Staircase Compatibility and its Applications in Scheduling and Piecewise Linearization

Andreas Bärmann[1], Thorsten Gellermann[2],
Maximilian Merkert[3] and Oskar Schneider[4]

[1]Andreas.Baermann@math.uni-erlangen.de
[2]Thorsten.Gellermann@math.uni-erlangen.de
[3]Maximilian.Merkert@math.uni-erlangen.de
[4]Oskar.Schneider@fau.de

6th September 2016

Lehrstuhl für Wirtschaftsmathematik
Department Mathematik
Friedrich-Alexander-Universität Erlangen-Nürnberg
Cauerstraße 11, 91058 Erlangen, Germany

Abstract

We consider the clique problem with multiple-choice constraints (CPMC) and characterize a case where it is possible to give an efficient description of the convex hull of its feasible solutions. This case, which we call *staircase compatibility*, generalizes common properties in applications and allows for a linear description of the integer feasible solutions to (CPMC) with a totally unimodular constraint matrix of polynomial size. We derive two such totally unimodular reformulations for the problem: one that is obtained by a strengthening of the compatibility constraints and one that is based on a representation as a dual network flow problem. Furthermore, we show a natural way to derive integral solutions from fractional solutions to the problem by determining integral extreme points generating this fractional solution. We also evaluate our reformulations from a computational point of view by applying them to two different real-world applications. The first one is a problem in railway timetabling where we try to adapt a given timetable slightly such that energy costs from operating the trains are reduced. The second one is the piecewise linearization of non-linear flow problems on a gas network. In both cases, we are able to reduce the solution times significantly by passing to the theoretically stronger formulations of the problem.

Keywords: Clique Problem, Multiple-Choice Constraints, Total Unimodularity, Compatibility Graph, Scheduling, Piecewise Linearization

Mathematics Subject Classification: 90C27 - 90C57 - 90C35 - 90C90

1 Introduction

Compatibility structures are prevalent in many combinatorial optimization problems. In fact, they arise whenever the choice of one solution element implies the choice of other elements – in the sense of an inequality of the form

$$x \leq \sum_{i=1}^{n} y_i,$$

with binary variables x and y_i for $i = 1, \ldots, n \in \mathbb{N}$. Such compatibilities are the core of the clique problem, which consists in finding a clique of a certain size in a given undirected graph.

In this work, we consider the combination of compatibility constraints together with another frequently-ocurring structure, namely so-called multiple-choice constraints:

$$\sum_{i=1}^{n} x_i = 1,$$

where the x_i are binary variables for $i = 1, \ldots, n \in \mathbb{N}$. These constraints are present whenever there is a partition of the set of eligible elements into subsets, such that it is required to choose exactly one element from each subset.

Altogether, this leads to a problem that can be classified as a clique problem with multiple-choice constraints (CPMC). While this problem is obviously NP-hard in general, we want to investigate here a special case where it is solvable in polynomial time. This is possible for a restriction of the compatibility graphs to graphs with a certain compatibility structure which we call *staircase compatibility*. For clique problems exhibiting this special structure, we will be able to state efficient binary programming (BP) formulations of which we can show that the corresponding constraint matrix is totally unimodular.

In order to demonstrate that there is great benefit from studying this structure, we present two very distinct real-world applications which are special cases of (CPMC) under staircase compatibility. There first one is a problem in railway timetabling which is a special case of the project scheduling problem. The second application arises in the context of piecewise-linear approximation of nonlinear functions in gas routing. In both cases, the resulting model reformulations are already known (see Möhring et al. (2001) and Correa-Posada and Sánchez-Martín (2014) respectively). However, our notion of staircase compatibility provides a common, more general framework to study the underlying clique problem with multiple-choice constraints. In particular, we are able to show that the derived integrality results hold for a wider class of compatibility graphs.

We begin with the definition of staircase compatibility in Section 2, which is accompanied by a first discussion of its presence in project scheduling problems and flow problems with piecewise-linear objective function as well as similar structures in the literature. Following that, we derive two totally unimodular BP formulations for the clique problem with multiple-choice constraints in the case of staircase compatibility in Section 3. The second of these two reformulations takes the form of a dual network flow problem. It will give rise to a very natural way of generating heuristic solutions from fractional solutions to the problem by determining the integral extreme points which generate this fractional solution. In Section 4, we present our computational

results for the two application mentioned above, showing that the better understanding of their structure directly translates into vastly shorter solution times. Finally, in Section 5, we summarize the findings of this paper and give possible directions for an extension of our results.

2 Staircase Compatibility

In the following, we define the clique problem with multiple-choice constraints as it is considered here as well as the notion of staircase compatibility. We will see later that the presence of such staircase structures in the problem allows for an efficient description of the convex hull of its feasible solutions.

Definition 2.1 (The Clique Problem under Multiple-Choice Constraints)**.** *Consider a finite basic set S together with a partition $S = \bigcup_{i=1}^{m} S_i$ of S into m disjoint subsets $S_1, \dots, S_m$ as well as a with a symmetric relation*

$$R \subseteq (S \times S) \setminus \bigcup_{i=1}^{m} (S_i \times S_i).$$

Two elements $s \in S_i$, $t \in S_j$ are said to be compatible *if and only if $(s,t) \in R$ holds. The* clique problem under multiple-choice constraints *(CPMC) is then given by the task to*

$$\text{chose exactly one element from each subset } S_i, \tag{rule 1}$$

such that the selected elements are pairwise compatible.

The clique problem under multiple-choice constraints amounts to finding a clique in the undirected graph $G = (S, R)$ whose nodes are the elements of S and whose arcs connect exactly the pairwise compatible elements in S, such that exactly one element from each subset in the partition of S is chosen. We call G the *compatibility graph* associated with relation R.

In this work, we focus on a special case of (CPMC) with a certain "connectedness" structure in the underlying compatibility relation:

Definition 2.2 (Staircase Relations)**.** *Let each subset S_i in the partition of S be an ordered set according to a total order $<_i$, which allows us to denote the elements of S_i by $s_{i,1}, \dots, s_{i,n_i}$ with $n_i = |S_i|$. In the following, we omit the index i and simply write $<$ whenever no confusion is possible. We then call a symmetric relation R on S a* staircase relation *if two conditions hold. The first condition states the connectedness of the compatible choices for a given element:*

$$(a, b_{k_1}) \in R \wedge (a, b_{k_3}) \in R \;\Rightarrow\; (a, b_{k_2}) \in R, \tag{rule 2a}$$

whenever $a \in S_i, b_{k_1}, b_{k_2}, b_{k_3} \in S_j, b_{k_1} < b_{k_2} < b_{k_3}$. The second condition forces some kind of monotonic behavior of R:

$$(a_{l_1}, b_{k_2}) \in R \wedge (a_{l_2}, b_{k_1}) \in R \;\Rightarrow\; (a_{l_1}, b_{k_1}) \in R \wedge (a_{l_2}, b_{k_2}) \in R \tag{rule 2b}$$

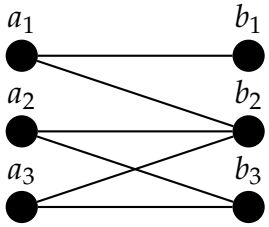
for $a_{l_1}, a_{l_2} \in S_i,\ b_{k_1}, b_{k_2} \in S_j,\ a_{l_1} < a_{l_2},\ b_{k_1} < b_{k_2}$.

The choice of the term "staircase relation" becomes clear when considering the adjacency matrix of the compatibility graph corresponding to such a relation: each submatrix that describes the compatibility between the elements of two subsets of the partition is a staircase matrix if its rows and columns are ordered according to the $<_i$ (see Fourer (1984) for an extensive compilation of the properties of staircase matrices).

Note that we assume in this article that each element of a subset S_i in the partition of S has as least one element in each of the remaining subsets with which it is compatible. Otherwise, this element may be eliminated as it cannot belong to a feasible selection. Note also that in this case (rule 2b) implies (rule 2a).

The above definitions prepare us to consider the problem of interest in this paper: if the relation R in the setting of Definition 2.1 is a staircase relation, we call the arising special case of (CPMC) the *clique problem with multiple-choice constraints under staircase compatibility* (CPMCS). Before we present interesting applications and a further discussion of this problem, we give an example which is taken up again in Section 3.

Example 2.3. *Consider the following example:*



$$\begin{array}{c} \\ a_1 \\ a_2 \\ a_3 \end{array} \begin{array}{c} \begin{array}{ccc} b_1 & b_2 & b_3 \end{array} \\ \begin{pmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \\ 0 & 1 & 1 \end{pmatrix} \end{array}$$

It shows the compatibility graph for (CPMCS) as well as the corresponding adjancency matrix for a certain staircase relation R on the set $S = S_1 \cup S_2$ with $S_1 = \{a_1, a_2, a_3\}$ and $S_2 = \{b_1, b_2, b_3\}$. We see that removing edge $\{a_2, b_2\}$ would destroy (rule 2a) *(and also* (rule 2b)*, see above), while removing $\{a_3, b_3\}$ would destroy* (rule 2b)*. Any selection $\{a, b\}$ with $(a, b) \in R$ would be feasible for (CPMCS).*

2.1 Two Applications of (CPMCS)

In the following, we give two example applications (CPMCS) may arise from. In both examples, the project scheduling problem and interval compatibilities in path flows, it is a possible way to characterize the set of feasible solutions.

Project Scheduling Let m tasks $j = j_1, \ldots, j_m$ be given. Each task has to be carried out at exactly one time slot, where we assume a discrete set $T_j = \{t_{j,1}, \ldots, t_{j,n_j}\}$ of possible execution times to be given that may differ for different jobs. Additionally, pairs of tasks may have precedence restrictions requiring one of them to start in a predefined time window relative to the other (if no relation is given, they may be done in any order, or possibly in parallel). This problem is called the *project scheduling problem with precedence constraints*. For further information and examples, see Schwindt and Zimmermann (2015) and the references therein.

The following is a possible formulation for the above scheduling problem:

$$\begin{aligned} \text{find} \quad & x & & & & & \text{(1a)} \\ \text{s.t.} \quad & x_k - x_l & \leq & \bar{d}_{k,l} & 1 \leq k < l \leq m & & \text{(1b)} \\ & x_k - x_l & \geq & \bar{d}_{k,l} & 1 \leq k < l \leq m & & \text{(1c)} \\ & x_j & \in & T_j & j = 1, \ldots, m & & \text{(1d)} \end{aligned}$$

for some $\bar{d}_{k,l}, \underline{d}_{k,l}$ with $k = 1, \ldots, m$ and $l = k+1, \ldots, m$.

We can model this problem as (CPMCS) as follows: each subset S_i represents a job j_i, where the elements in each subset are identified with the possible execution times $\{t_{j,1}, \ldots, t_{j,n_j}\}$. Consequently, the subsets come with an obvious chronological ordering. For different jobs j_k, j_l with $k \neq l$, we have

$$(t_{k,i_k}, t_{l,i_l}) \in R \Leftrightarrow \underline{d}_{k,l} \leq t_{k,i_k} - t_{l,i_l} \leq \bar{d}_{k,l}$$

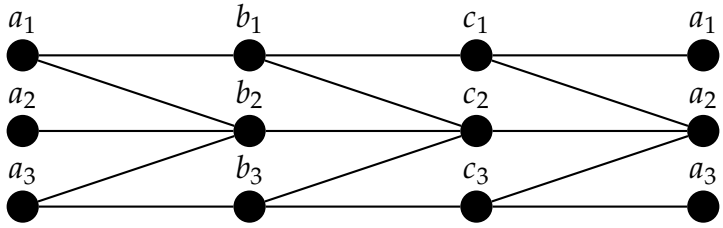
It is easily seen that (rule 2a) is satisfied due to the convexity of the relative time window defined by $\bar{d}_{k,l}$ and $\underline{d}_{k,l}$. Furthermore, violating (rule 2b) would contradict the temporal ordering. Therefore, R as defined here is a staircase relation.

Interval Compatibilities in Path Flows Let a path network consisting of m edges $e_1, \ldots, e_m$ be given. Each edge has an interval for the feasible flow on the edge which furthermore is subdivided into n_i subintervals, $i = 1, \ldots, m$. This scenario appears as a substructure in network flow problems where the flow has been piecewise linearized (see Section 4.2 for more details; cf. also Liers and Merkert (2015)). The task is to describe the set of feasible combinations of flow intervals.

It represents a special case of (CPMCS) as can be seen as follows: define S as the set of all intervals, where subset S_i includes all intervals belonging to edge i of the path. As those intervals are obtained from subdividing a larger interval, a canonical ordering is available. Intervals belonging to different (not necessarily adjacent) edges are *compatible* if and only if it is possible for the path flow to satisfy the bounds of both intervals. If the demand of all intermediate nodes of the path is zero, this is true if and only if they have nonempty intersection. Nonzero demands on path nodes can be reduced to this case by simple interval arithmetic which amounts to shifting intervals appropriately. An important observation is that the resulting relation R completely describes the problem, as a set of intervals is guaranteed to be compatible altogether if each pair of intervals is compatible (which is basically Helly's Theorem in dimension 1). Finally, R is a staircase relation, where (rule 2a) follows from the fact that intervals are convex, and (rule 2b) can easily be seen to hold from the way intervals can be sorted for each network edge.

Relation to General (CPMCS) The set of staircase relations that may originate from one of the two special cases of (CPMCS) forms a strict subclass of general staircase relations as defined in Definition 2.2. Intuitively, this is explained by the fact that most applications – including the two above – allow for some "transitivity reasoning", i.e. the compatibilities between subsets S_1 and S_2 together with those between S_2 and S_3 restrict the possible compatibilities between S_1 and S_3. However, according to the definition, both (rule 2a) and (rule 2b) only consider two subsets at a time. The following gives an example for a compatibility graph that does not originate from either of the two special cases mentioned above.

Example 2.4. *Consider the following compatibility graph G belonging to an instance of Problem (CPMCS) with three subsets $A = \{a_1, a_2, a_3\}$, $B = \{b_1, b_2, b_3\}$ and $C = \{c_1, c_2, c_3\}$, each of which has three elements. Note that there is another copy of partition A in the figure below to represent the compatibilities with partition C in order to highlight the symmetric structure of the example.*



Suppose G was obtained from an instance of Model (1)*. Then we could identify each partition with a job and each element of the partition with a possible execution time. We denote by $d_{A,B} := \bar{d}_{A,B} - \underline{d}_{A,B}$ the length of the time window between jobs A and B and similarly for the other relations. As $(a_2, b_2) \in R$, but $(a_2, b_1) \notin R$, $(a_2, b_3) \notin R$, we can conclude that $d_{A,B}$ is less than the time difference between b_1 and b_3, by slight abuse of notation denoted by $b_3 - b_1 > d_{A,B}$. Due to c_2 being connected to all nodes in B, the time window of length $d_{B,C}$ has to include b_1 as well as b_3 and hence $d_{B,C} \geq b_3 - b_1$, implying $d_{B,C} > d_{A,B}$. As the instance is symmetric, we can repeat this argument to obtain $d_{A,C} > d_{B,C}$ and $d_{A,B} > d_{A,C}$, which leads to the contradiction $d_{A,B} < d_{B,C} < d_{A,C} < d_{A,B}$.*

Similar reasoning shows that G also cannot be obtained from an instance of interval compatibilities on a path flow network (as it is described above): the argument is completely analogous, but uses the diameter of the intervals belonging to a_2, b_2, c_2 instead of $d_{A,B}, d_{B,C}, d_{A,C}$.

Moreover, the situation is no different if we do not assume the ordering of the elements within each partition to be given. This is because there is no other ordering that makes R a staircase relation apart from reversing all partition orderings.

3 Structural Properties

The problem (CPMCS) introduced in the previous section can be modelled as a mixed-integer program (MIP) in a straightforward fashion. We introduce a variable $x_s \in \{0, 1\}$ for each element $s \in S$ which takes a value of 1 if this element is chosen and 0 if not. A vector x is then a feasible selection if and only if it is a solution to the following feasibility problem:

$$
\begin{aligned}
&\text{find} \quad x \\
&\text{s.t.} \quad \sum_{s \in S_i} x_s = 1 \qquad (\forall S_i \in \mathcal{S}) && \text{(2a)} \\
&\qquad x_s \leq \sum_{\substack{t \in S_j: \\ (s,t) \in R}} x_t \qquad (\forall S_i \in \mathcal{S})(\forall s \in S_i)(\forall S_j \in \mathcal{S}, j > i) && \text{(2b)} \\
&\qquad x \in \{0,1\}^{|S|}, && \text{(2c)}
\end{aligned}
$$

where $\mathcal{S}$ denotes the given partition consisting of subsets $S_1, \ldots, S_m$, $m \in \mathbb{N}$. Multiple-Choice Constraints (2a) ensure that exactly one element of each subset in $\mathcal{S}$ is chosen, while Compatibility Constraints (2b) enforce the pairwise compatibility of the chosen

elements according to the relation R: choosing an element s from one subset S_i implies that we have to choose one of the elements compatible to s in each of the remaining subsets S_j. Integrality Constraints (2c) finally restrict variables x to take binary values. Note that Constraints (2b) for two subsets S_i, S_j are redundant if $(s,t) \in R$ for all $s \in S_i$ and $t \in S_j$.

Remark 3.1. *It is easy to find examples where Constraints* (2c) *are actually needed as Constraints* (2a) *and* (2b) *are not sufficient to ensure integrality of the solution. For the instance presented in Example 2.3, Model* (2) *reads:*

$$\begin{array}{lrcl} \text{find} & x & & \\ \text{s.t.} & x_1 + x_2 + x_3 & = & 1 \\ & x_4 + x_5 + x_6 & = & 1 \\ & x_1 & \leq & x_4 + x_5 \\ & x_2 & \leq & x_5 + x_6 \\ & x_3 & \leq & x_5 + x_6 \\ & x & \in & \{0,1\}^6. \end{array}$$

It allows for the fractional solution $(0, \frac{1}{2}, \frac{1}{2}, \frac{1}{2}, 0, \frac{1}{2})$ *if* $x \in \{0,1\}^6$ *is relaxed to* $x \geq 0$ *(observe that* $x \leq 1$ *is redundant). This solution is easily checked to be an extreme point of the corresponding polyhedron.*

Obviously, the polytope underlying Model (2) is not integral. However, we will see now that a small adaption leads to a totally unimodular description of the feasible set. We introduce the notation $\min(s, S_j, R)$ for some $s \in S_i \in \mathcal{S}$ and some S_j with $j \neq i$ to denote the minimal element in S_j which is compatible to s. Likewise, $\max(s, S_j, R)$ shall denote the maximal such element in S_j. Consider then the feasibility problem given by

$$\begin{array}{lrcll} \text{find} & x & & & \\ \text{s.t.} & \displaystyle\sum_{s \in S_i} x_s & = & 1 & (\forall S_i \in \mathcal{S}) \qquad (3a) \\ & \displaystyle\sum_{\substack{u \in S_i: \\ u \geq s}} x_u & \leq & \displaystyle\sum_{\substack{t \in S_j: \\ t \geq t_{\min(s,S_j)}}} x_t & (\forall S_i \in \mathcal{S})(\forall s \in S_i)(\forall S_j \in \mathcal{S}, j \neq i) \qquad (3b) \\ & x & \in & \{0,1\}^{|S|}. & \qquad (3c) \end{array}$$

It uses the same set of variables as Model (2) as well as the same multiple-choice constraints to enforce (rule 1). However, it features new compatibility constraints whose left-hand side arises by summing all x_u for $u \in S_i$ with $u > s$ onto the left-hand side of the old compatibility constraint corresponding to element $s \in S_i$ and some S_j with $j \neq i$. Its right-hand side arises by taking the old right-hand side and adding all variables x_t for $t \in S_j$ and $t > \max(s, S_j, R)$. Furthermore, this new model also incorporates compatibility constraints for subsets S_i and S_j with $i < j$. In the following, we show that the two models are in fact equivalent.

Proposition 3.2. *The respective feasible sets of Models* (2) *and* (3) *coincide.*

Proof. We begin by showing that each feasible solution to Model (2) is also feasible for Model (3). To see this, consider an element s of a subset S_i and its corresponding compatibility constraint (2b) with the elements of another subset S_j, which reads

$$x_s \leq \sum_{\substack{t \in S_j: \\ (s,t) \in R}} x_t.$$

By summing up these constraints for all elements $u \geq s$, we obtain

$$\sum_{\substack{u \in S_i: \\ u \geq s}} x_u \leq \sum_{\substack{u \in S_i: \\ u \geq s}} \sum_{\substack{t \in S_j: \\ (u,t) \in R}} x_t = \sum_{t \in S_j} |\{u \in S_i \mid u \geq s, (u,t) \in R\}| \cdot x_t,$$

using (rule 2b). Due to (rule 2a), all coefficients for variable x_t with $\min(s, S_j, R) \leq t \leq t_{j,n_j}$ on the right-hand side of this inequality are exactly those which are non-zero, i.e. 1 or greater. As its left-hand side can at most take a value of 1 due to the multiple-choice constraint for subset S_i, all coefficients on the right-hand side greater than 1 can be reduced to 1 without changing the set of integer solutions fulfilling the inequality. This exactly yields Compatibility Constraints (3b), which proves that the feasible set of Model (2) is included in that of Model (3).

To prove the opposite inclusion, consider the case of a feasible solution to Model (3) which violates some Compatibility Constraint (2b) of Model (2). That would mean, we have selected two elements $s \in S_i$ and $t \in S_j$ with $(s,t) \notin R$. In other words, either $t < \min(s, S_j, R)$ or $t > \max(s, S_j, R)$ holds due to (rule 2a). Assuming $i < j$, the first case can be ruled out, as a corresponding selection would violate Compatibility Constraint (3b) for element s and subset S_j. So let $t > \max(s, S_j, R)$. According to Compatibility Constraint (3b) for element t and subset S_i, we have $s > \min(t, S_i, R)$ and thus $s > \max(t, S_i, R)$ because of (rule 2a). Now, as $(s, \max(s, S_j, R)) \in R$ and $(t, \max(t, S_i, R)) \in R$, we find $(s,t) \in R$ according to (rule 2b), which is a contradiction. Consequently, we have shown that each feasible solution to Model (3) is feasible for Model (2). □

Note that similar as in the above proof, it can be shown that Compatibility Constraint (3b) for subsets S_i, S_j with $j < i$ is redundant to the corresponding constraint for $j > i$ if $\max(s, S_j, R) = t_{n_j,S_j}$.

Remark 3.3. *Continuing the discussion of Example 2.3, we consider Model* (3) *for the associated problem instance:*

$$\begin{aligned}
\text{find} \quad & x \\
\text{s.t.} \quad & x_1 + x_2 + x_3 &&= 1 \\
& x_4 + x_5 + x_6 &&= 1 \\
& x_1 + x_2 + x_3 &&\leq x_4 + x_5 + x_6 \\
& x_2 + x_3 &&\leq x_5 + x_6 \\
& x_3 &&\leq x_5 + x_6 \\
& x_4 + x_5 + x_6 &&\leq x_1 + x_2 + x_3 \\
& x_5 + x_6 &&\leq x_1 + x_2 + x_3 \\
& x_6 &&\leq x_2 + x_3 \\
& x &&\in \{0,1\}^6.
\end{aligned}$$

This feasibility problem no longer allows for the fractional solution $(0, \frac{1}{2}, \frac{1}{2}, \frac{1}{2}, 0, \frac{1}{2})$ *if relaxed to an LP. In fact, it can be checked that the corresponding polyhedron is integral.*

Generalizing the observation of Remark 3.3, we now show that the underlying polyhedron of Model (3) is always integral.

Theorem 3.4. *The constraint matrix of Model* (3) *is totally unimodular.*

Proof. In our proof, we use the following equivalent characterization of total unimodularity:

> A matrix A is totally unimodular, i.e. each square submatrix of A has determinant 0, $+1$ or -1, if and only if each collection of columns of A can be split into two parts, such that the sum of the columns in one part minus the sum of the columns in the other part is a vector with entries in $\{0, +1, -1\}$ only (see Ghouila-Houri (1962) and (Schrijver, 1998, Theorem 19.3 (iv), p. 269)).

We begin by showing the total unimodularity of the constraint matrix of Model (3) for the case of Example 2.3. We will then see that the idea behind the proof directly extends to the general case. Observe that the constraint matrix has a very special structure:

$$\left(\begin{array}{ccc|ccc||c|c|c|c} \multicolumn{3}{c|}{S_1} & \multicolumn{3}{c||}{S_2} & \sum_{\text{alt},S_1} & + & \sum_{\text{alt},S_2} & \sum \\ \hline 1 & 1 & 1 & 0 & 0 & 0 & 1 & + & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & + & 1 & 1 \\ 1 & 1 & 1 & -1 & -1 & -1 & 1 & + & -1 & 0 \\ 0 & 1 & 1 & 0 & -1 & -1 & 0 & + & 0 & 0 \\ 0 & 0 & 1 & 0 & -1 & -1 & 1 & + & 0 & 1 \\ -1 & -1 & -1 & 1 & 1 & 1 & -1 & + & 1 & 0 \\ -1 & -1 & -1 & 0 & 1 & 1 & -1 & + & 0 & -1 \\ 0 & -1 & -1 & 0 & 0 & 1 & 0 & + & 1 & 1 \end{array}\right),$$

where we have left out the submatrices I and $-I$ for the variable bounds, as they have no effect on total unimodularity. When computing the alternating sum of the columns corresponding to the elements of subset S_1, going backwards and starting with a positive sign in the last column, we observe that this yields a column vector that only consists of entries in $\{0, +1, -1\}$. The same holds for the columns corresponding to the elements of subset S_2. For the rows corresponding to Multiple-Choice Constraints (3a), exactly one of the two column vectors contains an entry $+1$ and the other one an entry 0. For the rows corresponding to Compatibility Constraint (3b) for the elements of S_1, the S_1-column vector contains either a $+1$ or a 0 and the S_2-column vector either a -1 or a 0, and vice versa for the elements of S_2. Thus, when adding the two column vectors, the result is a new column vector whose entries are in $\{0, -1, +1\}$ only. This property still holds when forming a submatrix by deleting individual columns of the constraint matrix due to the staircase structures in the compatibility constraint. Therefore, we have shown the total unimodularity of the matrix.

Now, when considering an arbitrary instance to Model (3), we can use the same strategy as above. Given an arbitrary subset of the columns of the constraint matrix, we partition it according to partition of S and compute the m column vectors arising when summing the columns in such a partition in a backwards fashion (exploiting

the ordering of the subsets S_i), starting with a positive sign for the last element. For the rows belonging to the multiple-choice constraints, exactly one resulting column vector will have an entry of 1, the other an entry of 0. As each row belonging to the compatibility constraint corresponds to the elements of exactly two subsets, at most one column vector will have an entry $+1$, and at most one column vector will have an entry -1. The other entries will be 0. As a result, when summing all the column vectors, the result will be a column vector with entries in $\{0, -1, +1\}$ only. This concludes the proof.

□

In many cases, totally unimodular constraint matrices correspond to problems defined on a network. More precisely, the matroid formed by a totally unimodular constraint matrix can be decomposed into matroids that are graphic, cographic, or isomorphic to the special matroid R_{10} (on the decomposition of regular matroids, see Seymour (1980)) – which is neither graphic nor cographic and rarely occurs in practical applications. Thus, it is natural to ask the question whether the constraint matrix of Model (3) is graphic or cographic (i.e., the linear matroid obtained from the matrix is a graphic or cographic matroid), in which case (CPMCS) is equivalent to a network flow problem or a dual network flow problem ("potential problem") respectively. The reader not familiar with those notions of matroid theory may consult Oxley (2006).

Theorem 3.5. *The constraint matrix of Model* (3) *is cographic.*

Proof. We show this by transforming Model (3) into a dual network flow problem. Given a graph $G = (V, A)$, such a problem is of the form

$$\begin{aligned} \min \quad & c^T \pi \\ \text{s.t.} \quad & \pi_j - \pi_i \leq d_{ij} \qquad (\forall\, a = (i,j) \in A) \\ & \pi \in \mathbb{R}^{|V|}. \end{aligned} \tag{4a}$$

To obtain this form, we use the following variable transformation: let

$$y_{i,j} \coloneqq \sum_{k=j}^{n_i} x_{i,k} \quad (\forall i = 1, \ldots, m)(\forall j = 1, \ldots, n_i).$$

Note that this transformation is bijective with $x_{i,j} = y_{i,j} - y_{i,j+1}$ if $j < n_i$, and $x_{i,n_i} = y_{i,n_i}$. Stating Model (3) in terms of the y-variables, we see that both sides form telescope sums, leaving only one variable on each side. Thus, Compatibility Constraint (3b) for two subsets S_i and S_l and some $j \in S_i$ now reads

$$y_{i,j} - y_{l,\min(j,S_l,R)} \leq 0,$$

which has the form of (4a). Constraints (rule 1) translate to

$$y_{i,1} = 1 \quad (\forall\, i = 1, \ldots, m). \tag{5}$$

This also implies upper bounds on the x-variables. Their lower bounds can be expressed via

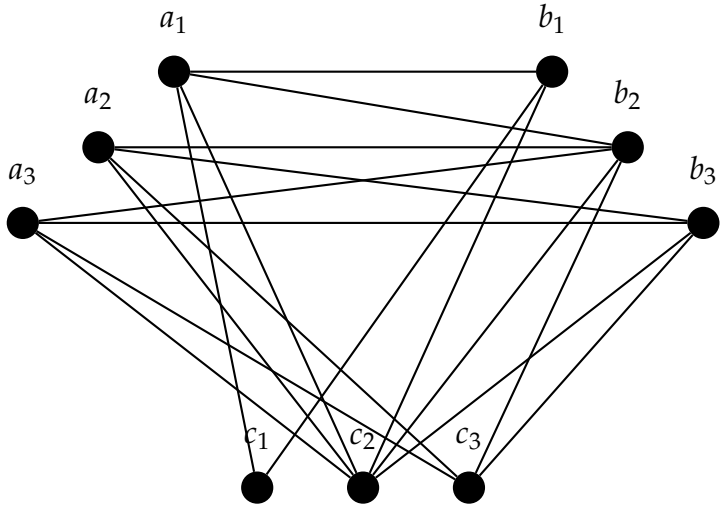
$$y_{i,j+1} - y_{i,j} \leq 0 \quad \text{if } j < n_i, \quad \text{and } -y_{i,n_i} \leq 0 \quad (\forall i = 1, \ldots, m). \tag{6}$$

G is simply defined to have a vertex for every y-variable and an arc (i, j) if and only if there is a constraint $y_j - y_i \leq 0$. □

Remark 3.6. *The y-variables have the following interpretation:* $y_{i,j} = 1$ *means: "from* S_i*, pick an element with index* j *or greater". This is very similar to the* Incremental Method *for linearizing a univariate function. Furthermore, the above transformation is well-known from this context (Vielma (2015)), where it is used to connect the Incremental Method to, for example, the* Convex Combination Method*, and vice versa. We can also recognize* (6) *as a* filling condition*.*

The following example illustrates the transformation of (CPMCS) to a dual network flow problem. It will also show that the constraint matrix is not graphic in general.

Example 3.7. *Let* S *be partitioned into three subsets* $A = \{a_1, a_2, a_3\}$*,* $B = \{b_1, b_2, b_3\}$ *and* $C = \{c_1, c_2, c_3\}$*. Let* R *be given by the following compatibility graph. Note that each pair of subsets behaves as in Example 2.3.*



As described in the proof of Theorem 3.5, Compatibility constraints (3b) *transform into Inequalities* (3)*, e.g. considering node* a_2 *together with subset* B*, the corresponding inequality*

$$x_{a_2} + x_{a_3} \leq x_{b_2} + x_{b_3}$$

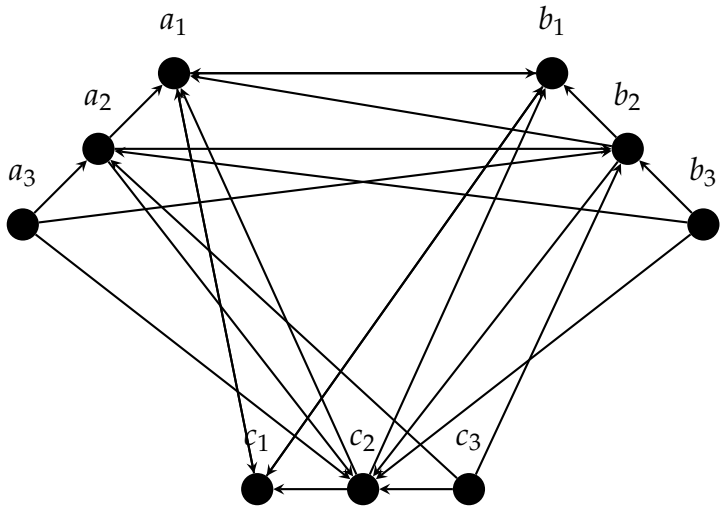
in terms of the y-variables now reads

$$y_{a_2} \leq y_{b_2}.$$

More generally, using the notation from the proof of Proposition 3.2, for every element $s \in S_i$ *and every subset* S_j*,* $j \neq i$ *we have*

$$y_s \leq y_{\min(s,S_j,R)}.$$

Due to (6)*, there are additional constraints ordering the y-variables within each subset. Therefore, by the proof of Theorem 3.5 we can formulate the given instance of (CPMCS) as a dual network flow problem on the following directed graph (arcs may be read as implications).*



The example shows that the constraint matrix of Model (3) *is not graphic in general, as this would require the above graph to be planar. However, this is not the case, as, for example, it has $K_{3,3}$ as a subgraph using nodes $\{a_2, b_2, c_2\}$ and $\{a_3, b_3, c_3\}$.*

As the constraint matrix is totally unimodular, we are guaranteed that each fractional point is the convex combination of integral solutions. Next, we will show how to find such a convex combination. In the case where (CPMCS) forms a substructure of a more complex problem, this may be useful for constructing a heuristic, as the integer points spanning a fractional solution are candidates for good feasible solutions.

The following is an easy way to obtain integral solutions from a fractional one. It generalizes the well-known fact that for dual network flow problems, rounding all components up or rounding all component down preserves feasibility.

Definition 3.8. *Let $\hat{y}$ be a solution to (CPMCS), and let $\lambda \in (0, 1]$ be some threshold value. We define $\hat{y}_\lambda$ to denote the integer point obtained from rounding all components of $\hat{y}$ according to the following rule:*

$$\hat{y}_{\lambda,i} = \begin{cases} 1 & \text{if } \hat{y}_i \geq \lambda \\ 0 & \text{if } \hat{y}_i < \lambda \end{cases},$$

and say that $\hat{y}_\lambda$ is obtained from λ-rounding *$\hat{y}$.*

It is easy to see that $\hat{y}_\lambda$ is also a solution to (CPMCS), for all $\lambda \in (0, 1]$. The key observation is that every operation that does not change the relative ordering of the y_{ij} (and also does not violate the $0-1$-bounds), preserves feasibility, as $d_{ij} = 0$ in (4a) whenever there are two variables present in the constraint.

Theorem 3.9. *Let $\hat{y}$ be a solution to (CPMCS). Then $\hat{y}$ is a convex combination of the integral solutions*

$$\{\hat{y}_\lambda \mid \lambda \in \{\hat{y}_i,\ i = 1, \ldots, |S|\}\}.$$

Proof. Let $\Lambda \coloneqq \{\hat{y}_i,\ i = 1, \ldots, |S|\}$ denote the set of values occurring in $\hat{y}$. We denote them by $\lambda_1, \ldots, \lambda_{|S|}$ and assume they are ordered increasingly, i.e. $\lambda_i \leq \lambda_j$ whenever $i \leq j$. We claim that

$$\hat{y} = \sum_{k=1}^{|S|} (\lambda_k - \lambda_{k-1}) \hat{y}_{\lambda_k}, \tag{7}$$

where λ_0 is interpreted as 0. Indeed, the i-th component of $\sum_{k=1}^{|S|}(\lambda_k - \lambda_{k-1})\hat{y}_{\lambda_k}$ is equal to

$$\begin{aligned}\sum_{k=1}^{|S|}(\lambda_k - \lambda_{k-1})\hat{y}_{\lambda_k,i} &\overset{\text{Def. 3.8}}{=} \sum_{k:\lambda_k \le \hat{y}_i}(\lambda_k - \lambda_{k-1})1 \\ &\overset{\text{telescope sum}}{=} \max_{k:\lambda_k \le \hat{y}_i} \lambda_k - \underbrace{\lambda_0}_{=0} \\ &= \hat{y}_i.\end{aligned}$$

Furthermore, we have

$$\lambda_k - \lambda_{k-1} \ge 0 \text{ for all } k = 1, \dots, |S|,$$

and also

$$\sum_{k=1}^{|S|}(\lambda_k - \lambda_{k-1}) = \lambda_{|S|} - \lambda_0 = 1,$$

since (5) implies $1 \in \Lambda$, and therefore $\lambda_{|S|} = 1$. Thus, Equation (7) describes $\hat{y}$ as a convex combination of $\{\hat{y}_\lambda \mid \lambda \in \Lambda\}$. □

4 Computational Results

In this section, we compare the efficiency of the three MIP formulations for (CPMCS) we have discussed previously: the first, naive compatibility formulation (2), the totally unimodular compatibility formulation (3) and the formulation as a dual network flow problem. We do this by evaluating them on real-world benchmark instances arising from two different applications: energy-efficient railway timetabling and piecewise linearization of the physical flow constraints on gas networks. We will see that passing from the original to the unimodular formulation already brings a significant computational advantage, but that the sparsity of the dual-flow formulation allows for the best results by far. Our computational study thus immediately shows two more things: staircase structures are present in real-world application problems and their exploitation is very beneficial in terms of computation time.

4.1 Computational Results for Energy-Efficient Timetabling

The first example for a successful exploitation of staircase compatibility we present here is a problem in railway timetabling. The aim is to take a preliminary timetable which is currently in the planning phase (typically towards the end) and to use the remaining degrees of freedom to allow for a reduction of the energy costs of the involved train operating companies (TOCs). This is possible by taking into account that a big consumer of electricity (as a TOC undoubtedly is) typically has an electricity contract consisting of two price components: the overall energy consumption and the maximum average power consumption over all 15-minute intervals in the billing period. In the special case of a German TOC, the electricity provider charges the collective consumption of all the trains operated by this TOC. This is done by summing up their individual power consumption profiles as measured by the electricity meters in the locomotives and computing both the area under the resulting curve (i.e. the total energy consumption) as

well as the maximum 15-minute average. Both values are multiplied with some cost factor and summed to obtain the final electricity bill. One possibility for optimization via timetabling now lies in adjusting the departure times of the trains in the stations. A train generally draws most power while accelerating. Thus, high peaks in consumption can be avoided if too many simultaneous departures are desynchronized, which can be used to decrease the price componet based on peak consumption. In many cases, this effect can already be achieved via small shifts in the departure times and is thus an interesting trade-off to be'considered: the power-based price component typically makes up for 20–25 % of the energy bill.

We illustrate the effect of this optimization in Figure 1. It shows the power con-

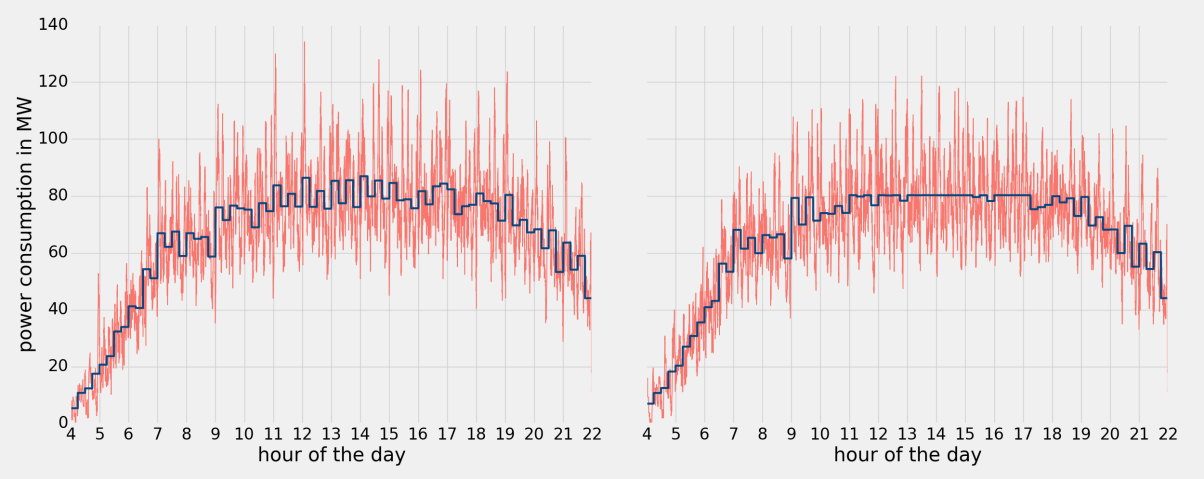


Figure 1: Power consumption profile of timetabling instance *Würzburg* before (left) and after (right) optimization

sumption profile before and after optimization for one of the benchmark instances introduced later (*Würzburg*) on a sample day. The curves in red show the power consumption in each second, while the blue curves show their consecutive 15-minute averages. As stated, the TOC is charged proportionally to the highest such average over the billing period (typically one year). According to the official price sheet by DB Energie GmbH for 2016, the cost factor is 120.83 € per kW and year, such that the demonstrated reduction from 87 to 80 MW in peak consumption equals an annual cost saving of around 850,000 € (and this is a rather small instance). Note that the energy recuperated from braking trains is refunded separately and not offset against the power drawn from the power supply. Thus, we can assume that the consumption profile is always non-negative.

In the following, we give a statement of the problem in terms of staircase compatibility and present our computational study on problem instances of different sizes.

The Problem as a Special Case of (CPMCS) We consider a given initial timetable in which each departure time of a train from a station may be shifted within some interval around the current departure time. We assume that the travel times of the trains on the tracks as well as the corresponding power consumptions are fix. Furthermore, we assume that the temporal order of the trains passing a certain track may not be changed by the optimization and that all connections between different trains in a station must be

preserved in order to maintain the structure of the original timetable as far as possible. Assuming a fixed order of the trains on each track, we also know the safety distances to respect between each consecutive pair of trains. The problem is now to find an adjusted timetable that minimizes the maximum average power consumption.

In order to state this problem in terms of (CPMCS), we need to define the basic set S and the compatibility relation R. Let D be the set of all trains, V^d the set of all stations from which train $d \in D$ departs A^d the tracks it uses. Let furthermore $J^{dv} \subseteq T$ denote the set of all feasible departure times for train $d \in D$ from station $v \in V^d$ within a given planning horizon T. We choose S to be the set of all triples (r, v, j) of a train $d \in D$ and its feasible departure times $j \in J^{dv}$ from some of its stations $v \in V^d$. It is then natural to choose the partition $S = \bigcup_{(d,v):d\in D, v\in V^r} S_{dv}$, where S^{dv} are all feasible triples (d, v, j) for some fixed d and v. A feasible timetable is then made up of a selection of exactly one element from each subset S^{dv}:

$$\sum_{j\in S^{dv}} x_j^{dv} = 1 \quad (\forall d \in D)(\forall v \in V^d).$$

To be feasible, this selection has to respect several further constraints which are stated in the following. The travel time for a train $d \in D$ to pass a track $a = (v, w) \in A^d$ on a journey between two stations $v, w \in V^d$ is Γ^{da}, and at after arriving at startion w it has to stop for a minimum time of c^{dv}. For each pair of consecutive trains (d_1, d_2) on a track between to stations v and w, as given by a set L^{vw}, we have to keep to a minimum headway time of $s^{d_1 d_2 vw}$. Finally, for each station $v \in \bigcup_{d\in D} V^d$ where a pair of trains (d_1, d_2) meets such that the time that passes between the arrival of d_1 and the departure of d_2 is at least $\rho^{d_1 d_2 v}$ and at most $\theta^{d_1 d_2 v}$, as given by a set U_v, this property has to be preserved in the new timetable to maintain the possibility to change between the two trains.

The relation R stating the compatibility between two elements $r_1 = (d_1, v_1, j_1), r_2 = (d_2, v_2, j_2) \in S$ is now given by

$$R = R_1 \cap R_2 \cap R_3.$$

Here, relation R_1 models the compatibility according to the minimum stopping times:

$$R_1 = \Big\{(r_1, r_2) \in S \times S \mid d_1 = d_2 =: d, v_1 = v_2 =: v \in V^d, (v, w) =: a \in A^d, \\ j_2 \geq j_1 + \Gamma^{da} + c^{dw}\Big\},$$

relation R_2 models the compatibility according to the minimum headway times:

$$R_2 = \Big\{(r_1, r_2) \in S \times S \mid v_1 = v_2 =: v \in V^d, (v, w) := a \in A^{d_1} \cap A^{d_2}, (d_1, d_2) \in L^{d_1 d_2 a}, \\ j_2 \geq j_1 + s^{d_1 d_2 a} + \min(\Gamma^{d_1 a} - \Gamma^{d_2 a}, 0)\Big\}$$

and relation R_3 models the compatibility according to the connection times:

$$R_3 = \Big\{(r_1, r_2) \in S \times S \mid (v_1, v_2) \in A^{d_1}, (d_1, d_2) \in U^{v_2}, \\ j_2 \geq j_1 + \Gamma^{d_1 a} + \rho^{d_1 d_2 v} \wedge j_2 \leq j_1 + \Gamma^{d_1 a} + \theta^{d_1 d_2 v}\Big\}.$$

It is easy to check that each of the three relations R_1, R_2 and R_3 is a staircase relations on S. Likewise, it is easy to check that the intersection of any number of staircase relations is again staircase. Consequently, R is a staircase relation on S, which allows us to to formulate the set of feasible selections according to each of the three models derived in Section 3.

What is left to define is the objective function. Let $p^{dat} \geq 0$ be the consumption of train d when passing track $a = (v, w) \in A^r$ at point $0 \leq t \leq \Gamma^{da}$ after departure. Comnsequently, if train d departs from station v at time j, the consumption at point $t \in T$ is given by:

$$\bar{p}_j^{dat} = \begin{cases} \max(p^{dat}, 0), & 0 \leq t - j \leq \Gamma_a^d \\ 0, & \text{otherwise.} \end{cases}$$

Let $I = \{1, 2, \ldots, m\}$ be the set of the m consecutive 15-minute (= 900-second) intervals in T (where the last interval may actually be somewhat shorter). The total energy consumption of a train $d \in D$ on track $a \in A^r$ within an averaging interval $i \in I$ when choosing departure time $j \in J^{dv}$ is then given by $e_j^{dai} = \frac{1}{2}(\bar{p}_j^{da,900i} + \bar{p}_j^{da,900(i+1)}) + \sum_{900i+1 \leq t \leq 900(i+1)-1} \bar{p}_j^{dat}$ (we consider the consumption p as a piecewise-linear function over time). The average power consumption over an interval $i \in I$ by all trains $d \in D$ depending on the chosen departure times is then given by

$$z_i(x) = \frac{1}{900} \sum_{d \in D} \sum_{a=(v,w) \in A^d} \sum_{j \in J_v^d} e_j^{dai} x_j^{dv}.$$

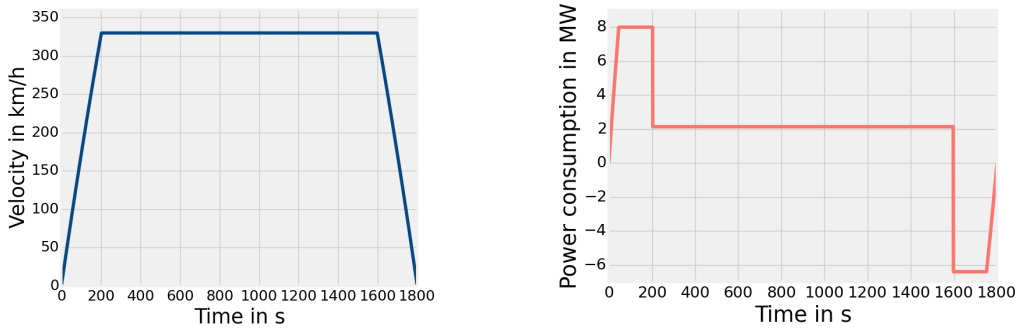
This leads to the following opptimization problem to minimize the highest of these averages:

$$\min_{x \in X} \max_{i \in I} z_i(x),$$

where X is the set of all feasible timetables. For this set X, we can now chosen between one of the three models for staircase compatibility derived in Section 3. Note that this timetabling problem is NP-hard even if all trains only have one track and $m = 2$ as this case can easily be reduced to the subset sum problem (cf. (Garey and Johnson, 1979, SP13)).

Computational Comparison of the Models for (CPMCS) We now present a computational study that compares the different formulations for staircase compatibility considered before as a part of the timetabling problem introduced above. We do this on real-world instances derived from the 2015 timetable for the German passenger traffic operated by our industry partner Deutsche Bahn AG (DB). We complemented this data by power consumption profiles based on height data of the stations as well as simplified speed profiles taking into account train characteristics. An example is depicted in Figure 2 which shows an assumed speed profile for an ICE-3 on a journey of 30 minutes in Figure 2a and the corresponding power profile on a track with an upwards inclination in Figure 2b. The minimum headway times we chose are based on (Pachl, 2016, Table 5.4) by rounding up the given values to full minutes.

Altogether, we have created 30 instances of different sizes, each for a planning horizon of 18 hours (4am to 10pm). These contain 18 *local instances* which contain all trains passing a certain station in Germany, 1 *Fernverkehr instance* covering the German long-distance traffic, 10 *regional instances* which contain all short-distance trains circulating



(a) Speed profile

(b) Power consumption profile

Figure 2: Example profiles for an ICE-3 on a 30 minutes journey climbing an inclination

in a given region of Germany, as well 1 *Deutschland instance* covering all German DB passenger trains. Each instance contains those parts of the journeys of the involved trains which fall within the planning horizon. The allowable shift in departure time was uniformly chosen to be ± 3 minutes around the current departure time. The sizes of the created instances, the computation times of the three different models as well as the achieved savings in peak power consumption are shown in Table 1. Here, NA denotes the naive formulation (2), TU stands for the totally unimodular formulation (3), and DF represents the formulation as a dual network flow problem. The result is very clear: Formulation DF is by far the best way to formulate the compatibilities as it leads to the fastest solution times on all but one instance. In many cases, the benefit is very significant. Most notably, for the Germany-wide instance *Deutschland* the computation time could be decreased from somewhat more than 3 hours to under 3 minutes – a factor of more than 60. The table also shows that the computation time of Formulation TU is usually between the solution times required for Formulations NA and DF. This shows the general benefit of passing to a totally unimodular description of the set of feasible timetables. However, the sparsity of Formulation DF leads to much lower node solution times in the branch-and-bound tree and is therefore vastly superior. We remark here that the stated reduction of about 4 % in peak power consumption for the Germany-wide instance would allow for cost savings of several million euros per year. More detailed information on the problem can be found in Bärmann et al. (2015).

4.2 Computational Results for Piecewise Linearization of Path Flows

Our second example for a staircase compatibility structure originates from piecewise linearization of a nonlinear network flow problem. In gas network optimization, in addition to a classical network flow problem for the mass flow, pressure has to be considered as well. Therefore, pressure variables are introduced for network nodes and additional constraints describe the nonlinear pressure loss along pipes. (Pfetsch et al., 2015, Equation (7)) give a commonly used algebraic approximation of the underlying physics. For a constant compressibility factor (see (Pfetsch et al., 2015, Equation (20))), the reduction in squared pressure Δp^2 on a pipe is a nonlinear univariate function of the flow q on that pipe, given by

$$\Delta p^2 = \lambda q|q|,$$

Instance	#Trains	#Trips	Computation time [s] NA	TU	DF	Saving [%]
Zeil	42	764	78.59	8.71	**0.91**	14.11
Bayreuth Hbf	69	329	1.16	0.96	**0.27**	22.67
Passau Hbf	75	1048	447.66	65.06	**7.81**	13.57
Jena Paradies	78	1109	353.71	11.57	**2.35**	12.05
Lichtenfels	113	1659	1307.67	137.49	**19.82**	16.45
Erlangen	142	2978	6185.18	2593.09	**28.17**	16.27
Bamberg	210	3659	7434.75	83.55	**24.06**	13.51
Aschaffenburg Hbf	245	3480	16.43	16.28	**2.67**	13.24
Kiel Hbf	298	2145	117.83	17.63	**2.63**	11.68
Leipzig Hbf (tief)	372	6828	5.44	12.72	**1.57**	5.43
Würzburg Hbf	373	4486	–	15614.77	**1417.03**	7.67
Dresden Hbf	422	6964	7932.98	173.49	**23.75**	6.53
Ulm Hbf	469	5753	647.74	76.24	**7.52**	11.36
Stuttgart Hbf (tief)	628	11612	1655.78	1185.40	**11.55**	1.03
Berlin Hbf (S-Bahn)	642	16146	26.76	77.09	**10.57**	3.03
Hamburg-Altona(S)	722	12391	609.79	210.54	**13.35**	1.31
Frankfurt(Main)Hbf	728	8687	3569.75	394.19	**47.95**	9.66
Nürnberg Hbf	952	12236	26039.74	30.06	**3.18**	5.62
Fernverkehr	671	7214	536.31	32.78	**7.34**	5.04
S-Bahn Hamburg	1209	17562	1305.85	315.20	**24.61**	2.59
Regio Nord	1477	13434	311.23	42.67	**8.49**	12.46
Regio Nordost	1495	16543	3974.56	97.17	**15.41**	15.75
Regio Hessen	1547	25135	52.37	82.42	**16.63**	5.51
Regio Südwest	1864	24244	399.49	47.21	**13.07**	12.48
Regio Südost	2361	32009	914.66	94.63	**34.23**	8.61
Regio BW	2385	30227	8535.13	277.87	**28.95**	13.73
S-Bahn Berlin	2584	53449	**93.16**	1153.95	345.81	1.42
Regio Nordwest	2828	47130	68.00	127.45	**31.71**	5.05
Regio Bayern	3560	49371	693.92	221.26	**48.04**	11.03
Deutschland	21981	316318	11847.46	3292.64	176.48	4.23

Table 1: Computational results for the three problem formulations for the energy-efficient timetabling problem

for some $\lambda > 0$.

In order to deal with those nonlinearities, a common and successful approach consists of constructing piecewise linearizations or relaxations of the involved nonlinear functions (see Figure 3), which makes the problem accessible to general purpose MIP solvers. It involves subdividing the feasible set into several intervals and introducing binary variables that indicate which interval the argument value is in. For a detailed description of the this technique, see Geißler et al. (2012).

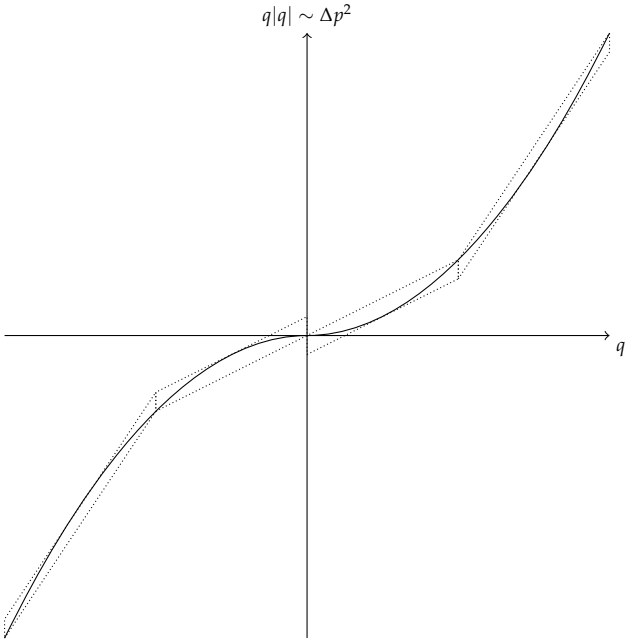


Figure 3: Illustration of a piecewise linear relaxation (dotted parallelograms) of a univariate nonlinear function.

It should be mentioned that gas network optimization problems usually contain additional challenges connected with the operation of active elements such as valves and compressors. However, as the substructure we are interested in can be motivated just from passive networks, we do not want to go into detail here and instead refer the reader to the literature, e.g. Koch et al. (2015).

For constructing a piecewise linear approximation, i.e. for modelling a piecewise linear function, several useful formulation methods are known. For now, we assume that we use a method where there is a binary variable z_I for each interval $I = [l_I, u_I]$ with the meaning

$$z_I = 1 \quad \Rightarrow \quad q_a \in [l_I, u_I], \tag{8}$$

as is true, for example, for the *Multiple-Choice Method* (MCM) (Jeroslow and Lowe (1984)) as well as for the *Convex-Combination Method* (CC). For a single piecewise linear function, MCM leads to *locally ideal* formulations, i.e. their linear relaxation is equal to the convex hull of feasible points. The basic version of CC is not locally ideal; however, a locally ideal improved variant is proposed in Padberg (2000). However, the situation is different when we consider multiple nonlinear functions that influence each other: in general, the formulation loses its desired property of being ideal.

For a network that is a path of length k, we can regain a complete description in the context of staircase compatibility. This is also covered as a special case in Liers

and Merkert (2015), though using different reasoning. We have already seen in Subsection 2.1 that this represents a special case of (CPMCS).

Computational Comparison of the Models for (CPMCS) We will also test the impact the different formulations for staircase compatibility have on instances arising from piecewise linearized network flow problems. We use the following setting for our test instances: given a network, for all arcs a we have real-valued flow variables q_a that have to satisfy flow conservation and demand satisfaction equations

$$\sum_{a \in \delta^+(v)} q_a - \sum_{a \in \delta^-(v)} q_a = d_v$$

for all network nodes, where d_v denotes the given demand for node v.

Furthermore, for each arc a we have several possible intervals $I_{a,1}, \dots, I_{a,n_a}$ for the arc flow, where n_a denotes the number of intervals belonging to arc a. Intervals belonging to the same arc arise from subdividing a larger interval such that they intersect in at most a point and come with a natural ordering. Let $z_{a,j}$ denote binary indicator variables for using the j-th interval on arc a. Lower and upper bounds of an interval $I_{a,j}$ are denoted by $l_{a,j}$ and $u_{a,j}$ respectively. Only one interval per arc can be active – and at least one has to. Therefore, the corresponding z-variables are connected by the constraint

$$\sum_{j=1}^{n_a} z_{a,j} = 1,$$

which represents (rule 1).

For implementing (8), the Multiple-Choice Method is used. Recall that a piecewise linear approximation of a univariate function f of q modelled by MCM on a connected domain $[l, u]$ with breakpoints $B_1 = l, B_2, \dots, B_k, B_{k+1} = u$ is obtained for $l_{a_i} = B_i, u_{a_i} = B_{i+1}, i = 1, \dots, k$. We create a "copy" q_i of the arc flow q for every interval and ensure (8) by the constraints

$$l_i z_i \leq q_i \leq u_i z_i \quad \forall\, i = 1, \dots, k,$$

where arc indices of the z-variables are omitted for better readability as we only consider a single arc. Then, q and $f(q)$ can be expressed as follows:

$$q = \sum_{i=1}^{k} q_i, \quad f(q) = \sum_{i=1}^{k} f(B_i) z_i + (q_i - B_i z_i) \frac{f(B_{i+1}) - f(B_i)}{B_{i+1} - B_i}.$$

Those constraints constitute our polyhedron of interest. The underlying network is given by the topology of a real-world gas network by the German gas network operator *Open Grid Europe* (OGE) consisting of 592 nodes and 623 arcs. As the network is not a path, there is no complete description available. However, (CPMCS) is present as a substructure, e.g. at each induced path of degree-two nodes in the network. 224 nodes have degree two and there are 128 paths of degree-two-nodes, which amounts to an average length of 2.75. The longest of those paths has length 8. In the following, we want to test the effect of using our improved formulations of (CPMCS) in those places.

In order to do this, we first identify all suitable subpaths of degree-two nodes in the network, construct the corresponding compatibility graphs and precompute the unimodular formulation of Model (3) for each of the detected subpaths. This description is quadratic in the length of the path and linear in the number of intervals per arc. We

do not add all those constraints right from the start as in practice many of them are redundant. Instead, we use a separation callback at every 50th branch-and-bound node that finds all violated inequalities and adds them to the model. The callback is called at most 100 times.

For all test instances additional input data is generated at random. This includes the vector d of demands as well as the initial arc capacities c. Capacities were scaled in such a way that feasibility of all instances is guaranteed. We then chose a random partition of the interval $[-c_a, c_a]$ into a given constant number of intervals for each network arc. This number varies across different instance sets and is meant to roughly represent the accuracy of the linearization. The objective function is constructed by drawing integer coefficients for the z-variables. This is done uniformly at random from the interval from 0 to twice the number of intervals per arc, with the restriction that there is an upper bound on the resulting "slope" of the objective function. As the generation of instances includes randomness, we always generated sets of five instances with the same number of intervals per arc. The solution times given in the following are always (geometric) averages over five instances each. If only a subset of the five instances was solvable within the time and memory limitations, the average is taken over this subset only. In any case we also state the number of instances that could be solved.

The computational experiments have been performed on a queuing cluster of Intel Xeon E5-2690 3.00 GHz computers with 25 MB cache and 128 GB RAM, running Version 7 of Debian GNU/Linux. Our implementation uses the C++-API of Gurobi 6.0.0. For large numbers of intervals per arc, we encountered numerical difficulties on the test set. These numerical issues are already observed in the standard formulation. To overcome this, we increased Gurobi's parameter *NumericFocus* to the value of 3 in order tell to the solver to be more careful regarding numerical issues. As a result, we did not observe numerical difficulties for any of the instances any more. However, this choice results in longer running times. Apart from that, we use Gurobi's standard parameter settings, except for turning on *PreCrush* for our cutting plane methods, which is mandatory if we want to add user cuts. Each job was run on 4 cores and with a time limit of 40 hours CPU-time.

# intervals	MCM		MCM + TU-paths	
per arc	solved	CPU[s]	solved	CPU[s]
3	**5**	**66.63**	5	73.13
4	5	4 943.87	**5**	**468.66**
5	5	9 001.10	**5**	**1 627.76**
6	2	31 384.31	**3**	**10 089.66**
7	1	103 191.92	**2**	**122 299.54**
8	0	∞	0	∞

Table 2: Number of instances solved and average solution times for instances on a gas network topology with 592 nodes and a varying number of intervals per arc

As can be seen from Table 2, adding constraints from the TU formulation of the (CPMCS)-substructures (i.e. paths of degree-two nodes) improves the runtime of the solver considerably for most test sets. This effect increases with a growing number of intervals per arc, resulting in a total of 2 more instances that can be solved within the

time limit.

Next, we want to apply the dual-flow formulation. Note, however, that the transformation used to obtain this formulation and to prove Theorem 3.5 in the present context of piecewise linearization is well-known for connecting the Incremental Method to methods using (8), as already mentioned in Remark 3.6. This suggests using the Incremental Method as a linearization method.

Let us quickly recall the Incremental Method (or δ-Method), first introduced in Markowitz and Manne (1957): let an interval $[l, u]$ for the flow value and breakpoints $B_1 = l, B_2, \ldots, B_n, B_{n+1} = u$ be given. We have continuous $[0, 1]$-variables δ_i, and the constraint

$$q = B_1 y_1 + \sum_{i=1}^{n} (B_{i+1} - B_i)\delta_i$$

together with the *filling condition* constraints $y_i \geq \delta_i, i = 1, \ldots, n$ and $\delta_i \geq y_{i+1}, i = 1, \ldots, n-1, \delta_n \geq 0$. A piecewise-linear function f of q can then be written as

$$f(q) = f(B_1)y_1 + \sum_{i=1}^{n} (f(B_{i+1}) - f(B_i))\delta_i.$$

The Incremental Method is widely used in practice and has proved very useful in the context of gas networks, see e.g. Correa-Posada and Sánchez-Martín (2014). Like MCM, it leads to locally ideal formulations, but of course also only in case of a single arc. In the following, we compare the standard formulation with and without adding constraints from the totally unimodular dual-flow formulation (which naturally uses the binary y-variables of the Incremental Method).

# intervals	INC		INC + TU-paths	
per arc	solved	CPU[s]	solved	CPU[s]
4	**5**	**5.61**	5	6.10
5	5	13.73	**5**	**10.50**
6	5	141.02	**5**	**41.96**
7	5	197.94	**5**	**68.49**
8	5	1424.02	**5**	**195.95**
9	5	1144.44	**5**	**857.59**
10	5	25506.75	**5**	**837.45**
12	3	85712.83	**5**	**3048.45**
15	0	∞	**5**	**44275.51**
20	0	∞	**1**	**824.18**
25	0	∞	0	∞

Table 3: Number of instances solved and average solution times for instances on a gas network topology with 592 nodes and a varying number of intervals per arc, using the Incremental Method

The results can be found in Table 3. Using the Incremental Method reduces the overall runtime by a large factor, such that instances up to 12 intervals per arc (20 with the TU formulation on paths) can now be solved. This agrees with Correa-Posada and Sánchez-Martín (2014), where a recent in-depth computational study for piecewise-linear functions in the context of gas network optimization sees the Incremental Method

coming out on top, outperforming the Multiple-Choice Method by several orders of magnitude for some test sets. It also gives an additional argument for the dual-flow formulation, as its variables seem to suit solvers well in this context. Providing the solver with the TU-formulation on paths again increases the performance of the solver significantly. For further related computational experiments see Liers and Merkert (2015).

The results of this subsection show that the TU-formulation can have a large benefit, not only if the feasible set – as in the last subsection – can be described as (CPMCS) as a whole, but also if (CPMCS) is present as a substructure.

5 Conclusion

In this paper, we introduced the notion of staircase compatibility, which generalizes compatibility structures known from different areas of application, such as project scheduling and piecewise linearization. We showed that the convex hull of feasible solutions of the clique problem with multiple-choice constraints can be described by a totally unimodular constraint matrix of polynomial size if the compatibility graph is given by a staircase relation. Furthermore, we showed that the constraint matrix is cographic, which yields a dual-flow formulation for the problem.

For two example applications, where (CPMCS) is present as a substructure, we showed that using unimodular formulations for (CPMCS) represents a significant improvement over a naive formulation and can vastly reduce solution time.

With these insights, future research may aim to identify (CPMCS) within more applications or even automatically in general MIPs in order to do a reformulation. Connected to this is the question whether – or in which cases – staircase compatibility structure can be recognized from a compatibility graph if the partitioning is not given. This might also be interesting from a graph theoretic point of view.

Acknowledgements

We gratefully acknowledge the computing resources provided by the group of Michael Jünger in Cologne. In particular, we thank Thomas Lange for his technical support. We also thank Rodrigo Alexander Castro Campos, Sergio Luis Pérez Pérez, Gualberto Vazquez Casas and Francisco Javier Zaragoza Martínez for our fruitful discussions on the topic. Furthermore, we acknowledge financial support by the BMBF under grant 05M13WEE. Moreover, we thank the EnCN for support within research focus Simulation, Project TP6, and the DFG for their support within Projects A05, B06, and B07 in CRC TRR 154.

References

Bärmann, A., Martin, A., and Schneider, O. (2015). Optimal balancing of the power consumption of trains in a railway network via timetabling. In *Proceedings of CASPT 2015*.

Correa-Posada, C. M. and Sánchez-Martín, P. (2014). Gas network optimization: A comparison of piecewise linear models. Optimization Online.

Fourer, R. (1984). Staircase matrices and systems. *SIAM Review*, 26(1):1–70.

Garey, M. R. and Johnson, D. S. (1979). *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W.H. Freeman and Company, New York.

Geißler, B., Martin, A., Morsi, A., and Schewe, L. (2012). Using piecewise linear functions for solving MINLPs. In Lee, J. and Leyffer, S., editors, *Mixed Integer Nonlinear Programming*, volume 154 of *The IMA Volumes in Mathematics and its Applications*, pages 287–314. Springer New York.

Ghouila-Houri, A. (1962). Caracterisation des matrices totalement unimodulaires. *Comptes Rendus Hebdomadaires des Séances de 1'Académie des Sciences (Paris)*, 254(1):1192–1194.

Jeroslow, R. G. and Lowe, J. K. (1984). Modelling with integer variables. In Korte, B. and Ritter, K., editors, *Mathematical Programming at Oberwolfach II*, volume 22 of *Mathematical Programming Studies*, pages 167–184. Springer Berlin Heidelberg.

Koch, T., Hiller, B., Pfetsch, M., and Schewe, L., editors (2015). *Evaluating Gas Network Capacities*. MOS-SIAM Series on Optimization.

Liers, F. and Merkert, M. (2015). Structural investigation of piecewise linearized network flow problems. Technical report, FAU Erlangen-Nürnberg.

Markowitz, H. M. and Manne, A. S. (1957). On the solution of discrete programming problems. *Econometrica*, 25(1):pp. 84–110.

Möhring, R. H., Schulz, A. S., Stork, F., and Uetz, M. (2001). On project scheduling with irregular starting time costs. *Operations Research Letters*, 28(4):149–154.

Oxley, J. G. (2006). *Matroid Theory (Oxford Graduate Texts in Mathematics)*. Oxford University Press, Inc., New York, NY, USA.

Pachl, J. (2016). *Systemtechnik des Schienenverkehrs: Bahnbetrieb planen, steuern und sichern*. Springer Vieweg.

Padberg, M. (2000). Approximating separable nonlinear functions via mixed zero-one programs. *Oper. Res. Lett.*, 27(1):1–5.

Pfetsch, M. E., Fügenschuh, A., Geißler, B., Geißler, N., Gollmer, R., Hiller, B., Humpola, J., Koch, T., Lehmann, T., Martin, A., Morsi, A., Rövekamp, J., Schewe, L., Schmidt, M., Schultz, R., Schwarz, R., Schweiger, J., Stangl, C., Steinbach, M. C., Vigerske, S., and Willert, B. M. (2015). Validation of nominations in gas network optimization: Models, methods, and solutions. *Optimization Methods and Software*, 30(1):15–53.

Schrijver, A. (1998). *Theory of Linear and Integer Programming*. John Wiley & Sons Ltd.

Schwindt, C. and Zimmermann, J., editors (2015). *Handbook on Project Scheduling (Vol. 1 + Vol. 2)*. Springer.

Seymour, P. (1980). Decomposition of regular matroids. *Journal of Combinatorial Theory, Series B*, 28(3):305 – 359.

Vielma, J. P. (2015). Mixed integer linear programming formulation techniques. *SIAM Review*, 57(1):3–57.