Optimal University Course Timetables and the Partial Transversal Polytope

Gerald Lach* Marco E. Lübbecke*

November 15, 2007

Abstract

University course timetabling is the conflict-free assignment of courses to weekly time slots and rooms subject to various hard and soft constraints. One goal is to meet as closely as possible professors' preferences. Building on an intuitive integer program (IP), we develop an exact decomposition approach which schedules courses first, and matches courses/times to rooms in a second stage. The subset of constraints which ensures a feasible room assignment defines the well-known partial transversal polytope. We describe it as a polymatroid, and thereby obtain a complete characterization of its facets. This enables us to add only strong valid inequalities to the first stage IP. In fact, for all practical purposes the number of facets is small. We present encouraging computational results on real-world and simulated timetabling data. The sizes of our optimally solvable instances (respecting all hard constraints) are the largest reported in the literature by far.

Keywords: integer programming; partial transversal polytope; university course timetabling

1 Introduction

Timetabling comes in many flavors, in education and sports, in industry and public transport. This diversity and its relevance in practice made timetabling an active research area in operations research; a series of conferences (Practice and Theory of Automated Timetabling, PATAT) is devoted to the topic [4]. In this paper, we aim for optimal solutions to one of the core problems of the field, the NP-complete *university course timetabling problem.*

A university timetable is an assignment of an appropriate number of time slots, or *periods*, and rooms to each weekly occurrence of each course. It is usually valid for one term. Customarily, one distinguishes between hard and soft constraints which have to be respected [2]. Typical hard constraints are: A professor cannot teach two classes at the same time; lectures belonging to the same curriculum must not be scheduled simultaneously; a room cannot be assigned to different courses in the same period; etc. A timetable is infeasible if one of these requirements is violated (which frequently occurs in practice). Soft constraints e.g., call for not exceeding a room's capacity; to provide the necessary equipment like beamer/PC; to spread the lectures of one course over the week; etc. A violation of these constraints is tolerated but penalized. Professors express preferences as to when to teach; an optimal timetable minimizes the total deviation from these preferences.

*Technische Universität Berlin, Institut für Mathematik, MA 6-1, Straße des 17. Juni 136, 10623 Berlin, Germany, Email: {lach, m.luebbecke}@math.tu-berlin.de.

1.1 Our Contribution

This paper makes a contribution to practical problem solving via integer programming, as well as it adds to the theory of combinatorial optimization.

On the practical side, we give a proof-of-concept that optimal timetables can be computed for larger universities in acceptable time. Our focus in on meeting all hard constraints, where we take some of the constraints traditionally considered soft (like room capacity) as hard ones.

In the integer program we propose, instead of simultaneously assigning courses to time slots and rooms, we only schedule rooms, providing for a later feasible room assignment. This is done by interpreting feasible course/room pairs on a bipartite graph, and enforcing the classical Hall's conditions [13] on the existence of perfectly matchable sets (or transversals). This allows a simpler formulation and results in much improved solution times.

Hall's conditions directly lead us to an investigation of the partial transversal polytope [18]. We obtain a complete description of its facets by stating it as a polymatroid. Thereby, on the theoretical side, we obtain an interesting strengthening of Hall's Theorem. Finally, we are interested in the number of facets of the partial transversal polytope, and obtain a generating set of facets, of linear size. All facets can be obtained from this set by an intuitive operation.

Currently, there is an international timetabling competition [10] and we tested our approach on the given instances. It turns out that we are able to compute optimal solutions within negligible running times. We therefore tested our approach on simulated data which is almost identical to real data from Technical University of Berlin. To the best of our knowledge, we are the first to obtain optimal solutions to university course timetabling instances of this size.

It is our impression that integer programming has been used for timetabling only because of its modeling power. It was not realized that a deeper understanding of combinatorial properties of the problem may be the key to actually solving large instances to proven optimality. In this sense, we consider our work a significant step forward in this field of research.

1.2 Related Work

University course timetabling problems are well studied, see e.g., the surveys [3, 16]. Much has been written about practical details [7], and the non-negligible human factor of timetabling [17].

Meta heuristics clearly constitute the main solution approach, see [2, 12, 14], and the references therein. Several integer programs were suggested as well [5, 6, 7, 15, 17], however, optimally solvable problem instances are (i) smaller than ours by at least an order of magnitude, or (ii) are much simpler (and thus less realistic) than ours.

Interestingly, complete polyhedral descriptions of problems closely related to finding transversals are well known. We have Edmonds' seminal work on the matching polytope [8]. Also, the perfectly matchable subgraph polytope for bipartite graphs is fully characterized [1]. Yet, we are not aware of any previous attempts to give a strong formulation of the partial transversal polytope.

2 Integer Programs and Decomposition

2.1 An Intuitive Integer Program

We give a generic integer program (IP) for the university course timetabling problem which concentrates on hard constraints (time conflicts and room conflicts). However, it is easy to enhance this IP by soft constraints.

Denote by $\mathcal{C}$ the set of courses, by $\mathcal{R}$ the set of rooms, and by $\mathcal{T}$ the set of time slots. For each course $c \in \mathcal{C}$ we know its eligible time slots $T(c) \subseteq \mathcal{T}$, and eligible rooms $R(c) \subseteq \mathcal{R}$. Further, $R^{-1}(r) \subseteq \mathcal{C}$ is the set of all courses which may take place in room $r \in \mathcal{R}$. Each course $c \in \mathcal{C}$ consists of $\ell(c)$ lectures, that is, we have to provide $\ell(c)$ different time slots for course c. The instructor of course $c \in \mathcal{C}$ assigns a preference $prio(c, t)$ to all eligible time slots $t \in T(c)$; the smaller it is, the better.

Time conflicts of any kind are represented via a conflict graph $G_{\text{conf}} = (V_{\text{conf}}, E_{\text{conf}})$: A vertex (c, t) represents an eligible combination of a course c and a timeslot t. Two nodes (c_1, t_1) and (c_2, t_2) are adjacent iff it is forbidden that c_1 is scheduled at t_1 and c_2 at t_2 (typically, $t_1 = t_2$). We see that time conflicts introduce a stable set flavor into our problem.

A binary variable $x_{c,t,r}$ represents whether course c is scheduled at time t in room r, or not. The following IP for the generic university course timetabling problem guarantees a sufficient number of time slots per course (2), avoids room conflicts (3), and time conflicts (4).

$$\min \sum_{c,t,r} prio(c,t) \cdot x_{c,t,r} \tag{1}$$

$$s.t. \sum_{t \in T(c), r \in R(c)} x_{c,t,r} = \ell(c) \qquad \forall c \in \mathcal{C} \tag{2}$$

$$\sum_{c \in R^{-1}(r)} x_{c,t,r} \leq 1 \qquad \forall t \in \mathcal{T},\ r \in \mathcal{R} \tag{3}$$

$$\sum_{r \in R(c_1)} x_{c_1,t_1,r} + \sum_{r \in R(c_2)} x_{c_2,t_2,r} \leq 1 \qquad \forall ((c_1,t_1),(c_2,t_2)) \in E_{\text{conf}} \tag{4}$$

$$x_{c,t,r} \in \{0,1\} \qquad \forall (c,t) \in V_{\text{conf}},\ r \in \mathcal{R} \tag{5}$$

This integer program will be infeasible for any reasonable practical data since usually some courses cannot be scheduled without conflicts. Thus, one tries to schedule as many courses as possible; a modification to accomplish this is straight forward. However, the computation times and solution qualities (cf. Table 3) do not advise to actually work with this formulation.

2.2 Decomposition into Time and Room Assignment

Instead, we reduce the problem in three dimensions to a problem in two dimensions, implicitly taking care of room conflicts. To this end, we represent eligible combinations of courses and rooms as undirected bipartite graphs $G_t = (\mathcal{C}_t \cup \mathcal{R}_t, E_t)$, one for every time slot $t \in \mathcal{T}$. Courses which may be scheduled at t are given in set $\mathcal{C}_t$; and $\mathcal{R}_t$ denotes the set of all eligible rooms for all courses in $\mathcal{C}_t$. A course c and a room r are adjacent iff r is eligible for c. For ease of exposition let $G = (\mathcal{C} \cup \mathcal{R}, E)$ be the graph consisting of all components G_t, $t \in \mathcal{T}$.

For any subset U of vertices, denote by $\Gamma(U) := \{i \in \mathcal{C} \cup \mathcal{R} \mid j \in U, (i,j) \in E\}$ the neighborhood of U; in particular, $\Gamma(U) \subseteq \mathcal{R}$ for any $U \subseteq \mathcal{C}$. The set of all vertices which are adjacent *only* to vertices in U is denoted by $\Gamma^{-1}(U) := \{i \in \mathcal{U} \cup \mathcal{R} \mid \Gamma(\{i\}) \subseteq U\}$. In particular, $\Gamma^{-1}(U) \subseteq \mathcal{C}$ for any $U \subseteq \mathcal{R}$.

Hall's Theorem [13] states that a bipartite graph $G = (\mathcal{C} \cup \mathcal{R}, E)$ has a matching of all vertices in $\mathcal{C}$ into $\mathcal{R}$ if and only if $|\Gamma(U)| \geq |U|$ for all $U \subset \mathcal{C}$. This enables us to state a simpler integer program which schedules courses in such a way that a later assignment of rooms is possible. It is thus based on binary variables $x_{c,t}$, but obviously has an exponential number of constraints.

$$\min \sum_{(c,t) \in V_{\text{conf}}} \mathit{prio}(c,t) \cdot x_{c,t} \tag{6}$$

$$\text{s.t.} \sum_{t \in T(c)} x_{c,t} = \ell(c) \qquad \forall c \in \mathcal{C} \tag{7}$$

$$\sum_{c \in U} x_{c,t} \leq |\Gamma(U)| \qquad \forall U \subseteq \mathcal{C},\ t \in \mathcal{T} \tag{8}$$

$$x_{c_1,t_1} + x_{c_2,t_2} \leq 1 \qquad \forall ((c_1,t_1),(c_2,t_2)) \in E_{\text{conf}} \tag{9}$$

$$x_{c,t} \in \{0,1\} \qquad \forall (c,t) \in V_{\text{conf}} \tag{10}$$

Once this IP is solved, the second stage merely consists of solving a sequence of minimum weight bipartite matching problems; clearly, this decomposition approach is exact.

Even though Hall's inequalities (8) can be separated in polynomial time via a maximum flow computation, we would like to work with a strongest possible formulation: We are interested in the facets of the polytope defined by (8) (and non-negativity).

3 The Partial Transversal Polytope

In the context of Hall's Theorem, $\mathcal{C}$ is known as *system of distinct representatives* or *transversal*. A *partial transversal* is a subset of $\mathcal{C}$ which can be perfectly matched (we may assume that all $r \in \mathcal{R}$ will be matched). The *partial transversal polytope* $P(\mathcal{C})$ is the convex hull of all incidence vectors of partial transversals of $\mathcal{C}$. It is full dimensional in $\mathbb{R}^{|\mathcal{C}|}$.

The *deficiency* of a vertex set $U \subseteq \mathcal{C}$ is defined as $\text{def}_G(U) := |U| - |\Gamma(U)|$. The *deficiency of a graph* G is $\text{def}(G) := \max_{U \subseteq \mathcal{C}} \text{def}_G(U)$. We will often consider the deficiency of induced subgraphs $(U \cup \Gamma(U), E)$, and denote it by $\text{def}(U)$, slightly abusing notation. Graph deficiency is known to be supermodular [13], that is, $\text{def}(U \cup V) + \text{def}(U \cap V) \geq \text{def}(U) + \text{def}(V)$ for $U, V \subseteq \mathcal{C}$. Finally, denote by $\nu(G)$ the cardinality of a maximum matching in G.

We consider two equivalent descriptions of the partial transversal polytope $P(\mathcal{C})$. We use the common shorthand notation $x(U) := \sum_{i \in U} x_i$.

Lemma 1 (The Partial Transversal Polytope) *Given a bipartite graph $G = (\mathcal{C} \cup \mathcal{R}, E)$, the partial transversal polytope $P(\mathcal{C}) \subseteq \mathbb{R}^{|\mathcal{C}|}$ is defined by*

$$x(U) \leq |\Gamma(U)| \qquad \forall U \subseteq \mathcal{C} \tag{11}$$

$$0 \leq x \leq 1 \tag{12}$$

or equivalently by

$$\begin{aligned} x(U) &\leq |U| - \mathrm{def}(U) \qquad \forall U \subseteq \mathcal{C} \qquad (13)\\ x &\geq 0 \ . \qquad (14) \end{aligned}$$

The advantage of the latter description is that $x \leq 1$ is not explicitly required. This will facilitate characterizing facets.

3.1 Facets

A theorem by Edmonds on the facets of polymatroids [18, Thm. 44.4] allows us to easily give a complete and non-redundant description of the partial transversal polytope. For a consistent presentation we define the set function

$$f : 2^{\mathcal{C}} \to \mathbb{N},\ \ U \mapsto f(U) := |U| - \mathrm{def}(U)\ \ , \qquad (15)$$

which is submodular by supermodularity of the deficiency. Note also that f is nondecreasing, that is, $f(U) \leq f(T)$ for $U \subseteq T$. Further, $f(\emptyset) = 0$ and $f(\{i\}) > 0$ for $i \in \mathcal{C}$.

A subset $U \subseteq \mathcal{C}$ is called an *f-flat* if $f(U \cup \{i\}) > f(U)$ for all $i \in \mathcal{C} \setminus U$; and U is *f-inseparable* if there are no $U_1, U_2 \neq \emptyset$ with $U_1 \cap U_2 = \emptyset$ and $U_1 \cup U_2 = U$ such that $f(U) = f(U_1) + f(U_2)$.

Edmonds has the following theorem: With the properties of a set function f as given in (15), the facets of $\{x \in \mathbb{R}^{|\mathcal{C}|} \mid x \geq 0,\ x(U) \leq f(U) \text{ for } U \subseteq \mathcal{C}\}$ are given by (i) $x \geq 0$, and (ii) $x(U) \leq f(U)$ for each nonempty f-inseparable f-flat $U \subseteq \mathcal{C}$.

Definition (Defining $\mathcal{C}$-set) Given a bipartite graph $G = (\mathcal{C} \cup \mathcal{R}, E)$, and f as defined in (15). A set $\emptyset \neq U \subseteq \mathcal{C}$ is called *defining $\mathcal{C}$-set*, iff U is an f-flat, and

$$\mathrm{def}(U) > \max_{\substack{U_1, U_2 \subseteq U \\ U_1 \cap U_2 = \emptyset}} \{\mathrm{def}(U_1) + \mathrm{def}(U_2)\}\ \ . \qquad (16)$$

This definition reflects the intuition that a $\mathcal{C}$-set is important, if it bears more information than the union of its parts. Inequality (16) will guarantee f-inseparability.

Theorem 2 *Given a bipartite graph $G = (\mathcal{C} \cup \mathcal{R}, E)$, then a set $U \subseteq \mathcal{C}$ is facet inducing for the partial transversal polytope $P(\mathcal{C})$, if and only if U is a defining $\mathcal{C}$-set.*

Proof: To prove necessity, let $V \subseteq \mathcal{C}$ be facet inducing. V is an f-inseparable f-flat by definition. Hence there are no disjoint $V_1, V_2 \neq \emptyset$ with $V = V_1 \cup V_2$ with

$$|V_1| - \mathrm{def}(V_1) + |V_2| - \mathrm{def}(V_2) = |V| - \mathrm{def}(V)\ \ .$$

Equivalently, for all disjoint $\emptyset \neq V_1, V_2 \subseteq V$:

$$\begin{aligned} |V| - \mathrm{def}(V) &< |V_1| - \mathrm{def}(V_1) + |V_2| - \mathrm{def}(V_2) \\ \mathrm{def}(V) &> \mathrm{def}(V_1) + \mathrm{def}(V_2)\ \ , \end{aligned}$$

so V is a defining $\mathcal{C}$-set. For sufficiency, let V be a defining $\mathcal{C}$-set. V is an f-flat by definition. Further it holds that

$$\mathrm{def}(V) > \max_{\substack{U_1, U_2 \subseteq U \\ U_1 \cap U_2 = \emptyset}} \{\mathrm{def}(U_1) + \mathrm{def}(U_2)\}\ \ .$$

That is, for arbitrary disjoint $\emptyset \neq V_1, V_2 \subseteq V$ with $V_1 \cup V_2 = V$ we have

$$\begin{aligned} \text{def}(V) &> \text{def}(V_1) + \text{def}(V_2) \\ |V| - \text{def}(V) &< |V_1| - \text{def}(V_1) + |V_2| - \text{def}(V_2) \\ f(V) &< f(V_1) + f(V_2) \ . \end{aligned}$$

So V is facet inducing for the partial transversal polytope. □

Corollary 3 (Strengthening of Hall's Conditions) *Let $G = (\mathcal{C} \cup \mathcal{R}, E)$ be a bipartite graph, and $D_1, .., D_n \subseteq \mathcal{C}$ the collection of all defining $\mathcal{C}$-sets. There exists a matching covering all elements of $A \subseteq \mathcal{C}$, if and only if for all D_i and for all $X \subseteq A$*

$$|D_i \cap X| \ \leq \ |\Gamma(D_i)| \ . \tag{17}$$

3.2 Generating all Facets, and a Generating Subset

Now that we know how to strengthen constraints (8), we would like to make algorithmic use of this knowledge. We will first see that taking unions of defining $\mathcal{C}$-sets again yields a defining $\mathcal{C}$-set, if we preserve the f-flat property.

Definition (The flat-union $\sqcup$) Given a bipartite graph $G = (\mathcal{C} \cup \mathcal{R}, E)$ and two sets $U_1, U_2 \subseteq \mathcal{C}$, then the flat-union $\sqcup$ is defined as follows:

$$U_1 \sqcup U_2 := U_1 \cup U_2 \cup \{c \in \mathcal{C} : \Gamma(\{c\}) \in \Gamma(U_1) \cup \Gamma(U_2)\} \ .$$

Lemma 4 (The flat-union of defining $\mathcal{C}$-sets) *Given a bipartite graph $G = (\mathcal{C} \cup \mathcal{R}, E)$, a set function f as in (15), and two defining $\mathcal{C}$-sets $U_1, U_2 \subseteq \mathcal{C}$ such that*

$$f(U_1) + f(U_2) > f(U_1 \cup U_2) \ .$$

Then $U = U_1 \sqcup U_2$ is a defining $\mathcal{C}$-set.

Proof: By definition, U is an f-flat. We assume for contradiction that there are disjoint $V_1, V_2 \neq \emptyset$ with

$$U \ = \ V_1 \cup V_2 \tag{18}$$

$$f(U) \ = \ f(V_1) + f(V_2) \ . \tag{19}$$

U_1, U_2 and V_1, V_2 both partition U. Thus, U_1 or U_2 cannot be completly contained in V_1 or V_2, so at least one of U_1, U_2 has to have a non trivial intersection with V_1 and V_2. W.l.o.g., $U_1 \cap V_1 \neq \emptyset$ and $U_1 \cap V_2 \neq \emptyset$. A consequence of (19) is

$$\nu((U_1 \cup \Gamma(U_1))) = \nu(((U_1 \cap V_1) \cup \Gamma(U_1 \cap V_1))) + \nu(((U_1 \cap V_2) \cup \Gamma(U_1 \cap V_2))) \tag{20}$$

which is equivalent to

$$\begin{aligned} |U_1 \cup \Gamma(U_1)| - \text{def}((U_1 \cup \Gamma(U_1), E)) \ = \ & |U_1 \cap V_1| - \text{def}(((U_1 \cap V_1) \cup \Gamma(U_1 \cap V_1), E)) + \\ & |U_1 \cap V_2| - \text{def}(((U_1 \cap V_2) \cup \Gamma(U_1 \cap V_2), E)) \ . \end{aligned}$$

That is, U_1 is not f-inseparable, hence it is not facet inducing, and thus no defining $\mathcal{C}$-set. □

The number of defining $\mathcal{C}$-sets can be as large as $2^{|\mathcal{R}|}-1$. Consider $G=(\mathcal{C}\cup\mathcal{R},E)$ described by the incidence matrix

$$A=\begin{pmatrix}1&1&1&0&0&0&1&0&0&0&0&1&0&0\\0&0&1&1&1&1&0&0&0&0&1&0&0&0\\0&0&0&0&0&1&1&1&1&1&0&0&0&0\\0&0&0&0&0&0&0&0&0&1&1&1&1&1\end{pmatrix} .$$

One can see, that for all $R\subseteq\mathcal{R}$, with $|R|\le 2$, $\Gamma^{-1}(R)$ is a defining $\mathcal{C}$-set. It is a consequence of Lemma 4 that $\Gamma^{-1}(R)$ is a defining $\mathcal{C}$-set for all $R\subseteq\mathcal{R}$.

Even though the number of facets can be large, we will show we can obtain all facets from a (practically small) subset via Lemma 4.

Definition (Atomic defining $\mathcal{C}$-set) Given a bipartite graph $G=(\mathcal{C}\cup\mathcal{R},E)$. A defining $\mathcal{C}$-set A is called *atomic*, if $|A|>1$, and no two defining $\mathcal{C}$-sets $U_1,U_2\subseteq A$ exist, such that

$$\begin{aligned}\Gamma(U_1)\cup\Gamma(U_2) &= \Gamma(A)\\ \Gamma(U_1)\cap\Gamma(U_2) &\neq \emptyset\ .\end{aligned}$$

All other defining $\mathcal{C}$-sets are called *non-atomic*.

Theorem 5 (Number of atomic defining $\mathcal{C}$-sets) *Given a bipartite Graph $G=(\mathcal{C}\cup\mathcal{R},E)$. The number of atomic defining $\mathcal{C}$-sets is at most* $\operatorname{def}(\mathcal{C})=\operatorname{def}(G)$.

Proof: Proof by induction on $r=|\mathcal{R}|$:
The assertion is easily verified for $r=1$ and $r=2$. For the induction step, let $\mathcal{A}$ be the set of all atomic defining $\mathcal{C}$-set and $h:2^{\mathcal{C}}\to\mathbb{N},\ U\mapsto|\{C\in\mathcal{A}:C\subseteq U\}|$.

Case I: $\mathcal{C}$ is an atomic defining $\mathcal{C}$-set.
Let $A_1,\ldots,A_k\subset\mathcal{C}$ be all inclusion maximal subsets of $\mathcal{C}$, from Lemma 4 we know that

$$\forall A\in\mathcal{A}\exists D_i : A\subset D_i\ . \tag{21}$$

Now the assertion is easy to show.

$$\begin{aligned}\operatorname{def}(\mathcal{C}) &\overset{\text{Lemma 12}}{\geq} \sum_{i=1}^{k}\operatorname{def}(D_i)+1\\ &\overset{\text{by ind. hyp.}}{\geq} \sum_{i=1}^{k} h(D_i)+1\\ &\overset{(21)}{=} |\mathcal{A}|\end{aligned}$$

Case II: $\mathcal{C}$ is not a defining $\mathcal{C}$-set or a non-atomic defining $\mathcal{C}$-set.
Let $C_i=\mathcal{C}\backslash\{c\in\mathcal{C}:i\in\Gamma(\{c\})\}$. Clearly, we have

$$\mathcal{C}\geq\bigcup_{i=1}^{r+1}C_i\ . \tag{22}$$

Further, one can conclude from Lemma 10 and the f-flat condition of a defining $\mathcal{C}$-set

$$\forall D \in \mathcal{A} \;\; \exists R \subset \mathcal{R} : D = \Gamma^{-1}(R) \ . \tag{23}$$

By supermodularity of def$(\cdot)$ we can proof the assertion.

$$\begin{aligned}
\operatorname{def}(\mathcal{C}) &\overset{(22)}{\geq} \operatorname{def}(\bigcup_{i=1}^{r+1} C_i) \\
&\overset{\text{Lemma 8}}{\geq} \sum_{k=1}^{r+1} (-1)^{k-1} \sum_{1 \leq i_1 < \ldots < i_k \leq r+1} \operatorname{def}(C_{i_1} \cap \ldots \cap C_{i_k}) \\
&\overset{\text{by ind. hyp.}}{\geq} \sum_{k=1}^{r+1} (-1)^{k-1} \sum_{1 \leq i_1 < \ldots < i_k \leq r+1} h(C_{i_1} \cap \ldots \cap C_{i_k}) \\
&\overset{(23)}{=} \sum_{C \in \mathcal{A}} (\sum_{k=1}^{r+1-|\Gamma(C)|} (-1)^{k-1} \sum_{1 \leq i_1 < \ldots < i_k \leq r+1} 1) \\
&\overset{\text{Lemma 9}}{=} \sum_{C \in \mathcal{A}} 1 \\
&= |\mathcal{A}|
\end{aligned}$$

□

3.3 Facet Enumeration

In our present implementation we enumerate all defining $\mathcal{C}$-sets, basically using Lemmas 4 and 11. If the number of defining $\mathcal{C}$-sets is polynomially bounded, the running time of the algorithm is polynomial. As pointed out below, for real-world instances this is a reasonable assumption. Theorem 5 suggests an algorithm which first constructs all atomic $\mathcal{C}$-sets, and repeatedly takes all non disjoint flat-unions. We postpone a detailed description of such an algorithm to the full paper.

4 Consequences

In real-world instances of the *university course timetabling problem* a room can be described by various attributes (or *features*). These may be capacity, location, seating, beamer, blackboard, etc. We distinguish between two types of features, *exclusive* and *inclusive*. Exclusive features cannot be requested at the same time (e.g., different room capacities). It is characteristic to exclusive features that the graph $G_t = (\mathcal{C}_t \cup \mathcal{R}_t, E_t)$ decomposes into independent components. We will show that for each component of G_t the maximum number of defining $\mathcal{C}$-sets only depends on the number of different (inclusive) features.

Lemma 6 (Number of defining $\mathcal{C}$-sets) *If the number of different features in a connected bipartite graph $G = (\mathcal{C} \cup \mathcal{R}, E)$ is ϕ, then the number of defining $\mathcal{C}$-sets in G is at most $2^\phi - 1$.*

Proof: Let $\mathcal{F}$ be the set of features and $F \subseteq \mathcal{F}$. We then denote with C_F all the courses, that apply for a room, which has to be provided with all features $f \in F$. Let

$$\mathcal{D} = \{C_F : F \subseteq \mathcal{F}\} \ .$$

We show that if $A \subseteq \mathcal{C}$ is a defining $\mathcal{C}$-set and $|A| > 1$, then $A \in \mathcal{D}$. We assume that $A \subseteq \mathcal{C}$ is a defining $\mathcal{C}$-set and $A \notin \mathcal{D}$. Then there exists a $c \in \mathcal{C} \backslash A$, such that $\Gamma(c) \subseteq \Gamma(A)$.

Case I: $\nu((A \cup \Gamma(A), E)) = |\Gamma(A)|$. Then, $|A \cup c| - \operatorname{def}(A \cup c) = |A| - \operatorname{def}(A)$, so A is not an f-flat and hence no defining $\mathcal{C}$-set.

Case II: $\nu((A \cup \Gamma(A), E)) < |\Gamma(A)|$. Then, A cannot be a defining $\mathcal{C}$-set (Lemma 10). □

This has important consequences for the applicability of our apporach to real-world instances.

Corollary 7 *For a fixed number of features, the number of defining $\mathcal{C}$-sets is $O(1)$.*

Practical evidence shows that the number of defining $\mathcal{C}$-sets is in fact small. For example, we added a total of about 6400 non-trivial facets to our largest instance, cf. Table 2.

5 Computational Results

All our results were obtained on a 3.2GHz Pentium 4 Linux PC with 1GB memory. Integer programs are solved using CPLEX 10.1. We separately list running times for three steps: (i) facet generation, (ii) solution of the integer program (6)–(10), and (iii) allocating rooms to all assigned periods of all courses via a sequence of perfect matching calculations.

5.1 The PATAT08 International Timetabling Competition

Accompanying the PATAT08 conference, there is an international timetabling competition. The data of seven problems have been published [10]. We present the statistics of our approach for these instances in Table 1. Note that we only report computation times for respecting all given hard constraints. Almost all soft constraints can be easily included in our IP without significantly worsen the running time.

Name	Courses	Course-Slots	Rooms	Violations	Step1	Step2	Step3
comp01	30	160	6	0	$<$ 0.01 sec.	0.05 sec.	0.02 sec.
comp02	82	283	16	0	$<$ 0.01 sec.	0.19 sec.	0.02 sec.
comp03	72	251	16	0	$<$ 0.01 sec.	0.17 sec.	0.02 sec.
comp04	79	286	18	0	$<$ 0.01 sec.	0.24 sec.	0.03 sec.
comp05	54	152	9	0	$<$ 0.01 sec.	0.56 sec.	0.02 sec.
comp06	108	361	18	0	$<$ 0.01 sec.	0.43 sec.	0.04 sec.
comp07	131	434	20	0	$<$ 0.01 sec.	0.57 sec.	0.04 sec.

Table 1: Statistics and results for PATAT08 instances

5.2 Statistics and Results Corresponding to Simulated Data

As we can see, the PATAT08 instances are no challenge to our approach. To obtain a better idea of its potential performance, we developed a simulation tool which is able to create large problem instances with near real-world character. We present statistics of three representative instances of different sizes, cf. Table 2. The key data (not listed here) of the large instance is almost identical to that of Technical University of Berlin (which is a rather large university). Computation times are acceptable, even though for an interactive timetable design, some tuning is necessary. Almost 80% of instructors teach at their first choice time slots.

Name	Courses	Course-Slots	Rooms	Violations	Step1	Step2	Step3
small	180	420	35	0	45 sec.	9 sec.	3 sec.
medium	950	2100	165	0	307 sec.	52 sec.	6 sec.
large	2100	4640	345	0	1235 sec.	5106 sec.	5 sec.

Table 2: Statistics and results for simulated instances

For comparison, we list in Table 3 the results for the same instances when using the intuitive integer program (1)–(5).

Name	# Variables	# Constraints	Runtime	Gap
small	13 000	7000	30 sec.	< 2%
medium	100 000	31 000	510 sec.	7 %
large	240 000	80 000	1 day	no solution

Table 3: Sizes, solution times, and quality for the intuitive integer program (1)–(5).

6 Discussion

We did not discuss several extensions, which are (or can easily be) incorporated in our practical implementation, most notably practical soft constraints. We believe that our generic model (in particular using the concept of a conflict graph) is well suited for this purpose. One can model e.g., that two courses have to be scheduled on consecutive time slots, or that no two lectures of the same course are given on the same day, etc.

One could think of solving the integer program (6)–(10) via branch-and-cut. However, even for our largest instances, the number of facet inducing Hall inequalities (8) was rather small. This is why we simply added all facet inducing inequalities up-front. A true branch-and-cut implementation is under way for an examination timetabling problem (which has a somewhat different flavor). We will report experiences with a soft constraint solver in a separate paper.

We have access to the courses database at Technical University of Berlin. It comprises 2100 courses (to be scheduled to about 4500 time slots), 345 rooms of about 50 types, and 1550 instructors; there are seven time periods each day. The only reason for using simulated data instead of the real instance, is that the database is severely inconsistent and incomplete [11]. It is planned to manually repair and complete the necessary data in the near future, and to test our implementation for the construction of timetables for the whole university.

References

[1] E. Balas and W. Pulleyblank. The perfectly matchable subgraph polytope of a bipartite graph. *Networks*, 13:495–516, 1983.

[2] E. Burke, K. Jackson, J.H. Kingston, and R. Weare. Automated university timetabling: The state of the art. *Comput. J.*, 40(9):565–571, 1997.

[3] E.K. Burke and S. Petrovic. Recent research directions in automated timetabling. *European J. Oper. Res.*, 140(2):266–280, 2002.

[4] E.K. Burke and H. Rudová, editors. *PATAT 2006: Proceedings of the 6th International Conference on the Practice and Theory of Automated Timetabling*, Lect. Notes Comp. Science, Berlin, 2007. Springer. To appear; available online at `http://patat06.muni.cz/proceedings.html`.

[5] M.W. Carter. A comprehensive course timetabling and student scheduling system at the university of Waterloo. In E. Burke and W. Erben, editors, *PATAT 2000: Proceedings of the 3th International Conference on the Practice and Theory of Automated Timetabling*, volume 2079 of *Lect. Notes Comp. Science*, pages 64–82, Berlin, 2001. Springer.

[6] S. Daskalaki and T. Birbas. Efficient solutions for a university timetabling problem through integer programming. *European J. Oper. Res.*, 127(1):106–120, January 2005.

[7] S. Daskalaki, T. Birbas, and E. Housos. An integer programming formulation for a case study in university timetabling. *European J. Oper. Res.*, 153:117–135, 2004.

[8] J. Edmonds. Maximum matching and a polyhedron with 0,1-vertices. *J. Res. Nat. Bur. Standards*, 69B:125–130, 1965.

[9] C.M. Grinstead and J.L. Snell. *Introduction to Probability.* American Mathematical Society, second edition, 2003. Available at `http://math.dartmouth.edu/~prob/prob/prob.pdf`.

[10] ITC2007: 2nd International Timetabling Competition. `http://www.cs.qub.ac.uk/itc2007/`.

[11] G. Lach. Modelle und Algorithmen zur Optimierung der Raumvergabe der TU Berlin. Master's thesis, Technische Universität Berlin, Institut für Mathematik, 2007. In German.

[12] R. Lewis. A survey of metaheuristic-based techniques for university timetabling problems. *OR Spectrum*, 2007. In press.

[13] L. Lovász and M.D. Plummer. *Matching Theory.* North-Holland, Amsterdam, 1986.

[14] C. Meyers and J.B. Orlin. Very large-scale neighborhood search techniques in timetabling problems. In Burke and Rudová [4], pages 36–52.

[15] A. Qualizza and P. Serafini. A column generation scheme for faculty timetabling. In E.K. Burke and M.A. Trick, editors, *PATAT 2004: Proceedings of the 5th International Conference on the Practice and Theory of Automated Timetabling*, volume 3616 of *Lect. Notes Comp. Science*, pages 161–173, Berlin, 2005. Springer.

[16] A. Schaerf. A survey of automated timetabling. *Artificial Intelligence Review*, 13(2):87–127, 1999.

[17] K. Schimmelpfeng and S. Helber. Application of a real-world university-course timetabling model solved by integer programming. *OR Spectrum*, 29:783–803, 2007.

[18] A. Schrijver. *Combinatorial Optimization Polyhedra and Efficiency.* Springer, Berlin, 2003.

A Auxiliary Results

The first two lemmas follow directly from the inclusion-exclusion principle in probability theory [9].

Lemma 8 *Given a set V and a supermodular function $f : 2^V \to \mathbb{N}$, then for arbitrary $A_1, \ldots, A_n \subseteq V$ the following inequality holds:*

$$f(\bigcup_{i=1}^{n} A_i) \geq \sum_{k=1}^{n} (-1)^{k-1} \sum_{1 \leq i_1 < \ldots < i_k \leq n} f(A_{i_1} \cap \ldots \cap A_{i_k}) \tag{24}$$

Proof: We proof the inequality by induction on n. For the base case, (24) trivially holds for $n = 1$; and for $n = 2$, (24) is accomplished because of the supermodularity of f. For the inductive step we have

$$\begin{aligned}
f(\bigcup_{i=1}^{n+1} A_i) &= f(\bigcup_{i=1}^{n} A_i \cup A_{n+1}) \\
&\geq f(\bigcup_{i=1}^{n} A_i) + f(A_{n+1}) - f(\bigcup_{i=1}^{n} A_i \cap A_{n+1}) \\
&\geq \sum_{k=1}^{n} (-1)^{k-1} \sum_{1 \leq i_1 < \ldots < i_k \leq n} f(A_{i_1} \cap \ldots \cap A_{i_k}) + f(A_{n+1}) \\
&\quad - \sum_{k=1}^{n} (-1)^{k-1} \sum_{1 \leq i_1 < \ldots < i_k \leq n} f(A_{i_1} \cap \ldots \cap A_{i_k} \cap A_{n+1}) \\
&= \sum_{k=1}^{n+1} (-1)^{k-1} \sum_{1 \leq i_1 < \ldots < i_k \leq n+1} f(A_{i_1} \cap \ldots \cap A_{i_k})
\end{aligned}$$

□

Lemma 9 *For each $n \in \mathbb{N}$ the following equation holds:*

$$\sum_{k=1}^{n} (-1)^{k-1} \sum_{1 \leq i_1 < \ldots < i_k \leq n} 1 = 1 \tag{25}$$

Lemma 10 *Given a bipartite graph $G = (\mathcal{C} \cup \mathcal{R}, E)$ and a defining $\mathcal{C}$-set $U \subseteq \mathcal{C}$ with cardinality larger one, then we have*

$$\nu((U \cup \Gamma(U), E)) = |\Gamma(U)| \ .$$

Proof: Assume for contradiction that $\nu((U \cup \Gamma(U), E)) < |\Gamma(U)|$. Then there exists at least one unmatched $r \in \mathcal{R}$ for all maximal matchings. So we choose $c \in \Gamma(\{r\})$ and define:

$$\begin{aligned}
U_1 &= c \\
U_2 &= U \backslash \{c\}
\end{aligned}$$

Clearly, U_1 and U_2 are disjoint. Furthermore,

$$\begin{aligned} f(U_1) + f(U_2) &= |U_1| - \operatorname{def}(U_1) + |U_2| - \operatorname{def}(U_2) \\ &= |U| - \operatorname{def}(U_1) \\ &= f(U) \ . \end{aligned}$$

Thus, U does not induce a facet of $P(\mathcal{C})$. This is a contradiction to Lemma 2. $\square$

Lemma 11 *Given a bipartite graph $G = (\mathcal{C} \cup \mathcal{R}, E)$ and an atomic $\mathcal{C}$-set $U \subseteq \mathcal{C}$, then all inclusion maximal defining subsets $D_1, \dots, D_k \subseteq U$ are disjoint, and furthermore their neighbourhoods $\Gamma(D_1), \dots, \Gamma(D_k)$ are disjoint.*

Proof: Assume for contradiction that there exist two inclusion maximal defining disjoint subsets D_1, D_2 with $\Gamma(D_1) \cap \Gamma(D_2) \neq \emptyset$.
Case I: $\Gamma(D_1) \cup \Gamma(D_2) = \Gamma(U)$
Then U is not atomic.
Case II: $\Gamma(D_1) \cup \Gamma(D_2) \subsetneq \Gamma(U)$
Then $D_1 \sqcup D_2$ is defining and D_1 is not an inclusion maximal defining subset of U. $\square$

Lemma 12 *Given a bipartite graph $G = (\mathcal{C} \cup \mathcal{R}, E)$, an atomic $\mathcal{C}$-set $U \subseteq \mathcal{C}$ and all inclusion maximal defining subsets $D_1, \dots, D_k \subsetneq U$ of U, then the following inequality holds:*

$$\operatorname{def}(U) \geq \sum_{i=1}^{k} \operatorname{def}(D_i) + 1 \ .$$

Proof: For all defining $\mathcal{C}$-sets D we have

$$\begin{aligned} \operatorname{def}(U) &= |U| - |\Gamma(U)| \\ &= \sum_{i=1}^{k} |D_i| - |\Gamma(D_i)| + |U \setminus \bigcup_{i=1}^{k} D_i| - |\Gamma(U) \setminus \bigcup_{i=1}^{k} \Gamma(D_i)| \\ &\geq \sum_{i=1}^{k} \operatorname{def}(D_i) + 1 \end{aligned}$$

$\square$