Separation of Cycle Inequalities for the Periodic Timetabling Problem*

Ralf Borndörfer[1], Heide Hoppmann[2], and Marika Karbstein[3]

1 **Zuse Institute Berlin, Berlin, Germany**
`borndoerfer@zib.de`
2 **Zuse Institute Berlin, Berlin, Germany**
`hoppmann@zib.de`
3 **Zuse Institute Berlin, Berlin, Germany**
`karbstein@zib.de`

Abstract

Cycle inequalities play an important role in the polyhedral study of the periodic timetabling problem. We give the first pseudo-polynomial time separation algorithm for cycle inequalities, and we give a rigorous proof for the pseudo-polynomial time separability of the change-cycle inequalities. The efficiency of these cutting planes is demonstrated on real-world instances of the periodic timetabling problem.

1998 ACM Subject Classification F.2.2. Nonnumerical Algorithms and Problems, G.1.6 Optimization

Keywords and phrases periodic timetabling, cycle inequalities, separation algorithm

Digital Object Identifier 10.4230/LIPIcs.ESA.2016.21

1 Introduction

Periodic timetable construction is a fascinating problem because it is intuitively understood and mathematically formulated, but very hard to solve. In fact, even though real world instances give rise to relatively small optimization models, branch-and-bound based methods can easily stall with large duality gaps. The likely reasons for this resistivity are the occurrence of genuine integer variables, symmetries, and modulo constraints.

The classical approach to periodic timetabling is to use a formulation in terms of the periodic event scheduling problem (PESP) by Serafini and Ukovich [14]. This model has been the basis for the development of a variety of exact and heuristic solution methods for the optimization and the feasibility version. Integer programming approaches were proposed, e.g., by Nachtigall [9], Lindner [7], and Liebchen [3]. A topological search method based on cohomology feasibility was used by Schrijver [12] to optimize a Dutch railway timetable. A modulo network simplex heuristic was invented by Nachtigall and Opitz [8]. Liebchen and Peeters [5] studied the relation to integral cycle bases to find tighter lower bounds. A SAT approach for the feasibility problem was developed by Kümmling et al. [2]. A comprehensive and up-to-date survey of the literature on mathematical timetable optimization and its applications is summarized in Sels et al. [13].

The best method to compute lower bounds for the optimization problem is to study the polyhedral structure of the periodic timetabling problem (PTP) associated with the PESP.

* This research was carried out in the framework of Matheon supported by Einstein Foundation Berlin.

24th Annual European Symposium on Algorithms (ESA 2016).
Editors: Piotr Sankowski and Christos Zaroliagis; Article No. 21; pp. 21:1–21:13
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

Several classes of valid inequalities have been identified, namely, chain, cycle, change-cycle, flow, and multi-circuit inequalities, see [3, 7, 9, 10, 11]. Some of them are known to be facet defining or in some Chvátal closure for the PESP polytope or relaxations of it under certain conditions [6]. It is also known that the change-cycle inequalities can be separated in pseudo-polynomial time [9]. The cycle inequalities have been used computationally by means of heuristic separation. They improve the lower bound significantly and are considered to be "the computationally most interesting cuts" [3, p. 210].

We study in this paper the separation problem for the cycle and the change-cycle inequalities for the periodic timetabling problem. We give the first pseudo-polynomial time separation algorithm for cycle inequalities. Its complexity is $\mathcal{O}(Tn^2m)$, where T is the period time, n the number of nodes, and m the number of arcs. The change-cycle inequalities have been studied by Nachtigall [9], who gave a rough sketch of a pseudo-polynomial algorithm and claimed a complexity of $\mathcal{O}(T(mn+n^2))$. We cannot follow this argument, but give a precise description of the algorithm and prove a complexity of $\mathcal{O}(T^2n^2m)$. Computational results on real world instances from a Dutch railway system and two German cities corroborate the efficiency of these cuts.

The paper is structured as follows. Section 2 gives a mathematical statement of the problem. Section 3 introduces the periodic slack polyhedron and states the cycle and the change-cycle inequalities. Sections 4 and 5 contain the separation algorithms for the change-cycle inequalities and the cycle inequalities. They are based on similar ideas, but cycle separation requires the setup of an additional auxiliary graph. Section 6 concludes with computational results.

2 Periodic Timetabling Problem and PESP

Most models in the literature about periodic timetabling are based on the *periodic event scheduling problem* (PESP) developed by [14]. In this problem, we are given a directed graph $\mathcal{N} = (V, A)$, the *event-activity network*. The nodes V are called *events* and represent arrivals and departures of lines at their stations. The arcs $A \subseteq V \times V$ are called *activities* and model lines driving between stations, waiting at stations, and possible transfers for passengers between lines at stations. Further, each activity $a \in A$ is associated with a lower and an upper time bound $\ell_a, u_a \in \mathbb{Q}_{\geq 0}$, respectively, on its duration. Let $n = |V|$ be the number of events and $m = |A|$ be the number of activities.

A *periodic timetable* $\pi : V \to [0, T)$ determines the timings of all events, which are assumed to repeat periodically w.r.t. a *period time* $T \in \mathbb{N}$. Given $x \in \mathbb{Q}$, we define the modulo operator by $[x]_T := \min\{x + zT : x + zT \geq 0, z \in \mathbb{Z}\}$. We call a timetable *feasible* if the *periodic interval constraints*

$$[\pi_w - \pi_v - \ell_a]_T \in [0, u_a - \ell_a] \quad \forall\, a = (v, w) \in A \tag{1}$$

are satisfied. We assume w.l.o.g. that $\ell_a < T$ and $u_a - \ell_a < T$ for all $a \in A$. Many operational requirements can be modeled with the constraints (1), see [4]. For a feasible timetable π, the *periodic tension* of activity $a \in A$ is defined by $x_a := \ell_a + [\pi_w - \pi_v - \ell_a]_T$ and corresponds to its duration. The *periodic slack* of activity $a \in A$ is defined by $y_a := [\pi_w - \pi_v - \ell_a]_T$. Given activity weights $w \in \mathbb{Q}^A$, the goal of the periodic timetabling problem is to find a feasible timetable that minimizes the weighted sum of the periodic slacks, i.e., $\min \sum_{a \in A} w_a\, y_a$.

An *oriented cycle* C in $\mathcal{N}$ is a sequence $C = (v_0, a_1, v_1, \ldots, a_k, v_k)$, where $k \geq 1$, $v_1, \ldots, v_k \in V$, $a_1, \ldots, a_k \in A$, $v_0 = v_k$, and $a_i \in \{(v_{i-1}, v_i), (v_i, v_{i-1})\}$. Activities with $a_i = (v_{i-1}, v_i) \in C$ are called *forward directed* and activities with $a_i = (v_i, v_{i-1}) \in C$

backward directed. An oriented cycle containing only forward directed activities is called a *circuit*. An oriented cycle is *elementary* if no event appears more than once in the sequence. For an oriented cycle C in $\mathcal{N}$, we define its incident vector $\gamma_C \in \{-1,0,1\}^A$ as

$$\gamma_{C_a} = \begin{cases} 1 & \text{if } a \in C \text{ and } a \text{ is forward directed,} \\ -1 & \text{if } a \in C \text{ and } a \text{ is backward directed,} \\ 0 & \text{if } a \notin C. \end{cases}$$

For convenience, we will refer interchangeably to C and γ_C. Let $\mathcal{B} = \{C_1, \ldots, C_\nu\}$, $\nu = m - n + 1$, be a cycle basis of $\mathcal{N}$ and denote by $\Gamma \in \mathbb{Z}^{\mathcal{B}\times A}$ the corresponding cycle matrix, i.e., the rows of Γ correspond to the characteristic vectors $\gamma_{C_i} \in \{-1,0,1\}^A$, $i \in \{1,\ldots,\nu\}$. Introducing periodic slack variables $y \in \mathbb{R}^A$ and *periodic offset* variables $z \in \mathbb{Z}^A$, we can state the periodic timetabling problem as the following mixed-integer program [9, 3]:

$$\begin{aligned} \text{(PTP)} \quad \min \quad & \sum_{a\in A} w_a y_a \\ \text{s.t.} \quad & \Gamma\, y - T\,\Gamma\, z = -\Gamma\ell && (2) \\ & 0 \le y \le u - \ell && (3) \\ & z \in \mathbb{Z}^A && (4) \end{aligned}$$

3 Periodic Slack Polyhedron

The literature considers different versions of the PTP polyhedron, e.g., the projection on the space of the periodic slack variables or the periodic offset variables, see [9, 3, 6]. Nachtigall [9] also considers the polyhedron that is obtained when the upper bounds in constraints (3) are omitted. In the following, we study the polyhedron $P_{IP}(\text{PTP})$ associated with the feasible solutions of (PTP), i.e., a polyhedron defined in the slack and offset space. We recall the cycle and change-cycle inequalities in a unified notation.

▶ **Definition 1.** The *periodic slack and offset space* is defined by

$$\mathcal{S} = \left\{(y,z) \in \mathbb{R}^A \times \mathbb{Z}^A | \Gamma\, y - T\,\Gamma\, z = -\Gamma\ell, 0 \le y \le u - \ell\right\}.$$

The *periodic slack polyhedron* is defined by

$$P_{IP}(\text{PTP}) = \operatorname{conv}(\mathcal{S})$$

and the corresponding LP relaxation by

$$P_{LP}(\text{PTP}) = \left\{(y,z) \in \mathbb{R}^A \times \mathbb{R}^A | \Gamma\, y - T\,\Gamma\, z = -\Gamma\ell, 0 \le y \le u - \ell\right\}.$$

The following lemma shows that the cycle equations (2) do not only hold for integer periodic offset variables and the cycles of the cycle basis but for any feasible solution of the LP relaxation of (PTP) and any cycle in the event-activity network.

▶ **Lemma 2.** *Let $(y,z) \in P_{LP}(\text{PTP})$ and let $\gamma \in \mathbb{Z}^A$ be an oriented cycle in $\mathcal{N}$. Then we have*

$$\gamma^t y = -\gamma^t \ell + T\,\gamma^t z.$$

Proof. Since $\mathcal{B}$ is a cycle basis, there exists a vector $\lambda \in \mathbb{R}^\nu$ such that $\gamma = \Gamma^t\,\lambda$. Hence, we get

$$\gamma^t y = \lambda^t\,\Gamma\, y = \lambda^t\,(-\Gamma\,\ell + T\,\Gamma\, z) = -(\Gamma^t\,\lambda)^t\ell + T\,(\Gamma^t\,\lambda)^t\, z = -\gamma^t\ell + T\,\gamma^t\, z. \qquad ◀$$

Lemma 2 implies that for any feasible solution of (PTP) the following modulo cycle equations hold.

▶ **Corollary 3.** *Let $(y, z) \in \mathcal{S}$ and let $\gamma \in \mathbb{Z}^A$ be an oriented cycle in $\mathcal{N}$. Then we have*

$$\gamma^t y \equiv_T -\gamma^t \ell.$$

Proof. The corollary follows directly from $\gamma^t z \in \mathbb{Z}$ and Lemma 2. ◀

For convenience, we introduce further notation. For an oriented cycle $\gamma \in \mathbb{Z}^m$ in $\mathcal{N}$ we define the positive part $\gamma_+ \in \mathbb{Z}^m$ and the negative part $\gamma_- \in \mathbb{Z}^m$, respectively, by

$$\gamma_{+,a} = \begin{cases} 1 & \text{if } \gamma_a = 1 \\ 0 & \text{else} \end{cases} \quad \text{and} \quad \gamma_{-,a} = \begin{cases} 1 & \text{if } \gamma_a = -1 \\ 0 & \text{else} \end{cases}$$

for all $a \in A$ i.e., $\gamma = \gamma_+ - \gamma_-$.

The following class of valid inequalities was introduced by Nachtigall [9] and are defined for every oriented cycle in the event-activity network.

▶ **Theorem 4.** *Let $\gamma \in \mathbb{Z}^A$ be an oriented cycle in $\mathcal{N}$ and define $\alpha = [-\gamma^t \ell]_T$. Then the* change-cycle inequality

$$(T - \alpha)\, \gamma_+^t y + \alpha\, \gamma_-^t y \geq \alpha\, (T - \alpha) \tag{5}$$

is valid for P_{IP}(PTP).

A second class of inequalities are also induced by the oriented cycles in the event-activity network and were first described by Odijk [10]. These inequalities are denoted as *cycle inequalities*. They are usually defined in terms of the periodic offset variables. We will show next that they can also be defined in terms of the slack variables.

▶ **Theorem 5.** *Let $\gamma \in \mathbb{Z}^A$ be an oriented cycle in $\mathcal{N}$. Then the z*-cycle inequality

$$\gamma^t\, z \geq \left\lceil \frac{1}{T} \left(\gamma_+^t \ell - \gamma_-^t u \right) \right\rceil \tag{6}$$

is valid for P_{IP}(PTP).

Proof. Let $(y, z) \in \mathcal{S}$. We have with Lemma 2

$$T\, \gamma^t\, z = \gamma^t y + \gamma^t \ell = \gamma_+^t y - \gamma_-^t y + \gamma^t \ell \geq -\gamma_-^t\, (u - \ell) + \gamma^t \ell = \gamma_+^t \ell - \gamma_-^t u.$$

Since $\gamma^t\, z \in \mathbb{Z}$, the inequality (6) follows. ◀

▶ **Lemma 6.** *Let $\alpha \in \mathbb{R}$, then*

$$[-\alpha]_T + \alpha = T \left\lceil \frac{1}{T} \alpha \right\rceil. \tag{7}$$

Proof. Let $z \in \mathbb{Z}$ and $\alpha \in \mathbb{R}$ then $-\alpha - T\, z = [-\alpha]_T$ and $-1 < -\frac{1}{T}\, [-\alpha]_T \leq 0$. We get

$$\begin{aligned} [-\alpha]_T + \alpha = -T\, z &= T\Big(- z + \underbrace{\left\lceil -\frac{1}{T}\, [-\alpha]_T \right\rceil}_{=0} \Big) \\ &= T \left\lceil -z - \frac{1}{T}\, [-\alpha]_T \right\rceil = T \left\lceil \frac{1}{T} (-T\, z - [-\alpha]_T) \right\rceil = T \left\lceil \frac{1}{T} \alpha \right\rceil. \end{aligned}$$

◀

With Lemma 6 we can show that the z-cycle inequalities can be expressed equivalently in terms of the slack variables.

▶ **Theorem 7.** *Let $\gamma \in \mathbb{Z}^A$ be an oriented cycle in $\mathcal{N}$. Let $(y, z) \in P_{LP}(\mathrm{PTP})$, then the* z-cycle inequality (6) *holds if and only if the the* y-cycle inequality

$$\gamma^t y \geq \left[-\gamma_+^t \ell + \gamma_-^t u\right]_T + \gamma_-^t(\ell - u) \tag{8}$$

holds.

Proof. Let $(y, z) \in P_{LP}(\mathrm{PTP})$ and assume that z satisfies the *z-cycle inequality* (6) for γ. Using first Lemma 6 and then Lemma 2 we have

$$\begin{aligned}
& & \gamma^t z &\geq \left\lceil \frac{1}{T}\left(\gamma_+^t \ell - \gamma_-^t u\right)\right\rceil \\
& & &= \frac{1}{T}\left(\left[-\gamma_+^t \ell + \gamma_-^t u\right]_T + \gamma_+^t \ell - \gamma_-^t u\right) \\
\Leftrightarrow & & \gamma^t y + \gamma^t \ell &\geq \left[-\gamma_+^t \ell + \gamma_-^t u\right]_T + \gamma_+^t \ell - \gamma_-^t u \\
\Leftrightarrow & & \gamma^t y &\geq \left[-\gamma_+^t \ell + \gamma_-^t u\right]_T + \gamma_+^t \ell - \gamma_-^t u - \gamma^t \ell \\
& & &= \left[-\gamma_+^t \ell + \gamma_-^t u\right]_T + \gamma_+^t \ell - \gamma_-^t u - \gamma_+^t \ell + \gamma_-^t \ell \\
& & &= \left[-\gamma_+^t \ell + \gamma_-^t u\right]_T - \gamma_-^t u + \gamma_-^t \ell \\
& & &= \left[-\gamma_+^t \ell + \gamma_-^t u\right]_T + \gamma_-^t(\ell - u).
\end{aligned}$$

◀

▶ **Corollary 8.** *Let $\gamma \in \mathbb{Z}^A$ be an oriented cycle in $\mathcal{N}$. Then the* y-cycle inequality

$$\gamma^t y \geq \left[-\gamma_+^t \ell + \gamma_-^t u\right]_T + \gamma_-^t(\ell - u)$$

is valid for $P_{IP}(\mathrm{PTP})$.

4 Separation of Change-Cycle Inequalities

In this section we describe a pseudo-polynomial dynamic programming procedure to separate violated change-cycle inequalities (5). The idea of this algorithm was originally proposed by Nachtigall [9]. He claimed a running time of $\mathcal{O}(T(mn + n^2)$, but did not give a proof. We prove a complexity of $\mathcal{O}(T^2 n^2 m)$.

Given a point $(y^*, z^*) \in P_{LP}(\mathrm{PTP})$, the separation problem is to find an oriented cycle γ in $\mathcal{N}$ such that the change-cycle inequality (5) induced by γ is violated, i.e., for $\alpha_0 = [-\gamma^t \ell]_T$ it holds

$$(T - \alpha_0)\gamma_+^t y^* + \alpha_0\, \gamma_-^t y^* < \alpha_0\,(T - \alpha_0),$$

or to conclude that no such cycle exists. The idea is to solve for each fixed $\alpha_0 \in \{0, \dots, T-1\}$ the problem

$$f^*(\alpha_0) = \min\{(T - \alpha_0)\gamma_+^t y^* + \alpha_0\, \gamma_-^t y^* | \gamma \text{ oriented cycle in } \mathcal{N}, \left[-\gamma^t \ell\right]_T = \alpha_0\}, \tag{9}$$

which is to find the minimum cost cycle w.r.t.

$$c_a = \begin{cases} (T - \alpha_0) y_a^* & \text{if } \gamma_a = 1 \\ \alpha_0 y_a^* & \text{if } \gamma_a = -1 \\ 0 & \text{else} \end{cases}$$

of all oriented cycles with $[-\gamma^t \ell]_T = \alpha_0$. Note that a violated change-cycle inequality exists if and only if for some $\alpha_0 \in \{0, \dots, T-1\}$ it holds $f^*(\alpha_0) < \alpha_0\,(T - \alpha_0)$.

The minimization problem (9), again, can be solved with a dynamic program that iterates over the cycle lengths w.r.t. the number of activities. We denote by a *chain* a path that can contain forward directed activities as well as backward directed activities. Let $\mathcal{C}_{ij}^k$ be the set of all chains in $\mathcal{N}$ from event $i \in V$ to event $j \in V$ that contain exactly k activities, given by their characteristic vectors. For $\alpha \in \{0, \dots, T-1\}$, let

$$f_{ij}^k(\alpha_0, \alpha) := \min \left\{ (T - \alpha_0) \sum_{\substack{a \in A: \\ p_a > 0}} y_a^* + \alpha_0 \sum_{\substack{a \in A: \\ p_a < 0}} y_a^* \middle| p \in \mathcal{C}_{ij}^k, \alpha = \left[-p^t \ell\right]_T \right\}$$

be the minimum length w.r.t. c_a of all chains in $\mathcal{C}_{ij}^k$ with $\alpha = [-p^t \ell]_T$. Since a chain of length $k \geq 2$ consists of a chain of length $k-1$ and an additional activity, the following recursive equation holds

$$f_{ij}^{k+1}(\alpha_0, \alpha) := \min \left\{ \min_{\substack{a=(u,j) \\ \left[\alpha' - \ell_a\right]_T = \alpha}} f_{iu}^k(\alpha_0, \alpha') + (T - \alpha_0) y_a^*, \min_{\substack{a=(j,u) \\ \left[\alpha' + \ell_a\right]_T = \alpha}} f_{iu}^k(\alpha_0, \alpha') + \alpha_0 y_a^* \right\}, \tag{10}$$

for all $k \geq 0$ with

$$f_{ij}^0(\alpha_0, \alpha) = \begin{cases} 0 & \text{if } i = j, \alpha = 0 \\ \infty & \text{else.} \end{cases}$$

Since every elementary cycle has at most n activities and $c_a \geq 0$ for all $a \in A$, the minimum length w.r.t. c of all oriented cycles γ with $\alpha_0 = [-\gamma^t \ell]_T$ is given by

$$f^*(\alpha_0) = \min_{i \in V} \min_{k=1}^{n} f_{ii}^k(\alpha_0, \alpha_0).$$

For fixed $k \in \{0, \dots, n-1\}$, the recursive equation (10) can be solved with Algorithm 1.

▶ **Theorem 9.** *For given $\alpha_0 \in \{0, \dots, T-1\}$, $k \in \{0, \dots, n-1\}$, and f_{ij}^k for all $i, j \in V$, Algorithm 1 computes $f_{ij}^{k+1}(\alpha_0, \alpha)$ for all $i, j \in V$ and $\alpha \in \{0, \dots, T-1\}$ in $\mathcal{O}(Tmn)$.*

Proof. For a given $k \in \{0, \dots, n-1\}$ and $\alpha_0 \in \{0, \dots, T-1\}$, Algorithm 1 obviously solves equation (10) for all $i, j \in V$ and for all $\alpha \in \{0, \dots, T-1\}$. The computation involves $\mathcal{O}(Tmn)$ elementary operations. ◀

The described separation algorithm is given in pseudocode in Algorithm 2.

▶ **Theorem 10.** *Algorithm 2 solves the separation problem for the change-cycle inequalities* (5) *in $\mathcal{O}(T^2 n^2 m)$.*

Proof. The Algorithm 2 solves for each $\alpha_0 \in \{0, \dots, T-1\}$ the minimization problem (9) and correctly reports if there exists an $\alpha_0 \in \{0, \dots, T-1\}$ such that $f^*(\alpha_0) < \alpha_0\,(T - \alpha_0)$. Hence, with the previous argumentation, the correctness of the algorithm follows. The algorithm needs to call Algorithm 1, see line 6 of Algorithm 2, in total kT-times. By Theorem 9, this results in a running time in $\mathcal{O}(T^2 n^2 m)$. ◀

Algorithm 1: Computing $f_{ij}^{k+1}(\alpha_0, \alpha)$ for all $\alpha \in \{0, \dots, T-1\}$

Input : $(y^*, z^*) \in P_{LP}(\text{PTP})$, $\alpha_0 \in \{0, \dots, T-1\}$, $k \in \{0, \dots, n-1\}$, $f_{ij}^k, \forall i, j \in V$

Output : $f_{ij}^{k+1}, \forall i, j \in V$

- **for** $a = (u, v) \in A$ **do**
  - **for** $\alpha' := 0, \dots, T-1$ **do**
    - **for** $i \in V$ **do**
      - $\alpha := [\alpha' - \ell_a]_T$
      - **if** $f_{iv}^{k+1}(\alpha_0, \alpha) > f_{iu}^k(\alpha_0, \alpha') + (T - \alpha_0)y_a^*$ **then**
        - $f_{iv}^{k+1}(\alpha_0, \alpha) := f_{iu}^k(\alpha_0, \alpha') + (T - \alpha_0)y_a^*$
      - **end**
      - $\alpha := [\alpha' + \ell_a]_T$
      - **if** $f_{iu}^{k+1}(\alpha_0, \alpha) > f_{iv}^k(\alpha_0, \alpha') + \alpha_0 y_a^*$ **then**
        - $f_{iu}^{k+1}(\alpha_0, \alpha) := f_{iv}^k(\alpha_0, \alpha') + \alpha_0 y_a^*$
      - **end**
    - **end**
  - **end**
- **end**
- **return** $f_{ij}^{k+1}, \forall i, j \in V$

Algorithm 2: Separation of Change-Cycle Inequalities

Input : LP-point $(y^*, z^*) \in P_{LP}(\text{PTP})$

Output : Cycle $\gamma \in \mathcal{N}$ such that the change-cycle inequality (5) is violated, or *NULL* if no such cycle exists.

- $f^* := \infty$
- $\rho^* := 0$
- **for** $\alpha_0 := 0, \dots, T-1$ **do**
  - $f_{ik}^k(\alpha_0, \alpha) := \begin{cases} 0 & \text{if } k = 0, i = j, \alpha = 0 \\ \infty & \text{else} \end{cases}$
  - **for** $k := 1, \dots, n$ **do**
    - compute $f_{ij}^k(\alpha_0, \alpha)$ for all $\alpha \in \{0, \dots, T-1\}$, for all $i, j \in V$
  - **end**
  - $f^*(\alpha_0) := \min_{i \in V} \min_{k=1}^n f_{ii}^k(\alpha_0, \alpha_0)$
  - **if** $f^*(\alpha_0) - \alpha_0(T - \alpha_0) < \rho^*$ **then**
    - $f^* := f^*(\alpha_0)$
    - $\rho^* := f^*(\alpha_0) - \alpha_0(T - \alpha_0)$
  - **end**
- **end**
- **if** $\rho^* < 0$ **then**
  - **return** $\gamma(f^*)$
- **else**
  - **return** *NULL*
- **end**

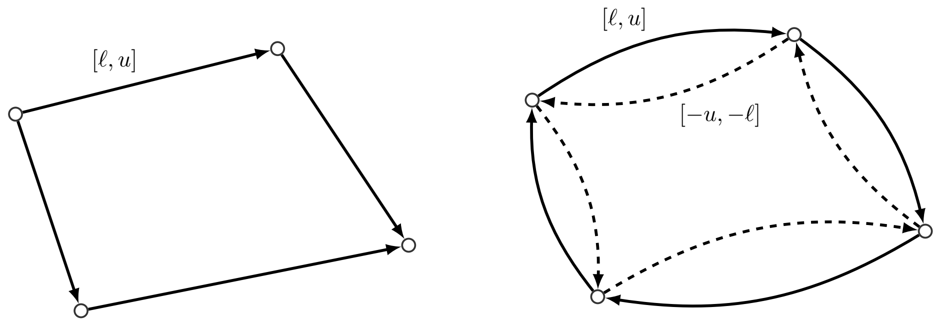

Figure 1 *Left*: the network $\mathcal{N}$, *Right*: the auxiliary network $\tilde{\mathcal{N}}$. The solid arcs correspond to the cycle γ and the circuit $\tilde{\gamma}$ from Theorem 11, respectively.

5 Separation of Cycle Inequalities

In this section we describe a pseudo-polynomial dynamic programming procedure to separate violated cycle inequalities (8) using an auxiliary network that was proposed by Nachtigall [9]. The auxiliary network allows to reduce cycle separation to finding certain modulo constrained circuits. Such a circuit can be found by a modification of the change-cycle separation Algorithm 2.

We obtain $\tilde{\mathcal{N}}$ by copying $\mathcal{N}$ and additionally introducing for each activity $a = (i, j)$ the back activity $\bar{a} = (j, i)$. The copies of the original activities keep their bounds, the bounds of the back activities are set to $\tilde{\ell}_{\bar{a}} = -u_a$ and $\tilde{u}_{\bar{a}} = -\ell_a$, see Figure 1. For a point $(y, z) \in P_{LP}(\text{PTP})$ we define $\tilde{y}$ on the activities of $\tilde{\mathcal{N}}$ by $\tilde{y}_a = y_a$ and $\tilde{y}_{\bar{a}} = u_a - \ell_a - y_a$.

▶ **Theorem 11.** *Let be $(y^*, z^*) \in P_{LP}(\text{PTP})$. Then, $\mathcal{N}$ contains a cycle γ that violates the y-cycle inequality (8), i.e.,*

$$\gamma^t y^* < \left[-\gamma_+^t \ell + \gamma_-^t u\right]_T + \gamma_-^t(\ell - u),$$

if and only if $\tilde{\mathcal{N}}$ contains a circuit $\tilde{\gamma}$ (cycle containing only forward directed activities) with

$$\tilde{\gamma}^t \tilde{y}^* < \left[-\tilde{\gamma}^t \tilde{\ell}\right]_T.$$

Proof. Let γ be a cycle in $\mathcal{N}$ and $\tilde{\gamma}$ be the circuit in $\tilde{\mathcal{N}}$ obtained by replacing all backward directed activities in γ with their auxiliary back activities, see Figure 1. Then we get

$$\begin{aligned}
& & \gamma^t y^* &< \left[-\gamma_+^t \ell + \gamma_-^t u\right]_T + \gamma_-^t(\ell - u) \\
\Leftrightarrow & & \gamma_+^t y^* + \gamma_-^t(u - \ell - y^*) &< \left[-\gamma_+^t \ell + \gamma_-^t u\right]_T \\
\Leftrightarrow & & \gamma_+^t \tilde{y}^* + \gamma_-^t \tilde{y}^* &< \left[-\gamma_+^t \ell - \gamma_-^t(-u)\right]_T \\
\Leftrightarrow & & \tilde{\gamma}^t \tilde{y}^* &< \left[-\tilde{\gamma}^t \tilde{\ell}\right]_T
\end{aligned}$$ ◀

By Theorem 11, there exists a violated cycle inequality (8) if and only if

$$\delta^* := \min\{\tilde{\gamma}^t \tilde{y}^* - \left[-\tilde{\gamma}^t \tilde{\ell}\right]_T | \tilde{\gamma} \text{ directed circuit in } \tilde{\mathcal{N}}\} < 0.$$

Hence, we can solve the separation problem by minimizing $\delta(\tilde{\gamma}) := \tilde{\gamma}^t \tilde{y}^* - \left[-\tilde{\gamma}^t \tilde{\ell}\right]_T$ over all directed circuits in the auxiliary network $\tilde{\mathcal{N}}$. We describe in the following the idea of the algorithm, which is given in pseudocode in Algorithm 3.

Let $\mathcal{P}_{ij}^k$ be the set of all (i,j)-paths in $\tilde{\mathcal{N}}$ that contain exactly k activities, given by their characteristic vectors. For $\alpha \in \{0,\dots,T-1\}$, let

$$d_{ij}^k(\alpha) := \min \left\{ \sum_{a \in A: p_a > 0} \tilde{y}_a^* \middle| p \in \mathcal{P}_{ij}^k, \alpha = \left[-p^t \tilde{\ell}\right]_T \right\} \tag{11}$$

be the minimum length with respect to $\tilde{y}^*$ of all paths in $\mathcal{P}_{ij}^k$ with $\alpha = \left[-p^t\tilde{\ell}\right]_T$. We can use the following recursive equation to compute (11)

$$d_{ij}^{k+1}(\alpha) := \min_{\substack{a=(u,j) \\ [\alpha' - \ell_a]_T = \alpha}} d_{iu}^k(\alpha') + \tilde{y}_a^*, \quad \forall\, k \geq 0$$

with

$$d_{ij}^0(\alpha) = \begin{cases} 0 & \text{if } i = j, \alpha = 0 \\ \infty & \text{else.} \end{cases}$$

Since every elementary circuit has at most n activities and $\tilde{y}_a^* \geq 0$ for all $a \in A$, the minimum length w.r.t. $\tilde{y}^*$ of all directed circuits $\tilde{\gamma}$ with $\alpha = \left[-\tilde{\gamma}^t\tilde{\ell}\right]_T$ is given by

$$d^*(\alpha) = \min_{i \in V} \min_{k=1}^{n} d_{ii}^k(\alpha)$$

and we have

$$\delta^* = \min\{d^*(\alpha) - \alpha | \alpha \in \{0,\dots,T-1\}\}.$$

▶ **Theorem 12.** *Algorithm 3 detects a violated cycle inequality in $\mathcal{O}(Tn^2m)$.*

Proof. The argumentation in this section proves that the algorithm computes δ^* and, thus, correctly detects violated cycle inequalities. The algorithm involves in total $\mathcal{O}(Tn^2m)$ elementary operations. ◀

6 Computational Results

This section gives some indication of the computational usefulness of cycle-separation compared to a heuristic separation.

As far as we know, the cycle inequalities (6) are added in cutting-plane algorithms only with heuristic separation algorithms, see [9, 3, 6]. In the so-called spanning tree heuristic, a minimum spanning tree of the event-activity network weighted with the slack values of the LP-solution is computed and the fundamental cycles of this tree are checked for violated inequalities.

We have implemented a full separation algorithm according to Algorithm 3 to separate all violated cycle inequalities (8) with a given maximum length. Such a length restriction is necessary to handle the memory consumption and the computation time of the separation algorithm. We tested a length restriction of 10, 15, and 20.

Our test set consists of seven instances, which are given in Table 1. The instance Wuppertal is based on the real multi-modal public transportation network of the city of Wuppertal for 2013. The remaining Wuppertal-instances are obtained by selecting a subset of lines of this instance. The Dutch instance is based on a network that was introduced

Algorithm 3: Separation of Cycle Inequalities

Input : LP-point $(y^*, z^*) \in P_{LP}(\text{PTP})$
Output : Cycle $\gamma \in \mathcal{N}$ such that the cycle inequality (8) is violated, or *NULL* if no such cycle exists.

$d^* := \infty$
$\delta^* = 0$
$d_{ij}^k(\alpha) := \begin{cases} 0 & \text{if } k = 0, i = j, \alpha = 0 \\ \infty & \text{else} \end{cases}$
for $k := 1, \dots, n$ **do**
  for $a = (u, v) \in \tilde{A}$ **do**
    for $\alpha' \in \{0, \dots, T-1\}$, $i \in V$ *with* $d_{iu}^k(\alpha') < \infty$ **do**
      $\alpha := [\alpha' - \tilde{\ell}_a]_T$
      if $d_{iv}^k(\alpha) > d_{iu}^{k-1}(\alpha') + \tilde{y}_a^*$ **then**
        $d_{iv}^k(\alpha) := d_{iu}^{k-1}(\alpha') + \tilde{y}_a^*$
        if $i = v$ *and* $d_{ii}^k(\alpha) - \alpha < \delta^*$ **then**
          $d^* := d_{ii}^k(\alpha)$
          $\delta^* := d_{ii}^k(\alpha) - \alpha$
        end
      end
    end
  end
end
if $\delta^* < 0$ **then**
  return $\gamma(d^*)$
else
  return *NULL*
end

Table 1 Statistics on the test instances. The columns list the instance name, the number of stations and lines of the transportation network, the number of events and activities of the event-activity network, the number of slack variables, periodic offset variables, and constraints in the original problem, and the number of variables and constraints after preprocessing.

name	$\lvert\mathcal{S}\rvert$	$\lvert\mathcal{L}\rvert$	n	m	$\#y$	$\#z$	#cons	#vars*	#cons*
Wuppertal 14	28	14	168	499	52	39	39	52	39
Wuppertal 44	64	44	395	1 426	122	85	85	106	77
Wuppertal 98	123	98	1 242	8 997	1 299	1 208	1 208	1 294	1 204
Wuppertal core	148	154	1 677	14 446	2 048	1 903	1 903	2 044	1 902
Wuppertal	1 582	311	13 202	79 251	3 188	2 886	2 886	3 150	2 862
Dutch	23	58	419	3 138	115	70	70	111	70
Potsdam	320	164	8 092	99 103	1 413	1 262	1 262	1 400	1 255

Table 2 Statistics on the computations. The columns list the instance name, the used cut separation, the solving time, the separation time, the number of separated cycle cuts, the number of cycle cuts selected by SCIP to be applied to the LP, the dual bound of the root node, the dual bound after termination, the best known primal bound, and the primal-dual gap in %.

name	method	solving time	sepa. time	cuts	applied cuts	root dual	dual	primal	gap in %
	no add. cuts	0.06s	-	-	-	16 231.80	24 074.55	24 074.55	0.00
	heuristic	0.04s	0.00s	2	2	16 499.35	24 074.55	24 074.55	0.00
Wuppertal 14	length $\leq$ 10	0.10s	0.03s	28	9	23 050.60	24 074.55	24 074.55	0.00
	length $\leq$ 15	0.18s	0.12s	84	16	23 088.89	24 074.55	24 074.55	0.00
	length $\leq$ 20	0.28s	0.22s	129	18	20 775.85	24 074.55	24 074.55	0.00
	no add. cuts	0.10s	-	-	-	28 778.74	37 755.40	37 755.40	0.00
Wuppertal 44	heuristic	0.11s	0.00s	1	1	28 669.75	37 755.40	37 755.40	0.00
	length $\leq$ 10	0.19s	0.05s	18	5	31 846.58	37 755.40	37 755.40	0.00
	length $\leq$ 15	0.46s	0.29s	40	9	31 953.26	37 755.40	37 755.40	0.00
	length $\leq$ 20	1.05s	0.80s	72	10	31 953.26	37 755.40	37 755.40	0.00
	no add. cuts	1h	-	-	-	81 940.30	112 023.51	477 161.17	325.95
Wuppertal 98	heuristic	1h	0.02s	20	20	89 284.15	124 697.64	468 467.85	275.68
	length $\leq$ 10	1h	1.16s	747	354	128 857.01	161 485.01	477 161.17	195.48
	length $\leq$ 15	1h	11.52s	2 413	887	149 847.94	173 291.01	477 161.17	175.35
	length $\leq$ 20	1h	41.34s	3 644	1 128	155 819.69	180 986.98	477 161.17	163.64
	no add. cuts	1h	-	-	-	98 654.95	118 462.71	464 533.25	292.13
Wuppertal core	heuristic	1h	0.02s	24	22	99 896.39	117 042.04	464 533.25	296.89
	length $\leq$ 10	1h	2.40s	949	448	137 337.00	155 433.06	464 533.25	198.86
	length $\leq$ 15	1h	25.90s	3 211	1 186	167 898.07	187 486.25	464 533.25	147.77
	length $\leq$ 20	1h	82.94s	4 886	1 488	175 122.92	184 265.70	464 533.25	153.09
	no add. cuts	1h	-	-	-	190 989.51	235 669.35	997 285.99	323.17
Wuppertal	heuristic	1h	0.08s	65	63	198 269.80	248 616.97	997 285.99	301.13
	length $\leq$ 10	1h	2.10s	1 082	402	232 178.52	273 620.80	997 285.99	264.48
	length $\leq$ 15	1h	21.55s	3 336	810	244 127.40	281 855.42	997 285.99	253.83
	length $\leq$ 20	1h	123.19s	5 307	1 098	255 288.10	290 249.68	997 285.99	243.60
	no add. cuts	7.06s	-	-	-	2 455.13	6 155.00	6 155.00	0.00
Dutch	heuristic	7.14s	0.00s	0	0	2 455.13	6 155.00	6 155.00	0.00
	length $\leq$ 10	7.99s	0.01s	0	0	2 455.13	6 155.00	6 155.00	0.00
	length $\leq$ 15	8.26s	0.04s	0	0	2 455.13	6 155.00	6 155.00	0.00
	length $\leq$ 20	8.24s	0.08s	0	0	2 455.13	6 155.00	6 155.00	0.00
	no add. cuts	1h	-	-	-	25 797.07	43 944.09	130 840.00	197.74
Potsdam	heuristic	1h	0.03s	10	10	28 407.66	46 545.79	130 840.00	181.10
	length $\leq$ 10	1h	0.34s	26	10	26 231.44	46 671.69	130 840.00	180.34
	length $\leq$ 15	1h	1.82s	106	33	27 115.22	45 784.24	130 840.00	185.76
	length $\leq$ 20	1h	8.04s	254	86	34 422.07	51 912.86	130 840.00	152.04

by Bussieck in the context of line planning [1]. The Potsdam instance is based on the real multi-modal public transportation network for 1998. We consider a period time of 20 for all instances. The activity weights are obtained by computing an uncapacitated multi-commodity flow in the event-activity network for a given passenger demand.

Our code is based on the constraint integer programming framework SCIP version 3.2.0 using Cplex 12.6.3 as an LP-solver. All computations were done on an Intel(R) Xeon(R) CPU E3-1245, 3.4 GHz computer (in 64 bit mode) with 8 MB cache, running Linux and 32 GB of memory. We set the time limit to one hour.

We compare the performance of the general MIP separators implemented in SCIP (*no add. cuts*), adding either the spanning-tree heuristic (*heuristic*) or our separation algorithm with a given length restriction (*length* ≤ 10, *length* ≤ 15, and *length* ≤ 20). The additional separators are only called at the root node. The results are listed in Table 2.

Looking at the root dual bound, one can see significant improvements, e.g., of up to 90% for Wuppertal 98, in comparison to the strategy without cycle cuts, and almost 75% over heuristic cycle cut separation. Hence, the separation algorithm has a greater effect on the dual bound than the heuristic, even though the separator only considers cycles of a restricted length. Only Wuppertal 14 has a smaller root dual bound if all cycles of maximum length 20 are separated compared to a cycle length of 10 or 15. This is not caused by the cycle inequalities, but by the additional "flow cover" and "strong cg" inequalities (heuristically) found by the default separator of SCIP. The given length restriction influences the performance of the separation algorithm: Separating cycle inequalities with higher length increases the computation time, but also, in general, the dual bound, especially for larger instances. In particular, the root dual bound for Potsdam can be further improved by 30% by using a length restriction of 20 compared to a length restriction of 10. Potsdam features the largest number of events, see Table 1, and benefits from a consideration of longer cycles.

References

1 Michael Bussieck. Gams – lop.gms: Line optimization. `http://www.gams.com/modlib/libhtml/lop.htm`.

2 M. Kümmling, P. Großmann, K. Nachtigall, J. Opitz, and R. Weiß. A state-of-the-art realization of cyclic railway timetable computation. *Public Transport*, 7(3):281–293, 2015. `doi:10.1007/s12469-015-0108-5`.

3 Christian Liebchen. *Periodic timetable optimization in public transport.* PhD thesis, Technische Universtität Berlin, 2006.

4 Christian Liebchen and Rolf H. Möhring. The modeling power of the periodic event scheduling problem: Railway timetables – and beyond. In Frank Geraets, Leo Kroon, Anita Schoebel, Dorothea Wagner, and Christos D. Zaroliagis, editors, *Algorithmic Methods for Railway Optimization*, volume 4359 of *Lecture Notes in Computer Science*, pages 3–40. Springer Berlin Heidelberg, 2007.

5 Christian Liebchen and Leon Peeters. Integral cycle bases for cyclic timetabling. *Discrete Optimization*, 6:98–109, 2009.

6 Christian Liebchen and Elmar Swarat. The Second Chvatal Closure Can Yield Better Railway Timetables. In Matteo Fischetti and Peter Widmayer, editors, *8th Workshop on Algorithmic Approaches for Transportation Modeling, Optimization, and Systems (ATMOS'08)*, volume 9 of *OpenAccess Series in Informatics (OASIcs)*, Dagstuhl, Germany, 2008. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik.

7 Thomas Lindner. *Train Schedule Optimization in Public Rail Transport.* PhD thesis, Technische Universtität Braunschweig, 2000.

8 K. Nachtigall and J. Opitz. Solving periodic timetable optimisation problems by modulo simplex calculations. In Matteo Fischetti and Peter Widmayer, editors, *ATMOS'08*, volume 9, 2008.

9 Karl Nachtigall. *Periodic Network Optimization and Fixed Interval Timetables.* Habilitation thesis, Universtität Hildesheim, 1998.

10 Michiel A. Odijk. Construction of periodic timetables, part 1: A cutting plane algorithm. Technical Report 94-61, TU Delft, 1994.

11 Leon W. P. Peeters. *Cyclic Railway and Timetable Optimization.* PhD thesis, Erasmus Universiteit Rotterdam, 2003.

12 Alexander Schrijver. Routing and timetabling by topological search. *Documenta Mathematica*, Extra Volume ICM 1998:1–9, 1998.

13 P. Sels, T. Dewilde, D. Cattrysse, and P. Vansteenwegen. Reducing the passenger travel time in practice by the automated construction of a robust railway timetable. *Transportation Research Part B: Methodological*, 84:124–156, 2016. doi:10.1016/j.trb.2015.12.007.

14 Paolo Serafini and Walter Ukovich. A mathematical model for periodic scheduling problems. *SIAM Journal on Discrete Mathematics*, 2(4):550–581, 1989.