $\pi$ , we have*

$$\forall i = 2, \dots, k : C_{\pi(i)} \setminus (C_{\pi(1)} \cup \dots \cup C_{\pi(i-1)}) \neq \emptyset,$$

*then  $\mathcal{C}$  is called a fundamental resp. strictly fundamental cycle basis.*

The following lemma provides a much more popular notion of strictly fundamental cycle bases.

**Lemma 9.** *The following properties of a directed cycle basis  $\mathcal{C}$  for a connected directed graph  $D$  are equivalent:*

1.  $\mathcal{C}$  is strictly fundamental.
2. The elements of  $\mathcal{C}$  are induced by the chords of some spanning tree.
3. There are at least  $k$  arcs that are part of exactly one cycle of  $\mathcal{C}$ .

We leave the simple formal proof to the reader.

Concerning fundamental cycle bases, by excluding minors, Hartvigsen and Zemel[9] gave a characterization of graphs in which every cycle basis is fundamental. Moreover, it is an elementary property of every fundamental cycle basis  $\mathcal{C}$ , that the first  $k$  rows of their arc-cycle incidence matrices  $\Gamma$  constitute an upper triangular matrix with diagonal elements in  $\{-1, +1\}$ , assuming  $\mathcal{C}$  being given such that  $\pi \equiv \text{id}$  complies with the definition. As an immediate consequence of theorem 7, we get



**Corollary 10.** *Fundamental cycle bases are integral cycle bases.*

The following example shows that the converse is not true.

**Example 2 (Liebchen and Peeters[13]).** *Consider the following 21 directed cycles in the digraph  $K_8$  with arcs directed arbitrarily:*

$$\begin{aligned} &(1, 2, 3) (1, 2, 4) (1, 2, 5) (1, 3, 4) (1, 3, 5) (1, 4, 6) (1, 5, 7) \\ &(1, 6, 8) (1, 7, 8) (2, 3, 6) (2, 3, 7) (2, 4, 7) (2, 5, 8) (2, 6, 8) \\ &(3, 4, 8) (3, 5, 6) (3, 7, 8) (4, 5, 6) (4, 5, 8) (4, 6, 7) (5, 6, 7) \end{aligned}$$

*This is a minimal directed cycle basis of  $K_8$ , because it involves only triangles. As there are  $15 \times 15$  submatrices with determinant one, it is in fact an integral cycle basis. But every arc is contained in more than one cycle, therefore this cycle basis cannot be fundamental. The node-minimality of this example has been discussed in Liebchen and Peeters[13].*

To complete our discussion, we provide another example.

**Example 3.** *Consider the directed cycle basis of  $K_5$  shown in figure 2. Permuting the cycles by  $\pi = (3, 4, 1, 2, 5, 6)$  establishes fundamentality. It is a minimal directed cycle basis, since it contains only triangles. Finally, this cycle basis is not strictly fundamental, as the set of arcs that*

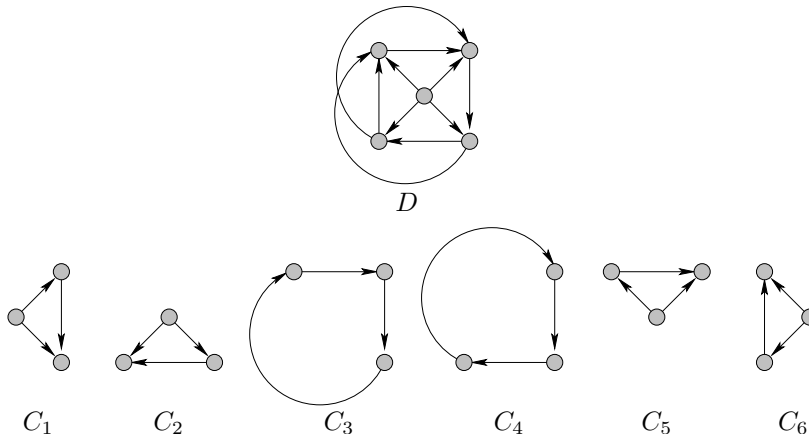


Figure 2: Fundamental cycle basis that is not strictly fundamental

*are part of several cycles contains a cycle itself.*

*To prove node-minimality of this example, consider  $K_4$ , which has exactly four triangles. Every minimal cycle basis selects three of them. In order that the set of arcs that are hit more than once contains a cycle, simple counting tells that such a cycle must have less than four arcs. But the arcs of the only non-basic triangle are exactly those that are covered only once by the three basic triangles.*

The Venn-diagram in figure 3 summarizes the relationship between the four major subclasses of directed cycle bases.

## 5 Algorithms

A first approach for constructing short integral cycle bases is to run one of the algorithms that construct a minimal undirected cycle basis. By orienting both edges and cycles arbitrarily, the determinant of the resulting directed cycle basis can be tested for being one. Notice that reversing an arc's or cycle's direction would translate into multiplying a row or column by minus one, which

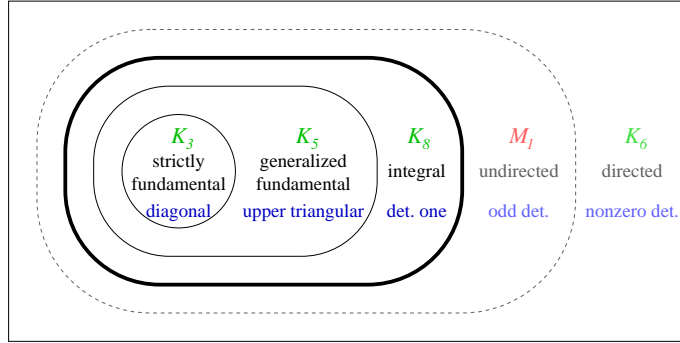


Figure 3: Map of directed cycle bases

is of no effect for the determinant of a cycle basis. But if our constructed minimal undirected cycle basis is not integral, it is worthless for us and we have to run other algorithms.

Deo et al.[6] introduced two sophisticated algorithms for constructing short strictly fundamental cycle bases: In the procedure *UV* (unexplored vertices), they grow the spanning tree by adding nodes that are adjacent to many non-tree nodes. In the procedure *NT* (non-tree edges), they grow the tree by selecting nodes that induce many non-tree edges in the current forest. Since every non-tree edge completes a fundamental cycle, they hope to get many short fundamental cycles from nodes added at the beginning, and only few long fundamental cycles from the last nodes.

But the computational results we are going to present in the next section demonstrate that we can do much better. The key are (generalized) fundamental cycle bases. As the complexity status of constructing a minimal cycle basis among the fundamental cycle bases is unknown to the author, we present several algorithms for constructing short fundamental – and thus integral – cycle bases, which will be formulated for undirected graphs.

## Fundamental Improvements to Spanning Trees

The first algorithm that we mention has been proposed by Berger[3]. To a certain extent, the ideas of de Pina[21] were simplified in order to maintain fundamentality. The algorithm is as follows:

1. Set  $\mathcal{C} := \emptyset$ .
2. Compute some spanning tree  $H$  with edges  $\{e_{k+1}, \dots, e_m\}$ .
3. For  $i = 1$  to  $k$  do
  - 3.1. For  $e_i = \{j, l\}$ , find a shortest path  $P_i$  between  $j$  and  $l$  which only uses arcs in  $\{e_1, \dots, e_{i-1}, e_{k+1}, \dots, e_m\}$ , and set  $C_i := e_i \cup P_i$ .
  - 3.2. Update  $\mathcal{C} := \mathcal{C} \cup C_i$ .

Obviously, the above procedure ensures  $e_i \in C_i \setminus \{C_1, \dots, C_{i-1}\}$ . Hence,  $\mathcal{C}$  is a fundamental cycle basis. Although this procedure is rather elementary, section 6 will point out the notable benefit it achieves even when starting with a rather good strictly fundamental cycle basis, e.g. the ones resulting from the procedures NT or UV. Similar ideas can be found in Nachtigall[18].

## Horton's Approximation Algorithm Produces a Fundamental Cycle Basis

The next algorithm for constructing a short fundamental cycle basis in an undirected graph  $G$  is due to Horton[10]. When he published the well-established first polynomial algorithm to construct a minimal cycle basis of an undirected graph, he has not been content with its running time  $\mathcal{O}(m^3n)$ .

This is why he added a *fast algorithm for a suboptimal cycle basis*. Its running time is only  $\mathcal{O}(n^2m)$ , and it is guaranteed to construct a cycle basis with no more than  $3(n-1)(n-2)/2$  edges in the unweighted case. In particular, when considering  $K_n$ , this algorithm even constructs a minimal fundamental cycle basis – but here the still simpler star trees achieve the minimum as well.

Further, the guarantee has to be compared to length  $\Omega(n^3)$  for the worst-case solutions that are known for the tree-heuristics proposed by Deo et al.[5]. We are going to show that Horton's heuristic always constructs a fundamental cycle basis for a weighted connected graph  $G$ .

1. Set  $\mathcal{C} := \emptyset$  and  $G' := G$ .
2. For  $i = 1$  to  $n - 1$  do
  - 2.1. Choose a vertex  $x_i$  of minimum degree  $\nu$  in  $G'$ .
  - 2.2. Find all shortest paths lengths in  $G' \setminus x_i$  between neighbors  $x_{i_1}, \dots, x_{i_\nu}$  of  $x_i$ .
  - 2.3. Define a new artificial network  $N_i$  by
    - 2.3.1. introducing a node  $s$  for every edge  $\{x_i, x_{i_s}\}$  in  $G'$  and
    - 2.3.2. defining the length of the *branch*  $\{s, t\}$  to be the length of a shortest path between  $x_{i_s}$  and  $x_{i_t}$  in  $G' \setminus x_i$ .
  - 2.4. Find a minimal spanning tree  $H_i$  for  $N_i$ .
  - 2.5. Let  $C_{i_1}, \dots, C_{i_{\nu-1}}$  be the cycles in  $G'$  that correspond to branches of  $H_i$ .
  - 2.6. Update  $\mathcal{C} := \mathcal{C} \cup \{C_{i_1}, \dots, C_{i_{\nu-1}}\}$  and  $G' := G' \setminus x_i$ .

First, observe that none of the edges  $\{x_i, x_{i_s}\}$  can be part of any cycle  $C_r$  of a later iteration  $r > i$ , because at the end of iteration  $i$  the vertex  $x_i$  is removed from  $G'$ . Hence, fundamentality follows by ordering, within each iteration  $i$ , the edges and cycles such that  $e_{i_j} \in C_{i_j} \setminus (C_{i_1}, \dots, C_{i_{j-1}})$  for all  $j = 2, \dots, \nu - 1$ . Moreover, every leaf  $s$  of  $H_i$  encodes an edge  $\{x_i, x_{i_s}\}$  that is part of only one cycle. Finally, as  $H_i$  is a tree, by recursively removing branches that are incident to a leaf of the remaining tree, we process every branch of the initial tree  $H_i$ .

We order the branches  $b_1, \dots, b_{\nu-1}$  of  $H_i$  according to such an elimination scheme, i.e. for every branch  $b_j = \{s_j, t_j\}$ , node  $s_j$  is a leaf subject to the subtree

$$H_i \setminus \bigcup_{l=1}^{j-1} \{b_l\}.$$

Migrating back to the original graph  $G'$ , for  $j = 1, \dots, \nu - 1$ , we define  $e_{i_j}$  to correspond to the leaf  $s_{\nu-j}$ , and  $C_{i_j}$  to be modeled by the branch  $b_{\nu-j}$ . This just complies with the definition.

## General Rules for Bases Changes

Finally, by interpreting an integral cycle basis  $\mathcal{C}$  of a directed graph  $D$  as a basis for the lattice of integer circulations in  $D$ , we develop a very general rule for improvements. Let  $\Gamma$  denote the cycle matrix of  $\mathcal{C}$ . Then, we can construct every integral cycle basis  $\mathcal{C}'$  of  $D$  by  $\Gamma' = \Gamma U$ , with  $U$  being an unimodular matrix. From lattice theory (cf. Schrijver[22]), we may borrow that unimodular matrices encode precisely elementary column operations.

The only elementary column operation relevant for our purpose is the addition of an integer multiple of one column to another column. One may think of it as a neighborhood, a rather small one, and far from being exact. But we may also combine several cycles within one step. The only restriction we have to obey is to exchange a basic cycle for another cycle, only if we must traverse the exiting basic cycle exactly once in the unique integer linear combination of the new cycle.

As this neighborhood becomes much bigger, we propose to analyze it – still heuristically – the other way round. Consider the current basis  $\mathcal{C} = \{C_1, \dots, C_k\}$  and detect some very short non-basic cycle  $C$ . By analyzing its integer combination of basic cycles

$$\gamma_C = \sum_{i=1}^k \lambda_i \gamma_{C_i}, \quad \lambda_i \in \mathbb{Z}, \quad (9)$$

we may exchange  $C$  for any basic cycle  $C_i$  with  $|\lambda_i| = 1$ . If  $C$  is shorter than the largest one of these cycles, the exchange will be effected. By the way, this approach is exactly what is realized in the fundamental improvement heuristic, in a prudent and well-defined order, of course.

## 6 Computational Results

At this point, we are able to report the benefit of the above considerations for the initial application. We are going to solve two instances of the cyclic timetabling problem.

The first one has been made available to us by Deutsche Bahn AG. In that, we want to minimize simultaneously both the number of vehicles required to operate ten given pairs of hourly served ICE/IC railway lines, and the waiting times faced by passengers along the 40 most important connections. Single tracks and optional additional stopping times of up to five minutes at major stations cause an average span of 75.9% of the period time for the 186 arcs that remain, after elimination of redundancies within the initial model with 4104 periodic events.

The second instance models the Berlin Underground. For the eight pairs of directed lines, which are operated every 10 minutes at a precision of 30 seconds, we consider any of the 144 connections for passengers. Additional stopping time is allowed to insert for 22 stopping activities. Hereby, the 188 arcs of the contracted graph have an average span of 69.5% of the period time. In an optimal solution, 3.5 minutes of additional stopping time will be inserted, and the weighted average passengers' effective waiting time is less than one and a half minutes.

In tables 1 and 2 we start by giving the base ten logarithm of the width of the cycle bases that are constructed by the heuristics proposed in Deo et al.[6] These have been applied for the arcs' weights chosen as one, the span  $d_a = u_a - l_a$ , or the negative of the span  $T - d_a$ . In addition, minimal spanning trees have been computed for two of these weight functions. Then, to each of those strictly fundamental cycle bases, the fundamental improvement heuristic has been applied. For sake of completeness, the width of a minimal cycle basis subject to the linearized objective (6) is given as well. Only the heuristic proposed by Horton has not been implemented, so far.

Subsequently, we report the behavior of CPLEX<sup>®</sup>[4] when faced with the different problem formulations. We use version 8.0 with standard parameters, except for *strong branching* as variable selection strategy and *aggressive cut generation*. The computations have been performed on an AMD Athlon<sup>®</sup> XP 1500+ with 512 MB main memory.

A key observation are the considerable positive correlations ( $> 0.44$  and  $> 0.67$ , resp.) between the base ten logarithm of the width of the cycle basis and the running time of the MIP solver. With the exception of only one case, the fundamental improvement either results in a notable speed-up, or enables an instance to be solved to optimality, in case that limits are reached when not applying the heuristic.

But there is another phenomenon, we want to point out. Compare the behavior of the MIP solver for MST span without improvement with the one for MST nspan plus the heuristic in table 1. Although the cycle bases involved differ significantly in size, their solution behavior is rather similar. We assume that this is due to the fact, that strictly fundamental cycle bases, which are encoded by cycle matrices that contain the unit matrix, might be much more advantageous for an LP-based MIP solver, than only triangular matrices which are contained in the cycle matrices of (generalized) fundamental cycle bases. Peeters[20] gives similar explanations when faced with this effect in his computations.

Now, let us have a detailed look at the cycle bases. We will investigate the distribution of the number of possible values for the integer variables. For the ICE/IC instance, in the initial

algorithm weight	global minima	MST		UV			NT
		span	nspan	unit	span	nspan	unit
initial width	34.3	65.9	88.4	59.7	58.6	61.2	58.5
fund. improve	–	41.0	43.2	42.9	42.2	42.9	42.7
without fundamental improvement							
time (s)	–	14720	> 28800	20029	23726	6388	> 28800
memory (MB)	–	13	113	29	30	10	48
status	–	opt	timelimit	opt	opt	opt	timelimit
solution	620486	667080					629993
fundamental improvement applied							
time (s)	–	807	11985	9305	17963	1103	> 28800
memory (MB)	–	1	23	24	30	3	114
status	–	opt	opt	opt	opt	opt	timelimit
solution	–						626051

Table 1: Influence of cycle bases on running times for timetabling (hourly served ICE/IC lines)

algorithm weight	global minima	MST		UV			NT
		span	nspan	unit	span	nspan	unit
initial width	39.4	62.7	84.9	67.0	71.1	65.9	67.0
fund. improve	–	46.6	48.2	46.4	46.6	46.5	46.4
without fundamental improvement							
time (s)	–	94	> 28800	747	9453	74	748
memory (MB)	–	1	87	1	10	1	1
status	–	opt	timelimit	opt	opt	opt	opt
solution	39820	39915					
fundamental improvement applied							
time (s)	–	137	159	65	265	19	65
memory (MB)	–	1	1	1	1	1	1
status	–	opt	opt	opt	opt	opt	opt
solution	–						

Table 2: Influence of cycle bases on running times for timetabling (Berlin Underground)

MST cycle basis with arcs' spans as weights, only 73% of the integer variables can be limited to at most four values. But after applying the fundamental improvement, there are more than 78% of the variables bounded to at most two values, cf. figure 4. This causes reductions of more than 92% for both, time and memory usage.

Notice that the reported running times are influenced by our quest for comparability of the results. In practice, our system will add a certain number of additional valid inequalities to the initial MIP formulation. But their coefficients depend on the problem formulation, and, in turn, the latter heavily depends on the cycle basis. Thus, we did not add valid inequalities in any of the above calculations. Let us however mention that even for the instance of Deutsche Bahn AG, an optimal solution has been achieved after only 66 seconds of CPU time for a formulation refined by 115 additional valid inequalities that were separated in less than 80 seconds.

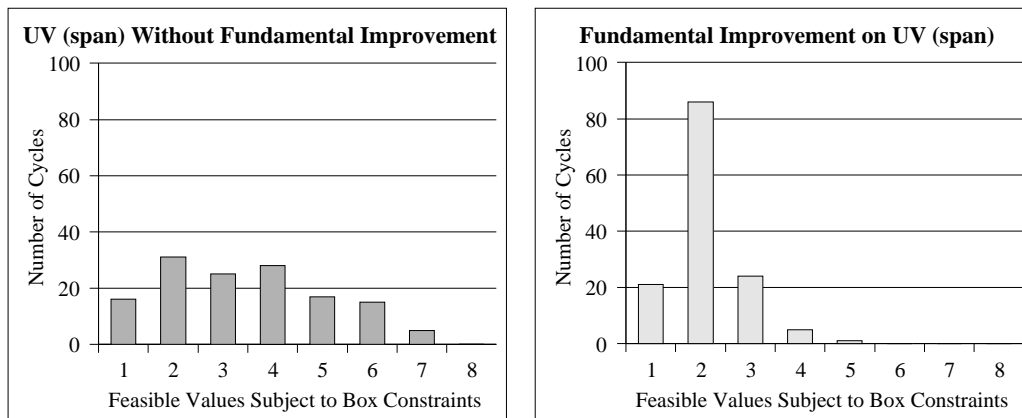


Figure 4: Shift in distribution of cycle widths due to the fundamental improvements

## 7 Conclusions

We generalized the standard approach for formulating instances of the cyclic timetabling problem, which is based on strictly fundamental cycle bases. Integral cycle bases have been established to be the most general class of directed cycle bases that enable the modeling of cyclic timetabling problems. Finally, we presented algorithms that construct short fundamental cycle bases with respect to a reliable empirical measure for estimating the running time of a mixed-integer solver for the originating application.

But some questions remain open. The greatest impact on cyclic timetabling would emerge from the classification of the computational complexity of minimizing a (linear) objective function over the class of fundamental, or even integral, cycle bases. Moreover, we would profit from progress achieved in the area of integer lattices. And, to adapt a question of Hartvigsen and Zemel[9]:

Does every graph have a fundamental cycle basis,  
which is minimal among *all* cycle bases?

## 8 Acknowledgments

First of all, I would like to thank Leon Peeters for many fruitful discussions. Sabine Cornelsen has been a valuable referee, in particular for several examples, which lead to deeper theoretical results in one case. The work of Berit Johannes simplified many phrases. Last but not least, Bob Bixby's hints helped increasing the performance of CPLEX.

## References

- [1] Amaldi, E. (2003) Personal Communication. Politecnico di Milano, Italy
- [2] Berge, C. (1962) The Theory of Graphs and its Applications. John Wiley & Sons
- [3] Berger, F. (2002) Minimale Kreisbasen in Graphen. Lecture on the annual meeting of the DMV in Halle, Germany
- [4] CPLEX 8.0 (2002) <http://www.ilog.com/products/cplex> ILOG SA, France.
- [5] Deo, N., Prabhu, M., Krishnamoorthy, M.S. (1982) Algorithms for Generating Fundamental Cycles in a Graph. ACM Transactions on Mathematical Software **8**, 26–42

- [6] Deo, N., Kumar, N., Parsons, J. (1995) Minimum-Length Fundamental-Cycle Set Problem: A New Heuristic and an SIMD Implementation. Technical Report CS-TR-95-04, University of Central Florida, Orlando
- [7] Gleiss, P. (2001) Short Cycles. Ph.D. Thesis, University of Vienna, Austria
- [8] Golynski, A., Horton, J.D. (2002) A Polynomial Time Algorithm to Find the Minimum Cycle Basis of a Regular Matroid. In: SWAT 2002, Springer LNCS 2368, edited by M. Penttonen and E. Meineche Schmidt
- [9] Hartvigsen, D., Zemel, E. (1989) Is Every Cycle Basis Fundamental? *Journal of Graph Theory* **13**, 117–137
- [10] Horton, J.D. (1987) A polynomial-time algorithm to find the shortest cycle basis of a graph. *SIAM Journal on Computing* **16**, 358–366
- [11] Krista, M. (1996) Verfahren zur Fahrplanoptimierung dargestellt am Beispiel der Synchronzeiten (Methods for Timetable Optimization Illustrated by Synchronous Times). Ph.D. Thesis, Technical University Braunschweig, Germany, In German
- [12] Lenstra, A.K., Lenstra, H.W., Lovász, L. (1982) Factoring polynomials with rational coefficients. *Mathematische Annalen* **261**, 515–534
- [13] Liebchen, C., Peeters, L. (2002) On Cyclic Timetabling and Cycles in Graphs. Technical Report 761/2002, TU Berlin
- [14] Liebchen, C., Peeters, L. (2002) Some Practical Aspects of Periodic Timetabling. In: Operations Research 2001, Springer, edited by P. Chamoni et al.
- [15] Lovász, L., Scarf, H.E. (1992) The Generalized Basis Reduction Algorithm. *Mathematics of Operations Research* **17**, 751–764
- [16] Nachtigall, K. (1994) A Branch and Cut Approach for Periodic Network Programming. *Hildesheimer Informatik-Berichte* 29
- [17] Nachtigall, K. (1996) Cutting planes for a polyhedron associated with a periodic network. *DLR Interner Bericht* 17
- [18] Nachtigall, K. (1996) Periodic network optimization with different arc frequencies. *Discrete Applied Mathematics* **69**, 1–17
- [19] Odijk, M. (1997) Railway Timetable Generation. Ph.D. Thesis, Delft University of Technology, The Netherlands
- [20] Peeters, L. (2003) Personal Communication. University of Constance, Germany
- [21] de Pina, J.C. (1995) Applications of Shortest Path Methods. Ph.D. Thesis, University of Amsterdam, The Netherlands
- [22] Schrijver, A. (1998) *Theory of Linear and Integer Programming*. Second Edition. John Wiley & Sons
- [23] Serafini, P., Ukovich, W. (1989) A mathematical model for periodic scheduling problems. *SIAM Journal on Discrete Mathematics* **2**, 550–581
- [24] Whitney, H. (1935) On the Abstract Properties of Linear Dependence. *American Journal of Mathematics* **57**, 509–533

Reports from the group

## “Combinatorial Optimization and Graph Algorithms”

of the Department of Mathematics, TU Berlin

- 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
- 723/2001** *Berit Johannes*: Scheduling Parallel Jobs to Minimize Makespan
- 716/2001** *Christian Liebchen*: The Periodic Assignment Problem (PAP) May Be Solved Greedily
- 711/2001** *Esther M. Arkin, Michael A. Bender, Sándor P. Fekete, Joseph S. B. Mitchell, and Martin Skutella*: The Freeze-Tag Problem: How to Wake Up a Swarm of Robots
- 710/2001** *Esther M. Arkin, Sándor P. Fekete, and Joseph S. B. Mitchell*: Algorithms for Manufacturing Paperclips and Sheet Metal Structures
- 705/2000** *Ekkehard Köhler*: Recognizing Graphs without Asteroidal Triples
- 704/2000** *Ekkehard Köhler*: AT-free, coAT-free Graphs and AT-free Posets
- 702/2000** *Frederik Stork*: Branch-and-Bound Algorithms for Stochastic Resource-Constrained Project Scheduling
- 700/2000** *Rolf H. Möhring*: Scheduling under uncertainty: Bounding the makespan distribution
- 698/2000** *Sándor P. Fekete, Ekkehard Köhler, and Jürgen Teich*: More-dimensional packing with order constraints
- 697/2000** *Sándor P. Fekete, Ekkehard Köhler, and Jürgen Teich*: Extending partial suborders and implication classes
- 696/2000** *Sándor P. Fekete, Ekkehard Köhler, and Jürgen Teich*: Optimal FPGA module placement with temporal precedence constraints
- 695/2000** *Sándor P. Fekete, Henk Meijer, André Rohe, and Walter Tietze*: Solving a “hard” problem to approximate an “easy” one: heuristics for maximum matchings and maximum Traveling Salesman Problems
- 694/2000** *Esther M. Arkin, Sándor P. Fekete, Ferran Hurtado, Joseph S. B. Mitchell, Marc Noy, Vera Sacristán and Saurabh Sethia*: On the reflexivity of point sets
- 693/2000** *Frederik Stork and Marc Uetz*: On the representation of resource constraints in project scheduling
- 691/2000** *Martin Skutella and Marc Uetz*: Scheduling precedence constrained jobs with stochastic processing times on parallel machines
- 689/2000** *Rolf H. Möhring, Martin Skutella, and Frederik Stork*: Scheduling with AND/OR precedence constraints
- 685/2000** *Martin Skutella*: Approximating the single source unsplittable min-cost flow problem
- 684/2000** *Han Hoogeveen, Martin Skutella, and Gerhard J. Woeginger*: Preemptive scheduling with rejection



- 683/2000** *Martin Skutella*: Convex quadratic and semidefinite programming relaxations in Scheduling
- 682/2000** *Rolf H. Möhring and Marc Uetz*: Scheduling scarce resources in chemical engineering
- 681/2000** *Rolf H. Möhring*: Scheduling under uncertainty: optimizing against a randomizing adversary
- 680/2000** *Rolf H. Möhring, Andreas S. Schulz, Frederik Stork, and Marc Uetz*: Solving project scheduling problems by minimum cut computations (Journal version for the previous Reports 620 and 661)
- 674/2000** *Esther M. Arkin, Michael A. Bender, Erik D. Demaine, Sándor P. Fekete, Joseph S. B. Mitchell, and Saurabh Sethia*: Optimal covering tours with turn costs
- 669/2000** *Michael Naatz*: A note on a question of C. D. Savage
- 667/2000** *Sándor P. Fekete and Henk Meijer*: On geometric maximum weight cliques
- 666/2000** *Sándor P. Fekete, Joseph S. B. Mitchell, and Karin Weinbrecht*: On the continuous Weber and  $k$ -median problems
- 664/2000** *Rolf H. Möhring, Andreas S. Schulz, Frederik Stork, and Marc Uetz*: On project scheduling with irregular starting time costs
- 661/2000** *Frederik Stork and Marc Uetz*: Resource-constrained project scheduling: from a Lagrangian relaxation to competitive solutions

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>