Computers & Operations Research 40 (2013) 1363–1370

Contents lists available at SciVerse ScienceDirect

Computers & Operations Research

journal homepage: www.elsevier.com/locate/caor

Improving the modulo simplex algorithm for large-scale periodic timetabling ☆

Marc Goerigk *, Anita Schöbel

Institut für Numerische und Angewandte Mathematik, Georg-August-Universität Göttingen, Germany

ARTICLE INFO

Available online 8 September 2012

Keywords:
Periodic event scheduling problem
Periodic timetabling
Combinatorial optimization
Large-scale optimization

ABSTRACT

The *periodic event scheduling problem* (PESP), in which events have to be scheduled repeatedly over a given period, is a complex and well-known discrete problem with numerous real-world applications. The most prominent of them is to find periodic timetables in public transport. Although even finding a feasible solution to the PESP is NP-hard, recent achievements demonstrate the applicability and practicability of the periodic event scheduling model. In this paper we propose different approaches to improve the *modulo network simplex algorithm* (Nachtigall and Opitz, 2008 [17]), which is a powerful heuristic for the PESP problem, by exploiting improved search methods in the modulo simplex tableau and larger classes of cuts to escape from the many local optima. Numerical experiments on large-scale railway instances show that our algorithms not only perform better than the original method, but even outperform a state-of-the-art commercial MIP solver.

1. Introduction

The *periodic event scheduling problem* (PESP) as introduced in [20] models periodically reoccurring events that have to be scheduled according to given feasible time spans. Examples for periodic scheduling involve problems from production, control, or link scheduling [2], but most importantly its general modeling power made it the model of choice for the computation of periodic timetables in public transport, see e.g., [13,18,16,11,19,1]. Recently, also connections to Graphical Diophantine Equations have been explored [3] in the case of multiple periods.

The applicability of the model to real-world problems has been impressively demonstrated by two recent milestones. In 2005, the new timetable for the underground railway of Berlin was introduced [12], being the first mathematically optimized railway timetable in practice. And in 2006, the largest Dutch railway company, the *Nederlandse Spoorwegen*, introduced a completely new timetable, with an estimated profit of 40 million Euro annually [8]. However, while the former work considered instances of comparatively small size, the latter focused on finding feasible solutions and only applied postoptimization methods.

The most common approach to solving PESPs is by mixed-integer programming techniques [15], using integral (minimum) cycle bases [14]. However, these approaches suffer from high computation times. In [17] a heuristic approach, the *modulo network simplex method*, is presented, which is based on the classic network simplex method. To the best of our knowledge, this heuristic is currently the most powerful method to solve large instances.

The focus of this paper is to close the currently existing gap in large-scale timetable optimization. To do so, we improve the modulo network simplex method's performance for practical timetabling instances, enabling us to compute solutions with both smaller runtimes and better objective values than both the original method as well as the commercial MIP solver Gurobi [7]. Specifically, our approach can on average improve the objective value of a starting solution about 80% faster than the original method, and about 90% faster than Gurobi, and improve about 7% more than the original method when no timelimit is imposed.

Overview: We introduce the periodic event scheduling problem in Section 2. In Section 3, the modulo simplex algorithm is described, and extended in Section 4. We present the application of PESP to periodic timetabling and an experimental evaluation in Section 6, and conclude this work in Section 7.

2. The periodic event scheduling problem

The *periodic event scheduling problem* (PESP) deals with a set of events, each of them being repeated whenever T time units have passed, and assigns a time to any of these events. Formally, we need the following notations, see [20].

☆ Partially supported by Grant SCHO 1140/3-1 within the DFG programme *Algorithm Engineering*.
* Corresponding author.
E-mail address: m.goerigk@math.uni-goettingen.de (M. Goerigk).

0305-0548/$ - see front matter © 2012 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.cor.2012.08.018

Let a *period* $T \in \mathbb{N}$ be given. A *periodic event* i is a countably infinite set of events i_p, $p \in \mathbb{Z}$, with occurrence times

$$t(i_p) = t(i) + p \cdot T.$$

The set of periodic events is denoted as $\mathcal{E}$. These events may be linked by activities $\mathcal{A} \subseteq \mathcal{E} \times \mathcal{E}$. For any activity $a=(i,j) \in \mathcal{A}$, a *span constraint* is given. It consists of an interval $[l_{ij}, u_{ij}] \subset \mathbb{R}$ and is satisfied if

$$(t(j)-t(i)) \bmod T \in [l_{ij}, u_{ij}],$$

i.e., if the periodic time difference between two events lies within the given interval. The interval bounds l_{ij} and u_{ij} represent the minimum and maximum duration of activity (i,j), respectively. The meaning of these bounds depends on the actual problem under consideration, see Section 5 as an example.

The PESP can now be stated as follows: For a given finite set $\mathcal{E}$ of events with a period T and a finite set of span constraints $\mathcal{A}$, find a time $t(i)$ for each periodic event i such that all span constraints are satisfied. It is shown [20] that PESP is NP-hard by transformation from the Hamiltonian Circuit Problem. Given a solution $t(i), i \in \mathcal{E}$ for PESP, the *duration* of an activity $a=(i,j) \in \mathcal{A}$ is given as $(t(j)-t(i)) \bmod T$.

The PESP can be further extended using a linear objective function on the activity durations. Then, we do not only search for a feasible solution, but instead for an optimal one. Let $\pi_i := t(i) \bmod T \in \mathbb{R}$ be the time assigned to the events $i \in \mathcal{E}$ for a given period T such that the span constraints are satisfied, i.e., $(\pi_j - \pi_i) \bmod T \in [l_{ij}, u_{ij}]$ for each activity $(i,j) \in \mathcal{A}$. For some given activity weights w_{ij}, we would like to minimize

$$\sum_{(i,j) \in \mathcal{A}} w_{ij}((\pi_j - \pi_i) \bmod T) - l_{ij}.$$

The PESP can be interpreted as a graph-theoretical problem by using the events $\mathcal{E}$ as nodes and the activities as edges between them, see Fig. 1. The resulting network $G=(\mathcal{E},\mathcal{A})$ is called *event-activity network*.

Example 1. The four events v_1, v_2, v_3, v_4 need to be scheduled within a period of $T=10$, fulfilling the six constraints as given in Fig. 1. A feasible solution is given by $\pi_1 = 0$, $\pi_2 = 5$, $\pi_3 = 0$ and $\pi_4 = 5$.

Instead of the event times $\pi_i, i \in \mathcal{E}$, one can equivalently determine the *slack* $y_{ij} = \pi_j - \pi_i - l_{ij}$ for any edge $(i,j) \in \mathcal{A}$ with lower bound l_{ij}. Generally speaking, the slack of an activity is the amount of time spent additionally to its minimum duration. Using this concept, an alternative formulation (used by the modulo network simplex) has been suggested in [16]. Let $\mathcal{T} = (\mathcal{E}, \mathcal{A}_{\mathcal{T}})$ be a spanning tree with its corresponding fundamental cycle matrix Γ, then the periodic event scheduling problem can be formulated as follows:

$$\text{(PESP)} \quad \min \sum_{(i,j) \in \mathcal{A}} \omega_{ij} y_{ij}$$
$$\text{s.t} \quad \Gamma(y+l) = Tz \quad 0 \le y_{ij} \le u_{ij} - l_{ij} \quad \forall (i,j) \in \mathcal{A} \quad y_{ij} \in \mathbb{R}$$
$$\forall (i,j) \in \mathcal{A} \quad z_{ij} \in \mathbb{Z} \quad \forall (i,j) \in \mathcal{A} \setminus \mathcal{A}_{\mathcal{T}}$$

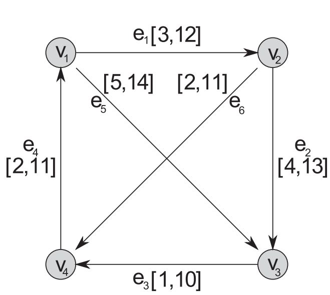

Fig. 1. An example PESP.

where $y=(y_{ij})_{(i,j) \in \mathcal{A}}$ and $l=(l_{ij})_{(i,j) \in \mathcal{A}}$. For details and correctness we refer to [16,11]. As the variables z_{ij} model the periodic character of the problem, they will be referred to as *modulo parameters*.

Note that the modulo parameters are the reason why this problem is NP-hard. For fixed variables z_{ij} the scheduling problem is called aperiodic and is the dual of a minimum cost flow problem that can be solved efficiently using the classical network simplex method.

3. The modulo network simplex method

In this section we briefly describe the method of [17]. Its main idea is to encode a solution as a spanning tree $\mathcal{T}_l \cup \mathcal{T}_u$ by setting the modulo parameters of the tree edges to 0 and the duration of these activities either to their respective lower or upper bound.

Definition 2 (*Nachtigall and Opitz [17]*). A *spanning tree structure* $(\mathcal{T}_l, \mathcal{T}_u)$ is a spanning tree $\mathcal{T} = \mathcal{T}_l \cup \mathcal{T}_u$ with an edge partition such that y_{ij} is set to 0 on all edges $(i,j) \in \mathcal{T}_l$ and set to $u_{ij} - l_{ij}$ for all edges $(i,j) \in \mathcal{T}_u$.

A spanning tree structure uniquely determines a periodic schedule by calculating the slack y_{ij} for the missing edges $(i,j) \notin \mathcal{T}$ such that the cycle condition $\Gamma(y+l) = Tz$ of (PESP) holds. On the other hand, it is shown in [16] that

$$\begin{pmatrix} \pi \\ z \end{pmatrix} \in \mathcal{Q} := \text{conv.hull}\left(\left\{ \begin{pmatrix} \pi \\ z \end{pmatrix} \middle| \, l_{ij} \le \pi_j - \pi_i + Tz_{ij} \le u_{ij}; z \in \mathbb{Z}^m; \pi \in \mathbb{R}^n \right\} \right)$$

is an extreme point of $\mathcal{Q}$ if and only if it is a solution that is given by a spanning tree structure. Thus it is sufficient to investigate only these solutions.

The modulo network simplex works as follows: as it is the case in the classic network simplex method, a given feasible spanning tree solution is gradually improved by exchanging tree and non-tree edges that lie in the same fundamental cycle, i.e., the cycle that consists of the non-tree edge and its unique path in the spanning tree. This is done with the help of a simplex-like tableau.

Example 3. We reconsider Example 1. In Fig. 2(a) the problem instance is given with period $T=10$, and in Fig. 2(b) a feasible spanning tree structure, where $\mathcal{T} = \mathcal{T}_l$. The corresponding modulo simplex tableau can be seen in Table 1. It contains the fundamental cycles that are induced by the non-tree arcs e_4, e_5 and e_6.

The objective value $w^t y$ is calculated by $\sum_{(i,j) \notin \mathcal{T}} w_{ij} y_{ij} + \sum_{(i,j) \in \mathcal{T}_u} w_{ij}(u_{ij} - l_{ij})$. Let y^{ij} be the slack vector after pivoting edges e_i and e_j. By writing $[y]_T := y \bmod T$ for short and denoting by b_{ij} the tableau entry for the edges e_i and e_j, the change in the objective value when pivoting a non-tree edge e_i and a tree edge e_j to $\mathcal{T}_l$ is

$$\begin{aligned}
\omega^t y^{ij} - \omega^t y &= \sum_{k \in \mathcal{A} \setminus (\mathcal{T} \cup \{i\})} \omega_k \left[y_k - \frac{b_{kj}}{b_{ij}} y_i \right]_T + \omega_j \left[\frac{y_i}{b_{ij}} + y_j \right]_T \\
&\quad + \sum_{k \in \mathcal{T}_u} \omega_k y_k - \sum_{k \in \mathcal{A}} \omega_k y_k \\
&= \sum_{k \in \mathcal{A} \setminus (\mathcal{T} \cup \{i\})} \omega_k \left(\left[y_k - \frac{b_{kj}}{b_{ij}} y_i \right]_T - y_k \right) \\
&\quad + \omega_j \left(\left[\frac{y_i}{b_{ij}} + y_j \right]_T - y_j \right) - \omega_i y_i,
\end{aligned}$$

while the change when pivoting to $\mathcal{T}_u$ is

$$\begin{aligned}
\Delta\omega_{ij} = \omega_{ij} - \omega &= \sum_{k \in \mathcal{A} \setminus (\mathcal{T} \cup \{i\})} \omega_k \left(\left[y_k - \frac{b_{kj}}{b_{ij}} (y_i - u_i + l_i) \right]_T - y_k \right) \\
&\quad + \omega_j \left(\left[\frac{y_i - u_i + l_i}{b_{ij}} + y_j \right]_T - y_j \right) - \omega_i (y_i - u_i + l_i).
\end{aligned}$$